เอกสารคำสอน

วิชา ความรู้เบื้องต้นทางปัญญาประดิษฐ์
(Introduction to Artificial Intelligence)

รหัสวิชา 030523111

โดย

ผู้ช่วยศาตราจารย์ ดร.ชูพันธุ์ รัตนโภคา

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์

วิทยาลัยเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

คำนำ

เอกสารคำสอน วิชา **ความรู้เบื้องต้นทางปัญญาประดิษฐ์ (Introduction to Artificial Intelligence)** รหัสวิชา 030523111 เล่มนี้ได้จัดทำขึ้นสำหรับการเรียนการสอนในหลักสูตรอุตสาหกรรมศาสตรบัณฑิต สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ (คอมพิวเตอร์) ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ เนื้อหาแบ่งออกเป็น 15 หัวเรื่อง ซึ่งเป็นการเรียนการสอนภาคทฤษฎีทั้งหมด แต่ละหัวเรื่องใช้เวลาในการสอนพร้อมทำแบบฝึกหัด 3 ชั่วโมง/สัปดาห์

โดยมีจุดประสงค์รายวิชาเพื่อให้ผู้เรียนเข้าใจถึงความรู้เบื้องต้นทางปัญญาประดิษฐ์ วิธีการค้นหาเพื่อนำทางไปสู่คำตอบ วิวัฒนาการของการนำเสนอองค์ความรู้และตรรกะ ฟัซซี่ลอจิก การเรียนรู้ของเครื่องจักร โครงข่ายประสาทเทียม การจดจำและแยกแยะรูปแบบ และการประยุกต์ใช้ในทางวิศวกรรม

เอกสารนี้ได้จัดทำและเริ่มใช้ในการเรียนการสอนครั้งแรกเมื่อวันที่ 1 เมษายน 2556 และได้รับการปรับปรุงล่าสุดเมื่อวันที่ 1 กรกฎาคม 2559 สำหรับการเรียนการสอนในภาคการศึกษาที่ 1 ปีการศึกษา 2559

ผู้ช่วยศาสตราจารย์ ดร.ชูพันธุ์ รัตนโภคา

อาจารย์ประจำภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์

1 กรกฎาคม 2559

สารบัญ

บทที่	หัวข้อ	หน้า
	รายละเอียดของรายวิชา	ก - ช
1	ประวัติของปัญญาประดิษฐ์และการประยุกต์ใช้งาน	2
2	การเขียนปริภูมิสถานะและเทคนิคการค้นหาโดยไม่มีการชี้แนะ	30
3	เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย	61
4	เทคนิคการค้นหาเมื่อมีคู่ปรปักษ์	89
5	เทคนิคการแก้ปัญหาแบบมีเงื่อนไข	106
6	ตรรกศาสตร์ที่ว่าด้วยประพจน์และการอนุมาน	132
7	ตรรกศาสตร์ที่ว่าด้วยภาคขยายและการอนุมาน	148
8	ตรรกศาสตร์กำกวม	168
9	ระบบผู้เชี่ยวชาญและการอนุมานแบบฐานกฎ	188
10	พื้นฐานทฤษฎีความน่าจะเป็น	208
11	ทฤษฎีของเบย์และเครือข่ายความเชื่อแบบเบย์	227
12	ความไม่แน่นอนและการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด	241
13	ต้นไม้การตัดสินใจ	253
14	โครงข่ายประสาทเทียม	279
15	อัลกอริทึมเชิงพันธุกรรม	293
ภาคผนวก	การประยุกต์ใช้ปัญญาประดิษฐ์ในทางวิศวกรรม	307

รายละเอียดของรายวิชา

030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์

ชื่อสถาบันอุดมศึกษา	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
วิทยาลัย/คณะ/ภาควิชา	วิทยาลัยเทคโนโลยีอุตสาหกรรม ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์

หมวดที่ 1 ข้อมูลทั่วไป

1 รหัสและชื่อรายวิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์
2 จำนวนหน่วยกิต 3(3-0-6)
3 หลักสูตรและประเภทของรายวิชา อุตสาหกรรมศาสตรบัณฑิต สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์
4 อาจารย์ผู้รับผิดชอบรายวิชา ผู้ช่วยศาสตราจารย์ ดร.ชูพันธุ์ รัตนโภคา
5 ภาคการศึกษา / ชั้นปีที่เรียน ภาคการศึกษาที่ 1 ชั้นปี 2
6 รายวิชาที่ต้องเรียนมาก่อน (Pre-requisite) (ถ้ามี) ไม่มี
7 รายวิชาที่ต้องเรียนพร้อมกัน (Co-requisite) (ถ้ามี) ไม่มี
8 สถานที่เรียน อาคาร 62 ชั้น 6 ห้องเรียนปกติ
9 วันที่จัดทำหรือปรับปรุงรายละเอียดของรายวิชาครั้งล่าสุด ภาคการศึกษาที่ 1 พ.ศ. 2555

หมวดที่ 2 จุดมุ่งหมายและวัตถุประสงค์

1 จุดมุ่งหมายของรายวิชา รู้และเข้าใจความรู้เบื้องต้นเกี่ยวกับวิวัฒนาการของการนำเสนอองค์ความรู้และตรรกะ ฟัซซี่ลอจิก การเรียนรู้ของเครื่องจักร โครงข่ายประสาทเทียม ระบบหลายตัวจัดการ การจดจำและแยกแยะรูปแบบ และการประยุกต์ใช้ในทางวิศวกรรม
2 วัตถุประสงค์ในการพัฒนา/ปรับปรุงรายวิชา -

หมวดที่ 3 ส่วนประกอบของรายวิชา

1 คำอธิบายรายวิชา ความรู้เบื้องต้นวิวัฒนาการของการนำเสนอองค์ความรู้และตรรกะ ฟัซซี่ลอจิก การเรียนรู้ของเครื่องจักร โครงข่ายประสาทเทียม ระบบหลายตัวจัดการ การจดจำและแยกแยะรูปแบบ และการประยุกต์ใช้ในทางวิศวกรรม			
2 จำนวนชั่วโมงที่ใช้/ภาคการศึกษา			
บรรยาย	**สอนเสริม**	**การฝึกปฏิบัติ/งานภาคสนาม/การฝึกงาน**	**การศึกษาด้วยตนเอง**
45 ชั่วโมง	ไม่มี	ไม่มี	90 ชั่วโมง
3 จำนวนชั่วโมงต่อสัปดาห์ที่อาจารย์จะให้คำปรึกษาและแนะนำทางวิชาการแก่นักศึกษาเป็นรายบุคคล ~ อาจารย์ประจำรายวิชา ประกาศเวลาให้คำปรึกษาแก่นักศึกษา ~ อาจารย์จัดเวลาให้คำปรึกษาเป็นรายบุคคล หรือ รายกลุ่มตามความต้องการ 1 ชั่วโมงต่อสัปดาห์ (เฉพาะรายที่ต้องการ)			

หมวดที่ 4 การพัฒนาผลการเรียนรู้ของนักศึกษา

1 คุณธรรม จริยธรรม
1.1 คุณธรรม จริยธรรมที่ต้องพัฒนา ~ พัฒนาผู้เรียนให้มีวินัย ตรงต่อเวลา รับผิดชอบต่อตนเองและสังคม เคารพกฎระเบียบและข้อบังคับต่างๆ ~ พัฒนาผู้เรียนให้มีภาวะความเป็นผู้นำและผู้ตาม สามารถทำงานเป็นหมู่คณะ สามารถแก้ไขข้อขัดแย้งตามลำดับความสำคัญ เคารพสิทธิและรับฟังความคิดเห็นของผู้อื่น

1.2 วิธีการสอนที่จะใช้พัฒนาการเรียนรู้

- ทำงานเดี่ยว โดยมุ่งเน้นให้มีวินัย ตรงต่อเวลา ความรับผิดชอบและความซื่อสัตย์ในการทำงาน
- ทำงานกลุ่ม โดยที่ทุกคนในกลุ่มจะต้องแบ่งภาระงานการทำ
- มีการกำหนดกฎข้อบังคับในการเรียน

1.3 วิธีการประเมินผล

- พฤติกรรมการเข้าเรียน และส่งงานที่ได้รับมอบหมายตามขอบเขตที่ให้และตรงเวลา
- มีการอ้างอิงเอกสารที่ได้นำมาทำรายงาน อย่างถูกต้องและเหมาะสม
- ประเมินผลการวิเคราะห์กรณีศึกษา
- ประเมินผลการนำเสนอรายงานที่มอบหมาย

2 ความรู้

2.1 ความรู้ที่จะได้รับ

ความรู้ที่จะได้รับจากวิชานี้เกี่ยวกับพื้นฐานปัญญาประดิษฐ์ ซึ่งจะเป็นพื้นฐานสำคัญสำหรับวิชาอื่น ๆ ในแขนงต่อไป ประกอบด้วยด้านต่าง ๆ ดังนี้

- ความรู้เบื้องต้นเกี่ยวกับวิวัฒนาการของการนำเสนอองค์ความรู้และตรรกะ ฟัซซี่ลอจิก
- การเรียนรู้ของเครื่องจักร โครงข่ายประสาทเทียม
- ระบบหลายตัวจัดการ การจดจำและแยกแยะรูปแบบ

2.2 วิธีการสอน

- บรรยายเนื้อหารายวิชา
- การทำงานเดี่ยวและงานกลุ่ม การนำเสนอและอภิปราย

2.3 วิธีการประเมินผล

- การสอบกลางภาคและการสอบปลายภาคด้วยข้อสอบ
- การทดสอบด้วยวิธีถามตอบในห้องเรียน
- การทำการบ้านโดยมุ่งเน้นการคำนวณและการประยุกต์ใช้งาน

3 ทักษะทางปัญญา

3.1 ทักษะทางปัญญาที่ต้องพัฒนา

- พัฒนาความสามารถในการคิดอย่างเป็นระบบ มีการวิเคราะห์ เพื่อการป้องกันและแก้ไขปัญหาที่เกิดขึ้น
- พัฒนาความสามารถในการสืบค้นและแสวงหาความรู้เพิ่มเติมได้ด้วยตนเอง

3.2 วิธีการสอน

- บรรยายเนื้อหารายวิชา
- แบบฝึกหัด

~ การทำงานเดี่ยวและการทำงานกลุ่ม การนำเสนอและอภิปราย

~ สอบกลางภาคและสอบปลายภาค ด้วยข้อสอบ

3.3 วิธีการประเมินผลทักษะทางปัญญาของนักศึกษา

~ ทดสอบย่อย สอบกลางภาคและสอบปลายภาค ด้วยข้อสอบ

~ จากแบบฝึกหัด ตรวจสอบถึงความถูกต้องตามหลักวิชาการ

~ จากรายงาน ตรวจสอบถึงความสามารถในการสืบค้นและแสวงหาความรู้เพิ่มเติมได้ด้วยตนเอง

~ จากงานที่มอบหมาย ดูวิธีการแก้ปัญหาและวิธีการแก้ปัญหาโดยใช้ความรู้ที่เรียนมา

4 ทักษะความสัมพันธ์ระหว่างบุคคลและความรับผิดชอบ

4.1 ทักษะความสัมพันธ์ระหว่างบุคคลและความสามารถในการรับผิดชอบ

~ พัฒนาทักษะในการสร้างสัมพันธภาพระหว่างผู้เรียนด้วยกัน

~ พัฒนาความเป็นผู้นำและผู้ตามในการทำงานเป็นทีม

~ มีความรับผิดชอบในการพัฒนาการเรียนรู้ของตนเอง

4.2 วิธีการสอน

~ จัดกิจกรรมกลุ่ม มอบหมายงานรายกลุ่ม การนำเสนอและอภิปราย

~ การทำรายงานเดี่ยว สืบค้นข้อมูลเพิ่มเติมด้วยตนเอง

4.3 วิธีการประเมิน

~ รายงานที่นำเสนอ และพฤติกรรมการทำงานเป็นทีม

~ สังเกตจากพฤติกรรมที่แสดงออกในการร่วมกิจกรรมต่างๆ

~ ความรับผิดชอบ ส่งงานตรงต่อเวลา สามารถอธิบายและเข้าใจข้อมูลที่เรียนรู้ด้วยตนเองได้

5 ทักษะการวิเคราะห์เชิงตัวเลข การสื่อสาร และการใช้เทคโนโลยีสารสนเทศ

5.1 ทักษะการวิเคราะห์เชิงตัวเลข การสื่อสาร และการใช้เทคโนโลยีสารสนเทศที่ต้องพัฒนา

~ พัฒนาทักษะในการสื่อสารทั้งการพูด การฟัง การเขียน โดยการทำรายงาน และนำเสนอในชั้นเรียน

~ พัฒนาทักษะในการวิเคราะห์ข้อมูลจากกรณีศึกษา และการสืบค้น ข้อมูลทางอินเทอร์เน็ต

~ ทักษะในการนำเสนอรายงานโดยใช้รูปแบบ เครื่องมือ และเทคโนโลยีที่เหมาะสม

5.2 วิธีการสอน

~ จัดกิจกรรมกลุ่ม มอบหมายงานรายกลุ่ม การนำเสนอและอภิปราย

~ ทำงานรายงานเดี่ยว สืบค้นข้อมูลเพิ่มเติมด้วยตนเอง

5.3 วิธีการประเมินผล

~ รายงานที่นำเสนอ และพฤติกรรมการทำงานเป็นทีม

~ สอบกลางภาคและสอบปลายภาค ด้วยข้อสอบ

หมวดที่ 5 แผนการสอนและการประเมินผล

1 แผนการสอน

สัปดาห์ที่	หัวข้อ/รายละเอียด	จำนวน (ชม.)	กิจกรรมการเรียนการสอนและสื่อที่ใช้	หมายเหตุ
1	ประวัติของปัญญาประดิษฐ์ และการประยุกต์ใช้งาน	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
2	การเขียนปริภูมิสถานะและเทคนิคการค้นหาโดยไม่มีการชี้แนะ	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
3	เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
4	เทคนิคการค้นหาเมื่อมีคู่ปรปักษ์	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
5	เทคนิคการแก้ปัญหาแบบมีเงื่อนไข	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
6	ตรรกศาสตร์ที่ว่าด้วยประพจน์ และการอนุมาน	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
7	ตรรกศาสตร์ที่ว่าด้วยภาคขยาย และการอนุมาน	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
8	ตรรกศาสตร์กำกวม	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
9	สอบกลางภาค			
10	ระบบผู้เชี่ยวชาญและการอนุมานแบบฐานกฎ	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
11	พื้นฐานทฤษฎีความน่าจะเป็น	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
12	ทฤษฎีของเบย์และเครือข่ายความเชื่อแบบเบย์	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
13	ความไม่แน่นอนและการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
14	ต้นไม้การตัดสินใจ	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
15	โครงข่ายประสาทเทียม	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
16	อัลกอริทึมเชิงพันธุกรรม	3	บรรยาย แบบฝึกหัด และใช้สื่อประสม	-
17	สอบปลายภาค			
		45		

2 แผนการประเมินผลการเรียนรู้

ผลการเรียนรู้*	กิจกรรมการประเมิน (เช่น การเขียนรายงาน โครงงาน การสอบย่อย การสอบกลางภาค การสอบปลายภาค)	กำหนดการประเมิน (สัปดาห์ที่)	สัดส่วนของการประเมินผล
2.1, 2.2, 5.5	สอบกลางภาค และสอบปลายภาค	9 และ 17	60%
1.2, 4.4	การเข้าชั้นเรียน	ตลอดภาคการศึกษา	10%
1.3, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 3.3, 3.4, 3.5, 4.3, 4.4, 5.1, 5.2, 5.3, 5.4	วิเคราะห์กรณีศึกษา ค้นคว้า การรายงานเดี่ยว การนำเสนอรายงาน การทำงานกลุ่ม		30%
* ระบุผลการเรียนรู้หัวข้อย่อยตามแผนที่แสดงการกระจายความรับผิดชอบต่อผลการเรียนรู้			

หมวดที่ 6 ทรัพยากรประกอบการเรียนการสอน

1 ตำราและเอกสารหลักที่กำหนด

- ชูพันธุ์ รัตนโภคา “**ความรู้เบื้องต้นทางปัญญาประดิษฐ์ (Introduction to Artificial Intelligence) เอกสารคำสอนวิชา 030523111**”, 1 กรกฎาคม 2559, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
- ณัฐพงษ์ วารีประเสริฐ และณรงค์ ล่ำดี “**ปัญญาประดิษฐ์ (Atrtificial Intelligence)**” พิมพ์ครั้งที่ 1 มิถุนายน 2552, สำนักพิมพ์ เคทีพี
- บุญเสริม กิจศิริกุล “**ปัญหาประดิษฐ์ (Aritificial Intelligence) เอกสารคำสอนวิชา 2110654**” เวอร์ชัน 1.0.2, 15 มีนาคม 2548, จุฬาลงกรณ์มหาวิทยาลัย
- Stuart J. Russell and Peter Norvig, “**Artificial Intelligence: A Modern Approach**”, Third Edition, Prentice-Hall, Inc., 2010
- George F. Luger, “**Artificial Intelligence: Structures and Strategies for Complex Problem Solving**”, Fifth Edition, Addison-Wesley, 2005

2 เอกสารและข้อมูลสำคัญ ไม่มี
3 เอกสารและข้อมูลแนะนำ ไม่มี

หมวดที่ 7 การประเมินรายวิชาและกระบวนการปรับปรุง

1 กลยุทธ์การประเมินประสิทธิผลโดยนักศึกษา ~ การสนทนากลุ่มระหว่างผู้สอนและผู้เรียน ~ การสะท้อนคิด จากพฤติกรรมของผู้เรียน ~ แบบประเมินผู้สอน และแบบประเมินรายวิชา
2 กลยุทธ์การประเมินการสอน ~ การสังเกตการณ์สอนของผู้ร่วมทีมการสอน ~ ผลการสอบ ~ การทวนสอบผลประเมินการเรียนรู้
3 การปรับปรุงการสอน ~ ประชุมการจัดการเรียนการสอน
4 การทวนสอบมาตรฐานผลสัมฤทธิ์รายวิชาของนักศึกษา ~ มีการตั้งคณะกรรมการในสาขาวิชา ตรวจสอบผลการประเมินการเรียนรู้ของนักศึกษา โดยตรวจสอบข้อสอบ รายงาน วิธีการให้คะแนนสอบ และการให้คะแนนพฤติกรรม
5 การดำเนินการทบทวนและวางแผนการปรับปรุงประสิทธิผลของรายวิชา ~ ปรับปรุงรายวิชาทุกปี ตามข้อเสนอแนะและผลการทวนสอบมาตรฐานผลสัมฤทธิ์ตามข้อ 4

แผนปฏิบัติการสอนสัปดาห์ที่ 1

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี
เรื่อง ประวัติของปัญญาประดิษฐ์และการประยุกต์ใช้งาน **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. ความหมายของปัญญาประดิษฐ์	IS 1.1
2. ประวัติของปัญญาประดิษฐ์	IS 1.2
3. การทดสอบปัญญาประดิษฐ์	IS 1.3
4. คุณสมบัติและข้อจำกัดของปัญญาประดิษฐ์	IS 1.4
5. การนำปัญญาประดิษฐ์มาประยุกต์ใช้งาน	IS 1.5
6. โปรแกรมภาษาที่ใช้ในปัญญาประดิษฐ์	IS 1.6

ข. การจัดการเรียนการสอน

เวลา – นาที		0 – 60	60 – 120	120 – 180
วัตถุประสงค์		**1-2-3-4-5-6**		
การนำเข้าสู่บทเรียน				
ให้เนื้อหา				
สรุปเนื้อหา				
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**		
วิธีการสอน:	บรรยาย			
	อธิบาย			
	ถาม – ตอบ			
	ปฏิบัติด้วยตนเอง			
สื่อการสอน:	กระดานดำ			
	คอมพิวเตอร์			
	โปรเจคเตอร์			

หมายเหตุ IS หมายถึง Instruction sheet

บทที่ 1
ประวัติของปัญญาประดิษฐ์และการประยุกต์ใช้งาน

ปัจจุบันระบบงานและเทคโนโลยีถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกและสนับสนุนการทำงานหลายๆ ด้านให้กับมนุษย์ ถ้าระบบงานหรือเทคโนโลยีนั้นสามารถคิดวิเคราะห์อย่างมีเหตุผล ตัดสินใจ และเรียนรู้เองได้ จะนำไปสู่การประยุกต์ใช้แก้ปัญหาต่างๆ ซึ่งจะมีประโยชน์ต่อมนุษย์อย่างมาก สิ่งเหล่านี้เป็นเรื่องท้าทายความคิดของมนุษย์ในการสร้าง “ความฉลาด” ให้กับเทคโนโลยีมากกว่า 50 ปี กับแนวคิดที่จะสร้างสมองหรือปัญญาเทียม ที่มีความฉลาดเทียบเท่ากับมนุษย์ ทำให้เกิดศาสตร์แขนงหนึ่งทางด้านวิทยาการคอมพิวเตอร์ขึ้นมาเรียกว่า “**ปัญญาประดิษฐ์ (Artificial Intelligence; AI)**”

1.1 ความหมายของปัญญาประดิษฐ์

ในยุคแรกเครื่องคอมพิวเตอร์ส่วนใหญ่จะมีหน้าที่คำนวณ และประมวลผลโปรแกรมของระบบงานตามที่มนุษย์ต้องการโดยมีมนุษย์เป็นผู้ควบคุม ป้อนคำสั่ง และรอผลลัพธ์จากเครื่องคอมพิวเตอร์ การตัดสินใจในสิ่งต่างๆ ยังคงอยู่บนพื้นฐานการสังเคราะห์ของมนุษย์ ทำให้มีแนวคิดที่จะพัฒนาให้เครื่องคอมพิวเตอร์มีประสิทธิภาพสูงขึ้น เพียงพอที่จะอำนวยความสะดวกและลดกำลังการสนับสนุนจากมนุษย์ลงให้น้อยที่สุด ดังนั้นจึงจำเป็นต้องพัฒนาเครื่องคอมพิวเตอร์ให้มีความคิด มีหลักการที่สมเหตุสมผลเทียบเคียงกับสมองมนุษย์ เพื่อให้เครื่องคอมพิวเตอร์สามารถวิเคราะห์ข้อมูลเพื่อตัดสินใจว่าสิ่งใดถูกต้องตามจุดประสงค์ในการทำงาน ด้วยเหตุนี้จึงเป็นที่มาของการพัฒนาสมองของเครื่องคอมพิวเตอร์ ให้มีกระบวนการคิดที่สมบูรณ์แบบแทนการคิดของมนุษย์ ที่เรียกกันว่า “ปัญญาประดิษฐ์ (Artificial Intelligence)”

ปัญญาประดิษฐ์ หรือเรียกสั้นๆ ว่า AI (Artificial Intelligence) เป็นแนวทางการพัฒนาทางด้านคอมพิวเตอร์อีกรูปแบบหนึ่ง ซึ่งทำให้เครื่องคอมพิวเตอร์คิดและตัดสินใจได้ใกล้เคียงกับมนุษย์ โดยอาศัยหลักการจากการศึกษาวิธีคิด การตัดสินใจหรือหลักของเหตุผลจากมนุษย์ เพื่อนำไปใช้ในการพัฒนาศักยภาพของเครื่องคอมพิวเตอร์ให้สามารถตอบสนองการทำงานที่มากกว่าเป็นเพียงเครื่องจักรกลหรือโปรแกรมทั่วไป โดยเริ่มจากการนำแนวคิดดังกล่าวมากำหนดเป็นขั้นตอนให้เครื่องคอมพิวเตอร์ทำงานแก้ปัญหา ตัดสิน และเรียนรู้ได้ด้วยตนเอง ส่งผลให้เครื่องคอมพิวเตอร์มีความฉลาดมากขึ้น สามารถทำงานในระบบที่มีความซับซ้อนได้อย่างมีประสิทธิภาพโดยไม่ต้องอาศัยแรงงานจากมนุษย์

ปัญญาประดิษฐ์ (Artificial Intelligence) เป็นศาสตร์แขนงหนึ่งของวิทยาการคอมพิวเตอร์ที่ต้องการพัฒนาเครื่องจักรกลต่างๆ เช่น เครื่องคอมพิวเตอร์ หรือหุ่นยนต์ เป็นต้น ให้มีความสามารถทางด้านความคิด การตัดสินใจ และพฤติกรรมที่คล้ายกับมนุษย์มากที่สุด เพื่อการแก้ปัญหาหรือหาเหตุผลจากข้อมูลหรือองค์ความรู้ที่มีอยู่แล้วนำมาวิเคราะห์ และแปลความหมายให้ได้ผลสรุปหรือผลลัพธ์ที่ถูกต้องและมีความใกล้เคียงกับการตัดสินใจด้วยมนุษย์ ปัญญาประดิษฐ์เป็นศาสตร์ที่ช่วยสนับสนุนระบบงานที่อาศัยการทำงานของเครื่องจักร หรือเครื่องคอมพิวเตอร์ได้เป็นอย่างดี เมื่อมีการพัฒนาปัญญาประดิษฐ์ที่มีประสิทธิภาพการทำงานต่างๆ จะเกิดข้อผิดพลาดน้อย มีการแก้ไขปัญหาและตัดสินใจที่เร็วกว่าการใช้สมองมนุษย์ เนื่องจากมนุษย์ยังต้องอาศัยพื้นฐานทางกายภาพหรือร่างกาย ทำให้การทำงานอาจเกิดความผิดพลาดขึ้นได้เสมอ

Russell และ Norvig ได้จำแนกปัญญาประดิษฐ์ออกเป็น 4 ลักษณะ โดยสามารถสรุปได้ดังนี้

1. ระบบความคิดที่เลียนแบบมนุษย์

ปัญญาประดิษฐ์ที่ถูกพัฒนาขึ้น ทำให้เครื่องจักรมีสติปัญญาสามารถคิดเองได้และแสดงการกระทำออกมาได้โดยอัตโนมัติ ซึ่งเป็นการคิดที่อาศัยรูปแบบแนวคิดของมนุษย์ จึงส่งผลต่อการกระทำในด้านการตัดสินใจ การแก้ปัญหา และการเรียนรู้ที่คล้ายคลึงกับมนุษย์อย่างมาก

2. ระบบการกระทำที่เหนือมนุษย์

การกระทำที่เกิดขึ้นจากปัญญาประดิษ์ฐ์ ซึ่งอาศัยพื้นฐานการคิดที่เลียนแบบจากมนุษย์ ส่งผลให้การแสดงออกและการกระทำต่างๆ เต็มเปี่ยมไปด้วยรูปแบบการทำงานของมุนษย์อย่างชัดเจน โดยมีประสิทธิภาพเทียบเท่าหรืออาจมากกว่าการกระทำของมนุษย์

3. ระบบความคิดอย่างมีเหตุผล

กระบวนการคิดของมนุษย์ตั้งอยู่บนพื้นฐานของเหตุและผล ดังนั้นเมื่อปัญญาประดิษฐ์ถูกสร้างขึ้นเพื่อคิดแทนมนุษย์ระบบสติปัญญาต่างๆ จึงต้องอาศัยแบบจำลองการคำนวณที่จะช่วยให้ปัญญาประดิษฐ์สามารถเข้าใจหลักของเหตุและผลสามารถคำนวนเหตุการณ์ต่างๆ ตามความเป็นไปได้และที่คาดการณ์ไว้ล่วงหน้าได้

4. ระบบการกระทำอย่างมีเหตุผล

ด้วยกระบวนการคิดอย่างมีเหตุผลของปัญญาประดิษ์ฐ์ ส่งผลให้การแสดงออกและการกระทำต่างๆ สอดคล้องกันอย่างสมเหตุสมผล ซึ่งกระบวนการคำนวณที่มีประสิทธิภาพจะแสดงถึงความชาญฉลาดของปัญญาประดิษฐ์ที่กำหนดการกระทำหรือพฤติกรรมที่แสดงออกมาได้อย่างชัดเจน การกระทำอย่างมีเหตุผลนี้ยังเป็นแนวทางในการออกแบบ “ตัวแทนปัญญา (Intelligent Agent)” ด้วย

แม้ว่ามีผู้กำหนดคำนิยามของคำว่า “ปัญญาประดิษฐ์” ไว้มากมาย แต่โดยสรุปแล้วปัญญาประดิษฐ์เป็นศาสตร์แขนงหนึ่งของวิทยาการคอมพิวเตอร์ ที่ใช้สำหรับพัฒนาให้เครื่องคอมพิวเตอร์หรือเครื่องจักรมีกระบวนการคิดเทียบเท่ากับมนุษย์ เช่น การพัฒนาให้คอมพิวเตอร์หรือหุ่นยนต์สามารถคิดและมีพฤติกรรมเลียนแบบมนุษย์ ตั้งแต่กระบวนการตัดสินใจแก้ปัญหา วินิจฉัย หาเหตุผล จากความรู้ที่จัดเก็บไว้ และนำความรู้นั้นมารวมกันเพื่อหาข้อสรุปหรือผลลัพธ์ของปัญหาได้อย่างมีประสิทธิภาพ

1.2 ประวัติของปัญญาประดิษฐ์

แนวคิดที่ถือว่าเป็นรากฐานสำคัญของปัญญาประดิษฐ์เกิดขึ้นตั้งแต่ยุคอาณาจักรกรีก โดยอริสโตเติล (Aristotle) นักปราชญ์ในยุคนั้น ได้แบ่งแยกความแตกต่างระหว่างมวลสาร (Matter) กับรูปแบบ (Form) ซึ่งต่อมาได้พัฒนามาเป็นพื้นฐานการคำนวณเชิงสัญลักษณ์ จากแนวคิดของอริสโตเติลนี้เองที่ถูกนักคณิตศาสตร์หลายๆคนได้นำมาสานต่อ จนวงการปัญญาประดิษฐ์ถือว่าแนวคิดของอริสโตเติลเป็นจุดกำเนิดของศาสตร์แขนงนี้ ซึ่งแบ่งการพัฒนาของปัญญาประดิษฐ์ ได้ตามยุคสมัยต่าง ๆ ดังนี้

1.2.1 400 ปีก่อนคริสต์ศักราช

อริสโตเติลได้คิดค้นแนวคิดทางตรรกศาสตร์ชื่อ Syllogism ซึ่งประกอบด้วย 3 ส่วนคือ สมมุติฐานหลัก, สมมุติฐานรอง และข้อสรุป ต่อมาได้พัฒนามาเป็น พื้นฐานการคำนวณเชิงสัญลักษณ์ จากแนวคิดของอริสโตเติลถือว่าเป็นพื้นฐานที่สำคัญเกี่ยวกับปัญญาประดิษฐ์

1.2.2 ช่วงศตวรรษที่ 17

René Descartes ชาวฝรั่งเศสภาพที่ 1-1(ก) เป็นผู้ที่ให้ความสนใจกับเรื่องของจิตใจและร่างกายมนุษย์ จึงได้คิดทฤษฎีเรื่อง Dualism หรือ ทวินิยม คือ ทฤษฎีทางปรัชญาที่ถือว่าความแท้จริงปฐมธาตุมี 2 อย่างคือ จิตกับสสาร แยกออกเป็น 2 กลุ่ม คือกลุ่มที่ถือว่าจิตเป็นผู้สร้างสสารเรียกว่า รังสรรค์นิยม และกลุ่มที่ถือว่า จิตกับสสารมีควบคู่กันมาตั้งแต่ต้น จิตควบคุมสสารได้โดยรู้กฎองสสารเรียกว่าชีวสสารนิยม กล่าวคือ

- ร่างกายมนุษย์เหมือนเครื่องจักรที่สามารถยืดขยายและขยับได้ภายใต้กฎของฟิสิกส์
- จิตใจของมนุษย์ไม่สามารถจะมองเป็นวัสดุได้ไม่มีการยืดขยายหรือขยับและไม่อยู่ภายใต้กฎของฟิสิกส์
- จิตใจจะถูกผูกติดกับร่างกายและเป็นตัวสั่งงานให้ร่างกายทำงาน

ซึ่งจากทฤษฎีนี้เป็นแนวคิดของศาสตร์ทางด้านหุ่นยนต์ที่แยกส่วนทางกายภาพ (ตัวหุ่นยนต์) ออกจากส่วนที่ทำหน้าที่สั่งงาน (โปรแกรมควบคุม)

(ก) René Descartes

(ข) Wilhelm Schickard

ภาพที่ 1-1 รูปบุคคลสำคัญ René Descartes และ Wilhelm Schickard

ภาพที่ 1-2 เครื่อง speeding clock จำลอง

ต่อมา Wilhelm Schickard ชาวเยอรมัน ได้สร้างเครื่องจักรคำนวณ ชื่อ speeding clock แสดงในภาพที่ 1-2 ในปี ค.ศ. 1623 ใช้สาหรับคำนวณตำแหน่งของดวงดาว และสามารถใช้ บวก ลบ ตัวเลข 6 หลักได้ แต่ต่อไปมาในปี ค.ศ. 1624 ได้เกิดไฟไหม้ขึ้นทำให้เครื่องคำนวณที่สร้างขึ้นเสียหาย และ Schickard ได้เลิกล้มความพยายามที่จะสร้างมันขึ้นมาใหม่

(ก) Blaise Pascal

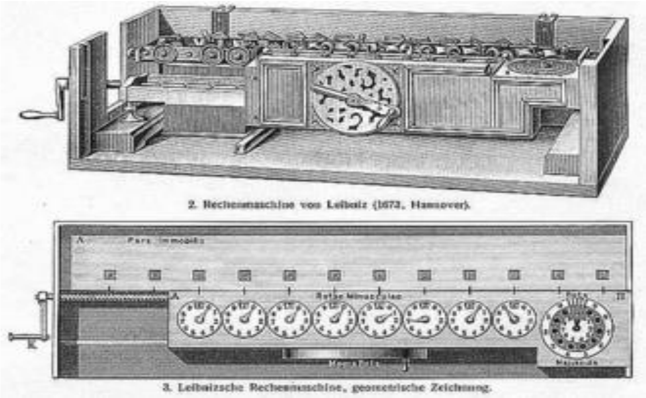
(ข) เครื่องคำนวณ Pascaline

ภาพที่ 1-3 นักคณิตศาสตร์ Blaise Pascal และเครื่องคำนวณ Pascaline

ในปี ค.ศ. 1642 นักคณิตศาสตร์ชาวฝรั่งเศสชื่อ Blaise Pascal (ภาพที่ 1-3(ก)) ได้สร้างเครื่องกลสำหรับการคำนวณชื่อ Pascaline ดังแสดงในภาพที่ 1-3(ข) โดยเครื่องนี้ได้ถูกสร้างขึ้นเนื่องจากความต้องการที่ Pascal อยากจะช่วยพ่อคำนวณเงินภาษีที่เก็บเข้ารัฐ Haute-Normandie โดยที่เครื่องนี้มีความสามารถในการคำนวณการบวกและการลบเท่านั้น สำหรับการคูณและหารนั้นต้องแปลงการคำนวณไปการทำงานต่อเนื่องกันของการบวกและลบแทน แต่อย่างไรก็ตามเครื่อง Pascaline ได้รับความนิยมและได้ถูกใช้งานกันอย่างแพร่หลาย รวมทั้งเครื่อง Pascaline เป็นพื้นฐานของเครื่องคำนวณในสมัยถัดมา

จากนั้นต่อมาในปี ค.ศ. 1672 นักคณิตศาสตร์ชาวเยอรมันชื่อ Gottfried Von Leibniz (ภาพที่ 1-4(ก)) ได้นำหลักการทำงานของเครื่อง Pascaline มาพัฒนาต่อให้สามารถทำการบวก ลบ คูณ และหารได้ รวมทั้งสามารถนำมาใช้ในการคำนวณค่ารากที่สองได้อีกด้วย โดยเครื่องที่ Leibniz พัฒนาขึ้นมานั้นมีชื่อว่า Step Reckoner ดังแสดงในภาพที่ 1-4(ข)

(ก) Gottfried Von Leibniz



(ข) เครื่องคำนวณ Step Reckoner

ภาพที่ 1-4 นักคณิตศาสตร์ Gottfried Von Leibniz และเครื่องคำนวณ Step Reckoner

ภาพที่ 1-5 แบบจำลองของ The Turk เครื่องเล่นหมากรุกอัตโนมัติ

1.2.3 ช่วงศตวรรษที่ 18

ในปี ค.ศ. 1726 นักกวีชาวไอริส Jonathan Swift ได้แต่งนิยายเรื่อง Gulliver’s Travels ที่พูดถึงมนุษย์ตัวเล็กๆ ที่ได้นำเอาเครื่องจักรกลเข้ามาประยุกต์ใช้งาน ซึ่งเครื่องจักรกลต่างๆที่กล่าวถึงในนิยาย ได้ถูกนำมาเป็นแรงบันดาลใจในการพัฒนาเครื่องจักรกลในยุคสมัยถัดๆ ไป

ต่อมาในปี ค.ศ. 1770 นักประดิษฐ์ชาวฮังกาเรียน Wolfgang von Kempelen ได้พัฒนาเครื่องเล่นหมากรุกแบบอัตโนมัติชื่อ The Turk ดังแสดงในภาพที่ 1-5 โดยสิ่งประดิษฐ์นี้ได้มีชื่อเสียงเป็นอย่างมาก เนื่องจากสามารถเล่นหมากรุกชนะได้ทั้ง นาโปเลียน และ เบนจามิน แฟรงกิ้น แต่ต่อมาในปี ค.ศ. 1820 ความลับของ The Turk ก็ถูกเปิดเผยซึ่งความจริงแล้ว The Turk ไม่ใช่เครื่องจักรกลที่สามารถคิดเองได้ แต่เป็นเพียงตุ๊กตาที่มีคนที่เล่นหมากรุกเก่งมากเข้าไปควบคุมข้างใน อย่างไรก็ตาม The Turk ได้ถูกนำไปแสดงในงานต่างๆ จนกระทั่งในปี ค.ศ. 1854 ได้เกิดเพลิงไหม้ที่งานแสดงทำให้ต้นแบบของ The Turk ก็ถูกทำลายไปด้วย

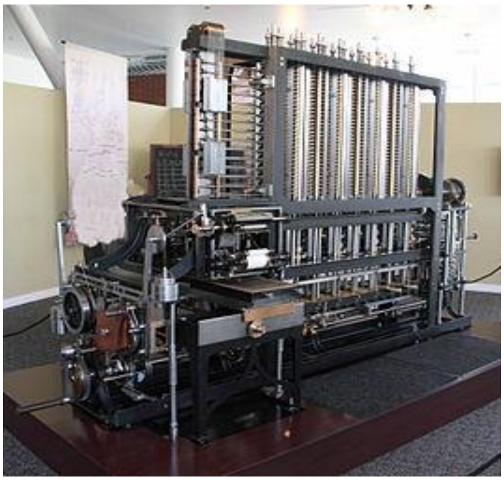
1.2.4 ช่วงศตวรรษที่ 19

ช่วงกลางศตวรรษที่ 18 จนถึงศตวรรษที่ 19 เป็นช่วงเวลาที่มีการปฏิวัติทางอุตสาหกรรมเกิดขึ้น ที่เป็นการเปลี่ยนแปลงครั้งใหญ่ทางด้านเศรษฐกิจ สังคม และเทคโนโลยี โดยเริ่มต้นที่กลุ่มประเทศในบริเตนใหญ่ ที่ต้องการนำเครื่องจักรไอน้ำที่สามารถทำงานอัตโนมัติมาใช้งาน ความก้าวหน้าทางเทคโนโลยีนี้ทำให้เกิด เรือ เรือกำปั่น รถยนต์ และรถไฟ ที่ทำงานด้วยเครื่องจักรไอน้ำ ความเจริญก้าวหน้าไปเผยแพร่ขยายไปสู่ยุโรปตะวันตกและทวีปอเมริกาเหนือและส่งผลกระทบทั่วโลกในที่สุด

ในขณะที่มีการปฏิวัติทางอุตสาหกรรม ก็ได้มีการพัฒนาเครื่องจักรเพื่อการคำนวณและหลักทฤษฎีที่เกี่ยวข้องเกิดควบคู่กันไปด้วย โดยมีบุคคลสำคัญหลายคนที่ช่วยผลักดันเทคโนโลยีการพัฒนาเครื่องจักรคำนวณและทฤษฎีที่เกี่ยวข้องดังนี้

(ก) Charles Babbage

(ข) Ada Lovelace



(ค) เครื่องแบบจำลอง Difference Engine

(ง) เครื่องแบบจำลอง Analytical Engine

ภาพที่ 1-6 นักคณิตศาสตร์ Charcles Babbage, Ada Lovelace และเครื่องคำนวณ Difference Engine, Analytical Engine

Charles Babbage (ภาพที่ 1-6(ก)) นักคณิตศาสตร์ นักปรัชญาวิเคราะห์ และเป็นนักวิทยาศาสตร์คอมพิวเตอร์ชาวอังกฤษ เนื่องจากเขาเป็นคนแรกที่มีแนวคิดเรื่องเครื่องคำนวณที่สามารถโปรแกรมหรือสั่งให้ทำงานได้ โดยเริ่มจากปี ค.ศ. 1822 Babbage ได้ยื่นข้อเสนอเพื่อจะพัฒนาเครื่องจักรที่มีความสามารถในการคำนวณเกี่ยวกับ logarithms และฟังก์ชันทางตรีโกณมิติได้ เพื่อมาช่วยในการคำนวณทางด้านดาราศาสตร์และ คณิตศาสตร์ โดยตั้งชื่อเครื่องนี้ว่า Difference Engine ซึ่งรัฐบาลอังกฤษให้ความสนใจเป็นอย่างมากและเริ่มให้เงินทุนในการพัฒนาในปี ค.ศ. 1823 ระหว่างการพัฒนาเครื่องนี้ Babbage ได้คิดถึงหลักการที่จะทำให้ เครื่องจักรนี้ทำงานเกี่ยวกับการคำนวณได้หลากหลายมากขึ้นโดยตั้งชื่อนี้ว่าเครื่อง Analytical Engine แต่อย่างไรก็ตามการพัฒนาเครื่องจักรนี้ยากมากและใช้งบประมาณที่สูงมาก โดยในปี 1842 รัฐบาลอังกฤษตัดสินใจตัดเงินสนับสนุนในการพัฒนาเครื่องนี้ เนื่องจากงบประมาณที่ให้

Babbage ในการพัฒนาตั้งแต่ต้นรวมแล้วมากกว่างบประมาณทางด้านทหารถึง 2 เท่า และยังไม่สามารถเห็นผลงานเป็นชิ้นเป็นอันได้ แต่อย่างไรก็ตามแนวคิดของเครื่องจักรนี้ได้ถูกยอมรับว่าเป็นต้นแบบของเครื่องคอมพิวเตอร์ในเวลาต่อมา และได้ถูกสร้างแบบจำลองจากเอกสารที่ได้ออกแบบเอาไว้ซึ่งมีลักษณะดังแสดงในภาพที่ 1-6(ค) และ 1-6(ง)

Ada Lovelace (ภาพที่ 1-6(ข)) นักคณิตศาสตร์ชาวอังกฤษ และได้ถูกรับการยกย่องให้เป็นนักเขียนโปรแกรมคนแรกของโลกจากผลงานที่ช่วยให้เครื่องจักร Analytical Engine ที่ออกแบบโดย Charles Babbage สามารถรับโปรแกรมและทำงานตามคำสั่งในโปรแกรมได้ โดย Lovelace ได้เสนอเทคนิคการเขียนโปรแกรมแบบวนรอบซ้ำ ๆ ที่เรียกว่า loop การเขียนโปรแกรมแบบวนรอบนี้สำคัญมาก เพราะถ้าหากไม่ใช้แนวคิดนี้แล้วโปรแกรมที่เขียนจะมีความยาวมาก และผิดพลาดได้ง่ายและการใช้โปรแกรมย่อย (subroutine) เพื่อคำนวณฟังก์ชันย่อยที่ได้ใช้งานบ่อย ๆ ฝังอยู่ในโปรแกรมหลัก โปรแกรมหลักสามารถเรียกใช้โปรแกรมย่อยทำงานได้เลยโดยไม่ต้องเขียนโปรแกรมใหม่ ซึ่งในปัจจุบันนี้ทุกภาษาต่างก็มีความสามารถในการวนซ้ำ และการใช้โปรแกรมย่อย เพื่อเป็นการ ยกย่อง Lovelace ต่อมาในปัจจุบันได้มีภาษาโปรแกรมชื่อ Ada

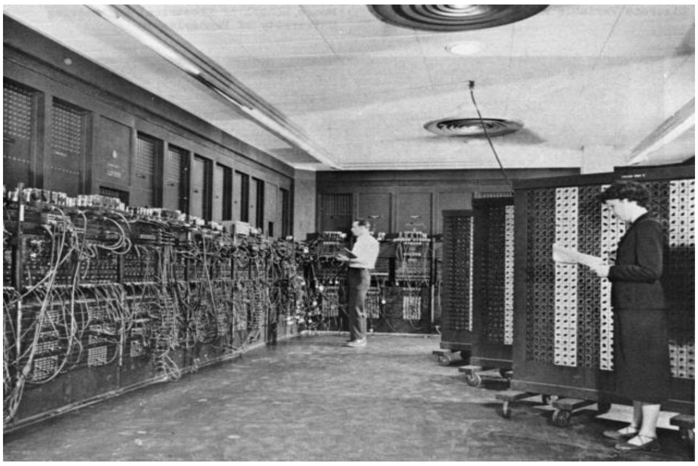
George Boole (ภาพที่ 1-7(ก)) นักคณิตศาสตร์ชาวอังกฤษ ได้ใช้เผยแพร่กฎของ Boolean Algebra ซึ่งเป็นคณิตศาสตร์ที่ใช้อธิบายเหตุผลของตรรกที่ตัวแปรมีค่าได้เพียง "จริง" หรือ "เท็จ" เท่านั้น (ใช้สภาวะสองอย่างคือ 0 กับ 1 ร่วมกับเครื่องหมายในเชิงตรรกพื้นฐาน AND, OR และ NOT)

Augustus De Morgan (ภาพที่ 1-7(ข)) นักคณิตศาสตร์และตรรกศาสตร์ชาวอังกฤษ คิดค้นทฤษฎี De Morgan’s Law เป็นกฎในวิชาตรรกศาสตร์ คือ ชุดของกฎในสาขาตรรกศาสตร์รูปนัยซึ่งแสดงความสัมพันธ์อย่างเป็นระบบระหว่างคู่ของตัวดำเนินการเชิงตรรกที่คู่กัน โดยแสดงในรูปนิเสธ ความสัมพันธ์เช่นนี้เรียกว่าภาวะคู่กันเดอมอร์แกน (De Morgan duality)

(ก) George Boole

(ข) Augustus De Morgan

ภาพที่ 1-7 รูปบุคคลสำคัญ George Boole และ Augustus De Morgan



ภาพที่ 1-8 เครื่องคอมพิวเตอร์ ENIAC

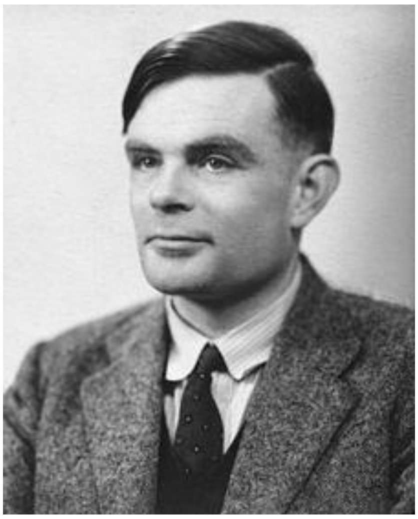
1.2.5 ช่วงศตวรรษที่ 20

ในช่วงศตวรรษนี้ได้เกิดสงครามโลกครั้งที่ 2 ในปี ค.ศ. 1946 ซึ่งจากแรงกระตุ้นของสงครามโลกนี้ได้กำเนิดคอมพิวเตอร์ที่ใช้ไฟฟ้าเครื่องแรกชื่อ “ENIAC” ดังภาพที่ 1-8 และได้มีการพัฒนาเครื่องมือเพื่อใช้คำนวนหาทิศทางการยิงตอบโต้ในสงคราม หลังจากนั้นปี ค.ศ.1950 ได้กำเนิดเครื่องคอมพิวเตอร์สำหรับใช้ในงานธุรกิจเครื่องแรกของโลกที่มีชื่อว่า “UNIVAV” แม้ว่ามนุษย์จะประดิษฐ์เครื่องคำนวนได้สำเร็จแต่ก็ยังห่างไกลกับคำว่า “เครื่องที่คิดได้เอง (Thinking Machine)” ลำดับจากยุคเครื่องคำนวนเข้าสู่ช่วงยุคสมัยต่างๆ ของปัญญาประดิษ์ฐ์ มีดังนี้

1.2.5.1 ยุคก่อนกำเนิดปัญญาประดิษฐ์ (ค.ศ. 1943-1955)

เป็นยุคก่อนที่จะเกิดคำว่า “ปัญญาประดิษฐ์” ในยุคนี้มีผลงานที่สำคัญ คือ ผลงานของ Warren McCulloch และ Walter Pits ซึ่งได้เริ่มวางระบบโครงข่ายใยประสาทเป็นครั้งแรก (ค.ศ. 1943) เป็นระบบประสาทเทียมมีลักษณะ “เปิด” และต่อมา Donald Hebb (ค.ศ. 1949) ได้ศึกษาโครงสร้างและพัฒนากฏเกณฑ์ต่างๆ กฏนี้เรียกว่า “Hebbian Learning” ซึ่งเป็นส่วนสำคัญในการพัฒนาโครงข่ายใยประสาท จากนั้นในปี ค.ศ.1951 Marvin กับ Dean Edmonds นักศึกษาจากคณะคณิตศาสตร์มหาวิทยาลัย Princeton ได้สร้างระบบโครงข่ายใยประสาทตัวแรกของโลกที่มีชื่อว่า “SNARC” โดยสร้างจากหลอดสูญญากาศ 3,000 ตัว และจำลองระบบใยประสาท 40 เส้นในระบบนักบินอัตโนมัติของเครื่องบินทิ้งระเบิด B-24 ถึงแม้คณะกรรมการยังสงสัยถึงการนำคณิตศาตร์มาใช้ในโครงข่ายใยประสาทที่ยังไม่สมบูรณ์ก็ตาม แต่ Von Neumann (ภาพที่ 1-9(ก)) กล่าวว่า “ถึงมันยังไม่ทำงานในตอนนี้ แต่สักวันหนึ่งมันจะทำได้แน่” ผลงานที่โดดเด่นในยุคนี้อีกชิ้นหนึง คือ ผลงานของ Alan Turing (ภาพที่ 1-9(ข)) ในปีค.ศ. 1950 เกี่ยวกับการกำหนดเกณฑ์ในการวัดผลที่ชัดเจน ทำให้มีมาตรวัดที่ชี้เป็นผลสำเร็จ ถือว่าจำเป็นมากในทุกศาสตร์ที่เกี่ยวกับวิทยาศาสตร์

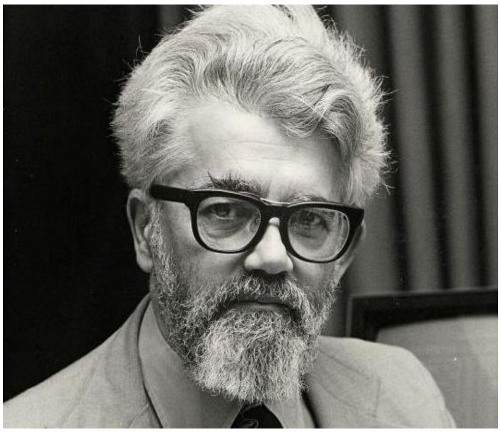
(ก) John von Neumann


(ข) Alan Turing

ภาพที่ 1-9 รูปบุคคลสำคัญ John von Neumann และ Alan Turing

1.2.5.2 ยุคกำเนิดปัญญาประดิษฐ์ (ค.ศ.1956)

John McCarthy (ภาพที่ 1-10) นับได้ว่าเป็นบุคคลที่สำคัญอีกหนึ่งคนที่มีผลงานทางด้านปัญญาประดิษฐ์ เมื่อ John สำเร็จการศึกษา ได้ย้ายมาที่วิทยาลัย Dartmouth ซึ่งถือเป็นจุดเริ่มต้นและการให้กำเนิดสาขาปัญญาประดิษ์ฐ์ โดย John ได้ชักชวนนักวิจัยทั่วอเมริกาที่สนใจศึกษาด้านนี้ เช่น Automata Theory, Neural Network เป็นต้น มาทำงานร่วมกันในปี ค.ศ.1956 หลังจากนั้นได้ร่วมกันจัดสัมมนาเชิงปฏิบัติการเป็นระยะเวลา 2 เดือน Dartmouth ซึ่งผู้เข้าร่วมงานล้วนเป็นบุคคลสำคัญในวงการนี้ทั้งสิ้น แม้ว่าการสัมมนาเชิงปฏิบัติการครั้งนี้ จะไม่ได้ไปสู้แนวความคิดใหม่ในสาขานี้ก็ตาม แต่ได้นำไปสู่รูปลักษณ์ที่สำคัญในที่สุด การสัมมนาครั้งนี้ได้กำหนดชื่อของสาขาวิชานี้ว่า “ปัญญาประดิษฐ์”



ภาพที่ 1-10 John MacCarth บิดาของศาสตร์ทางด้านปัญญาประดิษฐ์

1.2.5.3 ยุคตื่นตัว (ค.ศ.1952-1969)

ในปี ค.ศ.1957 Herbert Simon ได้ทำนายถึงอนาคตของปัญญาประดิษฐ์ไว้ว่า "อีก 10 ปี ปัญญาประดิษฐ์จะพิชิตแชมป์หมากรุก" แต่คำทำนายของเขามาเป็นจริงในภายหลัง ต่อมาในปี ค.ศ.1958 John McCarthy ได้ย้ายไปอยู่ที่ MIT (Massachusetts Institute of Technology) และได้สร้างโปรแกรมภาษา Lisp ซึ่งเป็นโปรแกรมภาษาที่เก่าแก่และใช้งานกันอย่างกว้างขวางในวงการปัญญาประดิษฐ์ จากนั้นในปี ค.ศ.1963 จึงย้ายไปทำงานที่มหาวิทยาลัย Stanford โดยได้คิดค้นปัญญาประดิษฐ์ที่สำคัญต่างๆ เช่น ระบบถามตอบและวางแผนของ Cordell Green, โปรแกรม Analogy โปรแกรมใช้ทดสอบ IQ ของ Tom Evan และโปรแกรมการเข้าใจภาษามนุษย์ Terry Winograd เป็นต้น

1.2.5.4 ยุคเผชิญปัญหาจริง (ค.ศ.1966-1973)

ยุคนี้เป็นยุคเผชิญปัญหาที่เกิดขึ้นในวงการปัญญาประดิษฐ์ ซึ่งตัวอย่างต่อไปนี้เป็นปัญหาหลักที่เกิดขึ้น

ปัญหาแรก ยุคนี้ไม่มีโปรแกรมใดที่เก็บองค์ความรู้ในรูปประโยคต่างๆ ได้อย่างชัดเจน โดยในปี ค.ศ.1966 สภาวิจัยแห่งชาติของอเมริกาพบว่าไม่มีเครื่องมือหรือเครื่องจักรที่แปลภาษาได้ ซึ่งเป็นช่วงสงครามเย็นที่อเมริกาพยายามแข่งขันด้านวิทยาศาสตร์และอวกาศกับรัสเซีย ทางรัฐบาลต้องการหาอุปกรณ์ที่แปลภาษารัสเซียให้เป็นภาษาอเมริกา และความล้มเหลวครั้งนี้ทให้ภาครัฐยกเลิกเงินลงทุนที่ใช้ในการพัฒนาปัญญาประดิษฐ์

ปัญหาที่สอง ยังไม่สามารถแก้ปัญหาทฤษฎีที่ซับซ้อนได้ เนื่องจากข้อจำกัดคือพื้นที่ใช้เก็บข้อมูลมีขนาดใหญ่ ทำให้เป็นอุปสรรคและทำให้ล้มเหลวในการพิสูจน์ทฤษฎีหลายๆเรื่อง

ปัญหาที่สาม ข้อจำกัดบนโครงสร้างพื้นฐานทำให้เกิดปัญหาในการใช้สร้างพฤติกรรมชาญฉลาด เช่น Perceptrons ของ Minsky กับ Papert (ค.ศ.1969) ซึ่งพบว่าไม่สามารถใส่ค่า Perceptron สองค่าที่แตกต่างกันได้ แต่ปัญหาลักษณะนี้สามารถแก้ไขได้ ค.ศ.1980

1.2.5.5 ยุคระบบฐานความรู้ (ค.ศ.1969-1979)

จากปัญหาในยุคที่แล้วทำให้นักวิจัยได้ข้อสรุปว่า วิธีแก้ปัญหายังไม่สมบูรณ์เพียงพอต่อการนำไปใช้งาน จึงหันมาหาความรู้เฉพาะด้านซึ่งนำไปสู่ขั้นตอนที่สมเหตุสมผลการแตกย่อยปัญหา และนำไปสู่ทางแก้ปัญหาในที่สุด โปรแกรม DENDRAL (1969) เป็นผลผลิตจากความร่วมมือของ Ed Firgenbaun, BruceBuchanan (นักปรัชญาที่ผันตัวเป็นนักคอมพิวเตอร์) และ Joshu Lederberg มาร่วมทีมแก้ปัญหาเรื่องการใช้โครงสร้างโมเลกุลหาค่า mass spectrometer โดยการแตกย่อยโครงสร้างโมเลกุลนั้นจะช่วยให้หาค่า mass spectrometer ได้ง่ายขึ้น ซึ่งโปรแกรมแรกของระบบ Knowledge-Intensive-System และนำไปสู่การแยกความรู้ (ในรูปของกฎต่างๆ) ส่งผลให้มีการพัฒนา Heuristics Programming Project ที่เป็นส่วนสำคัญของวิธีการแบบใหม่ๆในระบบผู้เชี่ยวชาญ (Expert System)

1.2.5.6 ยุคปัญญาประดิษฐ์เข้าสู่อุตสาหากรรม (ค.ศ.1980-ปัจจุบัน)

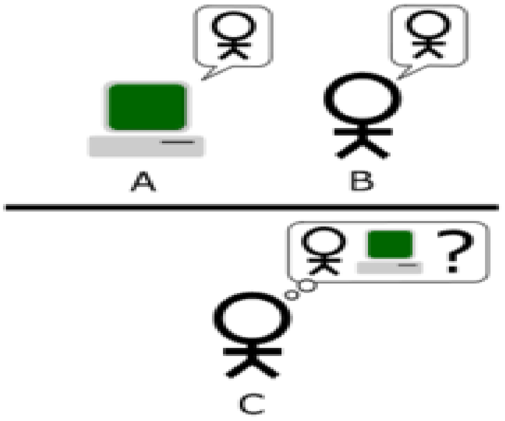
ในปี ค.ศ.1982 ผลงานที่ประสบความสำเร็จชิ้นแรก คือ ระบบผู้เชี่ยวชาญทางการค้าที่ชื่อว่า “R1” ซึ่งพัฒนาโดย McDermott ที่ Digital Equipment Corporation โปรแกรมนี้สามารถช่วยสั่งซื้อสินค้าในระบบคอมพิวเตอร์ และในปี ค.ศ.1986 R1 ได้ช่วยประหยัดเงินให้กับบริษัทได้ถึง 40 ล้านเหรียญสหรัฐต่อปี ในยุคนี้ปัญญาประดิษฐ์ในสาขาระบบผู้เชี่ยวชาญได้ถูกนำมาใช้ในองค์กรธุรกิจเอกชน ซึ่งมีมูลค่ามหาศาลต่อปี อุตสาหกรรมปัญญาประดิษฐ์เฟื่องฟูมากในปี ค.ศ.1980 และสร้างมูลค่ามหาศาลถึงหลายพันล้านดอลลาร์

1.2.5.7 ยุคของตัวแทนปัญญา (ค.ศ.1995-ปัจจุบัน)

ด้วยสาเหตุที่กระบวนการแก้ปัญหาโปรแกรมปัญญาประดิษฐ์มักพบปัญหาเรื่องเกี่ยวข้องกับ Agent อยู่เสมอ จึงส่งผลให้นักวิจัยกลับมาสนใจในเรื่องของตัวแทนปัญญามากขึ้น หนึ่งในสิ่งแวดล้อมที่รู้จักกันดีของ Agent คือ อินเทอร์เน็ตเช่น ระบบปัญญาประดิษฐ์ที่ใช้โดยทั่วไปใน Web-base Application ซึ่งนำ Agent มาใช้งานกันมากในอินเทอร์เน็ตและการค้าหาด้วย Search Engines เป็นต้น

1.3 การทดสอบปัญญาประดิษฐ์

วิทยาศาสตร์ทุกสาขาจำเป็นต้องมีมาตรวัด เพื่อใช้บ่งบอกระดับและคุณลักษณะในศาสตร์หรือสิ่งที่กำลังศึกษาอยู่ ปัญญาประดิษฐ์ก็เช่นกันหากไม่มีมาตรวัดไม่สามารถบอกได้ถึงสิ่งที่กำลังศึกษาและพัฒนาว่ามาถึงจุดที่ยอมรับว่ามี “ความฉลาด” หรือยัง โดยในปี ค.ศ. 1950 Alan Turing ได้ออกแบบการทดสอบปัญญาประดิษฐ์ หรือที่เรียกกันว่า “Turing Test” เพื่อพิสูจน์ว่าปัญญาประดิษฐ์ที่สร้างขึ้นถึงจุดที่ควรจะยอมรับถึง “ความฉลาด” ว่าทัดเทียมกับมนุษย์แล้ว โดยเป็นการนำฉากมากั้น ภายหลังฉากจะมีคอมพิวเตอร์และคนจริง ถ้าหากคนที่อยู่หน้าฉากโต้ตอบกับหลังฉากแล้วไม่สามารถแยกได้ว่า ผู้ตอบเป็นคอมพิวเตอร์หรือคนจริง แสดงว่า ปัญญาประดิษฐ์นั้นผ่านเกณฑ์วัดความฉลาด ดังภาพที่ 1-11



ภาพที่ 1-11 ตัวอย่างการทดสอบ Turing Test

1.4 คุณสมบัติและข้อจำกัดของปัญญาประดิษฐ์

หากกล่าวถึงการทำงานของปัญญาประดิษฐ์แล้วอาจเข้าใจได้ค่อนข้างยาก เนื่องจากเป็นการมองภาพรวมเกี่ยวกับการสร้างเครื่องคอมพิวเตอร์ให้มีความสามารถและฉลาดขึ้นเทียบเท่ากับมนุษย์หรือมากกว่าสำหรับความสามารถของปัญญาประดิษฐ์ เมื่อได้รับการพัฒนาแล้ว จะมีคุณสมบัติดังนี้

- ความสามารถในการเรียนรู้และทำความเข้าใจประสบการณ์ที่ผ่านมา
- ความสามารถในการทำความเข้าใจกับเนื้อหา และข้อมูลที่คลุมเครือ หรือมีความขัดแย้ง
- ความสามารถในการใช้หลักของเหตุและผล โดยนำมาใช้แก้ปัญหาต่างๆ พร้อมทั้งแนะนำวิธีการ หรือ แนวทางได้อย่างมีประสิทธิภาพ
- ความสามารถในการทำความเข้าใจปัญหาที่มีความซับซ้อน และวินิจฉัยเพื่อหาข้อสรุปที่เหมาะสมที่สุดได้
- ความสามารถในการนำองค์ความรู้มาประยุกต์ใช้ได้อย่างเหมาะสมกับสถานการณ์ และสภาพแวดล้อมแบบต่างๆ
- ความสามารถในการจำแนกความสำคัญของข้อมูล ปัญหา หรือองค์ประกอบที่แตกต่างกันตามสถานการณ์ได้
- ความสามารถในการรตอบสนองที่รวดเร็ว ถูกต้อง และแม่นยำ

จากที่กล่าวมาเป็นความสามารถหรือคุณสมบัติของปัญญาประดิษฐ์บางส่วนเท่านั้น ซึ่งในความเป็นจริงเมื่อนำไปพัฒนาและประยุกต์ใช้งานแล้วจะมีคุณสมบัติที่แตกต่างกันออกไปตามลักษณะงานที่นำไปใช้ ความสามารถส่วนใหญ่ของปัญญาประดิษฐ์มักต้องพึ่งพาการทำงานของเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง เพื่อใช้เป็นตัวควบคุมและสนับสนุนการทำงานในด้านต่างๆให้มีความถูกต้องได้อย่างที่ต้องการ

1.4.1 ข้อเปรียบเทียบและข้อจำกัดของปัญญาประดิษฐ์

เมื่อเปรียบเทียบปัญญาประดิษฐ์กับแรงงานมนุษย์แล้ว สามารถสรุปหน้าที่ของปัญญาประดิษฐ์ได้ดังนี้

1.4.1.1 ระบบการคำนวณ

โดยทั่วไปนิยมนำเครื่องคอมพิวเตอร์มาใช้ในการประมวลผลและคำนวณข้อมูลหรือตัวเลขที่ซับซ้อนเกินกว่าสมองมนุษย์จะคำนวณได้ ปัญญาประดิษฐ์มีส่วนช่วยให้วิเคราะห์ขั้นตอนการคำนวณได้อย่างถูกต้องและแม่นยำขึ้น นอกจากนี้ยังสามารถประมวลผลการคำนวณในปริมาณมากๆได้อย่างรวดเร็ว อีกทั้งยังเกิดข้อผิดพลาดน้อยด้วย หน้าที่นี้จึงมักเป็นหน้าที่ที่สำคัญอันดับแรกของการใช้งานปัญญาประดิษฐ์ ซึ่งจะช่วยประหยัดต้นทุนและเวลาได้เป็นอย่างมาก

1.4.1.2 ระบบการจัดเก็บข้อมูล

เมื่อในระบบมีข้อมูลมากขึ้น การจดจำด้วยสมองมนุษย์ย่อมไม่ใช่ทางเลือกที่ดีอย่างแน่นอน เครื่องคอมพิวเตอร์จึงมีส่วนสำคัญอย่างมากในการเก็บข้อมูลเหล่านี้เมื่อมีการนำปัญญาประดิษฐ์เข้ามาใช้ในระบบจัดเก็บข้อมูลจึงช่วยให้เครื่องคอมพิวเตอร์มีความสามารถมากขึ้น ไม่ได้เป็นเพียงพื้นที่จัดเก็บข้อมูลเท่านั้น ปัญญาประดิษฐ์จะช่วยเพิ่มประสิทธิภาพในการเรียนรู้ จดจำ หรือจำแนกข้อมูลต่างๆ ส่งผลให้เครื่องคอมพิวเตอร์สามารถจัดระเบียบการจัดเก็บข้อมูลได้ด้วยตนเอง โดยไม่ต้องอาศัยแรงงานมนุษย์เพื่อควบคุมการทำงาน แต่ในกรณีที่ข้อมูลเหล่านั้นเป็นความรู้หรือประสบการณ์ที่ต้องอาศัยการตีความก่อน เครื่องคอมพิวเตอร์จะไม่สามารถทำความเข้าใจได้อย่างลึกซึ้งคงทำได้แค่เพียงจัดเก็บสิ่งเหล่านี้ในลักษณะของข้อมูลเท่านั้น

1.4.1.3 ระบบการทำแบบวนซ้ำ

การทำงานที่มีรูปแบบเดิมๆวนซ้ำไปซ้ำมา อาจส่งผลต่อการทำงานและการตัดสินใจของมนุษย์ได้ เนื่องจากมนุษย์มีข้อจำกัดทางร่างกาย เมื่อทำงานเดิมๆ ในระยะเวลาหนึ่งอาจเกิดข้อผิดพลาด เกิดความเหนื่อยล้าทางร่างกายและความเบื่อหน่ายทางจิตใจ ซึ่งส่งผลเสียต่อระบบการทำงานทำให้ขาดความต่อเนื่อง เพราะต้องใช้ระยะเวลาในการทำงานเพิ่มมากขึ้น แต่สำหรับเครื่องคอมพิวเตอร์ที่มีปัญญาประดิษฐ์เหล่านี้มักนำไปใช้กับเครื่องจักรกลในสายการผลิตด้านต่างๆ เช่น อุปกรณ์อิเล็กทรอนิกส์ รถยนต์และอุตสาหกรรมอาหาร เป็นต้น นอกจากนี้คุณภาพงานที่ได้ ยังมีความสม่ำเสมอด้วยไม่แปรผันเหมือนการใช้แรงงานมนุษย์ ถึงแม้ว่าการใช้ปัญญาประดิษฐ์ในลักษณะนี้อาจต้องลงทุนสูง แต่เมื่อเทียบกับผลที่ได้ในระยะยาวถือว่าคุ้มค่ากับการลงทุน

นอกจากหน้าที่ของปัญญาประดิษฐ์ที่มีจุดเด่นในการนำมาใช้แทนแรงงานมนุษย์แล้ว ยังมีความสามารถของปัญญาประดิษฐ์ที่แตกต่างจากเครื่องคอมพิวเตอร์ทั่วไป ซึ่งเป็นส่วนหนึ่งที่ทำให้เห็นได้อย่างชัดเจนว่า เหตุใดปัญญาประดิษฐ์จึงมีความสำคัญและมีประสิทธิภาพมากกว่าและการใช้เครื่องคอมพิวเตอร์ ข้อเปรียบเทียบระหว่างเครื่องคอมพิวเตอร์ทั่วไปกับปัญญาประดิษฐ์แสดงในตารางที่ 1-1

จากที่กล่าวมาส่วนใหญ่ประสิทธิภาพของคอมพิวเตอร์ทั่วไปยังอิงตาม Algorithm ที่สร้างไว้ เนื่องจากการประมวลองค์ความรู้ของคอมพิวเตอร์ทั่วไปจะดำเนินตามลำดับขั้นตอนที่ถูกกำหนดไว้ใน Algorithm ที่สร้างขึ้นในโปรแกรม ในขณะที่คอมพิวเตอร์ปัญญาประดิษฐ์อาศัยการประมวลผลจากองค์ความรู้ที่เรียกว่า “การประมวลผลจากสัญลักษณ์ (Symbolic Processing)” เป็นการประมวลผลจากการเชื่อมโยงของข้อมูลในรูปสัญลักษณ์ที่แทนตัวอักษร คำ ข้อความ ประโยค หรือ ออบเจ็ค เพื่อใช้ค้นหาคำตอบที่เหมาะสมสำหรับแก้ไขปัญหาที่ต้องการ ดังนั้นการทำงานของคอมพิวเตอร์ทั่วไปกับคอมพิวเตอร์ปัญญาประดิษฐ์จึงมีความแตกต่างกันอย่างเห็นได้ชัด ทั้งนี้ควรพิจารณาความเหมาะสมของระบบงานว่าควรเลือกใช้คอมพิวเตอร์แบบใดเพื่อให้ได้ประโยชน์อย่างสูงสุดและไม่สิ้นเปลืองงบประมาณอย่างเปล่าประโยชน์

ตารางที่1-1 ข้อเปรียบเทียบระหว่างคอมพิวเตอร์ปัญญาประดิษฐ์และคอมพิวเตอร์ทั่วไป

ข้อเปรียบเทียบ	คอมพิวเตอร์ปัญญาประดิษฐ์	คอมพิวเตอร์ทั่วไป
การประมวลผล	ประมวลผลด้วยวิธีการวิเคราะห์รูปแบบสัญลักษณ์	ประมวลผลตาม Algorithm ที่สร้างไว้
ข้อมูลนำเข้า	ข้อมูลนำเข้าไม่จำเป็นต้องสมบูรณ์	ข้อมูลนำเข้าต้องมีความสมบูรณ์
วิธีการค้นหา	ใช้วิธีการที่ไม่อาศัยกฎเกณฑ์ตายตัวที่เรียกว่า Heuristic	ใช้วิธีการตามรูปแบบของ Algorithm
จุดมุ่งหมาย	การได้มาซึ่งองค์ความรู้	การได้มาซึ่งข้อมูล และสารสนเทศ
การให้เหตุผล	สามารถให้เหตุผลได้	ไม่สามารถหาเหตุผลได้
การตัดสินใจ	สามารถตัดสินใจได้ด้วยตนเอง	ต้องอาศัยมนุษย์ช่วยในการตัดสินใจหากอยู่นอกเหนือ Algorithm
การวิเคราะห์ข้อมูล	วิเคราะห์ข้อมูลอย่างสมเหตุสมผล	วิเคราะห์ข้อมูลตาม Algorithm
การเรียนรู้	สามารถเรียนรู้ได้ด้วยตนเอง	ไม่สามารถเรียนรู้ได้ด้วยตนเอง

ทั้งหมดที่กล่าวมานี้เป็นการเปรียบเทียบการทำงานของปัญญาประดิษฐ์กับมนุษย์ และคอมพิวเตอร์ทั่วไป ซึ่งในความเป็นจริงยังมีปัจจัยอีกหลายอย่างที่ปัญญาประดิษฐ์ไม่สามารถตอบสนองได้ดีเท่ากับมนุษย์ เนื่องจากมนุษย์แต่ละคนมีความรู้ ความสามารถและประสบการณ์ที่แตกต่างกัน ในบางกรณีผู้ที่มีประสบการณ์มากอาจเข้าใจปัญหาบางอย่างได้ลึกซึ้ง และหาวิธีแก้ไขได้อย่างรวดเร็ว อย่างไรก็ตามปัญญาประดิษฐ์ยังมีข้อจำกัดในการทำงานหลายๆด้าน ดังนี้

1. ปัญญาประดิษฐ์ไม่มีความคิดริเริ่มสร้างสรรค์ จึงไม่สามารถคิดค้นหรือสร้างสิ่งต่างๆ โดยปราศจากองค์ความรู้ หรือข้อมูลได้ แต่มนุษย์มีจินตนาการที่เป็นตัวขับเคลื่อนให้คิดค้นสิ่งใหม่ๆ ที่อยู่นอกเหนือจากข้อมูลหรือองค์ความรู้ได้อีกทั้งยังมีสภาพแวดล้อมที่เป็นแรงผลักดันและแรงบันดาลใจ ในขณะที่ปัญญาประดิษฐ์ไม่สามารถตอบสนองต่อสิ่งเหล่านี้ได้

2. ปัญญาประดิษฐ์อาศัยข้อมูล และสารสนเทศเป็นองค์ความรู้ในการตัดสินใจโดยต้องแปลข้อมูลก่อนนำไปใช้งานแต่มนุษย์สามารถเข้าใจข้อมูล หรือสารสนเทศนั้นได้เลย เนื่องจากมีประสบการณ์ในการเรียนรู้ข้อมูล ทำให้เข้าใจได้ง่ายกว่า ในบางกรณีสัญชาตญาณหรือความคุ้นเคยก็เป็นอีกปัจจัยหนึ่งที่ทำให้การรับรู้ของมนุษย์สามารถตอบสนองต่อข้อมูลได้อย่างรวดเร็ว

3. ปัญญาประดิษฐ์จะถูกพัฒนาให้มีหน้าที่เฉพาะด้านใดด้านหนึ่ง หากเป็นการทำงานที่นอกเหนือจุดประสงค์แล้ว การให้เหตุผลอาจมีประสิทธิภาพไม่เทียบเท่ากับมนุษย์ ในขณะที่มนุษย์สามารถรับรู้ระบบการทำงานในหลายๆด้านได้ตามประสบการณ์ ถึงแม้จะมีข้อจำกัดเรื่องความจำก็ตาม

1.4.2 ประโยชน์ของปัญญาประดิษฐ์

การนำปัญญาประดิษฐ์มาใช้ในระบบงานต่างๆ ขององค์กร นอกจากจะช่วยเพิ่มประสิทธิภาพของระบบงานที่ต้องการได้แล้ว ปัญญาประดิษฐ์ยังส่งผลต่อการทำงานในภาพรวมขององค์กรได้เป็นอย่างดี และยังสร้างประโยชน์ในด้านต่างๆอีกมากมาย ได้แก่

- ช่วยให้กระบวนการแก้ไขปัญหาทำได้อย่างรวดเร็ว และมีความสอดคล้องกันมากขึ้น
- ช่วยแก้ไขปัญหาต่างๆในกรณีที่ยากเกินความสามารถของมนุษย์ และเครื่องคอมพิวเตอร์ทั่วไป
- รองรับข้อมูลและองค์ความรู้จำนวนมากได้
- สามารถจัดเก็บข้อมูล และองค์ความรู้ได้แบบถาวร
- การเผยแพร่องค์ความรู้ภายในองค์กรทำได้ง่ายและทั่วถึง
- การวิเคราะห์ปัญหาทำได้ละเอียดรอบคอบ โดยเฉพาะปัญญาประดิษฐ์ที่ถูกพัฒนามาใช้งานเฉพาะด้าน
- มีความยืดหยุ่นต่อสถานการณ์และตอบสนองได้อย่างมีประสิทธิภาพ
- การควบคุมเครื่องจักรกลและเครื่องคอมพิวเตอร์ทำได้ง่ายขึ้น

คุณสมบัติ ข้อจำกัด และประโยชน์ของปัญญาประดิษฐ์เป็นอีกสิ่งหนึ่งที่แสดงให้เห็นว่าประสิทธิภาพการทำงานของปัญญาประดิษฐ์นั้นมีความสำคัญต่อระบบงานในองค์กรมากเพียงใด อีกทั้งยังเป็นปัจจัยสำคัญที่ชี้ให้เห็นได้อย่างชัดเจนว่าการนำปัญญาประดิษฐ์มาสนับสนุนงานในด้านต่างๆ จะช่วยให้ทราบถึงทิศทางความก้าวหน้าขององค์กรได้ในระดับหนึ่งด้วย นอกจากนี้ยังเป็นแนวทางในการพัฒนาปัญญาประดิษฐ์ให้มีความสามารถมากขึ้น และนำไปประยุกต์ใช้งานร่วมกับแขนงวิชาต่างๆได้อย่างมีประสิทธิภาพ

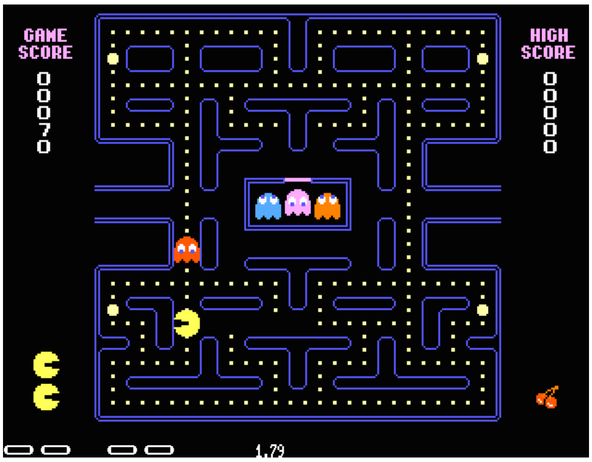
1.5 การนำปัญญาประดิษฐ์มาประยุกต์ใช้งาน

ขอบเขตในการประยุกต์ใช้งานทางปัญญาประดิษฐ์ไม่ได้จำกัดเฉพาะเพียงในสาขาที่เกี่ยวข้องกับปัญญาประดิษฐ์เท่านั้นแต่ยังขยายไปถึงสิ่งที่เกี่ยวข้องกับพฤติกรรมความชาญฉลาด ในหัวข้อนี้จะนำเสนอสาขาต่างๆ ที่ประสบผลสำเร็จในการพัฒนาและการประยุกต์ใช้งาน ดังต่อไปนี้

1.5.1 เกม (Game)

เป็นการนำปัญญาประดิษฐ์ไปใช้ในซอฟต์แวร์ด้านบันเทิง โดยส่วนใหญ่จะเป็นการนำไปใช้ในการเพิ่มความเสมือนจริงให้กับระบบการทำงานของเกมให้มีระดับความยากมากขึ้น เป็นการเพิ่มความท้าทายให้กับผู้เล่น ซึ่งการเพิ่มประสิทธิภาพในด้านนี้จะช่วยให้ผู้เล่นรู้สึกเหมือนกับเกมมีระบบการคิดเป็นของตนเอง โต้ตอบ ตอบสนองการสั่งการของผู้ใช้โดยราวกับมีตัวตน หรือมีมนุษย์ควบคุมอยู่ การพัฒนาของปัญญาประดิษฐ์อาจพัฒนาไปพร้อมๆกับการพัฒนาเกมเนื่องจากเป็นการนำเสนอลักษณะที่เกี่ยวเนื่องกับปัญหาขนาดเล็ก และมีกฎที่ชัดเจน ง่ายต่อการนำเสนอด้วยรูปแบบของสัญลักษณ์ ซึ่งคุณลักษณะที่กล่าวมานี้ ทำให้เกมเป็นแบบทดสอบที่ดีเยี่ยมทางปัญญาประดิษฐ์ทั้งในแง่แนวคิด และแง่เทคนิค ส่งผลให้เกมและปัญญาประดิษฐ์ต่างพัฒนาตนเองไปอย่างมาก Deep Blue เป็นโปรแกรมของ IBM ตัวแรกที่พิชิต Garry Kasparov (แชมป์โลกหมากรุกในปี 1997ด้วยคะแนน 3.5 ต่อ 2.5 โดย Kasparov ยังกล่าวยอมรับว่า “นี่เป็นความฉลาดชนิดใหม่” ที่โปรแกรมเหล่านี้สามารถเรียนรู้และหาทางแก้ไขสถานการณ์ต่างๆได้อย่างดีเยี่ยมในปัจจุบันวงการของซอฟต์แวร์เกมจำเป็นต้องมีระบบปัญญาประดิษฐ์ที่ดี เพื่อใช้เป็นจุดเด่นในการนำเสนอความน่าสนใจของเกม เพื่อให้อารมณ์ในการเล่นเกมมีความเสมือนจริง ราวกับผู้เล่นกำลังแข่งขันกับสิ่งที่หยั่งรู้และตัดสินใจได้รวดเร็วเทียบเคียงกับมนุษย์ ส่งผลให้เกมมีความสนุกสนานมากขึ้นด้วย ตัวอย่างของเกมที่ใช้ปัญญาประดิษฐ์แสดงดังภาพที่ 1-12

(ก) เกมหมากรุก



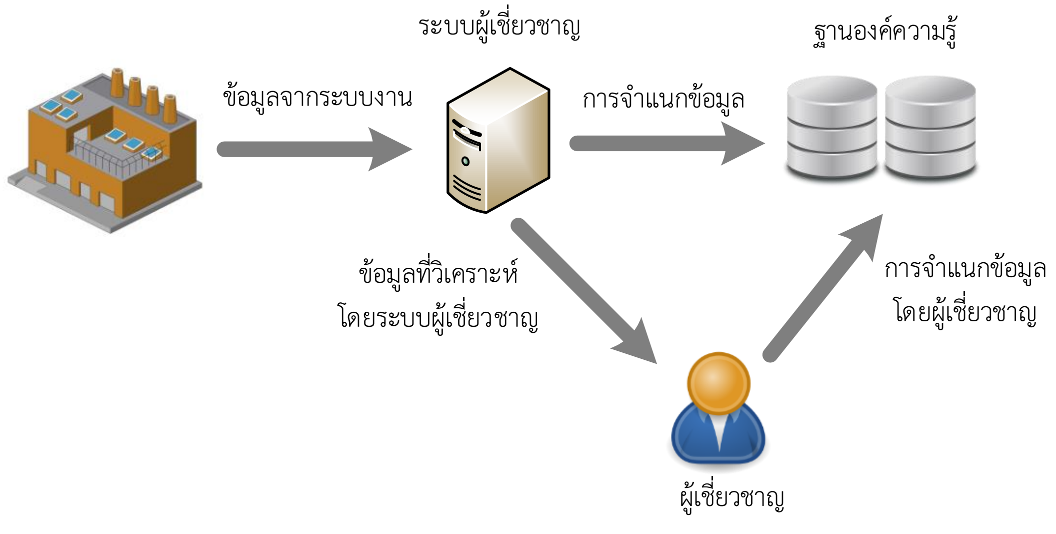
(ข) เกม Pacman

ภาพที่ 1-12 ตัวอย่างเกมที่นำปัญญาประดิษฐ์มาประยุกต์ใช้เพื่อเพิ่มความน่าสนใจ

1.5.2 ระบบผู้เชี่ยวชาญ (Expert System : ES)

เป็นอีกระบบหนึ่งที่พัฒนามาจากปัญญาประดิษฐ ซึ่งรู้จักกันอย่างกว้างขวางและนำไปใช้งานอย่างแพร่หลาย โดยระบบจะช่วยในการตัดสินใจเกี่ยวกับปัญหาหรือข้อมูลที่ซับซ้อน จึงต้องอาศัยองค์ความรู้ต่างๆที่เกี่ยวข้อง เพื่อใช้เป็นข้อมูลสนับสนุนในการแก้ไขปัญหาที่เกิดขึ้น จนได้คำตอบที่ดีที่สุดและเหมาะสมกับสถานการณ์ ระบบผู้เชี่ยวชาญเป็นการนำปัญญาประดิษฐ์มาใช้ในอีกรูปแบบหนึ่ง โดยให้เครื่องคอมพิวเตอร์ทำหน้าที่เสมือนเป็นที่ปรึกษา ซึ่งเดิมทีอาจใช้แรงงานมนุษย์ แต่เนื่องจากปัญหามีความซับซ้อนข้อมูลแวดล้อมมีปริมาณมาก การอาศัยแรงงานมนุษย์อาจทำได้ล่าช้างบประมาณที่ใช้สำหรับจ้างผู้เชี่ยวชาญก็ค่อนข้างสูง การนำปัญญาประดิษฐ์มาประยุกต์ใช้เป็นระบบผู้เชี่ยวชาญจึงเป็นอีกหนทางเลือกหนึ่งที่สามารถแก้ไขปัญหาดังกล่าวได้

ด้วยเหตุผลนี้จึงมีการพัฒนาระบบการจัดการองค์ความรู้ ที่เรียกว่า ระบบฐานองค์ความรู้ (Knowledge Base System) เพื่อนำมาใช้สนับสนุนด้านการจัดการองค์ความรู้ที่มีอยู่จำนวนมาก ซึ่งเป็นตัวสนับสนุนผู้เชี่ยวชาญให้สามารถตัดสินใจแก้ปัญหาที่ซับซ้อนได้ โดยมีปัญญาประดิษฐ์เป็นตัวขับเคลื่อนให้ระบบผู้เชี่ยวชาญสามารถแสดงศักยภาพในการตัดสินใจจากฐานความรู้ที่มีได้อย่างถูกต้องและแน่นอน ซึ่งสามารถทดแทนแรงงานมนุษย์ได้เป็นอย่างดี ระบบผู้เชี่ยวชาญถูกนำไปใช้ในสาขาวิชาชีพต่างๆเช่น ระบบผู้เชี่ยวชาญในการวิจัยทางการแพทย์ (MYCIN) และเครื่องมือวิเคราะห์ทางธรณีวิทยา (PROSPECTOR) เป็นต้น ภาพที่ 1-13 แสดงตัวอย่างกระบวนการทำงานของระบบผู้เชี่ยวชาญภายในองค์กร



ภาพที่ 1-13 ตัวอย่างกระบวนการทำงานของระบบผู้เชี่ยวชาญภายในองค์กร

1.5.3 การให้เหตุผลเชิงกรณีศึกษา (Case-based Resoning)

การให้เหตุผลเชิงกรณีศึกษา (Case-based Reasoning) หรือเรียกสั้นๆว่า CBR เป็นการประยุกต์ใช้งานปัญญาประดิษฐ์อีกวิธีหนึ่ง ที่อาศัยพื้นฐานทางความคิดและการเรียนรู้จากประสบการณ์ของมนุษย์ เพื่อเป็นแนวทางในการพัฒนา โดยจะอ้างอิงวิธีการแก้ไขปัญหาต่างจากกรณีศึกษาที่เคยผ่านมา CBR เป็นระบบที่มีกระบวนการแก้ไขปัญหาด้วยการสรุปความ และหาเหตุผลจากปัญหาหรือกรณีศึกษาเดิมที่เคยผ่านมา CBR เป็นระบบที่มีกระบวนการแก้ไขปัญหาด้วยการสรุปความ และหาเหตุผลจากปัญหาหรือกรณีศึกษาเดิมที่เคยพบในอดีต กรณีศึกษาเหล่านี้จะถูกจัดเก็บเป็นองค์ความรู้ไว้ เมื่อระบบต้องเผชิญกับปัญหา จะนำกรณีศึกษาที่มีความใกล้เคียงกันมาเป็นแนวทางในการแก้ไขปัญหาดังกล่าว โดยกรณีศึกษาที่จัดเก็บไว้อาจนำมาใช้แก้ปัญหาได้โดยตรง หรืออาจต้องดัดแปลง และเรียนรู้เพื่อหาวิธีการแก้ไขปัญหาดังกล่าวโดยกรณีศึกษามากที่สุด หลังจากระบบแก้ไขปัญหาเรียบร้อยแล้ว จึงดำเนินการจัดเก็บไว้ในฐานกรณีศึกษาทุกครั้ง โดยเฉพาะกรณีศึกษาใหม่ๆเพื่อเป็นแหล่งองค์ความรู้สำหรับใช้งานในอนาคตต่อไป

จากที่กล่าวมา CBR นิยมนำมาใช้กับรูปแบบงานที่มีโอกาสเผชิญกับปัญหาที่คล้ายกันอยู่เสมอ เพื่อให้ระบบหรือซอฟต์แวร์สามารถแก้ไขปัญหาได้ด้วยตนเอง โดยไม่ต้องอาศัยแรงงานจากมนุษย์ ทั้งนี้ระบบจะทำการเรียนรู้จากกรณีศึกษาที่จัดเก็บไว้ในฐานองค์ความรู้ การประยุกต์ใช้งานระบบ CBR มีอยู่หลายประเภท เช่น ระบบการตรวจจับข้อผิดพลาดในสายงานผลิตภายในโรงงาน ซึ่งเป็นการตรวจสอบโดยอาศัยกรณีศึกษา เพื่อลดข้อผิดพลาดที่อาจเกิดขึ้นซ้ำ ๆ ให้น้อยลงจากการเรียนรู้และหาแนวทางป้องกันจากฐานกรณีศึกษาที่มี เป็นต้น การแก้ไขปัญหาต่างๆ ด้วยกรณีศึกษา ขึ้นอยู่กับประสบการณ์ที่ผ่านมาว่าเผชิญกับรูปแบบปัญหามากน้อยเพียงใด ยิ่งพบปัญหามากก็ยิ่งมีกรณีศึกษาที่หลากหลาย การแก้ไขปัญหาก็ย่อมมีประสิทธิภาพมากขึ้น เช่นเดียวกับมนุษย์ที่มีประสบการณ์มากย่อมได้เปรียบผู้ที่มีประสบการณ์น้อยนั่นเอง

1.5.4 กลไกการเรียนรู้ (Machine Learning)

เป็นการนำปัญญาประดิษฐ์มาประยุกต์ใช้ในด้านการเรียนรู้ เพื่อเพิ่มประสิทธิภาพให้เครื่องจักรกลและเครื่องคอมพิวเตอร์มีความรู้ในระบบงานที่มีหน้าที่รับผิดชอบ ซึ่งเป็นการเรียนรู้จากสถานการณ์ หรือองค์ความรู้ใหม่ที่เกิดขึ้นตามสภาพแวดล้อมต่างๆโดยอาจเป็นสิ่งที่เครื่องคอมพิวเตอร์ไม่เคยพบมาก่อน ทำให้การทำงานมีประสิทธิภาพมากขึ้น อีกทั้งยัดลดความผิดพลาดลงได้อีกทางหนึ่ง เนื่องจากเครื่องคอมพิวเตอร์หรือเครื่องจักรมีองค์ความรู้ใหม่ที่มีข้อมูลครอบคลุมทุกสถานการณ์หรือปัญหา ปัญญาประดิษฐ์เป็นตัวช่วยให้ระบบของเครื่องคอมพิวเตอร์หรือเครื่องจักรเกิดการเรียนรู้สิ่งต่างๆ ได้ด้วยตนเองโดยไม่ต้องอาศัยการสั่งจากมนุษย์ กลไกการเรียนรู้มีจุดมุ่งหมายเพื่อค้นคว้าและเข้าถึงฐานความรู้เพื่อเพิ่มความรู้จากสถานการณ์ที่เกิดขึ้นได้ เช่น การเรียนรู้ในโปรแกรมหมากรุกให้สามารถคาดคะเนเกมการเล่นได้ ทำให้มีรูปแบบการเล่นบนกระดานที่น่าสนใจมากขึ้น คือมีรูปแบบในการแก้ไขสถานการณ์ที่หลากหลายมากขึ้น ทั้งแบบที่เพิ่มโอกาสชนะให้สูงขึ้น หรือลดโอกาสแพ้ให้น้อยลง ลักษณะนี้เป็นเพียง

ภาพที่ 1-14 ตัวอย่างการนำปัญญาประดิษฐ์มาใช้พัฒนาการเรียนรู้ให้กับเครื่องจักรกล

กลยุทธ์พื้นฐาน การวิจัยทางด้านนี้จึงมีลักษณะที่เรียนรู้จากผลสำเร็จที่ผ่านมาเพื่อค้นหารูปแบบหรือทฤษฎีที่ดีขึ้นต่อไป ภาพที่ 1-14 แสดงตัวอย่างการนำกลไกการเรียนรู้ของเครื่องจักรไปประยุกต์ใช้กับแขนกลที่ใช้ในการเชื่อมจุดต่อของตัวถังรถยนต์

1.5.5 การคำนวณเชิงวิวัฒนาการ (Evolutionary Computation)

เป็นการเรียนรู้อีกชนิดหนึ่งที่ประยุกต์ใช้งานกับเครื่องจักรหรือเครื่องคอมพิวเตอร์ เพื่อเพิ่มประสิทธิภาพในการเรียนรู้ให้สูงขึ้น โดยเลียนแบบวิวัฒนาการของสิ่งมีชีวิต การเรียนรู้รูปแบบนี้ทำงานอยู่บนพื้นฐานทฤษฎีวิวัฒนาการของCharles Darwin ที่นำเสนอเมื่อปี ค.ศ.1858 โดยนำมาประยุกต์ใช้กับการเรียนรู้ของเครื่องจักร เช่น ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm: GA) และ โปรแกรมเชิงพันธุกรรม (Genetic Programming: GP) เป็นต้น เป็นทฤษฎีการดำรงอยู่ของสิ่งมีชีวิตที่เกิดการเปลี่ยนแปลงตลอดเวลา ส่งผลให้สิ่งมีชีวิตมีวิวัฒนาการ (Evolution) โดยปรับเปลี่ยนลักษณะต่างๆทั้งทางกายภาพ และพฤติกรรมให้มีความเหมาะสม เพื่อให้สามารถอยู่รอดในสภาพแวดล้อมขณะนั้นได้

การคำนวณเชิงวิวัฒนาการจะมุ่งเน้นในการศึกษาด้านพันธุกรรมของสิ่งมีชีวิต เพื่อเป็นแนวทางในการพัฒนาการเรียนรู้ของเครื่องจักร เนื่องจากสิ่งมีชีวิตจะมีลักษณะทางพันธุกรรม (Genetic Character) ที่สามารถถ่ายทอดจากพ่อแม่ไปสู่ลูกหลานได้ด้วยหน่วยพันธุกรรมที่เรียกว่า “ยีน (Gene)” สำหรับลักษณะทางพันธุกรรมทั่วไป เช่น สีผม สีผิว สติปัญญา และความสูง เป็นต้น ยีนเหล่านี้เป็นส่วนหนึ่งของ DNA ที่ปรากฏอยู่บนตำแหน่งโครงสร้างของโครโมโซม (Chromosome) โดยมีหน้าที่ถ่ายทอดและควบคุมลักษณะทางพันธุกรรม ด้วยเหตุนี้รูปแบบของการวิวัฒนาการของสิ่งมีชีวิตจึงถูกนำมาประยุกต์ใช้ในกระบวนการเรียนรู้ ที่สามารถพัฒนาสติปัญญาของเครื่องจักรให้มีกระบงนการคิดที่ดีขึ้นและเรียนรู้สิ่งต่างๆได้จากการเปลี่ยนแปลงดังกล่าว การเรียนรู้ในการแก้ไขปัญหาต่างๆ จึงมีประสิทธิภาพสูงขึ้น การจัดการกับปัญหาจะพัฒนาขึ้นจากเดิม ช่วยให้รูปแบบการเรียนรู้และกระบวนการคิดใกล้เคียงกับมนุษย์มากยิ่งขึ้น

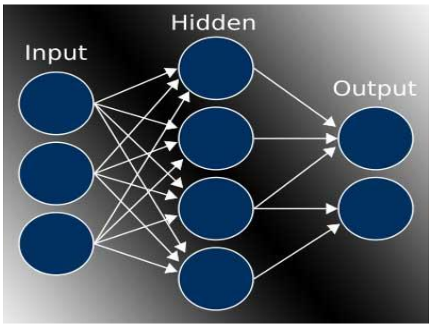
1.5.6 ระบบโครงข่ายใยประสาท (Neural Network)

เป็นการประยุกต์ใช้ปัญญาประดิษฐ์อีกแขนงหนึ่งที่มีสถาปัตยกรรมที่แตกต่างจากแขนงอื่นๆ โดยเป็นสาขาที่พยายามใช้โครงสร้างการทำงานที่มีลักษณะเลียนแบบระบบประสาทในสมองของมนุษย์ดังภาพที่ 1-15(ก) ซึ่งแต่ละส่วนของโครงข่ายใยประสาทจะเชื่อมต่อเข้าด้วยกันและทำงานคล้ายกับระบบประสาทที่มีการประมวลผลได้อย่างรวดเร็ว การประมวลผลองค์ความรู้นี้เรียกว่า การประมวลผลแบบขนาน (Parallel Processing) ซึ่งมีความสามารถในการประมวลองค์ความรู้จำนวนมากได้ในเวลาเดียวกัน ระบบโครงข่ายใยประสาทนี้สามารถเรียนรู้และจดจำสารสนเทศในรูปแบบที่ปัญญาประดิษฐ์ทั่วไปไม่สามารถทำได้ ซึ่งก็คือ “ประสบการณ์” เนื่องจากมีการเลียนแบบจากโครงข่ายประสาทของมนุษย์ทำให้สามารถเชื่อมโยงข้อเท็จจริงเข้าด้วยกัน เพื่อใช้หาข้อสรุปโดยอาศัยประสบการณ์ที่ได้จัดเก็บไว้มาเป็นข้อมูลในการเรียนรู้ และทำความเข้าใจจนได้ความเกี่ยวข้องสัมพันธ์กันระหว่างองค์ความรู้ทั้งหมด ซึ่งสามารถนำไปปรับปรุงองค์ความรู้เดิมให้มีประสิทธิภาพมากยิ่งขึ้น เพื่อความทันสมัยและรองรับสถานการณ์ที่จะเกิดขึ้นในอนาคต

ระบบโครงข่ายใยประสาทจะมีประสิทธิภาพสูงเมื่อนำมาใช้แก้ปัญหาที่มีลักษณะเป็นชุดข้อมูลหรือมีรูปแบบที่ชัดเจน โดยนำชุดข้อมูลที่รู้คำตอบแล้วมาหนึ่งชุด นำมาใช้เป็นแนวทางไปสู่การแก้ไขปัญหาที่ยังไม่ทราบรูปแบบ เช่น สร้างระบบโครงข่ายใยประสาทที่แยกแยะเพศได้จากรูปภาพ โดยรูปภาพใดที่เป็นผู้หญิงก็คือ “ใช่” นอกเหนือจากนั้นก็คือ “ไม่” และเมื่อใส่ชุดข้อมูลลงไปจะทำให้ระบบโครงข่ายใยประสาทวิเคราะห์ว่าชุดตัวอย่างนั้น สามารถระบุรูปแบบหน้าตาผู้หญิงได้หรือไม่ โดยจะหาตัวแปรที่ซ่อนภายในออกมาแล้วจึงแสดงผลลัพธ์ ด้วยเหตุนี้ระบบโครงข่ายใยประสาทจึงต้องการชุดข้อมูลใหม่เพิ่มลงไป แม้ผลลัพธ์จะไม่ถูกต้องทุกครั้งก็ตาม แต่ถ้ามีความแม่นยำ 95% ก็ถือว่ายอมรับได้

ระบบโครงข่ายใยประสาทประกอบด้วยโหนด (Node) ต่างๆ ที่เปรียบเสมือนเป็นเซลล์ประสาทของมนุษย โดยแต่ละโหนดจะเชื่อมโยงกันเป็นเครือข่าย และถูกแบ่งออกเป็นระดับชั้น (Layer) ในแต่ละระดับชั้นจะมีหน้าที่แตกต่างกันโดยอาศัยองค์ประกอบอื่นๆ เพื่อเริ่มกระบวนการ เช่น ข้อมูลการนำเข้า (Input) และฟังก์ชันการแปลง (Transformation Function) เป็นต้น ดังภาพที่ 1-15(ข)

(ก) โครงข่ายใยประสาทในสมอง



(ข) โครงข่ายใยประสาทเทียม

ภาพที่ 1-15 ตัวอย่างโครงสร้างโครงข่ายใยประสาทในสมองและโครงข่ายใยประสาทเทียม

1.5.7. ตัวแทนปัญญา (Intelligent Agent)

ตัวแทนปัญญา (Intelligent Agent) เป็นอีกหนึ่งเทคโนโลยีของปัญญาประดิษฐ์ที่ได้รับการพัฒนาอย่างกว้างขวาง โดยนำปัญญาประดิษฐ์มาประยุกต์ใช้งาน ในอนาคตจะกลายเป็นเทคโนโลยีที่สำคัญทางด้านระบบสารสนเทศ และงานด้านการวิจัยที่เกี่ยวกับคอมพิวเตอร์ เนื่องจากเป็นการพัฒนาตัวแทนปัญญาที่สามารถแก้ไขปัญหาต่างๆ และเพิ่มประสิทธิภาพในการทำงานได้ ตัวแทนในที่นี้อาจเรียกว่า “Agent (เอเจนต์)” หรือ “โปรแกรมตัวแทน” ซึ่งหมายถึง ผู้ที่ได้รับมอบหมายให้ทำหน้าที่ตามที่มนุษย์ต้องการ ซึ่ง Agent สามารถรับข้อมูลต่างๆ ได้อย่างเป็นลำดับจากสภาพแวดล้อมผ่านทางตัว Sensors หรือระบบที่สร้างไว้เพื่อการรับรู้ โดยสามารถตอบสนองกลับด้วยการกระทำแบบต่างๆ ตามรูปแบบของการโต้ตอบที่ระบบกำหนดไว้ ภาพที่ 1-16 แสดงโปรแกรมตัวอย่างที่สามารถโต้ตอบกับมนุษย์ได้

ตัวแทนปัญญา เป็นหน่วยของซอฟต์แวร์ที่สามารถดำเนินการหรือประมวลผลบางอย่างให้กับผู้ใช้หรือโปรแกรมอื่นได้อย่างเป็นอิสระ โดยอาศัยองค์ความรู้ที่สร้างขึ้นเพื่อใช้ในการรับรู้ข้อมูลจากสภาพแวดล้อม ตัวแทนปัญญามีวัตถุประสงค์หลัก คือ “การดำเนินการบางอย่างแทนมนุษย์ด้วยความรวดเร็ว และมีประสิทธิภาพมากกว่าที่มนุษย์ดำเนินการเอง” ด้วยเหตุนี้จึงมีการพัฒนาให้ตัวแทนปัญญาสามารถรับรู้สภาพแวดล้อมผ่านทางอุปกรณ์ต่างๆ เช่น หุ่นยนต์ที่มีตัว Sensors สำหรับข้อมูลต่างๆ จากสิ่งแวดล้อมได้ โดยอาศัยกล้อง หรือ Sensors วัดอุณหภูมิ เป็นต้น ทั้งนี้ขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน การพัฒนาตัวแทนปัญญาเหล่านี้ด้วยอุปกรณ์เสริมประสิทธิภาพในการเรียนรู้จะช่วยให้การตอบสนองมีความใกล้เคียงมนุษย์มากยิ่งขึ้น เปรียบเสมือนการสร้างอวัยวะต่างๆ ให้กับคอมพิวเตอร์หรือเครื่องจักร เช่น ตา หรือ หู ที่ใช้ Sensors แบบต่างๆ ในการรับรู้ เป็นต้น ตัวแทนปัญญาอาจถูกพัฒนาในรูปแบบของเครื่องจักรกล หุ่นยนต์ หรือซอฟต์แวร์ก็ได้ ขึ้นอยู่กับความต้องการในการนำไปประยุกต์ใช้งาน

ภาพที่ 1-16 ตัวอย่างซอฟต์แวร์ของตัวแทนปัญญาที่ประยุกต์ใช้ในระบบโต้ตอบกับมนุษย์

1.5.8 การวางแผน (Planning)

การวางแผน (Planning) เป็นอีกเทคนิคหนึ่งทางด้านปัญญาประดิษฐ์ ที่ใช้สำหรับกระบวนการวิเคราะห์ตัดสินใจล่วงหน้า เพื่อใช้ออกแบบแผนการที่จะนำไปสู่จุดมุ่งหมายตามที่ต้องการ การวางแผนจะเป็นลำดับขั้นตอนที่ใช้ในการแก้ไขปัญหาโดยต้องมีความสอดคล้องกันระหว่างการกระทำและผลลัพธ์ การวางแผนอาจถูกนำไปใช้ควบคู่กับตัวแทนปัญญา (AI) เพื่อให้ตัวแทนปัญญาทำหน้าที่เป็นผู้วางแผน (Planner) ปัญหาได้ การวางแผนปัญหา (Planning Problem) เป็นจุดเริ่มต้นของการออกแบบแผนการด้วยกระบวนการวิเคราะห์ตัดสินใจเพื่อรับมือกับปัญหา ซึ่งวิธีการค้นหา (Searching) ก็เป็นการวางแผนชนิดหนึ่ง เนื่องจากเป็นการหาเส้นทางเพื่อนำไปสู่การแก้ไขปัญหาตามที่ต้องการนั่นเอง

การวางแผนจำเป็นต้องคำนึงถึงสภาพแวดล้อมด้วย เนื่องจากเป็นปัจจัยที่สามารถส่งผลกระทบต่อการวางแผนได้ เช่น เวลา เป็นต้น สิ่งแวดล้อมเหล่านี้จำเป็นต้องมีความเหมาะสมกับการวางแผนการดำเนินการต่างๆ ตามแผนการจึงจะเป็นไปอย่างราบรื่น การวางแผนถูกนำมาประยุกต์ใช้งานในระบบต่างๆ ของปัญญาประดิษฐ์ ทั้งนี้เพื่อเพิ่มประสิทธิภาพในการดำเนินงาน หรือสนับสนุนความต้องการของระบบในรูปแบบต่างๆ เพื่อให้บรรลุเป้าหมายได้อย่างแท้จริง การวางแผนมักนำไปใช้กับระบบงานที่มีความซับซ้อน เพื่อให้ระบบสามารถขับเคลื่อนไปได้อย่างเป็นลำดับขั้นตอนที่ชัดเจน เช่น ระบบการขนส่งทางอากาศที่จำเป็นต้องอาศัยการเลือกเส้นทางและการวางแผนการขนส่งอย่างรอบคอบ เพื่อให้เครื่องบินแต่ละลำสามารถเดินทางไปยังสนามบินได้อย่างถูกต้อง ทำหน้าที่ของตนได้อย่างสมบูรณ์ตามเป้าหมาย และมีประสิทธิภาพสูงที่สุด เป็นต้น

1.5.9 การประมวลผลภาษาธรรมชาติ (Natural Language Processing : NLP)

การประมวลผลภาษาธรรมชาติ คือ รูปแบบการประมวลผลภาษาด้วยการวิเคราะห์จากเครื่องคอมพิวเตอร์ โดยมีจุดประสงค์ให้ผู้ใช้สามารถโต้ตอบกับคอมพิวเตอร์ได้โดยใช้ภาษาธรรมชาติ ซึ่งผู้ใช้เข้าใจเป็นอย่างดีโดยเครื่องคอมพิวเตอร์จะมีการจัดเก็บองค์ความรู้ที่จำเป็นสำหรับวิเคราะห์และแปลเสียงไว้เรียบร้อยแล้ว การประมวลผลภาษาธรรมชาติแบ่งเป็น 2 ประเภท ได้แก่

1) การทำความเข้าใจภาษาธรรมชาติ (Natural Language Understanding) เป็นกระบวนการที่ทำให้คอมพิวเตอร์สามารถวิเคราะห์คำสั่งหรือเสียงของผู้ใช้ และแปลเป็นข้อมูลที่คอมพิวเตอร์จะเข้าใจได้ จากนั้นจึงดำเนินตามคำสั่งดังกล่าว พร้อมทั้งโต้ตอบด้วยวิธีการแสดงผลแบบต่างๆ ซึ่งคอมพิวเตอร์จะแบ่งการทำงานของกระบวนการนี้ออกเป็น 2 ส่วน คือ ส่วนแรกจะทำหน้าที่ในการจดจำเสียง และจำแนกคำพูดออกตามรูปแบบของคำนั้น โดยไม่ต้องเข้าใจความหมาย ส่วนที่สองจะดำเนินการต่อจากการจดจำ ซึ่งจะทำหน้าที่ทำความเข้าใจกับคำที่จำแนกออกมาจากส่วนแรก เพื่อให้เข้าใจความหมายของคำดังกล่าวและสามารถตอบสนองได้อย่างถูกต้อง

2) การสร้างภาษาธรรมชาติ (Natural Language Generation) เป็นกระบวนการที่คอมพิวเตอร์ใช้สำหรับการสร้างภาษาธรรมชาติจากองค์ความรู้ที่รวบรวมไว้ เพื่อใช้โต้ตอบด้วยภาษาที่ผู้ใช้เข้าใจ โดยสามารถโต้ตอบได้หลายรูปแบบ เช่น ข้อความ และคำพูด เป็นต้น เทคโนโลยีการสร้างภาษาธรรมชาติที่สามารถสังเคราะห์เสียงได้ เรียกว่า “กระบวนการสังเคราะห์เสียง (Voice Synthesis)” ซึ่งเป็นการสร้างเสียงขึ้นใหม่จากคำพูดให้กลายเป็นเสียงอิเล็กทรอนิกส์ ที่มีเอกลักษณ์และคล้ายเสียงของมนุษย์มากที่สุด โดยสามารถนำมาใช้แปลงเสียงจากเสียงคนหนึ่งเป็นอีกเสียงคนหนึ่งได้ด้วย

การประมวลผลภาษาธรรมชาติสามารถนำไปประยุกต์ใช้กับโปรแกรมอื่นๆ ได้หลากหลาย เช่น การแปลงจากตัวอักษรเป็นคำพูดที่เปล่งออกมาเป็นภาษาให้กับคนตาบอดเข้าใจได้ เป็นต้น ทั้งนี้ขึ้นอยู่กับจุดประสงค์ในการนำไปใช้งาน จากที่กล่าวมาจะเห็นว่าปัญญาประดิษฐ์มีส่วนสำคัญในระบบงานประเภทนี้ ทั้งในการจดจำ เรียนรู้ และการจัดเก็บองค์ความรู้เสมือนกับคอมพิวเตอร์สามารถโต้ตอบกับมนุษย์ได้โดยตรง

1.5.10 หุ่นยนต์ (Robotic)

เป็นการประยุกต์ใช้งานปัญญาประดิษฐ์สำหรับพัฒนาเครื่องจักรกลให้มีความสามารถและฉลาดพอที่จะทำหน้าที่แทนมนุษย์ได้อย่างสมบูรณ์ ซึ่งเป็นการสร้างเครื่องจักรให้มีระบบการทำงานแบบอัตโนมัติที่มีความใกล้เคียงกับแรงงานมนุษย์ โดยทั่วไปนิยมใช้กับเครื่องจักรในโรงงานอุตสาหกรรม สำหรับทำงานในด้านต่างๆ แทนมนุษย์ เพื่อความปลอดภัยจากงานที่เสี่ยงอันตราย หรือเพื่อเพิ่มปริมาณการผลิต การประยุกต์ใช้ปัญญาประดิษฐ์กับเครื่องจักร ถือเป็นศาสตร์ที่เกี่ยวข้องกับการควบคุมพฤติกรรมของหุ่นยนต์ หรือเครื่องจักรในรูปแบบต่างๆ เช่น การวางแผน การเคลื่อนที่ การประกอบวัตถุ และการมองเห็นวัตถุ เป็นต้น ซึ่งจะนำไปสู่การบรรลุเป้าหมายที่ต้องการ โดยการควบคุมหุ่นยนต์ที่มีประสิทธิภาพและถูกต้องจะอาศัยองค์ความรู้ที่สร้างขึ้นผนวกเข้ากับระบบปัญญาประดิษฐ์ เช่น การเคลื่อนที่ไปหาเป้าหมายจะต้องคำนึงถึงความรู้ในด้านกลศาสตร์ จลศาสตร์ และการมองเห็น เป็นต้น หากมีวัตถุกีดขวางเส้นทางหุ่นยนต์จะใช้องค์ความรู้ดังกล่าวเพื่อหลบหลีกและหาเส้นทางที่เหมาะสม

ปัจจุบันศาสตร์ทางด้านนี้ได้รับความนิยมอย่างแพร่หลาย เนื่องจากสามารถนำไปประยุกต์ใช้กับองค์กร หรือระบบงานได้มากมาย ช่วยให้ประหยัดแรงงานหรือบุคลากร ลดอันตรายที่อาจเกิดขึ้น อำนวยความสะดวก เพิ่มประสิทธิภาพในการทำงาน และสร้างความบันเทิงให้กับมนุษย์ ในอนาคตเครื่องจักรหรือหุ่นยนต์เหล่านี้ อาจเป็นตัวแทนของมนุษย์ในการขับเคลื่อน และสร้างประโยชน์ได้ในทุกๆ องค์กร การพัฒนาของหุ่นยนต์ที่รู้จักกันอย่างแพร่หลาย เช่น ASIMO ชื่อหุ่นยนต์ต้นแบบที่ถูกพัฒนาขึ้นโดย HONDA ซึ่งมีรูปแบบการเคลื่อนไหวที่คล้ายกับมนุษย์ และสามารถตอบสนองพร้อมทั้งมีปฏิสัมพันธ์กับมนุษย์ได้ ดังแสดงในภาพที่ 1-17 เป็นต้น

ภาพที่ 1-17 หุ่นยนต์ ASIMO ที่ใช้ปัญญาประดิษฐ์ในการควบคุม

1.6 โปรแกรมภาษาที่ใช้ในปัญญาประดิษฐ์

มีโปรแกรมภาษาหลายชนิดที่เป็นผลมาจากการวิจัยทางปัญญาประดิษฐ์ โดยโปรแกรมภาษาที่ใช้สำหรับด้านปัญญาประดิษฐ์มีอยู่ 4 ชนิด ได้แก่ Logic Programming, Functional Programming, Rule-Based Programming และ Object-Oriented Programming

โปรแกรมภาษาหลักที่ใช้ทางด้านปัญญาประดิษฐ์ในปัจจุบัน ได้แก่ ภาษา Prolog และ ภาษา Lisp โดยภาษา Lisp ใช้กันอย่างกว้างขวางในอเมริกา ส่วนภาษา Prolog จะใช้ในแถบยุโรป และญี่ปุ่น นอกจากนี้ยังมีโปรแกรมภาษาพิเศษอื่นๆ ที่ใช้ในปัจจุบัน เช่น ภาษา OPS5 (Official Programming System Version 5) ซึ่งใช้สำหรับออกแบบระบบผู้เชี่ยวชาญ หรือภาษา MRS (Meta Reasoning System) เป็นต้น

1.6.1 Logic Programming

โครงสร้างทาง Logic เป็นพื้นฐานในการเขียนโปรแกรม ทุก Statement ในภาษานี้จะเขียนออกมาในรูปแบบ Logic ซึ่งภาษาใน Logic Programming ที่รู้จักกันดีคือ ภาษา Prolog โดยคำว่า Prolog มาจาก “Programmation en Logique” ในภาษาฝรั่งเศสซึ่งมีความหมายในภาษาอังกฤษว่า “Logic Programming” ภาษา Prolog จะใช้วิธีการทาง Logic แทนการกำหนดคำสั่งอย่างละเอียดให้กับคอมพิวเตอร์ ซึ่งพื้นฐานของภาษานี้มาจาก Predicate Calculus

1.6.2 Functional Programming

นำทฤษฎีทางคณิตศาสตร์มาใช้เป็นพื้นฐานในการเขียนโปรแกรม Functional Programming จะเขียนทุก Statement ในรูปของฟังก์ชัน ภาษาที่ใช้ทางด้านปัญญาประดิษฐ์ที่รู้จักกันดี คือ ภาษา Lisp (List Processing) เป็นภาษาที่ประมวลผลสัญลักษณ์ได้ดีและใช้โครงสร้างข้อมูลที่เป็นรายการ (List) ในการเขียนโปรแกรม อีกทั้งเป็นภาษาที่ใช้อย่างกว้างขวางในอเมริกา

1.6.3 Rule-Based Programming

มีลักษณะคล้ายกับ Logic Programming ที่นำโครงสร้าง Logic มาเขียนลงในโปรแกรม อย่างไรก็ตาม Rule-Based Programming มีข้อจำกัดมากกว่า Logic Programming ภาษาที่ใช้ใน Rule-Based Programming เช่น ภาษา Official Programming System Version 5 (OPS5) ที่มุ่งเน้นไปยังกฎที่เกี่ยวกับ IF-THEN โดย Rule-Based Programming จะเอื้อประโยชน์ให้แสดงผลได้หลายลักษณะตามระบบที่ใช้ รวมทั้งสามารถให้เหตุผลกับค่าข้อมูลที่มีความไม่แน่นอน (Uncertainty) ได้

1.6.4 Object-Oriented Programming

ใช้คุณสมบัติของ OO (Object-Oriented) ในการจัดแบ่งโปรแกรมออกเป็น คลาส (Class) และออบเจ็ค (Object) ภาษาที่รองรับรูปแบบการทำงานแบบ Object-Oriented Programming (OOP) ปัจจุบันมีค่อนข้างมาก เช่น C++, JAVA และ .NET เป็นต้น โปรแกรมลักษณะนี้กำลังเป็นที่นิยมอย่างแพร่หลายในปัจจุบัน

1.7 สรุป

ปัญญาประดิษฐ์ (Artificial Intelligence) หรือที่เรียกสั้นๆ ว่า “AI” เป็นศาสตร์แขนงหนึ่งของวิทยาการคอมพิวเตอร์ที่ใช้ในการพัฒนาเครื่องจักรกล ให้สามารถเรียนรู้เพื่อแก้ปัญหา โดยหาเหตุผลจากองค์ความรู้ที่มีอยู่ แล้วนำไปวิเคราะห์ จนได้ผลสรุปหรือผลลัพธ์ที่ต้องการ ปัญญาประดิษฐ์จึงเป็นศาสตร์ที่มีวัตถุประสงค์เพื่อใช้พัฒนาให้เครื่องจักรมีสมองและกระบวนการคิดเสมือนมนุษย์ มีประสิทธิภาพในการดำเนินงานต่างๆ ลดข้อผิดพลาดให้น้อยลง แก้ไขปัญหาและตัดสินใจด้วยความรวดเร็วเทียบเท่าหรือมากกว่ามนุษย์ ระบบการกระทำที่เหมือนมนุษย์ ระบบความคิดอย่างมีเหตุผล และระบบการกระทำอย่างมีเหตุผล

วิวัฒนาการของปัญญาประดิษฐ์ตามยุคสมัยจะเริ่มตั้งแต่การคิดค้นศาสตร์ในแขนงนี้ ซึ่งทำให้มีการพัฒนาของรูปแบบของเทคโนโลยีใหม่ๆ ขึ้นมากมาย ตั้งแต่การนำปัญหามาใช้ทดสอบกับระบบปัญญาประดิษฐ์ เพื่อวัดความสามารถและประสิทธิภาพต่อมาจึงมีการประยุกต์ใช้กับฐานองค์ความรู้ที่ช่วยให้ปัญญาประดิษฐ์สามารถเรียนรู้ได้ใกล้เคียงกับมนุษย์มากขึ้น มีการเข้าใจถึงปัญหาและเรียนรู้จากประสบการณ์ที่ผ่านมาได้ ด้วยการพัฒนาอย่างต่อเนื่องทำให้ธุรกิจและอุตสาหกรรมต่างๆ ตื่นตัวและหันมาสนใจในการพัฒนาปัญญาประดิษฐ์มากขึ้นจนถึงปัจจุบัน

การนำปัญญาประดิษฐ์เข้ามาใช้งานสามารถส่งเสริมการทำงานได้อย่างมากมาย ด้วยคุณสมบัติต่างๆ ของปัญญาประดิษฐ์ อาทิเช่น ทำความเข้าใจและเรียนรู้ได้จากประสบการณ์ วินิจฉัยปัญหาที่มีความซับซ้อน และสามารถตอบสนองต่อสถานการณ์ที่เกิดขึ้นได้หลากหลายรูปแบบ เป็นต้น สิ่งเหล่านี้ช่วยให้ปัญญาประดิษฐ์มีข้อได้เปรียบมากกว่ามนุษย์ในหลายด้าน ได้แก่ การคำนวณที่มีความซับซ้อนและปริมาณ

มากเกินกว่ามนุษย์จะทำได้ ระบบจัดเก็บข้อมูลที่อยู่แบบถาวรสามารถเรียกใช้ได้เมื่อต้องการ และการทำงานวนซ้ำที่เกิดความผิดพลาดน้อยกว่ามนุษย์ อย่างไรก็ตามปัญญาประดิษฐ์ก็มีข้อจำกัดในการทำงานหลายๆด้าน เช่น ปัญญาประดิษฐ์มักถูกสร้างขึ้นเพื่อนำมาใช้งานด้านใดด้านหนึ่ง ดังนั้นเมื่อพบกับหน้าที่ที่เกินความรับผิดชอบจะไม่สามารถตัดสินใจได้ หรือแก้ไขสถานการณ์ที่เหนือความคาดหมายได้ อีกทั้งยังต้องอาศัยองค์ความรู้เป็นหลัก เมื่อขาดสิ่งนี้ไปปัญญาประดิษฐ์จะไม่สามารถทำงานได้ตามเป้าหมาย เป็นต้น

การประยุกต์ใช้งานปัญญาประดิษฐ์สามารถทำได้หลายรูปแบบ ขึ้นอยู่กับจุดประสงค์ของระบบงานว่าต้องการให้ปัญญาประดิษฐ์ทำหน้าที่อะไร เช่น ทดแทนแรงงานมนุษย์ เพิ่มประสิทธิภาพให้ระบบงานและลดข้อผิดพลาดในการคำนวณหรือวิเคราะห์ข้อมูล เป็นต้น ศาสตร์ที่นำปัญญาประดิษฐ์ไปประยุกต์ใช้งานมีอยู่มากมาย เช่น ระบบผู้เชี่ยวชาญ ที่ใช้การวิเคราะห์หาวิธีการแก้ไขปัญหาต่างๆ ซึ่งมีความแน่นอนและรวดเร็วกว่ามนุษย์ สำหรับศาสตร์ทางด้านหุ่นยนต์จะนำปัญญาประดิษฐ์มาใช้ในการควบคุมเครื่องจักรและทำงานที่มีความเสี่ยงแทนแรงงานมนุษย์ เป็นต้น

แผนปฏิบัติการสอนสัปดาห์ที่ 2

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง การเขียนปริภูมิสถานะและเทคนิคการค้นหาโดยไม่มีการชี้แนะ **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. การแทนปัญหา	IS 2.1
2. การแก้ไขปัญหา	IS 2.2
3. ปริภูมิสถานะ	IS 2.3
4. เทคนิคการค้นหาโดยไม่มีการชี้แนะ	IS 2.4
5. การปรับปรุงเทคนิคการค้นหาในการจำเส้นทางในการเดินทางไปยังเป้าหมาย	IS 2.5

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		1-2-3-4-5			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 2
การเขียนปริภูมิสถานะและเทคนิคการค้นหาโดยไม่มีการชี้แนะ

การนำปัญญาประดิษฐ์มาช่วยในการอำนวยความสะดวกและแก้ปัญหาให้กับมนุษย์นั้น ต้องมีการแทนปัญหา (Preblem Representation) ให้กับปัญญาประดิษฐ์ก่อน โดยนำปัญหาที่ต้องการแก้ไขมาวิเคราะห์ และพิจารณาความเป็นไปได้ต่างๆ มาแทนให้อยู่ในรูปแบบที่ชัดเจน และนำไปสู่การแก้ไขปัญหาให้ได้ประสิทธิภาพมากที่สุด วิธีการแก้ไขปัญหาส่วนใหญ่ในปัญญาประดิษฐ์จะมองปัญหาให้อยู่ในรูปแบบการค้นหา โดยจะนำเทคนิคการค้นหาที่เหมาะสมมาประยุกต์ใช้ร่วมกัน โดยบทนี้จะกล่าวถึงหลักการแทนปัญหาและเทคนิคการค้นหาโดยไม่มีการชี้แนะ

2.1 การแทนปัญหา

ปัญหาส่วนใหญ่ที่พบในงานด้านปัญญาประดิษฐ์ มักเป็นปัญหาที่ต้องการคำตอบที่เหมาะสม โดยปกติปัญหาจะไม่ได้อยู่ในรูปแบบโครงสร้างที่ชัดเจนมากนัก แต่ศาสตร์ทางด้านปัญญาประดิษฐ์ได้คิดค้นเทคนิคการแปลงปัญหาให้อยู่ในรูปแบบโครงสร้างที่ชัดเจน จุดประสงค์เพื่อให้ปัญญาประดิษฐ์เองสามารถรับรู้และเข้าใจปัญหาที่พบได้ โดยเทคนิคดังกล่าวเรียกว่า “การแทนปัญหา (Problem Representation)” โดยการแทนปัญหาที่ดีจะต้องมีขอบเขตของปัญหาที่ชัดเจนเพื่อนำไปสู่กระบวนการแก้ไขปัญหาด้วยปัญญาประดิษฐ์ได้อย่างถูกต้อง ตัวอย่างการแทนปัญหาที่นิยมมาใช้ คือ การแทนปัญหาด้วยปริภูมิสถานะ (State Space Representation) เป็นการนำปัญหามาแจกแจงให้อยู่ในรูปโครงสร้างของกราฟ เพื่อให้สามารถแสดงลำดับการแก้ไขปัญหาได้อย่างชัดเจน

กระบวนการแปลงปัญหามี 3 ส่วนดังแสดงในภาพที่ 2-1 ได้แก่

- **ปัญหา (Problem)** ศึกษารายละเอียดของปัญหาที่เกิดขึ้น รวมทั้งต้องทราบเป้าหมายของปัญหาที่ต้องการ
- **โครงสร้างข้อมูล (Data Structure)** นำรายละเอียดของปัญหามาแปลงให้อยู่ในรูปแบบโครงสร้างข้อมูล เช่น โครงสร้างกราฟ เป็นต้น
- **อัลกอริธึม (Algorithm)** นำมาใช้จัดการภายในโครงสร้างข้อมูล เช่น การค้นหาแบบลึกก่อน (Depth-First Search) , การค้นหาแบบกว้างก่อน (Breath-First Search) เป็นต้น

ภาพที่ 2-1 กระบวนการแปลงปัญหา

ปัญหาในงานด้านปัญญาประดิษฐ์แบ่งได้เป็น 2 ประเภท คือ ปัญหาของเล่น (Toy Problem) และปัญหาในโลกความเป็นจริง (Real-World Problem)

- **ปัญหาของเล่น** ส่วนใหญ่เป็นปัญหาอย่างง่ายที่มีรายละเอียดสั้น ชัดเจน และไม่ซับซ้อน เหมาะสำหรับนำมาใช้เปรียบเทียบประสิทธิภาพของอัลกอริธึม เช่น เกม เป็นต้น
- **ปัญหาในโลกความเป็นจริง** เป็นปัญหาที่มีความซับซ้อนและรายละเอียดมาก บางปัญหาต้องทดลองด้วยสมการต่างๆ เพื่อใช้แก้ปัญหา เช่น ปัญหาการออกแบบวงจรรวมความจุสูงมาก (Very Large-Scale Integration: VLSI) เป็นการนำชิฟที่มีจำนวนมาก มาจัดวางตำแหน่งบนแผงบอร์ดที่มีพื้นที่จำกัด โดยอาจมีชิฟบางตัวที่จำเป็นต้องวางตำแหน่งใกล้กันเพื่อเชื่อมวงจรเข้าด้วยกัน ทำให้ต้องจัดกลุ่ม และจัดวางตำแหน่งให้เหมาะสมบนพื้นที่จำกัดโดยไม่ทับซ้อนกัน นอกจากนี้ปัญหาอื่นๆ เช่น ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem: TSP), และปัญหาหุ่นยนต์สำรวจ

2.2 การแก้ไขปัญหา

การแก้ไขปัญหา (Problem Solving) เป็นการนำปัญหามาวิเคราะห์เป็นขั้นตอน เพื่อค้นหาเป้าหมายที่ต้องการ โดยหลักการแก้ไขปัญหาในปัญญาประดิษฐ์จะมองปัญหาให้อยู่ในรูปแบบ ของการค้นหา เนื่องจากมีกระบวนการค้นหาคำตอบที่เป็นลำดับขั้นตอนที่ชัดเจน สะดวกต่อการเขียนโปรแกรม และหากกรณีที่พบคำตอบที่มากกว่าหนึ่งวิธี จะสามารถค้นหาเส้นทางเพื่อไปยังคำตอบที่ดีที่สุดได้ หากพิจารณาขั้นตอนการแก้ไขปัญหาจะพบว่ามีหลักการทั้งหมด 4 ขั้นตอน

1. **Goal Formulation** เป็นการกำหนดเป้าหมาย ซึ่งมีความสำคัญมาก เนื่องจากเป้าหมายจะช่วยให้การแก้ปัญหาเป็นไปตามขอบเขตที่วางไว้ ดังนั้นจึงควรกำหนดเป้าหมายให้ชัดเจน
2. **Problem Formulation** อธิบายปัญหาให้อยู่ในรูปแบบมาตรฐาน Well-Defined Problem
3. **Search for Solution** เลือกเทคนิคการค้นหาให้เหมาะสมกับปัญหา
4. **Execute** นำเทคนิคการค้นหามาเขียนเป็นโปรแกรมประยุกต์ใช้งานจริง

การแก้ปัญหาทั้ง 4 ขั้นตอนล้วนมีความสำคัญมากต่อการแก้ไขปัญหา แต่ขั้นตอนของ Problem Formulation มีความสำคัญมาก เนื่องจากหากระบุรายละเอียดปัญหาอย่างถูกต้องและครบถ้วนแล้ว จะทำให้ง่ายต่อการอธิบายหรือทำความเข้าใจกับปัญหา และสามารถแก้ปัญหาได้ตรงจุด

ขั้นตอนของ Problem Formulation เป็นการนำปัญหามาแปลงให้อยู่ในรูปมาตรฐาน Well-Defined Problem ซึ่งจะประกอบด้วย

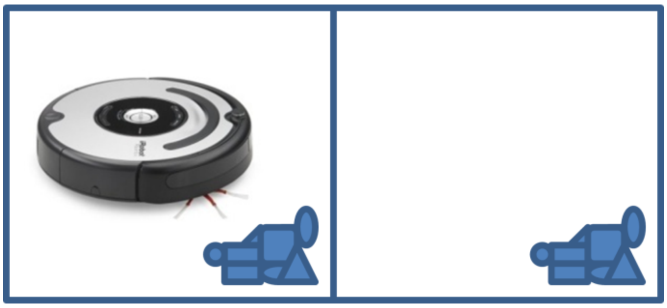
- Initial State กำหนดสถานะเริ่มต้นของปัญหา
- Successor Function กำหนดเซตการกระทำทั้งหมดที่เป็นไปได้
- Goal State กำหนดสถานะเป้าหมายของปัญหาที่เป็นไปได้
- Path Cost กำหนดค่าใช้จ่ายหรือทรัพยากรที่มใช้ทั้งหมด จากการกระทำแต่ละครั้ง

2.2.1 ตัวอย่าง: ปัญหาหุ่นยนต์ทำความสะอาด

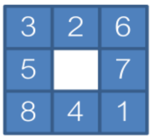
ปัญหาคือหุ่นยนต์จะทำหน้าที่ขจัดสิ่งสกปรกโดยการดูด (Suck) ด้วยอุปกรณ์ทำความสะอาดของหุ่นยนต์ โดยกำหนดให้มีห้องที่ต้องทำความสะอาด 2 ห้อง หุ่นยนต์สามารถเคลื่อนที่ไปห้องซ้าย(Left) และ ขวา (Right) ได้ กำหนดให้หุ่นยนต์เริ่มต้นอยู่ที่ห้องซ้าย และ ทั้งสองห้องมีสิ่งสกปรกอยู่ดังแสดงในภาพที่ 2-2

จากปัญหาข้างต้นสามารถนำมาเขียนให้อยู่ในรูปแบบ Problem Formulation ได้ดังนี้

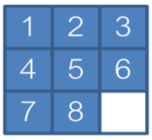
- Initial State กำหนดให้ตำแหน่งหุ่นยนต์เริ่มต้นอยู่ที่ห้องฝั่งซ้าย
- Successor Function เซตของการกระทำของหุ่นยนต์มี 3 การกระทำคือ {Left, Right, Suck}
- Goal State ห้องทั้ง 2 ห้องต้องสะอาด
- Path Cost การกระทำแต่ละครั้ง กำหนดให้ใช้ทรัพยากร 1 หน่วย



ภาพที่ 2-2 สถานะเริ่มต้นของปัญหาหุ่นยนต์ทำความสะอาด



(ก) สถานะเริ่มต้น



(ข) สถานะเป้าหมาย

ภาพที่ 2-3 สถานะเริ่มต้นและเป้าหมายของปัญหา 8-Puzzle

2.2.2 ตัวอย่าง: ปัญหาเกม 8-Puzzle

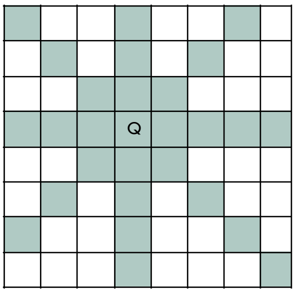
เกม 8-Puzzle ประกอบด้วยแผ่นกระดาน 3 x 3 โดยบรรจุแผ่นป้ายขนาด 1 x 1 ทั้งหมด 8 แผ่น โดยแต่ละแผ่นป้ายมีหมายเลขกำกับที่ไม่ซ้ำกันตั้งแต่ 1 ถึง 8 และมีช่องว่าง 1 ช่อง แผ่นป้ายที่ติดอยู่กับช่องว่างสามารถเคลื่อนมาแทนที่ช่องว่างได้ ปัญหาของเกม 8-Puzzle คือ เลื่อนตำแหน่งแผ่นป้ายหมายเลขจากสถานะเริ่มต้น ให้อยู่ในรูปแบบตามสถานะเป้าหมายที่กำหนด ดังแสดงในภาพที่ 2-3

จากปัญหาข้างต้นสามารถนำมาเขียนให้อยู่ในรูปแบบ Problem Formulation ได้ดังนี้

- Initial State สถานะเริ่มต้นดังภาพที่ 2-3(ก)
- Successor Function เซตการเคลื่อนที่ของช่องว่างสามารถเคลื่อนที่ได้ 4 ทาง คือ {Up, Down, Left, Right}
- Goal State สถานะเริ่มต้นดังภาพที่ 2-3(ข)
- Path Cost การเคลื่อนที่ของช่องว่างแต่ละครั้ง กำหนดให้ใช้ทรัพยากร 1 หน่วย

2.2.3 ตัวอย่าง: ปัญหาเกม 8-Queen

ปัญหาของเกม 8-Queen คือผู้เล่นต้องวางตัวหมากรุก Queen ลงบนกระดานหมากรุกที่มีขนาด 8 x 8 ช่อง โดยไม่ให้ Queen ทั้ง 8 ตัวนั้นสามารถโจมตีกันได้ โดยตัว Queen ในหมากรุกนั้นสามารถโจมตีได้ทั้ง 8 ทิศรอบกระดาน ดังแสดงในภาพที่ 2-4 และภาพที่ 2-5 แสดงหนึ่งในตัวอย่างของตำแหน่งการวาง Queen ทั้ง 8 ตัวไม่ให้โจมตีกัน



ภาพที่ 2-4 การโจมตีของ Queen ทั้ง 8 ทิศรอบกระดาน

Q							
						Q	
				Q			
							Q
	Q						
			Q				
					Q		
		Q					

ภาพที่ 2-5 ตัวอย่างการวาง Queen ทั้ง 8 ตัวโดยไม่โจมตีกัน

จากปัญหาข้างต้นสามารถนำมาเขียนให้อยู่ในรูปแบบ Problem Formulation ได้ดังนี้

- **Initial State** ไม่มี Queen ตัวใดอยู่บนกระดานหมากรุก
- **Successor Function** เพิ่มจำนวน Queen ทีละตัวลงบนช่องว่างของกระดานหมากรุก
- **Goal State** มี Queen 8 ตัวอยู่บนกระดาน โดยทั้ง 8 ตัวนั้นไม่สามารถโจมตีกันเองได้
- **Path Cost** เป็น 0 เนื่องจากปัญหานี้ไม่มีค่าใช้จ่ายในการเคลื่อนที่และไม่มีเวลากำหนด

การเขียนปัญหาให้อยู่ในรูปแบบของ Problem Formulation ข้างต้นนั้นถูกหลักตามมาตรฐานรูปแบบ Well-Defined Problem แต่อย่างไรก็ตาม ถ้านำปัญหานี้ไปเขียนโปรแกรมเพื่อทำการแก้ปัญหาจริงๆ จะไม่เหมาะสมมากนัก เนื่องจากช่องว่างในกระดานหมากรุกเริ่มต้นนั้นมีทั้งหมด 64 ช่อง ซึ่งหมายความว่าจาก Successor Function ที่กำหนด Queen ตัวแรกสามารถถูกวางได้ 64 วิธี (มีช่องให้วางทั้งหมด 64 ช่อง) และ Queen ตัวที่ 2 สามารถถูกวางได้ 63 วิธี (มีช่องว่างเหลือ 63 ช่องหลังจากวาง Queen ตัวแรกแล้ว) ซึ่งถ้าต้องวาง Queen ทั้ง 8 ตัวจะทำให้มีวิธีในการวาง Queen ทั้งหมด 64 x 63 x 62 x 61 x 60 x 59 x 58 x 57 วิธี หรือประมาณ 1.8×10^{14} วิธี ทำให้ปัญหานี้มีความซับซ้อนและใช้เวลาเป็นอย่างมากในการแก้ปัญหา

แต่ถ้ากลับมามองลักษณะของ Queen ซึ่งโจมตี 8 ทิศรอบกระดาน ถ้าหาก Queen ได้ถูกวางไว้ที่หลักหรือแถวใดแล้ว ตัว Queen ที่เหลือจะไม่สามารถวางที่หลักหรือแถวนั้นได้เนื่องจากจะถูก Queen ตัวที่วางแล้วโจมตี ดังนั้นเพื่อให้การแก้ปัญหาสะดวกขึ้น จึงควรเปลี่ยน Successor Function ใหม่เป็น

- **Successor Function** เพิ่มจำนวน Queen ทีละตัวจากหลักซ้ายไปขวา และตำแหน่งที่วางต้องไม่ถูก Queen บนกระดานโจมตี

2.3 ปริภูมิสถานะ

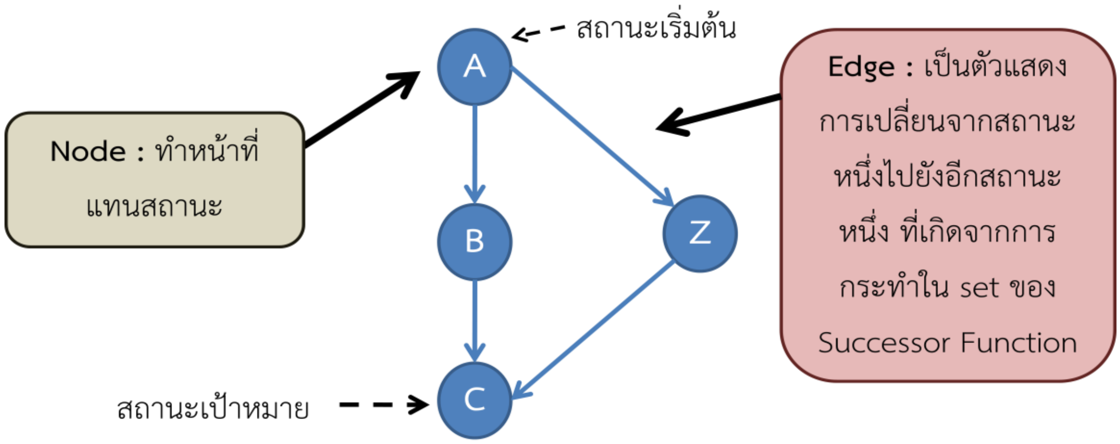
ส่วนประกอบหลักของปัญหาจะประกอบด้วย Initial State, Successor Function, Goal State โดยส่วนประกอบเหล่านี้ได้มาจากการพิจารณาปัญหาด้วย Problem Formulation หากนำส่วนหลักทั้ง 3 ส่วนนี้มาแสดงในรูปแบบของแผนภาพ จะทำให้เห็นการเปลี่ยนแปลงของสถานะต่างๆ และแสดงเส้นทางการแก้ปัญหาที่นำไปสู่เป้าหมายได้หลากหลายแนวทาง เพื่อที่จะค้นหาวิธีที่ดีที่สุด การแทนปัญหาด้วยแผนภาพเรียกว่า “การแทนด้วยปริภูมิสถานะ (State Space Representation)”

2.3.1 ส่วนประกอบของปริภูมิสถานะ

สถานะ (State) หมายถึง เหตุการณ์หรือสถานการณ์ของปัญหาที่เป็นไปได้ ดังนั้น ปริภูมิสถานะ (State Space) จึงหมายถึงเซตที่ประกอบด้วยทุกสถานการณ์ที่เป็นไปได้ตั้งแต่สถานะเริ่มต้น ช่วงดำเนินการที่ทำให้เกิดสถานะต่างๆ จนถึงสถานะเป้าหมาย

การแก้ไขปัญหาส่วนมากจะมองปัญหาอยู่ในรูปแบบของการค้นหาบนปริภูมิสถานะ (State Space Search) เนื่องจากภายในปริภูมิสถานะจะแสดงการเปลี่ยนแปลงของแต่ละสถานะอย่างชัดเจน และง่ายต่อการหาเส้นทางไปยังสถานะเป้าหมายได้ ปริภูมิสถานะจะนำเสนออยู่ในรูปแบบของโครงสร้างแบบกราฟ ดังแสดงในภาพที่ 2-6

ปริภูมิสถานะจะประกอบด้วยโหนด (Node) ที่ทำหน้าแทนสถานะ และเส้นเชื่อมระหว่างโหนดเรียกว่า เอจ (Edge) แทนเส้นทางการเปลี่ยนจากสถานะหนึ่งไปยังอีกสถานะหนึ่งที่เกิดจากการกระทำในเซตของ Successor Function และในการแก้ปัญหาทั่วไปต้องกำหนดสถานะเริ่มต้น (Initial State) และสถานะเป้าหมาย (Goal State) เข้าไปในปริภูมิสถานะด้วย



ภาพที่ 2-6 ส่วนประกอบของปริภูมิสถานะ

2.3.2 การประยุกต์ใช้งานปริภูมิสถานะ

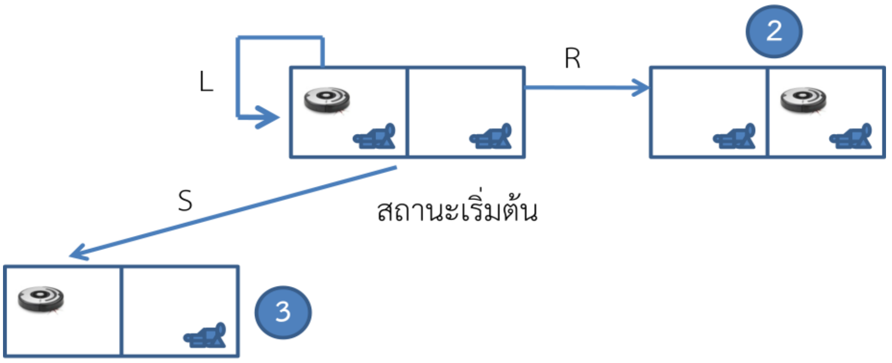
การนำปริภูมิสถานะไปประยุกต์ใช้แก้ปัญหา ต้องเข้าใจก่อนว่าสิ่งที่ทำให้เกิดสถานการณ์ต่างๆ บนปริภูมิสถานะนั้น เกิดมาจากการกระทำในเซตของ Successor Function ที่ได้มาจาก Problem Formulation มากระทำกับสถานะที่มี ทำให้เกิดผลลัพธ์เป็นสถานะต่างๆ เชื่อมต่อกันเป็นโครงสร้างกราฟ

2.3.2.1 ตัวอย่าง: ปริภูมิสถานะของปัญหาหุ่นยนต์ทำความสะอาด

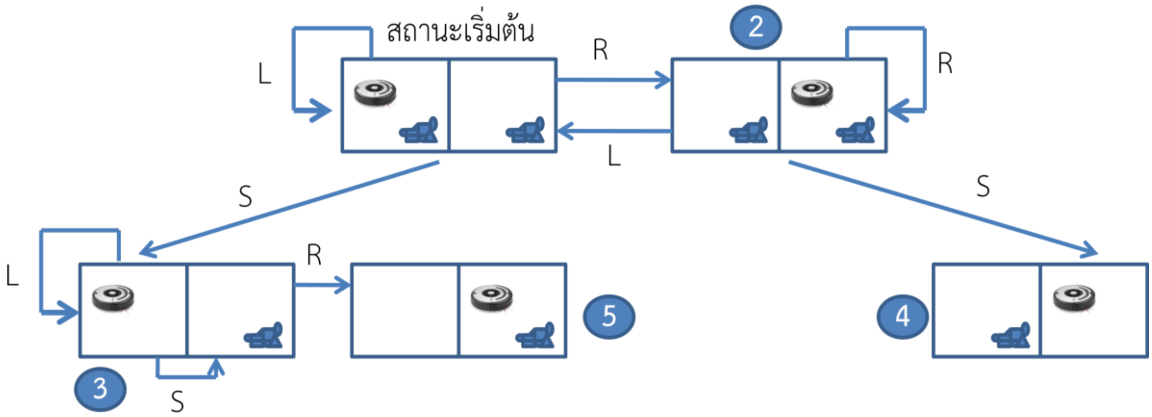
จากหัวข้อ 2.2.1 ได้กำหนด Problem Formulation เอาไว้แล้ว ซึ่งสถานะเริ่มต้นของปัญหาคือ หุ่นยนต์ทำความสะอาดอยู่ที่ห้องด้านซ้ายและห้องทั้ง 2 ห้องสกปรก ซึ่งเมื่อพิจารณาการกระทำในเซตของ Successor Function คือ Left (L), Right (R) และ Suck (S) กับสถานะเริ่มต้นนี้จะทำให้เกิดความสัมพันธ์ระหว่างสถานะดังภาพที่ 2-7

โดยจากสถานะเริ่มต้นเมื่อมีการกระทำจากสมาชิกในเซตของ Successor Function ตัวแรกคือ Left (L) จะไม่ทำให้เกิดการเปลี่ยนแปลงสถานะใดๆ (วนกลับมาที่สถานะเดิม) เนื่องจากหุ่นยนต์ได้อยู่ที่ตำแหน่งห้องด้านซ้ายอยู่แล้วจึงไม่สามารถเคลื่อนที่ต่อไปได้อีก ในขณะที่เมื่อมีการกระทำจากสมาชิกในเซตของ Successor Function ตัวที่สองคือ Right (R) กับสถานะเริ่มต้นจะส่งผลให้หุ่นยนต์เปลี่ยนตำแหน่งไปอยู่ที่ห้องทางด้านขวา ดังสถานะหมายเลข 2 ในภาพที่ 2-7 และสำหรับการกระทำจากสมาชิกในเซตของ Successor Function ตัวสุดท้ายคือ Suck (S) กับสถานะเริ่มต้นจะส่งผลให้หุ่นยนต์ทำความสะอาดห้องทางด้านซ้ายที่ตัวเองอยู่ ทำให้เกิดสถานะใหม่ดังสถานะหมายเลข 3 ในภาพที่ 2-7 ที่ห้องด้านซ้ายสะอาดและตัวหุ่นยนต์ยังคงอยู่ด้านซ้าย

จากนั้นเพื่อทำปริภูมิสถานะให้สมบูณ์ จึงต้องเขียนทุกสถานะที่เกิดขึ้นหลังจากเกิดการกระทำในเซตของ Successor Function กับสถานะที่เกิดใหม่ให้ครบ ภาพที่ 2-8 แสดงปริภูมิสถานะที่เกิดขึ้นหลังจากนำการกระทำทั้งหมดมากระทำกับโหนด 2 และ โหนด 3 จะเกิดสถานะโหนด 4 และโหนด 5 เพิ่มขึ้นตามลำดับ

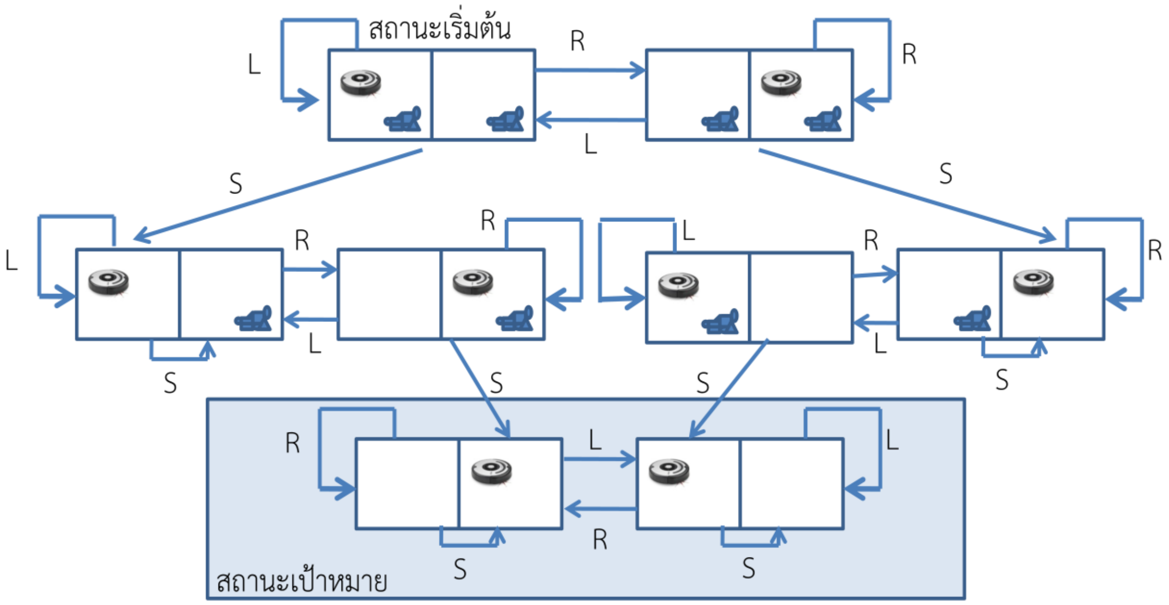


ภาพที่ 2-7 สถานะในปริภูมิสถานะจากการกระทำในเซตของ Successor Function กับสถานะเริ่มต้น

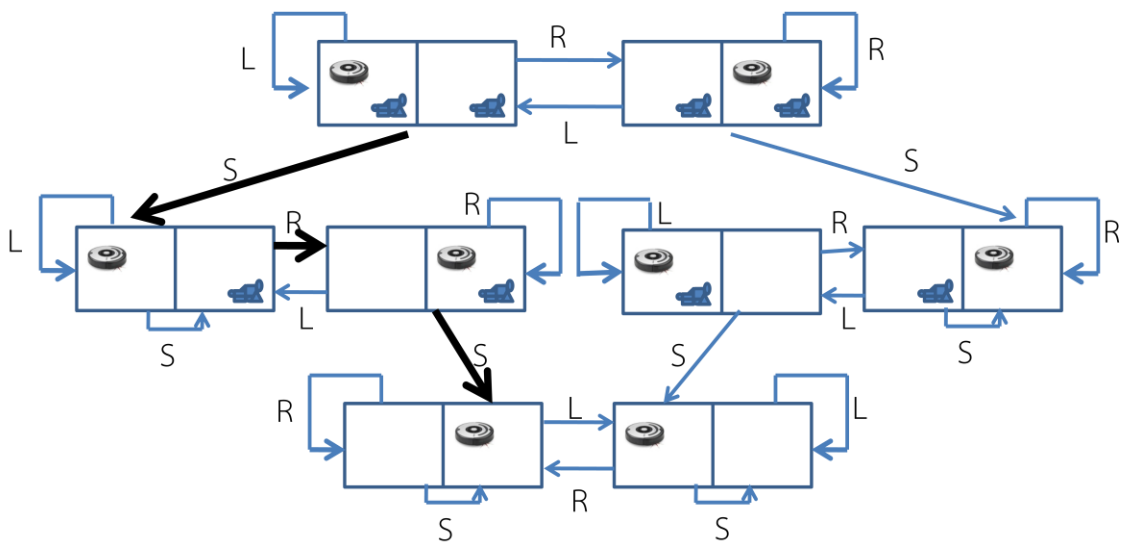


ภาพที่ 2-8 สถานะในปริภูมิสถานะจากการกระทำในเซตของ Successor Function กับสถานะ 2 และ 3

จากนั้นให้ทำการเพิ่มสถานะใหม่ที่เกิดจากกระทำของสมาชิกในเซตของ Successor Function กับสถานะที่มีอยู่ต่อไปเรื่อยๆ จนได้ปริภูมิสถานะที่สมบูรณ์ และสามารถที่จะเห็นสถานะเป้าหมายจากปริภูมิสถานะได้ ดังแสดงในภาพที่ 2-9 ซึ่งจากภาพจะเห็นได้ว่าเส้นทางการกระทำจากสถานะเริ่มต้นไปยังสถานะเป้าหมายนั้นมีได้หลายทาง ซึ่งการค้นหาเส้นทางนี้นั้นจะขึ้นอยู่กับเทคนิคหรือวิธีการค้นหาที่จะนำมาใช้งาน ตัวอย่างเส้นทางที่เหมาะสมในการเดินทางจากสถานะเริ่มต้นไปยังสถานะเป้าหมายที่สั้นที่สุดแสดงดังภาพที่ 2-10 โดยใช้การกระทำ Suck, Right และ Suck ตามลำดับ



ภาพที่ 2-9 ปริภูมิสถานะที่สมบูรณ์ของปัญหาหุ่นยนต์ทำความสะอาด



ภาพที่ 2-10 ตัวอย่างเส้นทางการกระทำที่ใช้จากสถานะเริ่มต้นไปยังสถานะเป้าหมาย

2.3.2.2 ตัวอย่าง: ปริภูมิสถานะของปัญหา 8-Puzzle

กำหนดในปัญหา 8-Puzzle มีสถานะเริ่มต้นดังภาพที่ 2-11(ก) และต้องการสถานะเป้าหมายดังภาพที่ 2-11(ข) โดยเซตของการกระทำ Successor Function ของปัญหาได้ถูกกำหนดไว้แล้วในหัวข้อที่ 2.2.2 โดยมีการกระทำที่สามารถทำกับสถานะใดๆ คือการเคลื่อนที่ช่องว่าง ซึ่งสามารถเคลื่อนที่ได้ 4 ทิศทาง Up (U), Down (D), Left (L) และ Right (R) ซึ่งเมื่อนำสมาชิกในเซตของการกระทำ Successor Function ทำกับสถานะเริ่มต้นก็จะเกิดสถานะใหม่ขึ้น จากนั้นนำสมาชิกในเซตของการกระทำมากระทำกับสถานะใหม่ที่เกิดขึ้นต่อไปเรื่อยๆ เนื่องจากปัญหานี้มีสถานะที่เป็นไปได้หลายสถานะ ดังนั้นภาพที่ 2-12 จะแสดงเพียงบางส่วนของปริภูมิสถานะที่เกิดขึ้นเท่านั้น แต่จากปริภูมิสถานะบางส่วนนี้ได้มีสถานะเป้าหมายรวมอยู่ด้วยซึ่งสามารถไปถึงได้ด้วยการกระทำ Down, Down, Right ตามลำดับ ดังแสดงในภาพที่ 2-13

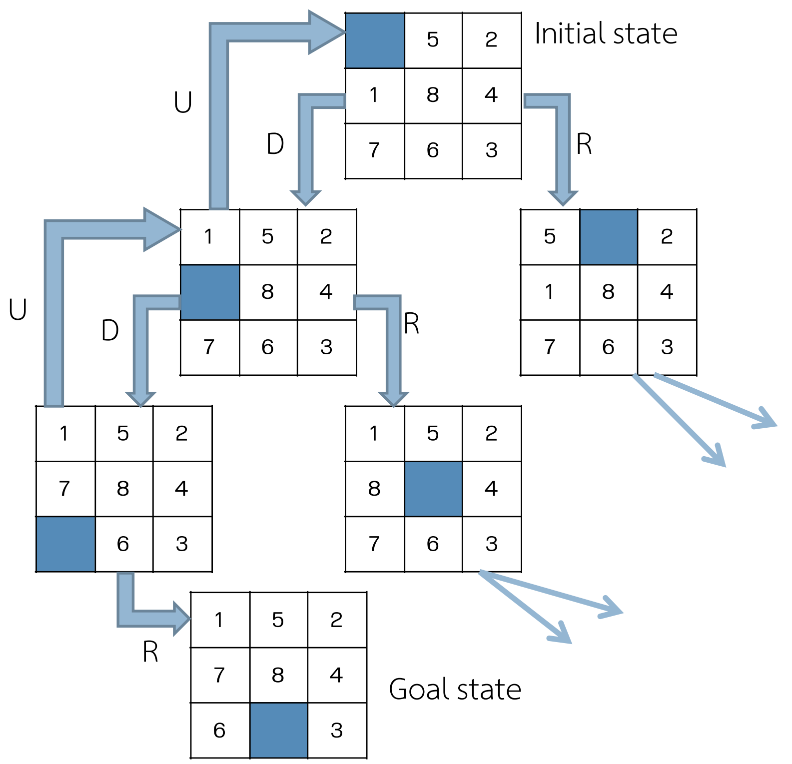
	5	2
1	8	4
7	6	3

(ก) สถานะเริ่มต้น

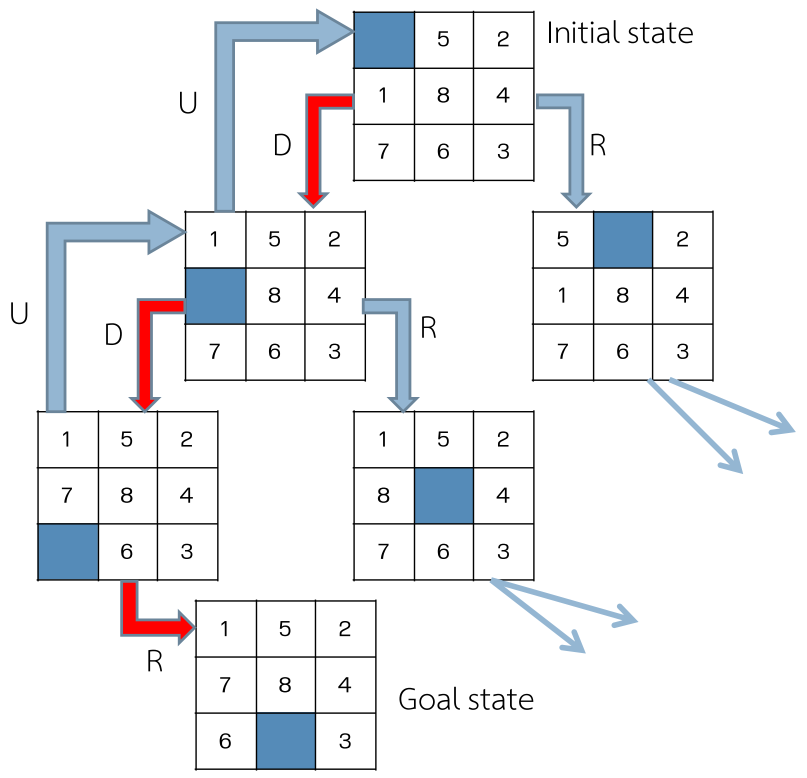
1	5	2
7	8	4
6		3

(ข) สถานะเป้าหมาย

ภาพที่ 2-11 ตัวอย่างสถานะของปัญหา 8-Puzzle



ภาพที่ 2-12 ปริภูมิสถานะของปัญหา 8-Puzzle



ภาพที่ 2-13 เส้นทางการกระทำที่ใช้จากสถานะเริ่มต้นไปยังสถานะเป้าหมาย

2.4 เทคนิคการค้นหาโดยไม่มีการชี้แนะ

เทคนิคการค้นหาโดยไม่มีการชี้แนะ (Uninformed Search) บางครั้งเรียกว่า เทคนิดการค้นหาแบบบอด (Blind search) ซึ่งเป็นเทคนิคการค้นหาที่ไม่มีตัวช่วยในการค้นหา แต่จะมีรูปแบบการค้นหาที่แน่นอนตายตัวเช่นจะค้นหาสถานะจากบนลงล่างในปริภูมิค้นหา เทคนิคการค้นหาประเภทนี้สามารถแบ่งออกเป็นเทคนิคหลักๆ ได้ 2 เทคนิค คือ เทคนิคการค้นหาแบบกว้างก่อน (Breath-First Search) และเทคนิคการค้นหาแบบลึกก่อน (Depth-First Search) อีกทั้งยังมีการพัฒนาเทคนิคการค้นหาใหม่โดยใช้พื้นฐานจากเทคนิคพื้นฐานนี้ ได้แก่ เทคนิคการค้นหา Depth-Limited Search (DLS) และ Iterative Deepening Search (IDS)

2.4.1 เทคนิคการค้นหาแบบกว้างก่อน (Breath-First Search)

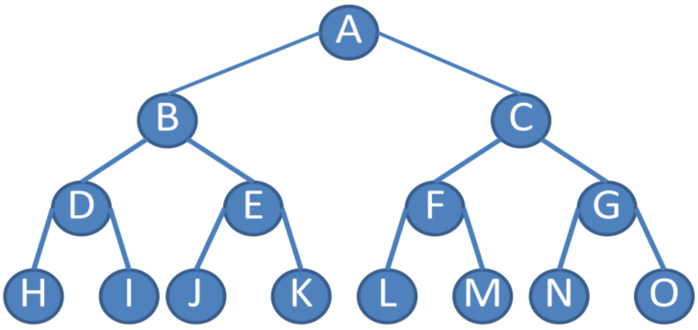
ในการค้นหาแบบกว้างก่อน (breadth-first search) นี้สถานะทุกตัวที่อยู่ในระดับเดียวกันของต้นไม้หรือกราฟ จะถูกตรวจสอบก่อนสถานะที่อยู่ในระดับถัดไปวิธีการของการค้นหาแบบนี้ทำโดยเริ่มจากการสร้างสถานะลึกของสถานะเริ่มต้นก่อนแล้วตรวจสอบว่ามีสถานะใดที่เป็นสถานะเป้าหมายหรือไม่ ถ้าหากว่ามีก็เป็นอันว่าการค้นหาสิ้นสุด ถ้าไม่มีก็จะสร้างสถานะลึกของสถานะเหล่านั้นต่อไปอีก และทำเช่นนี้ไปเรื่อยๆจนกว่าจะพบสถานะเป้าหมาย หรือจนไม่สามารถสร้างสถานะลึกใหม่ได้อีก ซึ่งอัลกอริธึมที่ใช้กับการจัดการเทคนิคการค้นหาประเภทนี้คือ เทคนิคแถวคอย (Queue) ซึ่งสามารถเขียนเป็นรหัสเทียม (Pseudo Code) ได้ดังภาพที่ 2-14

```
begin
   open := [start];
   close := [ ] ;
   while open ≠ [ ] do
   begin
            remove leftmost state from open, call it X;
            if X is a goal then return SUCCESS
            else begin
                        generate children of X;
                        put X on close;
                        discard children of X if already on open or closed
                        put remaining children on right end of open
             end
   end
   return FAIL
end
```

ภาพที่ 2-14 รหัสเทียมของเทคนิคการค้นหาแบบกว้างก่อน

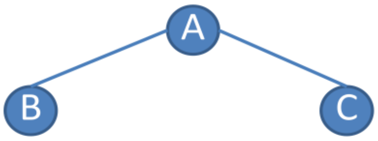
จากรหัสเทียมของเทคนิคการค้นหาแบบกว้างก่อนนั้น จะมีตัวแปรประเภทลิสท์อยู่ 2 ตัว คือ open และ closed โดยตัวแปร open จะใช้เก็บสถานะหรือโหนดที่ไม่ได้ถูกสำรวจ และตัวแปร closed จะใช้เก็บสถานะที่เคยถูกสำรวจผ่านมาแล้ว อัลกอริธึมจะเริ่มจากการใส่สถานะเริ่มต้นที่ตัวแปร open ตราบใดที่ยังมีสถานะรอการสำรวจในลิสท์ open ให้นำสถานะที่อยู่ซ้ายสุดของลิสท์ออกมาเพื่อดำเนินการกับเซต Successor Function เพื่อให้ได้สถานะลูกมา จากนั้นนำสถานะนั้นไปเก็บไว้ในลิสท์ closed สถานะลูกที่เกิดขึ้นจะนำไปแทรกที่ทางด้านขวาจะของลิสท์ open ถ้าสถานะลูกที่เกิดขึ้นมานั้นไม่เคยอยู่ในลิสท์ open หรือ closed มาก่อน

ตัวอย่างการใช้เทคนิคการค้นหาตามแบบกว้างก่อน โดยกำหนดปริภูมิของสถานะจำลองของปัญหาดังแสดงในภาพที่ 2-15 กำหนดให้สถานะเริ่มต้นคือ สถานะ “A” และสถานะเป้าหมายคือ สถานะ “E” โดยขั้นตอนการค้นหาสามารถทำได้ตามตารางที่ 2-1

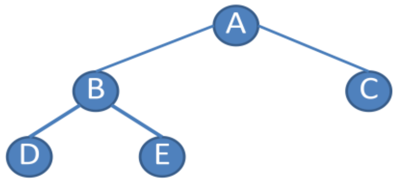
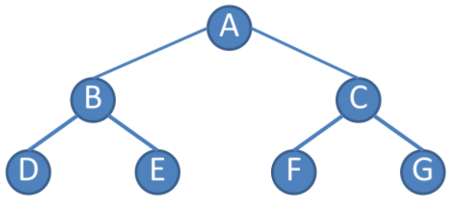
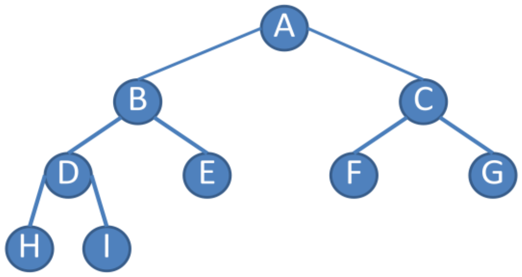
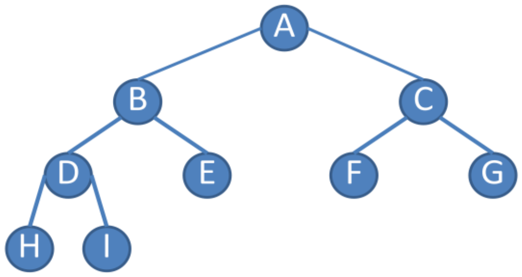


ภาพที่ 2-15 ปริภูมิสถานะจำลองของปัญหา

ตารางที่ 2-1 ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบกว้างก่อน

	นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open • open = [A] • closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือสถานะ “B” และ “C” จากนั้นนำสถานะ “A” ใส่ในตัวแปร closed และสถานะลูกของ “A” (“B” และ “C”) ใส่ที่ด้านขวาของตัวแปร open • open = [B, C] • closed = [A]

ตารางที่ 2-1 (ต่อ) ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบกว้างก่อน

	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “B” ซึ่งก็คือสถานะ “D” และ “E” จากนั้นนำสถานะ “B” ใส่ในตัวแปร closed และสถานะลูกของ “B” (“D” และ “E”) ใส่ที่ด้านขวาของตัวแปร open • open = [C, **D, E**] • closed = [**B,** A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “C” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “C” ซึ่งก็คือสถานะ “F” และ “G” จากนั้นนำสถานะ “C” ใส่ในตัวแปร closed และสถานะลูกของ “C” (“F” และ “G”) ใส่ที่ด้านขวาของตัวแปร open • open = [D, E, **F, G**] • closed = [**C,** B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “D” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “D” ซึ่งก็คือสถานะ “H” และ “I” จากนั้นนำสถานะ “D” ใส่ในตัวแปร closed และสถานะลูกของ “D” (“H” และ “I”) ใส่ที่ด้านขวาของตัวแปร open • open = [E, F, G, **H, I**] • closed = [**D,** C, B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “E” ออกจากลิสท์ open ตรวจสอบว่า “E” คือสถานะเป้าหมายหรือไม่ จากปัญหา E คือสถานะเป้าหมาย เพราะฉนั้นการค้นหาสถานะเป้าหมายจึงสิ้นสุดลง

เกณฑ์การวัดประสิทธิภาพของเทคนิคการค้นหาแต่ละประเภทจะแบ่งออกเป็น 4 ส่วนคือ ความสามารถจะค้นหาสถานะเป้าหมายได้อย่างแน่นอนถ้าสถานะเป้าหมายอยู่ในปริภูมิสถานะ (Completeness), ความสามารถที่จะค้นหาสถานะเป้าหมายพบโดยใช้จำนวนการกระทำของเซต Successor Function กับสถานะเริ่มต้นน้อยที่สุด (Optimality), ระยะเวลาที่ใช้ในการค้นหา (Time Complexity) และ ทรัพยากรที่ใช้ในการค้นหา (Space Complexity)

โดยเทคนิคการค้นหาแบบกว้างก่อนนั้น มีประสิทธิภาพดังนี้

- Completeness สามารถค้นหาสถานะเป้าหมายได้อย่างแน่นอน
- Optimality เส้นทางไปสู่สถานะเป้าหมายใช้การกระทำน้อยที่สุด
- Time Complexity $O(b^{d+1})$
- Space Complexity $O(b^{d+1})$

เมื่อ b = จำนวนกิ่งเฉลี่ยของโหนด, d = ความลึกของต้นไม้ในระดับที่หา

2.4.2 เทคนิคการค้นหาแบบลึกก่อน (Depth-First Search)

ในการค้นหาแบบลึกก่อน (depth-first search) นี้จะสร้างสถานะในแนวลึกทางด้านมุมซ้ายล่างก่อน ถ้าสถานะตามแนวดิ่งถูกสร้างหรือกระจายจนหมดและยังไม่ได้ คำตอบ ก็จะไล่กลับขึ้นด้านบนเพื่อหาเส้นทางอื่นที่จะเป็นไปได้ ซึ่งอัลกอริธึมที่ใช้กับการจัดการเทคนิคการค้นหาประเภทนี้คือเทคนิคสแต็ก (Stack) ซึ่งสามารถเขียนเป็นรหัสเทียม (Pseudo Code) ได้ดังภาพที่ 2-16

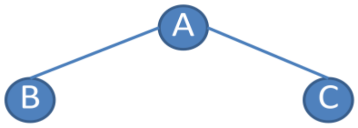
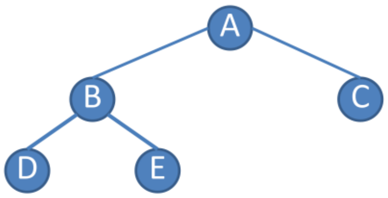
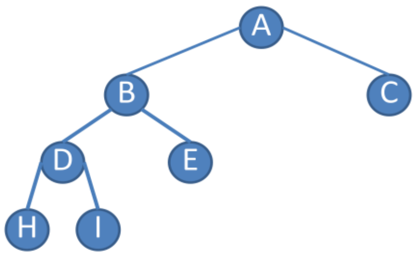
```
begin
   open := [start];
   close := [ ] ;
   while open ≠ [ ] do
   begin
            remove leftmost state from open, call it X;
            if X is a goal then return SUCCESS
            else begin
                        generate children of X;
                        put X on close;
                        discard children of X if already on open or closed
                        put remaining children on left end of open
             end
   end
   return FAIL
end
```

ภาพที่ 2-16 รหัสเทียมของเทคนิคการค้นหาแบบลึกก่อน

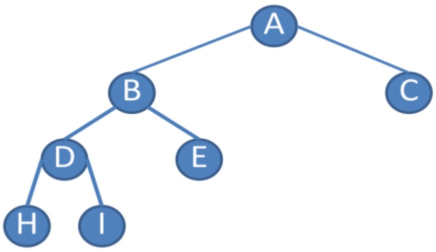
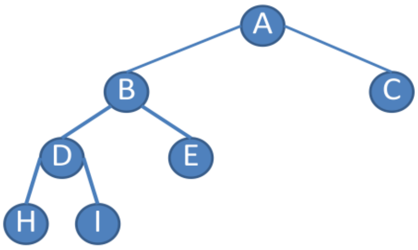
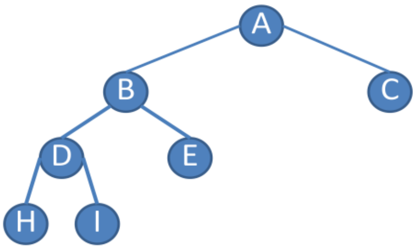
โดยจากรหัสเทียมจะเป็นได้ว่าเทคนิคการค้นหาแบบลึกก่อนนี้แตกต่างกับเทคนิคการค้นหาแบบกว้างก่อนเพียงแค่ตำแหน่งการใส่สถานะลูกที่เกิดจากการกระทำของสถานะแม่กับเซตของการกระทำ Successor Function โดยที่เทคนิคการค้นหาแบบกว้างก่อนนั้นจะนำสถานะลูก ใส่เข้าที่ด้านขวาของตัวแปรลิสท์ open แต่สำหรับเทคนิคการค้นแบบลึกก่อนจะนำสถานะลูกใส่เข้าที่ด้านซ้าย ของตัวแปรลิสท์ open แทน

ตัวอย่างการใช้เทคนิคการค้นหาตามแบบลึกก่อน โดยกำหนดปริภูมิของสถานะจำลองของปัญหาเดียวกับการใช้เทคนิคค้นหาแบบกว้างก่อนดังแสดงในภาพที่ 2-15 กำหนดให้สถานะเริ่มต้นคือสถานะ “A” และสถานะเป้าหมายคือ สถานะ “E” โดยขั้นตอนการค้นหาสามารถทำได้ตามตารางที่ 2-2

ตารางที่ 2-2 ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบลึกก่อน

	นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open • open = [A] • closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือ สถานะ “B” และ “C” จากนั้นนำสถานะ “A” ใส่ในตัวแปร closed และสถานะลูกของ “A” (“B” และ “C”) ใส่ที่ด้านซ้ายของตัวแปร open • open = [B, C] • closed = [A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “B” ซึ่งก็คือ สถานะ “D” และ “E” จากนั้นนำสถานะ “B” ใส่ในตัวแปร closed และสถานะลูกของ “B” (“D” และ “E”) ใส่ที่ด้านซ้ายของตัวแปร open • open = [D, E, C] • closed = [B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “D” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “D” ซึ่งก็คือ สถานะ “H” และ “I” จากนั้นนำสถานะ “D” ใส่ในตัวแปร closed และสถานะลูกของ “D” (“H” และ “I”) ใส่ที่ด้านซ้ายของตัวแปร open • open = [H, I, E, C] • closed = [D, B, A]

ตารางที่ 2-2 (ต่อ) ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบลึกก่อน

	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “H” ออกจากลิสท์ open เพื่อสร้างสถานะลูกของโหนด “H” แต่ทว่า “H” ไม่มีโหนดลูก จึงให้นำ “H” ไปใส่ในตัวแปร closed เพียงอย่างเดียว • open = [**I**, E, C] • closed = [**H**, D, B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “I” ออกจากลิสท์ open เพื่อสร้างสถานะลูกของโหนด “I” แต่ทว่า “I” ไม่มีโหนดลูก จึงให้นำ “I” ไปใส่ในตัวแปร closed เพียงอย่างเดียว • open = [E, C] • closed = [**I**, H, D, B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “E” ออกจากลิสท์ open ตรวจสอบว่า “E” คือสถานะเป้าหมายหรือไม่ จากปัญหา E คือสถานะเป้าหมาย เพราะฉนั้นการค้นหาสถานะเป้าหมายจึงสิ้นสุดลง

เทคนิคการค้นหาแบบลึกก่อนนั้น มีประสิทธิภาพดังนี้

- Completeness **ไม่**สามารถค้นหาสถานะเป้าหมายได้อย่างแน่นอน
- Optimality เส้นทางไปสู่สถานะเป้าหมาย**ไม่จำเป็น**ต้องใช้จำนวนการกระทำที่น้อยที่สุด
- Time Complexity $O(b^m)$
- Space Complexity $O(bm)$

เมื่อ b = จำนวนกิ่งเฉลี่ยของโหนด, m = ความลึกมากสุดของต้นไม้

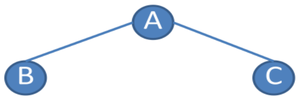
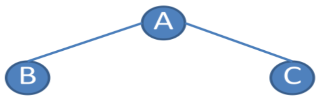
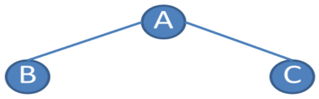
โดยจะเห็นได้ว่าการค้นหาแบบกว้างก่อนนั้นให้ประสิทธิภาพเรื่อง Completeness, Optimality ดีกว่าการค้นหาแบบลึกก่อน แต่อย่างไรก็ตามการใช้เทคนิคการค้นหาแบบลึกก่อนจะใช้ทรัพยากรในการค้นหาที่น้อยกว่าแบบกว้างก่อนเป็นอย่างมาก

2.4.3 เทคนิคการค้นหาแบบจำกัดความลึก (Depth-Limited Search)

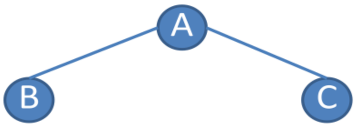
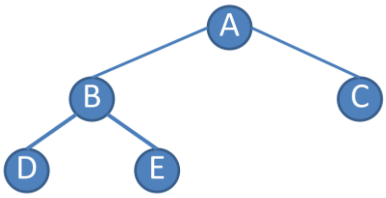
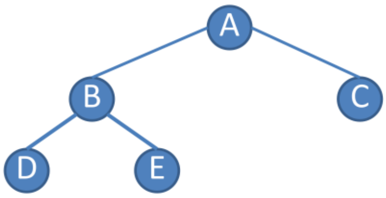
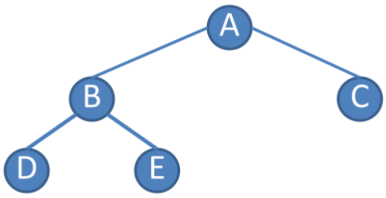
จากเทคนิคการค้นหาแบบลึกก่อนที่กล่าวมาข้างต้น จะเห็นได้ว่าวิธีนี้จะใช้ทรัพยากรในการค้นหาสถานะเป้าหมายน้อยกว่าเทคนิคการค้นหาแบบกว้างก่อน แต่ปัญหาของการค้นหาแบบลึกก่อน คือ จำนวนการกระทำจากสถานะเริ่มต้นไปยังสถานะเป้าหมายนั้นไม่ได้ใช้จำนวนการกระทำน้อยที่สุด และบางครั้งไม่สามารถหาสถานะเป้าหมายได้ทั้งๆ ที่สถานะเป้าหมายนั้นมีอยู่ในปริภูมิสถานะเนื่องจากเส้นทางในแนวลึกระหว่างการค้นหาอาจจะมีลูปอยู่ภายในทำให้วนลงลึกแบบไม่รู้จบ

เทคนิคการค้นหาแบบจำกัดความลึกจึงพัฒนาขึ้นมาเพื่อแก้ไขปัญหาดังกล่าวของเทคนิคการค้นหาแบบลึกก่อน โดยในการเริ่มต้นการค้นหานั้นจะมีการกำหนดระดับความลึกในการค้นหาเอาไว้ก่อน ตัวอย่างการใช้เทคนิคการค้นหาแบบจำกัดความลึก โดยกำหนดปริภูมิของสถานะจำลองของปัญหาดังแสดงในภาพที่ 2-14 กำหนดให้สถานะเริ่มต้นคือ สถานะ “A” และสถานะเป้าหมายคือ สถานะ “E” โดยขั้นตอนการค้นหาสามารถทำได้ตามตารางที่ 2-3 เมื่อจำกัดระดับความลึกที่ 1 ชั้น และตามตารางที่ 2-4 เมื่อจำกัดความลึกที่ 2 ชั้น (ต้นไม้ระดับแรกคือชั้นที่ 0)

ตารางที่ 2-3 ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบกำจัดความลึกที่ความลึก 1 ชั้น

	นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open • open = [A] • closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือ สถานะ “B” และ “C” จากนั้นนำสถานะ “A” ใส่ในตัวแปร closed และสถานะลูก “B” และ “C” ใส่ที่ด้านซ้ายของตัวแปร open • open = [B, C] • closed = [A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open เนื่องด้วย B อยู่ในชั้นที่ 1 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้ว จึงไม่สามารถสร้างสถานะลูก ของโหนด “B” ต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “B” ใส่ในตัวแปร closed ทันที • open = [C] • closed = [B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “C” ออกจากลิสท์ open เนื่องด้วย C อยู่ในชั้นที่ 1 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้ว จึงไม่สามารถสร้างสถานะลูก ของโหนด “C” ต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “C” ใส่ในตัวแปร closed ทันที เมื่อไม่มีสถานะใน open แล้วยังไม่พบเป้าหมาย ก็คือ**ไม่สามารถหาเป้าหมายได้** • open = [ ] • closed = [C, B, A]

ตารางที่ 2-4 ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบกำจัดความลึกที่ความลึก 2 ชั้น

	นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open • open = [A] • closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือ สถานะ “B” และ “C” จากนั้นนำสถานะ “A” ใส่ในตัวแปร closed และสถานะลูก “B” และ “C” ใส่ที่ด้านซ้ายของตัวแปร open • open = [B, C] • closed = [A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “B” ซึ่งก็คือ สถานะ “D” และ “E” จากนั้นนำสถานะ “B” ใส่ในตัวแปร closed และสถานะลูก “D” และ “E” ใส่ที่ด้านซ้ายของตัวแปร open • open = [D, E, C] • closed = [B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “D” ออกจากลิสท์ open เนื่องด้วย D อยู่ในชั้นที่ 2 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้วจึงไม่สามารถสร้างสถานะลูกของโหนด “D” ต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “D” ใส่ในตัวแปร closed ทันที • open = [E, C] • closed = [D, B, A]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “E” ออกจากลิสท์ open ตรวจสอบว่า “E” คือสถานะเป้าหมายหรือไม่ จากปัญหา E คือสถานะเป้าหมาย เพราะฉนั้นการค้นหาสถานะเป้าหมายจึงสิ้นสุดลง

เทคนิคการค้นหาแบบจำกัดลึกนั้น มีประสิทธิภาพดังนี้

- Completeness **ไม่**สามารถค้นหาสถานะเป้าหมายได้อย่างแน่นอน
- Optimality เส้นทางไปสู่สถานะเป้าหมาย**ไม่จำเป็น**ต้องใช้จำนวนการกระทำที่น้อยที่สุด
- Time Complexity $O(b^l)$
- Space Complexity O(bl)

เมื่อ b = จำนวนกิ่งเฉลี่ยของโหนด, l = ระดับความลึกของจำนวนชั้นที่ถูกจำกัดความลึก

2.4.4 เทคนิคการค้นหาแบบ Iterative Deepening Search

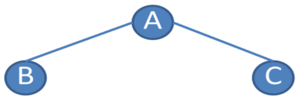
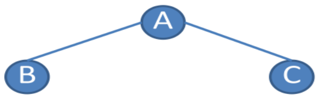
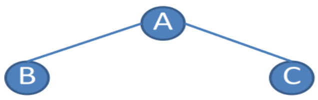
จากการปรับปรุงเทคนิคการค้นหาแบบลึกก่อนเป็นเทคนิคการค้นหาแบบจำกัดความลึกนั้น ยังไม่สามารถแก้ไขปัญหาเรื่อง Completeness และ Optimality ได้ ดังนั้นเพื่อให้แก้ปัญหาดังกล่าวได้ จึงได้มีการพัฒนาเทคนิคการค้นหาแบบ Iterative Deepening Search ขึ้นมาโดยเป็นการนำ เทคนิคการค้นหาแบบจำกัดความลึก มาพัฒนาต่อ ให้เริ่มทำการค้นหาโดยจำกัดความลึกที่ชั้น 0 ถ้าไม่พบสถานะเป้าหมายก็จะทำการเพิ่มระดับความลึกเป็น 1 ชั้น และเพิ่มขึ้นเรื่อยๆ ถ้าหากยังไม่สามารถพบ สถานะเป้าหมาย

ตัวอย่างการใช้เทคนิคการค้นหาแบบจำกัดความลึก โดยกำหนดปริภูมิของสถานะจำลองของปัญหาดังแสดงในภาพที่ 2-15 กำหนดให้สถานะเริ่มต้นคือ สถานะ “A” และสถานะเป้าหมายคือ สถานะ “E” โดยขั้นตอนการค้นหาสามารถทำได้ตามตารางที่ 2-5

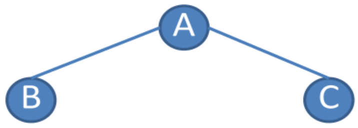
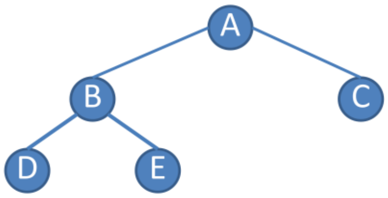
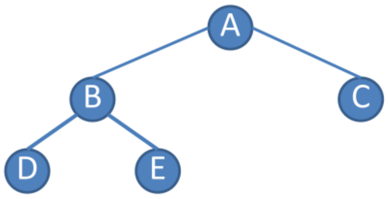
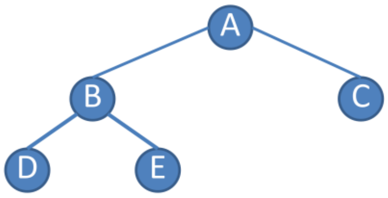
ตารางที่ 2-5 ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหา Iterative Deepening Search

	กำจัดความลึกที่ชั้น 0 : นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open • open = [A] • closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open เนื่องด้วย A อยู่ในชั้นที่ 0 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้ว จึงไม่สามารถสร้างสถานะลูกต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “A” ใส่ในตัวแปร closed ทันที เมื่อไม่มีสถานะใน open แล้วยังไม่พบเป้าหมาย ก็คือ**ไม่สามารถหาเป้าหมายได้** • open = [ ] • closed = [A]

ตารางที่ 2-5 (ต่อ) ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหา Iterative Deepening Search

 	เมื่อค้นหาด้วยระดับความลึกจำกัดที่ 0 ไม่พบเป้าหมายเพิ่มระดับความลึกจำกัดเป็น 1 **กำจัดความลึกที่ชั้น 1 :** นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open ● open = [A] ● closed = [ ]
 	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือ สถานะ “B” และ “C” จากนั้นนำสถานะ “A” ใส่ในตัวแปร closed และสถานะลูก “B” และ “C” ใส่ที่ด้านซ้ายของตัวแปร open ● open = [B, C] ● closed = [A]
 	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open เนื่องด้วย B อยู่ในชั้นที่ 1 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้ว จึงไม่สามารถสร้างสถานะลูก ของโหนด “B” ต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “B” ใส่ในตัวแปร closed ทันที ● open = [C] ● closed = [B, A]
 	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “C” ออกจากลิสท์ open เนื่องด้วย C อยู่ในชั้นที่ 1 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้ว จึงไม่สามารถสร้างสถานะลูก ของโหนด “C” ต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “C” ใส่ในตัวแปร closed ทันที เมื่อไม่มีสถานะใน open แล้วยังไม่พบเป้าหมาย ก็คือ**ไม่สามารถหาเป้าหมายได้** ● open = [ ] ● closed = [C, B, A]

ตารางที่ 2-5 (ต่อ) ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหา Iterative Deepening Search

<table>
<tr>
<td>
</td>
<td>เมื่อค้นหาด้วยระดับความลึกจำกัดที่ 1 ไม่พบเป้าหมายเพิ่มระดับความลึกจำกัดเป็น 2
กำจัดความลึกที่ชั้น 2 : นำสถานะเริ่มต้น “A” ใส่ในตัวแปร open
open = [A]
closed = [ ]</td>
</tr>
<tr>
<td>
</td>
<td>นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือ สถานะ “B” และ “C” จากนั้นนำสถานะ “A” ใส่ในตัวแปร closed และสถานะลูก “B” และ “C” ใส่ที่ด้านซ้ายของตัวแปร open
open = [B, C]
closed = [A]</td>
</tr>
<tr>
<td>
</td>
<td>นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “B” ซึ่งก็คือ สถานะ “D” และ “E” จากนั้นนำสถานะ “B” ใส่ในตัวแปร closed และสถานะลูก “D” และ “E” ใส่ที่ด้านซ้ายของตัวแปร open
open = [D, E, C]
closed = [B, A]</td>
</tr>
<tr>
<td>
</td>
<td>นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “D” ออกจากลิสท์ open เนื่องด้วย D อยู่ในชั้นที่ 2 ของต้นไม้ซึ่งได้ถูกจำกัดเอาไว้แล้วจึงไม่สามารถสร้างสถานะลูกของโหนด “D” ต่อไปได้อีกแล้ว ดังนั้นให้นำสถานะ “D” ใส่ในตัวแปร closed ทันที
open = [E, C]
closed = [D, B, A]</td>
</tr>
<tr>
<td>
</td>
<td>นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “E” ออกจากลิสท์ open ตรวจสอบว่า “E” คือสถานะเป้าหมายหรือไม่ จากปัญหา E คือสถานะเป้าหมาย เพราะฉนั้นการค้นหาสถานะเป้าหมายจึงสิ้นสุดลง</td>
</tr>
</table>

เทคนิคการค้นหาแบบ Iterative Deepening Search มีประสิทธิภาพดังนี้

- Completeness สามารถค้นหาสถานะเป้าหมายได้อย่างแน่นอน
- Optimality เส้นทางไปสู่สถานะเป้าหมายใช้การกระทำน้อยที่สุด
- Time Complexity $O(b^d)$
- Space Complexity $O(b^d)$

เมื่อ b = จำนวนกิ่งเฉลี่ยของโหนด, d = ความลึกของต้นไม้ในระดับที่หา

ซึ่งจะเห็นได้ว่าการเทคนิคการค้นหาแบบ Iterative Deepening Search ได้นำข้อดีของเทคนิคการค้นหาแบบกว้างก่อนและแบบลึกก่อนมาผสมผสานกัน ทำให้ได้เทคนิคการค้นหาที่สามารถค้นหาสถานะเป้าหมายได้อย่างแน่นอนถ้ามีสถานะเป้าหมายนั้นอยู่ในปริภูมิสถานะ และเส้นทางที่ได้จากการค้นหาเพื่อไปยังสถานะเป้าหมายจะใช้การกระทำของเซต Successor Function น้อยที่สุด รวมถึงการใช้เวลาและทรัพยากร ในการค้นหาเหมาะสมไม่มากเท่ากับการค้นหาแบบกว้างก่อน

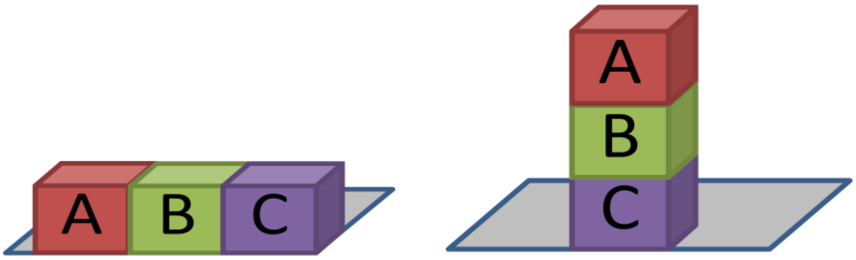
2.4.5 ตัวอย่างการประยุกต์ใช้เทคนิคการค้นหาแบบไม่มีการชี้แนะ

ปัญหาโลกของบล็อก (Block world problem) คือมีการกำหนดสถานะเริ่มต้นดังแสดงในภาพที่ 2-17(ก) และต้องการทางการกระทำที่จะทำให้ได้สถานะเป้าหมายดังภาพที่ 2-17(ข) โดยตัวปัญหาคือมีข้อกำหนดดังนี้

(1) ถ้าบล็อก X ไม่มีบล็อกอื่นทับสามารถวางบล็อก X บนโต๊ะได้

(2) ถ้าบล็อก X และ Y ไม่มีบล็อกอื่นทับสามารถวางบล็อกวาง X บนบล็อก Y ได้

โดยที่ X และ Y เป็นตัวแปรและสามารถแทนที่ด้วย 'A', 'B' หรือ 'C' เช่นเมื่อใช้ตัวกระทำการ (1) กับสถานะเริ่มต้น จะได้ว่าถ้าบล็อก 'A' ไม่มีบล็อกอื่นทับ แล้ววาง 'A' บนโต๊ะได้ เป็นต้น



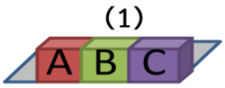
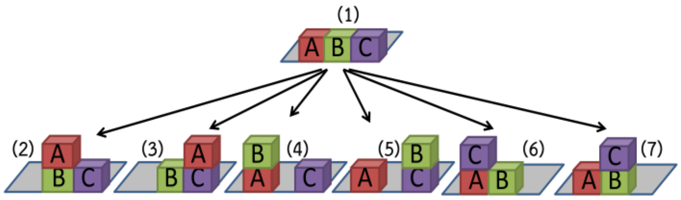
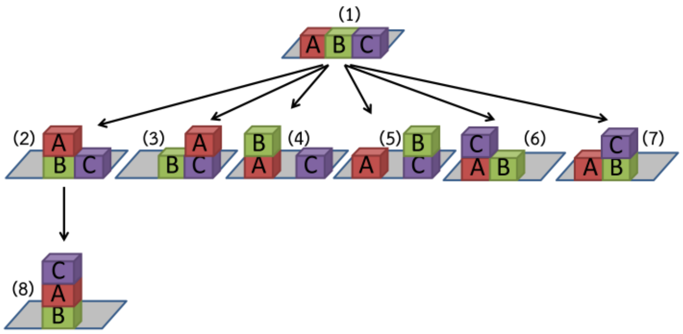
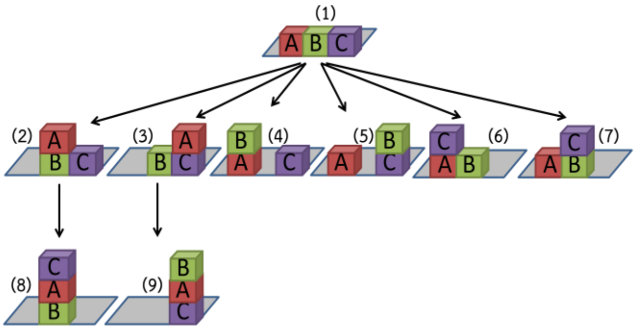
(ก) สถานะเริ่มต้น (ข) สถานะเป้าหมาย

ภาพที่ 2-17 ปัญหาโลกของบล็อก (Block World Problem)

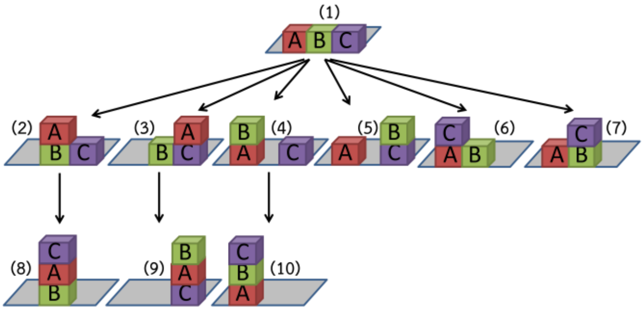
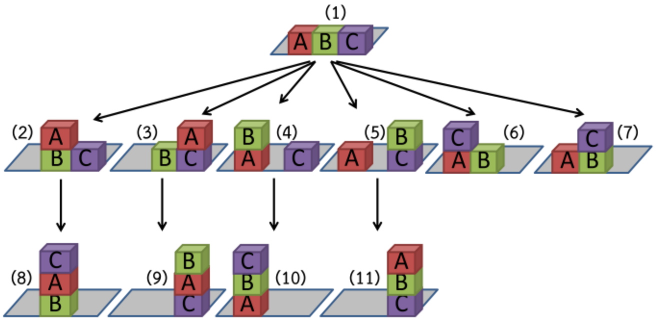
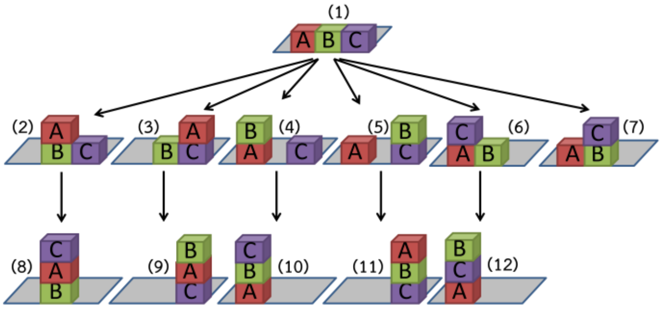
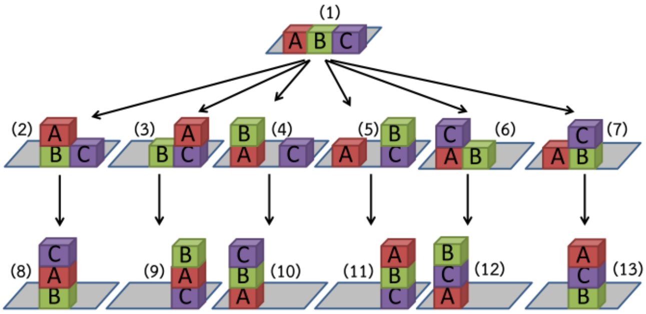
2.4.5.1 การค้นหาด้วยเทคนิคการค้นหาแบบกว้างก่อน

ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบกว้างก่อน เพื่อแก้ปัญหาโลกของบล็อก แสดงดังตารางที่ 2-6

ตารางที่ 2-6 ขั้นตอนการแก้ปัญหาโลกของบล็อกด้วยเทคนิคการค้นหาแบบกว้างก่อน

	นำสถานะเริ่มต้น (1) ใส่ในตัวแปร open • open = [1] • closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (1) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (1) ซึ่งก็คือ สถานะ (2), (3), (4), (5), (6) และ (7) จากนั้นนำสถานะ (1) ใส่ในตัวแปร closed และสถานะลูกของ (1) ใส่ที่ด้านขวาของตัวแปร open • open = [**2, 3, 4, 5, 6, 7**] • closed = [**1**]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (2) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (2) ซึ่งก็คือ สถานะ (8) จากนั้นนำสถานะ (2) ใส่ในตัวแปร closed และสถานะลูกของ (2) ใส่ที่ด้านขวาของตัวแปร open • open = [3, 4, 5, 6, 7, **8**] • closed = [**2**, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (3) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (3) ซึ่งก็คือ สถานะ (9) จากนั้นนำสถานะ (3) ใส่ในตัวแปร closed และสถานะลูกของ (3) ใส่ที่ด้านขวาของตัวแปร open • open = [4, 5, 6, 7, 8, **9**] • closed = [**3**, 2, 1]

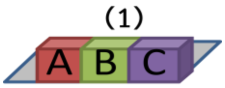
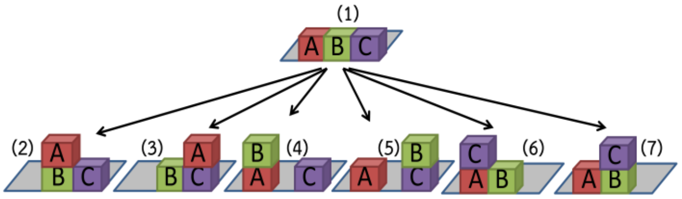
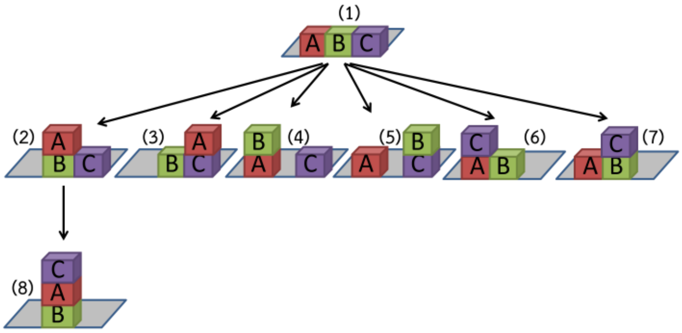
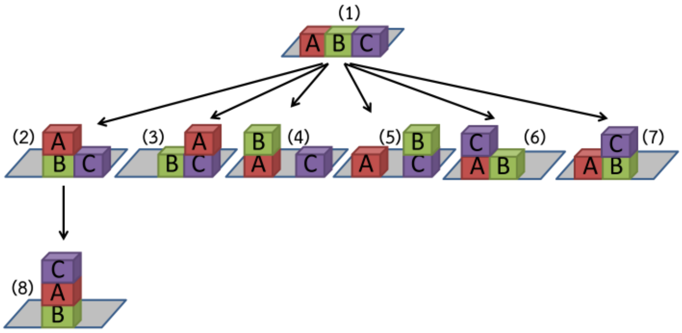
ตารางที่ 2-6 (ต่อ) ขั้นตอนการแก้ปัญหาโลกของบล็อกด้วยเทคนิคการค้นหาแบบกว้างก่อน

	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (4) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (4) ซึ่งก็คือ สถานะ (10) จากนั้นนำสถานะ (4) ใส่ในตัวแปร closed และสถานะลูกของ (4) ใส่ที่ด้านขวาของตัวแปร open • open = [5, 6, 7, 8, 9, **10**] • closed = [**4**, 3, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (5) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (5) ซึ่งก็คือ สถานะ (11) จากนั้นนำสถานะ (5) ใส่ในตัวแปร closed และสถานะลูกของ (5) ใส่ที่ด้านขวาของตัวแปร open • open = [6, 7, 8, 9, 10, **11**] • closed = [**5**, 4, 3, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (6) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (6) ซึ่งก็คือ สถานะ (12) จากนั้นนำสถานะ (6) ใส่ในตัวแปร closed และสถานะลูกของ (6) ใส่ที่ด้านขวาของตัวแปร open • open = [7, 8, 9, 10, 11, **12**] • closed = [**6**, 5, 4, 3, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (7) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (7) ซึ่งก็คือ สถานะ (13) จากนั้นนำสถานะ (7) ใส่ในตัวแปร closed และสถานะลูกของ (7) ใส่ที่ด้านขวาของตัวแปร open • open = [8, 9, 10, 11, 12, **13**] • closed = [**7**, 6, 5, 4, 3, 2, 1]
หลังจากขั้นตอนนี้จะได้ปริภูมิสถานะที่สมบูรณ์แล้วจึงไม่มีการแตกสถานะของลูกต่อและสถานะ (11) คือสถานะเป้าหมายเมื่อทำอัลกอริทึมให้สมบูรณ์จะได้ดังนี้	• open = [9, 10, 11, 12, 13] • closed = [**8**, 7, 6, 5, 4, 3, 2, 1] • open = [10, 11, 12, 13] • closed = [**9**, 8, 7, 6, 5, 4, 3, 2, 1] • open = [11, 12, 13] • closed = [**10**, 9, 8, 7, 6, 5, 4, 3, 2, 1]

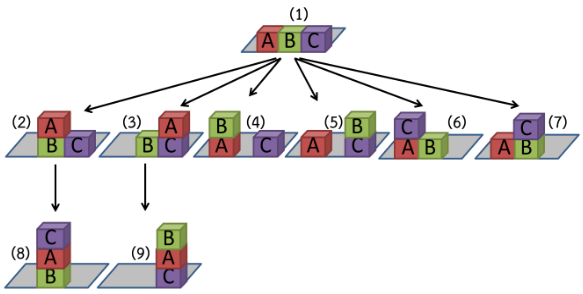
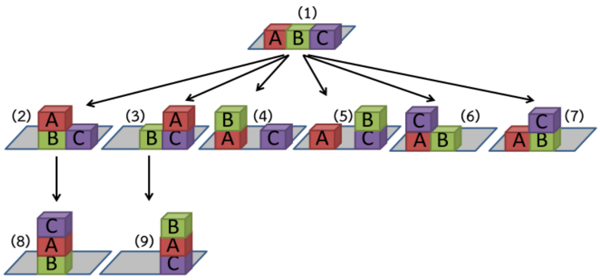
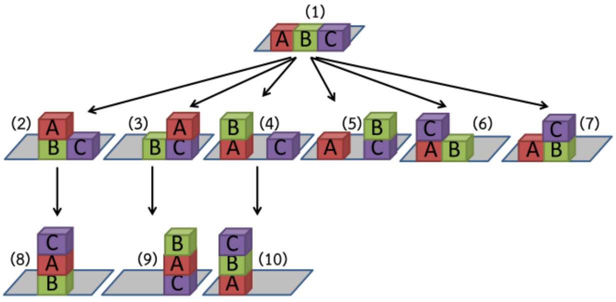
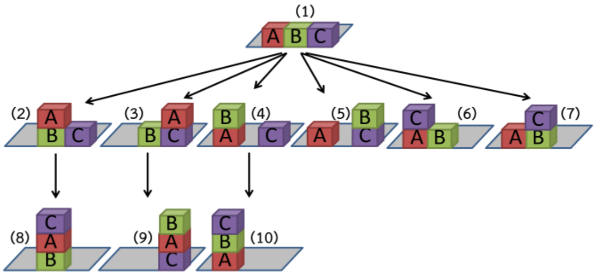
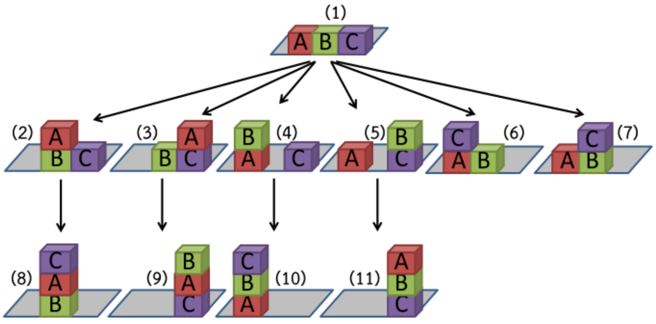
2.4.5.2 การค้นหาด้วยเทคนิคการค้นหาแบบลึกก่อน

ขั้นตอนการค้นหาสถานะเป้าหมายด้วยเทคนิคการค้นหาแบบลึกก่อน เพื่อแก้ปัญหาโลกของบล็อก แสดงดังตารางที่ 2-7

ตารางที่ 2-6 ขั้นตอนการแก้ปัญหาโลกของบล็อกด้วยเทคนิคการค้นหาแบบลึกก่อน

	นำสถานะเริ่มต้น (1) ใส่ในตัวแปร open ● open = [1] ● closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (1) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (1) ซึ่งก็คือ สถานะ (2), (3), (4), (5), (6) และ (7) จากนั้นนำสถานะ (1) ใส่ในตัวแปร closed และสถานะลูกของ (1) ใส่ที่ด้านซ้ายของตัวแปร open ● open = [2, 3, 4, 5, 6, 7] ● closed = [1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (2) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (2) ซึ่งก็คือ สถานะ (8) จากนั้นนำสถานะ (2) ใส่ในตัวแปร closed และสถานะลูกของ (2) ใส่ที่ด้านซ้ายของตัวแปร open ● open = [8, 3, 4, 5, 6, 7] ● closed = [2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (8) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (8) แต่ทว่าไม่มีสถานะลูกที่สามารถสร้างได้แล้ว จึงนำสถานะ (8) ใส่ในตัวแปร closed ● open = [3, 4, 5, 6, 7] ● closed = [8, 2, 1]

ตารางที่ 2-6 (ต่อ) ขั้นตอนการแก้ปัญหาโลกของบล็อกด้วยเทคนิคการค้นหาแบบลึกก่อน

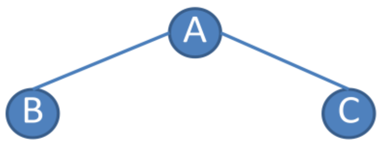
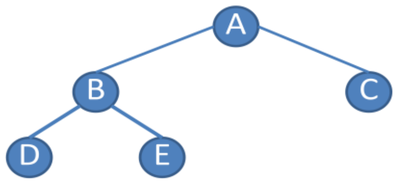
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (3) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (3) ซึ่งก็คือ สถานะ (9) จากนั้นนำสถานะ (3) ใส่ในตัวแปร closed และสถานะลูกของ (3) ใส่ที่ด้านซ้ายของตัวแปร open • open = [**9**, 4, 5, 6, 7] • closed = [**3**, 8, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (9) ออกจากลิสท์ open แต่สถานะ (9) ไม่มีสถานะลูกที่สามารถสร้างได้แล้ว จึงนำสถานะ (9) ใส่ในตัวแปร closed • open = [4, 5, 6, 7] • closed = [**9**, 3, 8, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (4) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (4) ซึ่งก็คือ สถานะ (10) จากนั้นนำสถานะ (10) ใส่ในตัวแปร closed และสถานะลูกของ (10) ใส่ที่ด้านซ้ายของตัวแปร open • open = [**10**, 5, 6, 7] • closed = [**4**, 9, 3, 8, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (10) ออกจากลิสท์ open แต่สถานะ (10) ไม่มีสถานะลูกที่สามารถสร้างได้แล้ว จึงนำสถานะ (10) ใส่ในตัวแปร closed • open = [5, 6, 7] • closed = [**10**, 4, 9, 3, 8, 2, 1]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ (5) ออกจากลิสท์ open แล้วสร้างสถานะลูกของสถานะ (5) ซึ่งก็คือ สถานะ (11) จากนั้นนำสถานะ (5) ใส่ในตัวแปร closed และสถานะลูกของ (5) ใส่ที่ด้านซ้ายของตัวแปร open • open = [**11**, 6, 7] • closed = [**5**, 10, 4, 9, 3, 8, 2, 1]

2.5 การปรับปรุงเทคนิคการค้นหาในการจำเส้นทางในการเดินทางไปยังเป้าหมาย

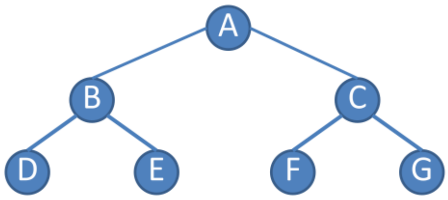
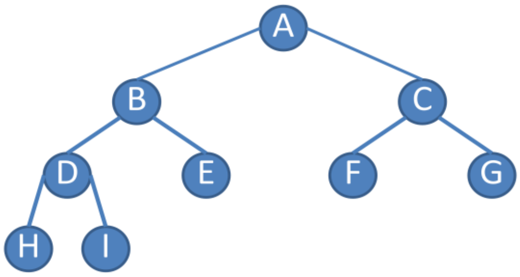
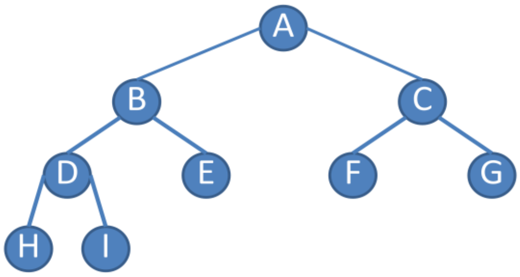
จากเทคนิคการค้นหาที่กล่าวมาทั้งหมด จะเห็นได้ว่าการค้นหาที่สามารถพบสถานะเป้าหมายได้นั้นไม่สามารถบ่งบอกได้ว่าจะต้องใช้เส้นทางการกระทำอะไรบ้างหรือต้องผ่านสถานะในบ้างเพื่อให้ไปถึงสถานะเป้าหมาย ดังนั้นจึงมีการปรับปรุงวิธีการเขียนค่าสถานะในตัวแปร open และ closed เพื่อให้รู้ถึงเส้นทางที่ใช้ในการเดินทางจากสถานะเริ่มต้นไปยังสถานะเป้าหมาย โดยปรับเปลี่ยนจากแทนที่จะใช้สถานะเพียงอย่างเดียว เช่น A, B, C, 1, 2, 3 ให้อยู่ในรูปแบบดังนี้ *(สถานะปัจจุบัน, สถานะก่อนหน้า)* โดยที่สถานะเริ่มต้นนั้นให้มี สถานะก่อนหน้า เป็น nil (หรือ null)

เพื่อให้เข้าใจได้ง่ายขึ้นจึงขอยกตัวอย่างปัญหาจากภาพที่ 2-15 ด้วยวิธีการค้นหาแบบกว้างก่อน ดังแสดงในตารางที่ 2-7

ตารางที่ 2-7 ขั้นตอนการค้นหาเทคนิคการค้นหาแบบกว้างก่อนโดยมีการกำกับสถานะก่อนหน้า

	นำสถานะเริ่มต้น “A” ที่ไม่มีสถานะก่อนหน้า จึงเป็น (A, nil) ใส่ในตัวแปร open ● open = [**(A, nil)**] ● closed = [ ]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “A” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “A” ซึ่งก็คือสถานะ (B, A) และ (C, A) จากนั้นนำสถานะ (A,nil) ใส่ในตัวแปร closed และสถานะ (B, A) และ (C, A) ใส่ที่ด้านขวาของตัวแปร open ● open = [**(B, A)**, **(C, A)**] ● closed = [**(A, nil)**]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “B” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “B” ซึ่งก็คือสถานะ (D, B) และ (E, B) จากนั้นนำสถานะ (B, A) ใส่ในตัวแปร closed และสถานะ (D, B) และ (E, B) ใส่ที่ด้านขวาของตัวแปร open ● open = [(C, A), **(D, B)**, **(E, B)**] ● closed = [**(B, A)**, (A, nil)]

ตารางที่ 2-7 (ต่อ) ขั้นตอนการค้นหาเทคนิคการค้นหาแบบกว้างก่อนโดยมีการกำกับสถานะก่อนหน้า

	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “C” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “C” ซึ่งก็คือสถานะ (F, C) และ (G, C) จากนั้นนำสถานะ (C, A) ใส่ในตัวแปร closed และสถานะ (F, C) และ (G, C) ใส่ที่ด้านขวาของตัวแปร open ● open = [(D, B), (E, B), **(F, C)**, **(G, C)**] ● closed = [**(C, A)**, (B, A), (A, nil)]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “D” ออกจากลิสท์ open แล้วสร้างสถานะลูกของโหนด “D” ซึ่งก็คือสถานะ (H, D) และ (I, D) จากนั้นนำสถานะ “D” ใส่ในตัวแปร closed และสถานะ (H, D) และ (I, D) ใส่ที่ด้านขวาของตัวแปร open ● open = [(E, B), (F, C), (G, C), **(H, D)**, **(I, D)**] ● closed = [**(D, B)**, (C, A), (B, A), (A, nil)]
	นำสถานะซ้ายสุดของ open ซึ่งก็คือสถานะ “E” ออกจากลิสท์ open ตรวจสอบว่า “E” คือสถานะเป้าหมายหรือไม่ จากปัญหา E คือสถานะเป้าหมาย เพราะฉนั้นการค้นหาสถานะเป้าหมายจึงสิ้นสุดลง

เมื่อการค้นหาสิ้นสุดลง ให้พิจารณาจากสถานะเป้าหมายที่ได้ ในที่นี้สถานะเป้าหมายก็คือ E ที่มีสถานะก่อนหน้าคือ B ที่เขียนอยู่ในรูป (E, B) จากนั้นให้พิจารณาที่สถานะในตัวแปร closed โดยขั้นตอนสุดท้ายจะได้สถานะที่อยู่ในตัวแปร closed ดังนี้

closed = [(D, B), (C, A), (B, A), (A, nil)]

ให้ดูจากสถานะใน closed ว่า สถานะ B ที่เป็นสถานะก่อนหน้าของสถานะ E นั้น มีสถานะก่อนหน้าคือสถานะอะไร ซึ่งจะพบจากคู่ (B, A) หมายความว่าสถานะ A คือสถานะก่อนหน้าของ B แล้วให้ไล่ต่อไปเพื่อหาสถานะก่อนหน้าของ A จะพบคู่ (A, nil) ซึ่งหมายถึงสถานะ A เป็นสถานะเริ่มต้นก็การค้นหาแล้ว ดังนั้นเส้นทางที่ใช้ในการเดินทางจากสถานะเริ่มต้นไปยังสถานะเป้าหมายให้ไล่กลับอีกรอบ ซึ่งก็คือ เริ่มจากสถานะ A แล้วไปยังสถานะ B แล้วไปยังสถานะ E ที่เป็นสถานะเป้าหมาย

2.6 สรุป

ในบทนี้ได้กล่าวถึงการแทนปัญหา (Problem Representation) ซึ่งเป็นแนวคิดที่ทำให้ปัญญาประดิษฐ์สามารถเข้าใจปัญหาต่างๆ เพื่อนำไปสู่วิธีการแก้ไขปัญหาได้ การนำปัญหามาแปลงให้อยู่ในรูปแบบ Problem Formulation จะช่วยทำให้เกิดความเข้าใจถึงสภาพแวดล้อมของปัญหาที่เกิดขึ้นอย่างชัดเจน โดยข้อมูลที่ได้จาก Problem Formulation สามารถนำมาสร้างเป็นปริภูมิสถานะ (State Space Representation) ซึ่งเป็นวิธีที่นิยมนำไปใช้แทนปัญหา เนื่องจากมีลักษณะเป็นกราฟสามารถนำไปสู่สถานะเป้าหมายได้

สำหรับเทคนิคการค้นหาแบบไม่มีการชี้แนะ เป็นเทคนิคการค้นหาที่ไม่มีข้อมูลมาร่วมประกอบการพิจารณา ได้แก่ เทคนิคการค้นหาแบบกว้างก่อน (Breath-First Search: BFS), เทคนิคการค้นหา แบบลึกก่อน (Depth-First Search: DFS), เทคนิคการค้นหาแบบจำกัดความลึก (Depth-Limited Search: DLS) และ เทคนิคการ ค้นหาแบบ Iterative Deepening Search (IDS) ซึ่งมีคุณสมบัติดังนี้

คุณสมบัติ	BFS	DFS	DLS	IDS
Completeness	Yes	No	No	Yes
Optimality	Yes	No	No	Yes
Time Complexity	$O(b^{d+1})$	$O(b^m)$	$O(b^l)$	$O(b^d)$
Space Complexity	$O(b^{d+1})$	O(bm)	O(bl)	O(bd)

โดย b คือ จำนวนกิ่งเฉลี่ยของโหนด

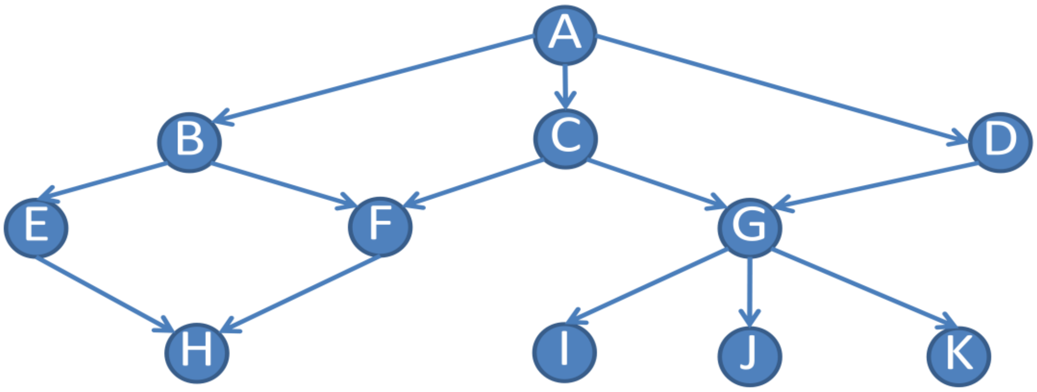
d คือ ความลึกของต้นไม้ในระดับที่หา

m คือ ความลึกมากสุดของต้นไม้

l คือ ระดับความลึกของจำนวนชั้นที่ถูกจำกัดความลึก

แบบฝึกหัด

1. อธิบายข้อแตกต่างระหว่างเทคนิคการค้นหาแบบ Breath-First Search และ Depth-First Search
2. เพราะเหตุใดการค้นหาด้วยวิธี Depth-First Search จึงไม่มีคุณสมบัติ Completeness และ Optimality
3. จาก graph ต่อไปนี้จงแสดงวิธีการหาโหนดเป้าหมาย G จากโหนดเริ่มต้น A ด้วยวิธี BFS และ DFS ในแต่ละขั้นตอนระบุหมายเลขขั้นตอน, ค่าใน open, และ ค่าใน close รวมถึงให้การเก็บข้อมูลรวมข้อมูลเกี่ยวกับ node แม่ด้วย และบอกเส้นทางที่ดีที่สุดในการไปถึงเป้าหมาย



แผนปฏิบัติการสอนสัปดาห์ที่ 3

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	รายละเอียดตามที่ระบุไว้ใน
1. ปัญหาของการค้นหาแบบไม่มีการชี้แนะ	IS 3.1
2. เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย (ฮิวริสติก)	IS 3.2
3. เทคนิคการค้นหาแบบ GBFS	IS 3.3
4. เทคนิคการค้นหาแบบ A*	IS 3.4
5. การกำหนดฟังก์ชันฮิวริสติก	IS 3.5
6. อัลกอริทึมแบบ Local Search	IS 3.6

ข. การจัดการเรียนการสอน

เวลา – นาที		0–60	60–120	120–180
วัตถุประสงค์		**1-2-3-4-5-6**		
การนำเข้าสู่บทเรียน		■ (0–15)		
ให้เนื้อหา		■ (15–60)	■	
สรุปเนื้อหา				■ (120–135)
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**		
วิธีการสอน:	บรรยาย	■ (15–40)		
	อธิบาย	■ (40–60)	■	
	ถาม – ตอบ		■ (90–120)	
	ปฏิบัติด้วยตนเอง			■ (135–180)
สื่อการสอน:	กระดานดำ	■	■	
	คอมพิวเตอร์	■	■	■
	โปรเจคเตอร์	■	■	■

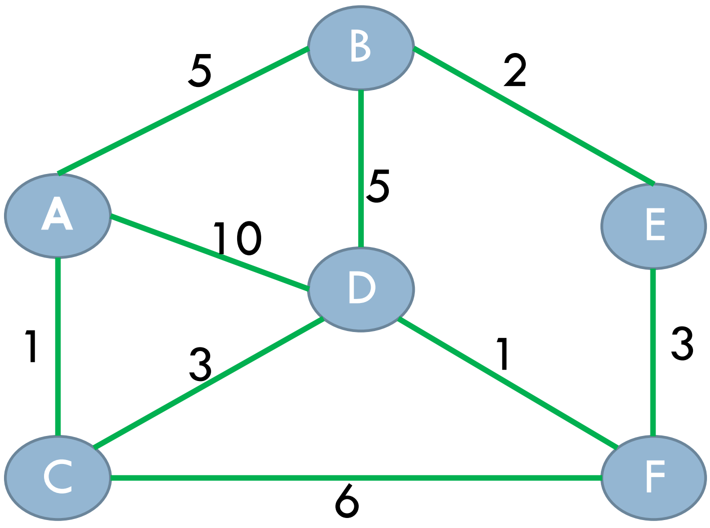
บทที่ 3
เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย

เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย หรือการค้นหาแบบฮิวริสติก (Heuristic Search) เป็นเทคนิคการค้นหาแบบมีข้อมูลช่วย (Informed) โดยนำข้อมูลมาประกอบการพิจารณาเพื่อช่วยเพิ่มประสิทธิภาพการแก้ปัญหาหรือหาคำตอบที่ดีที่สุด โดยอัลกอริทึมแบบฮิวริสติกที่จะกล่าวถึงในบทนี้ด้วยกันทั้งหมด 4 อัลกอริทึมคือ Greedy Best First Search, A*, Hill Climbing และ Simulated Annealing Search

3.1 ปัญหาของการค้นหาแบบไม่มีการชี้แนะ

เทคนิคการค้นหาแบบไม่มีการชี้แนะที่ได้กล่าวถึงในบทที่ 2 จะไม่สามารถแก้ปัญหาในกรณีที่ปัญหามีค่า Path Cost (ทรัพยากรที่ใช้ในการกระทำต่อสถานะเพื่อเปลี่ยนสถานะ) ไม่เท่ากัน

ยกตัวอย่างปัญหาการหาเส้นทางเดินทางจากเมือง A ไปยังเมือง F ดังแสดงในภาพที่ 3-1 โดยแต่ละเมืองจะมีระยะทางที่ใช้ในการเดินทางระหว่างเมือง เช่น ระยะทางจากเมือง A ไปยังเมือง B ใช้ระยะทาง 5 หน่วย และระยะทางระหว่างเมือง A ไปยังเมือง D คือ 10 หน่วย เป็นต้น



ภาพที่ 3-1 ตัวอย่างแผนผังเมืองและระยะทางที่ใช้ในการเดินทางระหว่างเมืองต่างๆ

3.1.1 แก้ปัญหาด้วยเทคนิคการค้นหาแบบลึกก่อน

เมื่อใช้เทคนิคการค้นหาแบบลึกก่อนโดยสถานะเริ่มต้นคือ เมือง A จะมีขั้นตอนในการค้นหาดังต่อไปนี้

ขั้นตอนที่	สถานะในตัวแปร open และ closed
1	open = [**(A, nil)**] closed = [ ]
2	open = [**(B, A)**, **(C, A)**, **(D, A)**] closed = [**(A, nil)**]
3	open = [**(E, B)**, (C, A), (D, A)] closed = [**(B, A)**, (A, nil)]
4	open = [**(F, E)**, (E, B), (C, A)] closed = [**(E, B)**, (B, A), (A, nil)]

โดยเมื่อพิจารณาเส้นทางที่ใช้ในการเดินทางจากสถานะเริ่มต้นถึงสถานะเป้าหมายจะได้ว่า สถานะเป้าหมาย F ที่ได้จาก (F, E) คือมีสถานะก่อนหน้าคือ E และสถานะ E มีสถานะก่อนหน้าคือ B จากคู่ (E, B) และสถานะ B มีสถานะก่อนหน้าคือ A จากคู่ (B, A) และสุดท้ายคือสถานะ (A, nil) หมายถึงสถานะ A เป็นสถานะเริ่มต้น

(F, E) → (E, B) → (B, A) → (A, nil) ก็คือ A → B → E → F

จึงทำให้รู้เส้นทางในการเดินทางโดยเริ่มต้นจากสถานะ A เดินทางไปยังสถานะ B เดินทางต่อไปยังสถานะ E แล้วเดินทางไปยังสถานะเป้าหมาย ซึ่งก็คือสถานะ F เมื่อทำการคำนวณค่า Path Cost ทั้งหมดจะได้ว่า

- ระยะทางจาก A ไป B คือ 5 หน่วย
- ระยะทางจาก B ไป E คือ 2 หน่วย
- ระยะทางจาก E ไป F คือ 3 หน่วย
- ดังนั้นระยะทางโดยรวม (Total Path Cost) คือ 5 + 2 + 3 = 10 หน่วย

3.1.2 แก้ปัญหาด้วยเทคนิคการค้นหาแบบกว้างก่อน

เมื่อใช้เทคนิคการค้นหาแบบกว้างก่อนโดยสถานะเริ่มต้นคือ เมือง A จะมีขั้นตอนในการค้นหาดังต่อไปนี้

ขั้นตอนที่	สถานะในตัวแปร open และ closed
1	open = [**(A, nil)**] closed = [ ]
2	open = [**(B, A), (C, A), (D, A)**] closed = [**(A, nil)**]
3	open = [(C, A), (D, A), **(E, B)**] closed = [**(B, A)**, (A, nil)]
4	open = [(D, A), (E, B), **(F, C)**] closed = [**(C, A)**, (B, A), (A, nil)]
5	open = [(E, B), (F, C)] closed = [**(D, A)**, (C, A), (B, A), (A, nil)]
6	open = [(F, C)] closed = [**(E, B)**, (D, A), (C, A), (B, A), (A, nil)]

โดยเมื่อพิจารณาเส้นทางที่ใช้ในการเดินทางจากสถานะเริ่มต้นถึงสถานะเป้าหมายจะได้ว่า สถานะเป้าหมาย F ที่ได้จาก (F, C) คือมีสถานะก่อนหน้าคือ C และสถานะ C มีสถานะก่อนหน้าคือ A จากคู่ (C, A) และ สถานะ (A, nil) หมายถึงสถานะ A เป็นสถานะเริ่มต้น

(F, C) → (C, A) → (A, nil) ก็คือ A → C → F

จึงทำให้รู้เส้นทางในการเดินทางโดยเริ่มต้นจากสถานะ A เดินทางไปยังสถานะ C แล้วเดินทางไปยังสถานะเป้าหมาย ซึ่งก็คือสถานะ F เมื่อทำการคำนวณค่า Path Cost ทั้งหมดจะได้ว่า

- ระยะทางจาก A ไป C คือ 1 หน่วย
- ระยะทางจาก C ไป F คือ 6 หน่วย
- ดังนั้นระยะทางโดยรวม (Total Path Cost) คือ 1 + 6 = 7 หน่วย

ในขณะที่การค้นหาแบบกว้างก่อนนั้น รับประกันการค้นหาเส้นทางที่สั้นที่สุดไปยังสถานะเป้าหมายซึ่งในปัญหานี้ก็คือ เป็นการค้นหเส้นทางที่ไปยังเป้าหมายโดยเดินทางผ่านเมืองให้น้อยที่สุดแต่ไม่ได้รับประกันว่า ระยะทางที่ใช้นั้นจะน้อยที่สุดเสมอ

3.1.3 สรุปปัญหาที่เกิดจากการใช้เทคนิคการค้นหาแบบไม่มีการชี้แนะ

จากที่เห็นผลลัพธ์ที่เกิดจากการแก้ปัญหาด้วยเทคนิคการค้นหาแบบกว้างก่อนและแบบลึกก่อนนั้น ผลลัพธ์ที่ได้ไม่ใช่ผลลัพธ์ที่ใช้ทรัพยากรในการเดินทางไปยังเป้าหมายที่น้อยที่สุด (Total Path Cost) เนื่องด้วยเทคนิคการค้นหาแบบลึกก่อนแนะนำให้เดินทางจากสถานะ A ไปยังสถานะ F ด้วยเส้นทาง A → B → E → F ที่ใช้ระยะทางโดยรวมเท่ากับ 10 หน่วย ส่วนเทคนิคการค้นหาแบบกว้างก่อนที่รับประกันว่าเส้นทางที่ได้จากการค้นหาจะใช้การกระทำน้อยที่สุดแนะนำให้ใช้เส้นทาง A → C → F ที่มีระยะทางโดยรวมเท่ากับ 7 หน่วย แต่อย่างไรก็ตามถ้าพิจารณาเส้นทางในภาพที่ 3-1 ให้ดี จะพบว่าเส้นทางที่ดีที่สุดที่จะเดินทางจากสถานะ A ไปยังสถานะ F ควรใช้เส้นทาง A → C → D → F เนื่องจากระยะทางที่ใช้จะมีดังนี้

- ระยะทางจาก A ไป C คือ 1 หน่วย
- ระยะทางจาก C ไป D คือ 3 หน่วย
- ระยะทางจาก D ไป F คือ 1 หน่วย
- ดังนั้นระยะทางโดยรวม (Total Path Cost) คือ 1 + 3 + 1 = 5 หน่วย

3.2 เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วย (ฮิวริสติก)

การค้นหาประเภทฮิวริสติกนี้จะใช้ความรู้แบบหนึ่งที่เรียกว่าฮิวริสติกมาช่วยในการค้นหาให้มีประสิทธิภาพมากขึ้น โดยฮิวริสติกตัวนี้จะช่วยชี้แนะว่ากระบวนการค้นหาควรจะเลือกเส้นทางใดหรือสถานะใดเพื่อทำการค้นหาต่อไปให้ได้คำตอบอย่างมีประสิทธิภาพ พิจารณาปัญหาการเดินทางของพนักงานขาย (traveling salesman problem) ตัวอย่างของปัญหานี้ เช่น มีเมือง 7 เมือง พนักงานขายต้องการเดินทางไปให้ได้ครบทั้ง 7 เมืองและกลับมายังจุดเริ่มต้นโดยให้ได้ระยะทางโดยรวมสั้นที่สุด วิธีหนึ่งที่ทำได้คือ หาเส้นทางทั้งหมดที่เป็นไปได้ซึ่งจะมีด้วยกันทั้งสิ้น (7-1)!/2 =360 แบบ จากนั้นวัดแต่ละเส้นทางว่าใช้ระยะทางเท่าไร แล้วก็เลือกเส้นทางที่สั้นที่สุด วิธีการนี้ไม่สามารถคำนวณได้อย่างมีประสิทธิภาพในทางปฏิบัติ เมื่อจำนวนเมืองมีมากขึ้น เช่นถ้ามีเมือง 100 เมือง จะมีเส้นทางที่เป็นไปได้ทั้งสิ้น 4.67×10^{155} แบบถ้าเราใช้สามัญสำนึกโดยคาดเดาอย่างมีเหตุผลว่า เมื่อเราต้องการระยะทางโดยรวมสั้นที่สุด เราก็น่าจะเลือกเมืองที่อยู่ใกล้มากที่สุดกับเมืองที่เราอยู่ในปัจจุบัน แล้วเดินทางไปเมืองนั้นก่อน เมื่อไปถึงเมืองนั้นแล้วค่อยทำในทำนองเดียวกันอีกว่า จะเดินไปยังเมืองที่ใกล้ที่สุดเมืองถัดไป ทำเช่นนี้จนกระทั่งเดินทางครบทุกเมือง ก็น่าจะได้ระยะทางโดยรวมสั้นที่สุดแม้ว่าวิธีการเช่นนี้จะทำงานได้อย่างมีประสิทธิภาพ และคำตอบที่ได้มีแนวโน้มว่าจะดี แต่อย่างไรก็ดี คำตอบที่ได้โดยวิธีนี้อาจไม่เป็นเส้นทางที่สั้นที่สุดก็ได้ วิธีการเช่นนี้ก็คือการนำความรู้แบบหนึ่งมาแก้ไขปัญหา ความรู้แบบนี้อาจไม่ใช่ความรู้ที่สมบูรณ์ แต่ก็พอที่จะนำมาแก้ไขปัญหาให้เราได้ และช่วยแนะให้เรารู้ว่าควรจะค้นหาเส้นทางอย่างไร เราเรียกว่าความรู้ที่ไม่สมบูรณ์หรือการคาดเดาอย่างมีเหตุผลแบบนี้ว่าฮิวริสติก

การค้นหาแบบฮิวริสติกคือ การค้นหาที่นำความรู้ประเภทนี้มาใช้ช่วยชี้แนะเส้นทางในการค้นหาคำตอบ โดยมีลักษณะเด่นดังนี้

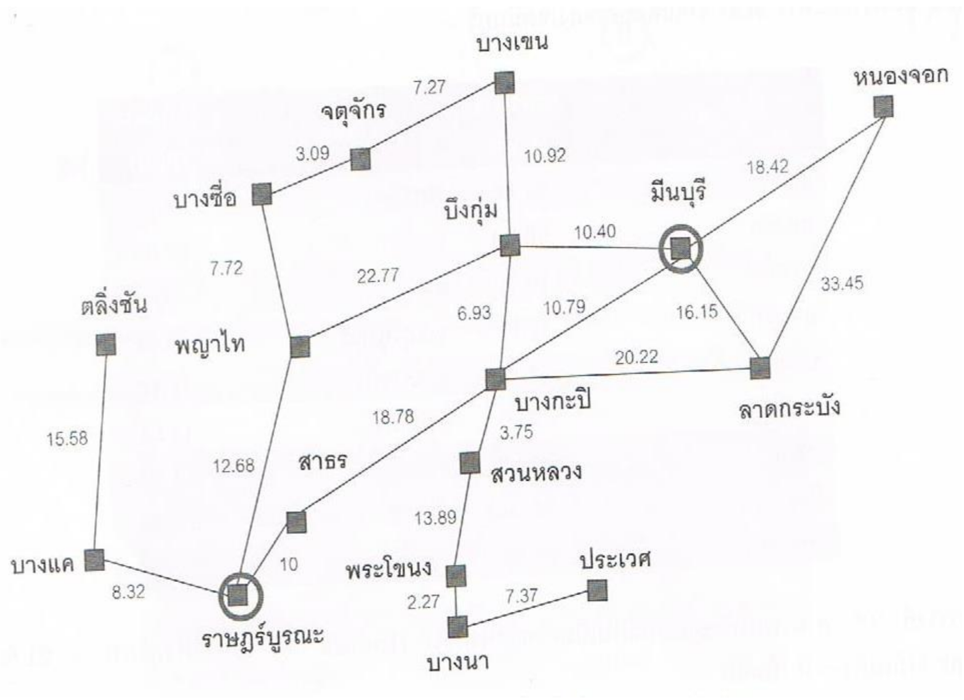
- เป็นเทคนิคที่ใช้เพิ่มประสิทธิภาพของกระบวนการค้นหา โดยอาจจะต้องยอมให้ขาดความสมบูรณ์ไปบ้าง คืออาจไม่พบคำตอบที่ถูกต้อง แม้ว่าในปริภูมิสถานะจะมีคำตอบนี้อยู่
- การนำฮิวริสติกมาใช้จะต้องนำมาใช้ในรูปแบบที่วัดค่าได้อย่างง่าย ซึ่งมักทำโดยนิยามฮิวริสติกให้อยู่ในรูปแบบของฟังก์ชัน เราเรียกว่าฟังก์ชันฮิวริสติก (heuristic function) ซึ่งเป็นฟังก์ชันที่คำนวณค่าจากสถานะไปยังตัวเลขที่ชี้ว่าสถานะนั้นเข้าใกล้สถานะเป้าหมายมากเท่าไร (ยิ่งมากเท่าไร ยิ่งมีโอกาสที่จะเปลี่ยนเป็นสถานะเป้าหมายมากเท่านั้น) การค้นหาก็จะมุ่งไปเส้นทางที่มีค่าฟังก์ชันฮิวริสติกที่ดีกว่า
- ฟังก์ชันฮิวริสติกนี้เป็นสิ่งที่ใช้ชี้แนะกระบวนการค้นหาว่าควรจะค้นหาไปในทิศทางใดซึ่งกระบวนการค้นหาที่ใช้ฟังก์ชันฮิวริสติกสามารถออกแบบได้หลายชนิด ดังจะกล่าวต่อไป
- ในบางกรณีที่เราสามารถนิยามฟังก์ชันฮิวริสติกได้อย่างสมบูรณ์แบบ การค้นหาก็จะสามารถมุ่งตรงไปยังสถานะเป้าหมายโดยไม่ผิดเส้นทางเลย แต่ถ้าฟังก์ชันฮิวริสติกไม่ดีก็อาจทำให้กระบวนการค้นหาหลงไปในทิศทางที่ผิดได้ ทำให้คำตอบ

ฟังก์ชันพื้นฐานที่นำมาใช้ประกอบกับการค้นหาแบบฮิวริสติกมี 2 ชนิด ได้แก่

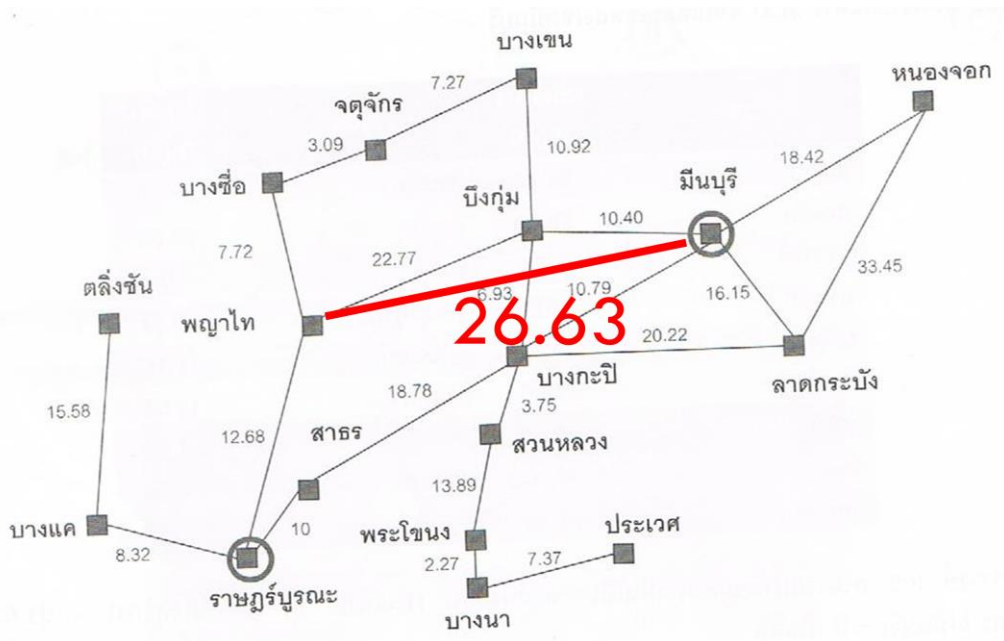
- ฟังก์ชัน Evaluation (Evaluation function) หรือ *f(n)* มีหน้าที่ประมาณทรัพยากรหรือค่าใช้จ่ายทั้งหมดบนเส้นทางจากสถานะ n ไปยังสถานะเป้าหมาย
- ฟังก์ชัน Heuristic (Heuristic function) หรือ *h(n)* มีหน้าที่บอกปริมาณทรัพยากรโดยประมาณที่จะใช้จากสถานะ n ไปยังสถานะเป้ามหาย

ตัวอย่างการหาค่าของฟังก์ชันฮิวริสติก

จากภาพที่ 3-2 กำหนดแผนที่ที่ตั้งของเขตต่างๆ ในกรุงเทพฯ พร้อมทั้งระยะทางจริงของเส้นทางที่ใช้ในการเชื่อมเขตต่างๆ เข้าด้วยกัน เช่น เส้นทางระหว่างตลิ่งชันและบางแค คือ 15.58 กิโลเมตร ในการหาฟังก์ชันฮิวริสติกนั้น จำเป็นจะต้องบ่งบอกสถานะเป้าหมายถึงก่อนถึงจะสามารถคำนวณค่าของฟังก์ชันฮิวริสติกได้ โดยในตัวอย่างนี้จะกำหนด *มีนบุรี* เป็นสถานะเป้าหมาย ฟังก์ชันฮิวริสติกที่ง่ายที่สุดของปัญหานี้คือ การคำนวณของค่าระยะทางของเส้นตรงจากเขตใดๆ (n) ไปยังเขตเป้าหมาย (มีนบุรี) โดยจากภาพที่ 3-3 แสดงค่า h(พญาไท) = 26.63 ซึ่งหมายถึงระยะทางเส้นตรงจากเขตพญาไทไปยังเขตมีนบุรี คือ 26.63 กิโลเมตร เป็นต้น โดยค่าฮิวริสติกของเขตทั้งหมดในแผนที่ที่มีสถานะเป้าหมายคือ เขตมีนบุรี แสดงในตารางที่ 3-1



ภาพที่ 3-2 แผนที่เขตและเส้นทางพร้อมระยะทางที่เชื่อมระหว่างเขตต่างๆ ในกรุงเทพฯ
(ที่มาจาก : ณัฐพงษ์ วารีประเสริฐ. (2552). ปัญญาประดิษฐ์, กรุงเทพฯ: เคทีพี คอมพ์ แอน คอนซัลท์)



ภาพที่ 3-3 ค่าฮิวริสติก h(พญาไท) ที่เกิดจากระยะทางเส้นทางจากพญาไทไปยังมีนบุรี
(ที่มาจาก : ณัฐพงษ์ วารีประเสริฐ. (2552). ปัญญาประดิษฐ์, กรุงเทพฯ: เคทีพี คอมพ์ แอน คอนซัลท์)

ตารางที่ 3-1 ค่าฮิวริสติก h(n) ของทุกเขตไปยังเขตมีนบุรี

เขต	h(n)	เขต	h(n)	เขต	h(n)
จตุจักร	21.86	ประเวศ	18.79	บางนา	26.16
ตลิ่งชัน	36.34	พญาไท	26.63	บึงกุ่ม	10.40
บางกะปิ	10.79	มีนบุรี	0	พระขโนง	28.44
บางเขน	17.59	ราษฎร์บูรณะ	36.84	สาธร	31.97
บางแค	42.11	ลาดกระบัง	16.15	หนองจอก	18.42
บางซื่อ	24.95	สวนหลวง	14.54		

3.3 เทคนิคการค้นหาแบบ Greedy Best First Search (GBFS)

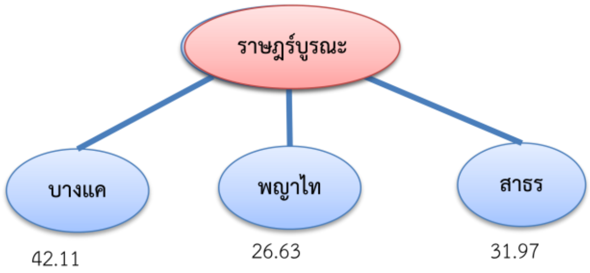
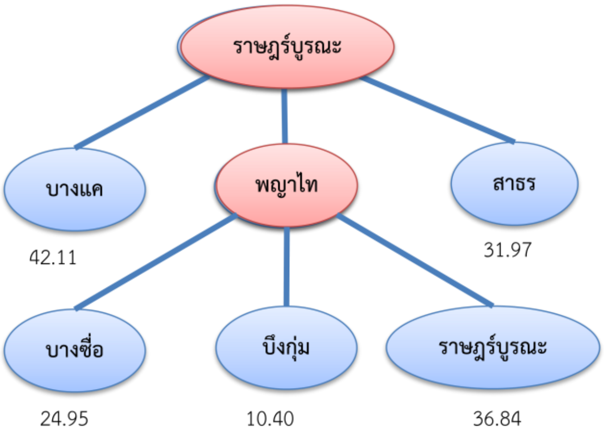
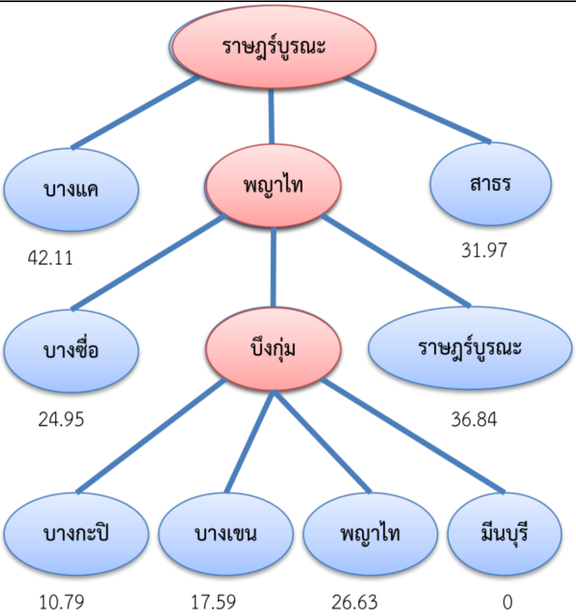
เป็นเทคนิคการค้นหาด้วยวิธีการเลือกเส้นทางที่ดีที่สุดก่อน ตามค่าของฟังก์ชัน Evaluation เพื่อให้ค้นหาเป้าหมายได้เร็วขึ้น โดยจะพิจารณาเฉพาะกับเส้นทางที่มองเห็นในขณะนั้นเท่านั้น และเลือกเส้นทางที่ใช้ทรัพยากรจากตำแหน่ง ณ สถานะนั้นจนถึงเป้าหมายให้น้อยที่สุด ดังนั้นฟังก์ชัน Evaluation ที่ได้จะพิจารณาจากค่าฮิวริสติกเพียงอย่างเดียว ทำให้สามารถเขียนอยู่ในรูปแบบสมการได้ดังนี้

$$f(n) = h(n)$$

ตัวอย่างการค้นหาด้วยวิธี Greedy Best First Search จากเขตราษฎร์บูรณะไปยังเขตมีนบุรี มีขั้นตอนดังตารางที่ 2-2 โดยสถานะสีแดงในต้นไม้แสดงถึงสถานะนั้นไปถูกทำการสำรวจแล้ว ในขณะที่สถานะที่น้ำเงินหมายถึงสถานะที่มองเห็นและรอการสำรวจ

ค่าฮิวริสติกนั้นใช้เพื่อบ่งบอกสถานะของปัญหาว่ามีโอกาสเข้าใกล้เป้าหมายมากน้อยเพียงใด หากค่าฮิวริสติกมีค่าที่ดี (กำหนดให้ค่าฮิวริสติกยิ่งน้อยยิ่งดี) แสดงว่าสถานะของปัญหานั้นใกล้ถึงเป้าหมาย มากขึ้น ดังนั้นการค้นหาจะพิจารณาจากสถานะที่มีค่าฮิวริสติกน้อย จนกว่าจะพบสถานะเป้าหมายที่ต้องการ (ค่าฮิวริสติกเป็น 0)

ตารางที่ 3-2 ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค GBFS จากราษฎร์บูรณะไปยังเขตมีนบุรี

	เริ่มค้นหาที่สถานะเริ่มต้น คือ เขตราษฎร์บูรณะ ที่มีค่า f(ราษฎร์บูรณะ) = h(ราษฎร์บูรณะ) = 36.84
	พิจารณาเส้นทางที่สามารถเดินทางได้จากราษฎร์บูรณะ ซึ่งจะพบว่าสามารถเดินทางไป บางแค พญาไท และ สาธร ได้ ซึ่งจากตารางฮิวริสติกนั้นพบว่า f(บางแค) = h(บางแค) = 42.11 f(พญาไท) = h(พญาไท) = 26.63 f(สาธร) = h(สาธร) = 31.97 โดยจะเห็นว่าค่าฮิวริสติกของพญาไทจะดีที่สุดให้เลือกเดินทางไปยัง เขตพญาไท
	พิจารณาเส้นทางที่สามารถเดินทางได้จากพญาไทซึ่งจะพบว่าสามารถเดินทางไป บางซื่อ บึงกุ่ม และกลับไปยังราษฎร์บูรณะได้ ซึ่งจากตารางฮิวริสติกนั้นพบว่า f(บางซื่อ) = h(บางซื่อ) = 24.95 f(บึงกุ่ม) = h(บึงกุ่ม) = 10.40 f(ราษฎร์บูรณะ) = h(ราษฎร์บูรณะ) = 36.84 โดยจะเห็นว่าค่าฮิวริสติกของบึงกุ่มจะดีที่สุดในสถานะที่เป็นสีน้ำเงินทั้งหมด (บางแค, สาธร, บางซื่อ และราษฎร์บูรณะ) ให้เลือกเดินทางไปยัง เขตบึงกุ่ม
	พิจารณาเส้นทางที่สามารถเดินทางได้จากบึงกุ่มซึ่งจะพบว่าสามารถเดินทางไป บางกะปิ บางเขน พญาไท และ มีนบุรีได้ ซึ่งจากตารางฮิวริสติกนั้นพบว่า f(บางกะปิ) = h(บางกะปิ) = 10.79 f(บางเขน) = h(บางเขน) = 17.59 f(พญาไท) = h(พญาไท) = 26.63 f(มีนบุรี) = h(มีนบุรี) = 0 โดยจะเห็นว่าค่าฮิวริสติกของมีนบุรีจะดีที่สุดในสถานะที่เป็นสีน้ำเงินทั้งหมด (บางแค, สาธร, บางซื่อ ราษฎร์บูรณะ บางกะปิ บางเขน และ พญาไท) ให้เลือกเดินทางไปยัง เขตมีนบุรี ซึ่งเป็นสถานะเป้าหมายที่ต้องการ

จากเทคนิคการค้นหาด้วยวิธี Greedy Best First Search นั้นจะพบว่า การเดินทางจากเขตราษฎร์บูรณะไปยังเขตมีนบุรีนั้นจะใช้เส้นทางดังนี้

ราษฎร์บูรณะ → พญาไท → บึงกุ่ม → มีนบุรี

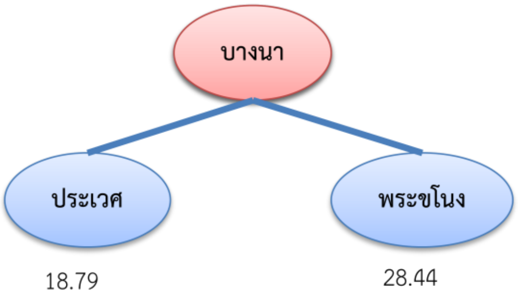
ซึ่งเมื่อพิจารณาระยะทางที่ใช้ในการเดินทางทั้งหมดจากระยะจริงในแผนที่ จะได้ระยะดังนี้

- ราษฎร์บูรณะ → พญาไท ใช้ระยะทาง 12.68 กิโลเมตร
- พญาไท → บึงกุ่ม ใช้ระยะทาง 22.77 กิโลเมตร
- บึงกุ่ม → มีนบุรี ใช้ระยะทาง 10.40 กิโลเมตร
- ดังนั้นระยะทางทั้งหมดคือ 12.68 + 22.77 + 10.40 = **45.85** กิโลเมตร

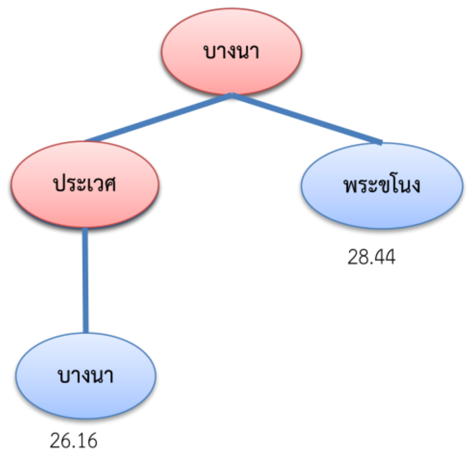
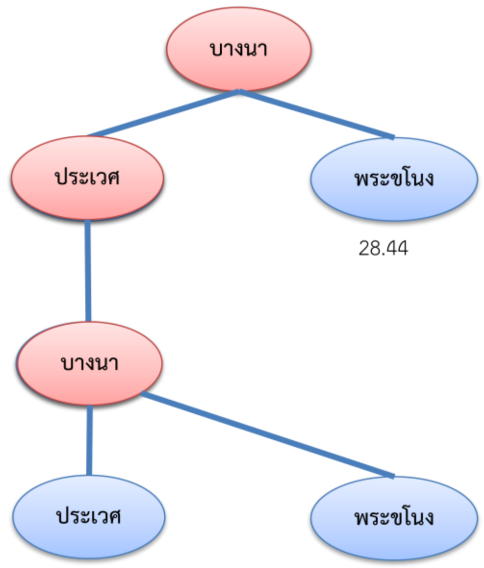
เส้นทางการเดินทางที่ได้จากการค้นหาด้วยเทคนิค Greedy Best First Search นั้นไม่ได้เส้นทางที่ใช้ระยะทางที่สั้นที่สุด แต่ช่วยทำให้การค้นหาเส้นทางไปสู่เป้าหมายได้เร็วขึ้นเนื่องด้วยมีการใช้งานฟังก์ชันฮิวริสติกที่ช่วยแนะนำสถานะที่ควรสำรวจก่อน

แต่อย่างไรก็ตามการค้นหาด้วยวิธี Greedy Best First Search อาจไม่ค้นพบเส้นทางไปสู่ สถานะเป้าหมายได้ทุกครั้ง โดยมีโอกาสที่จะค้นหาเส้นทางที่ซ้ำซ้อนจนเกิดเป็นวงเวียนไม่มีที่สิ้นสุด เช่น การค้นหาเส้นทางในการเดินทางจากเขตบางนา ไปยังเขตมีนบุรี ซึ่งแสดงขั้นตอนการค้นหาในตารางที่ 3-3

ตารางที่ 3-3 ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค GBFS จากบางนาไปมีนบุรี

	เริ่มค้นหาที่สถานะเริ่มต้น คือ เขตบางนา ที่มีค่า f(บางนา) = h(บางนา) = 26.16
	พิจารณาเส้นทางที่สามารถเดินทางได้จากบางนา ซึ่งจะพบว่าสามารถเดินทางไป ประเวศ และ พระขโนง ได้ ซึ่งจากตารางฮิวริสติกนั้นพบว่า f(ประเวศ) = h(ประเวศ) = 18.79 f(พระขโนง) = h(พระขโนง) = 28.44 โดยจะเห็นว่าค่าฮิวริสติกของพญาไทจะดีที่สุดให้เลือกเดินทางไปยัง เขตประเวศ

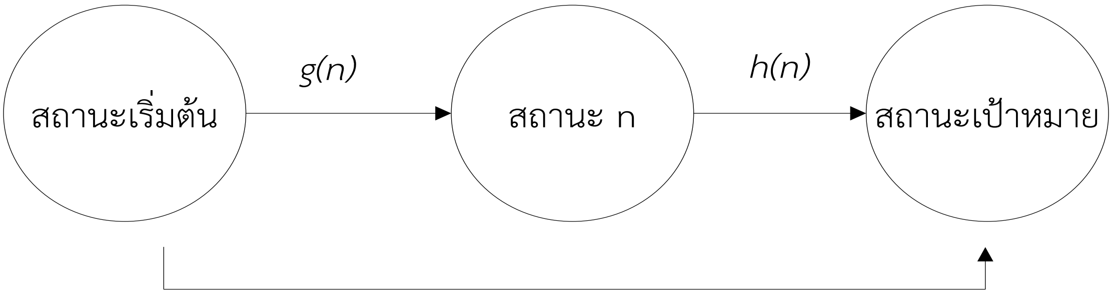
ตารางที่ 3-3 (ต่อ) ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค GBFS จากบางนาไปมีนบุรี

	พิจารณาเส้นทางที่สามารถเดินทางได้จากประเวศซึ่งจะพบว่าสามารถเดินทางไป บางนา ได้เพียงเส้นทางเดียว ซึ่งจากตารางฮิวริสติกนั้นพบว่า f(บางนา) = h(บางนา) = 26.16 โดยจะเห็นว่าค่าฮิวริสติกของบางนานั้นดีกว่าค่าฮิวริสติกของพระขโนง (ดูจากสถานะสีน้ำเงิน) ทำให้อัลกอริทึมของ GBFS ตัดสินใจเลือกเดินทางกลับไปเขตบางนาอีกครั้ง
	พิจารณาเส้นทางที่สามารถเดินทางได้จากบางนา ซึ่งจะพบว่าสามารถเดินทางไป ประเวศ และ พระขโนง ได้ ซึ่งปัญหาได้ถูกวนกลับมาเหมือนกับขั้นตอนแรกในการแก้ไขปัญหา ซึ่งจะทำให้การค้นหาต่อไปอัลกอริทึม GBFS ก็จะเลือกเดินทางไปยัง ประเวศ แล้วเดินทางกลับมายัง บางนา ใหม่อีกครั้ง และจะมีการเดินทางในลักษณะนี้ต่อไปเรื่อยๆ อย่างไม่มีสิ้นสุด

ดังนั้นจึงสามารถสรุปคุณสมบัติของการค้นหาแบบ Greedy Best First Search ได้ดังนี้

- Completeness ไม่สามารถค้นหาสถานะเป้าหมายพบทุกครั้ง
- Optimality เส้นทางไปสู่สถานะเป้าหมายไม่จำเป็นต้องใช้การกระทำน้อยที่สุด
- Time Complexity $O(b^m)$
- Space Complexity $O(bm)$

เมื่อ b = จำนวนกิ่งเฉลี่ยของโหนด, m = ความลึกมากที่สุดของต้นไม้



$$f(n) = g(n) + h(n)$$

ภาพที่ 3-4 ฟังก์ชัน Evaluation ของเทคนิคการค้นหาแบบ A*

3.4 เทคนิคการค้นหาแบบ A*

เทคนิคการค้นหาแบบ A* (อ่านว่า เอ-สตาร์) เป็นวิธีการค้นหาที่ถูกพัฒนาขึ้นมาเพื่อแก้ไขปัญหาที่เกิดขึ้นในเทคนิคการค้นหาแบบ Greedy Best First Search โดยการนำข้อมูลเข้ามาพิจารณาเพิ่มเติม คือ ข้อมูลทรัพยากรที่ใช้ตั้งแต่ตำแหน่งเริ่มต้นจนถึงตำแหน่งที่กำลังพิจารณาขณะนั้น หากนำมาเขียนสัญลักษณ์ $g(n)$ จะหมายถึง ทรัพยากรที่ใช้ตั้งแต่สถานะเริ่มต้นจนถึงสถานะ n ดังภาพที่ 3-4 ดังนั้นฟังก์ชัน Evaluation สามารถเขียนได้ในรูปแบบสมการดังนี้

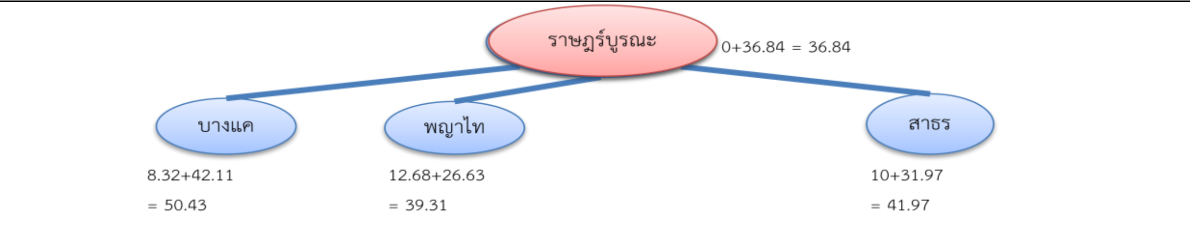
$$f(n) = g(n) + h(n)$$

หลักการค้นหาจะพิจารณาสถานะที่เป็นโหนดใบแต่ละสถานะในขณะนั้น ว่าสถานะใดที่มีค่าฟังก์ชัน Evalution ดีที่สุด และทำเช่นนี้จนกว่าจะพบเส้นทางไปยังสถานะเป้าหมายที่สั้นที่สุด จากตัวอย่างของแผนที่เดิมในกรุงเทพฯ ดังแสดงในภาพที่ 3-2 ถ้าต้องการทำการค้นหาเส้นทางจากจากเขตราษฎร์บูรณะไปยังเขตมีนบุรี จะมีขั้นตอนการค้นหาดังแสดงในตารางที่ 3-4 โดยสถานะสีแดงในต้นไม้แสดงถึงสถานะนั้นไปถูกทำการสำรวจแล้ว ในขณะที่สถานะที่น้ำเงินหมายถึงสถานะที่มองเห็นและรอการสำรวจ

ตารางที่ 3-4 ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค A* จากราษฎร์บูรณะไปยังเขตมีนบุรี

ขั้นตอนที่ 1: เริ่มค้นหาที่สถานะเริ่มต้น คือ ราษฎร์บูรณะ ที่มีค่า h(ราษฎร์บูรณะ) = 36.84 และค่า g(ราษฎร์บูรณะ) = 0 เนื่องด้วยเป็นสถานะแรกยังไม่มีค่าใช้จ่ายในการเดินทางเกิดขึ้น ดังนั้น f(ราษฎร์บูรณะ) = g(ราษฎร์บูรณะ) + h(ราษฎร์บูรณะ) = 0 + 36.84 = 36.84

ตารางที่ 3-4 (ต่อ) ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค A* จากราษฎร์บูรณะไปยังเขตมีนบุรี



ขั้นตอนที่ 2: พิจารณาเส้นทางที่สามารถเดินทางได้จากราษฎร์บูรณะ ซึ่งจะพบว่าสามารถเดินทางไป บางแค พญาไท และ สาธร ได้ โดยค่า g(n) ของแต่ละสถานะที่เกิดใหม่คือ

g(บางแค) = 8.32 เนื่องจากระยะทางจริงที่ใช้ในการเดินทางจากราษฎร์บูรณะไปยังบางแค คือ 8.32 กิโลเมตร

g(พญาไท) = 12.68 เนื่องจากระยะทางจริงที่ใช้ในการเดินทางจากราษฎร์บูรณะไปยังพญาไท คือ 12.68 กิโลเมตร

g(สาธร) = 10 เนื่องจากระยะทางจริงที่ใช้ในการเดินทางจากราษฎร์บูรณะไปยังสาธร คือ 10 กิโลเมตร

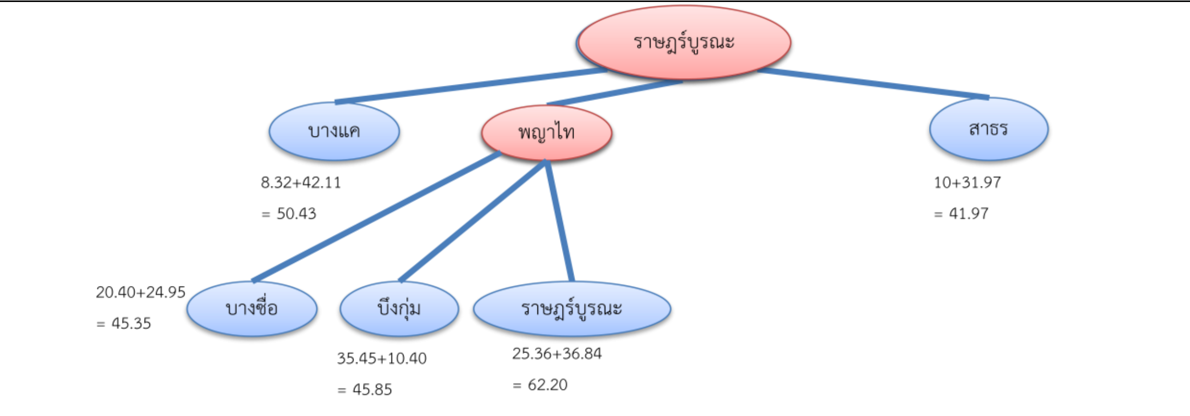
ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้

f(บางแค) = g(บางแค) + h(บางแค) = 8.32 + 42.11 = 50.43

f(พญาไท) = g(พญาไท) + h(พญาไท) = 12.68 + 26.63 = 39.31

f(สาธร) = g(สาธร) + h(สาธร) = 10 + 31.97 = 41.97

โดยจะเห็นว่าค่าฟังก์ชัน Evaluation พญาไทจะดีที่สุดให้เลือกเดินทางไปยัง เขตพญาไท



ขั้นตอนที่ 3: พิจารณาเส้นทางที่สามารถเดินทางได้จากพญาไท ซึ่งจะพบว่าสามารถเดินทางไป บางซื่อ บึงกุ่ม และกลับมายังราษฎร์บูรณะได้ โดยค่า g(n) ของแต่ละสถานะที่เกิดใหม่คือ

g(บางซื่อ) = 20.40 ระยะทางจริงจากราษฎร์บูรณะผ่านพญาไทไปยังบางซื่อ คือ 12.68 + 7.72 กิโลเมตร

g(บึงกุ่ม) = 35.45 ระยะทางจริงจากราษฎร์บูรณะผ่านพญาไทไปยังบึงกุ่ม คือ 12.68 + 22.77 กิโลเมตร

g(ราษฎร์บูรณะ) = 25.36 ระยะทางจริงจากราษฎร์บูรณะผ่านพญาไทไปยังราษฎร์บูรณะ คือ 12.68 + 12.68 กิโลเมตร

ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้

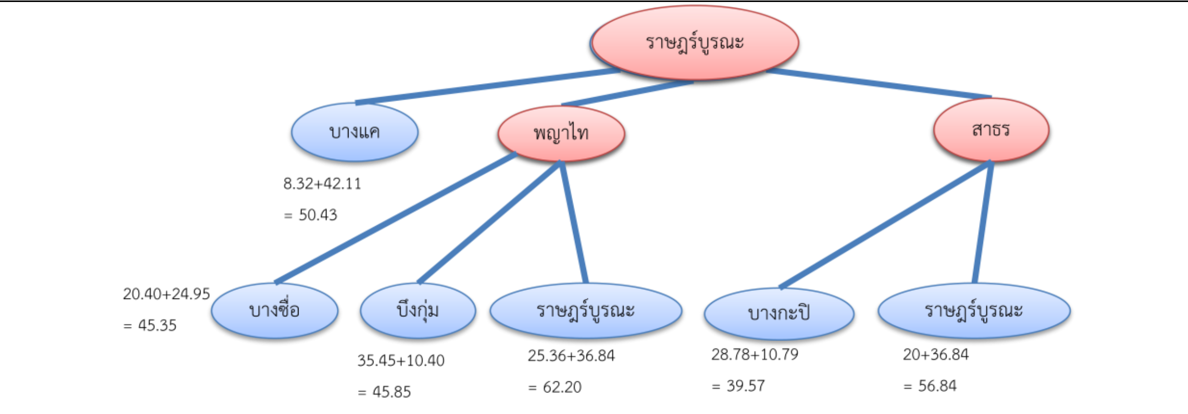
f(บางซื่อ) = g(บางซื่อ) + h(บางซื่อ) = 20.40 + 24.95 = 45.35

f(บึงกุ่ม) = g(บึงกุ่ม) + h(บึงกุ่ม) = 35.45 + 10.40 = 45.85

f(ราษฎร์บูรณะ) = g(ราษฎร์บูรณะ) + h(ราษฎร์บูรณะ) = 25.36 + 36.84 = 62.20

โดยจะเห็นว่าค่าฟังก์ชัน Evaluation ของสถานะสีน้ำเงินทั้งหมดของ สาธรดีที่สุดให้เลือกเดินทางไปยัง เขตสาธร

ตารางที่ 3-4 (ต่อ) ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค A* จากราษฎร์บูรณะไปยังเขตมีนบุรี



ขั้นตอนที่ 4: พิจารณาเส้นทางที่สามารถเดินทางได้จากสาธร ซึ่งจะพบว่าสามารถเดินทางไป บางกะปิ และ ราษฎร์บูรณะ ได้ โดยค่า g(n) ของแต่ละสถานะที่เกิดใหม่คือ

g(บางกะปิ) = 28.78 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรไปยังบางกะปิ คือ 10 + 18.78 กิโลเมตร

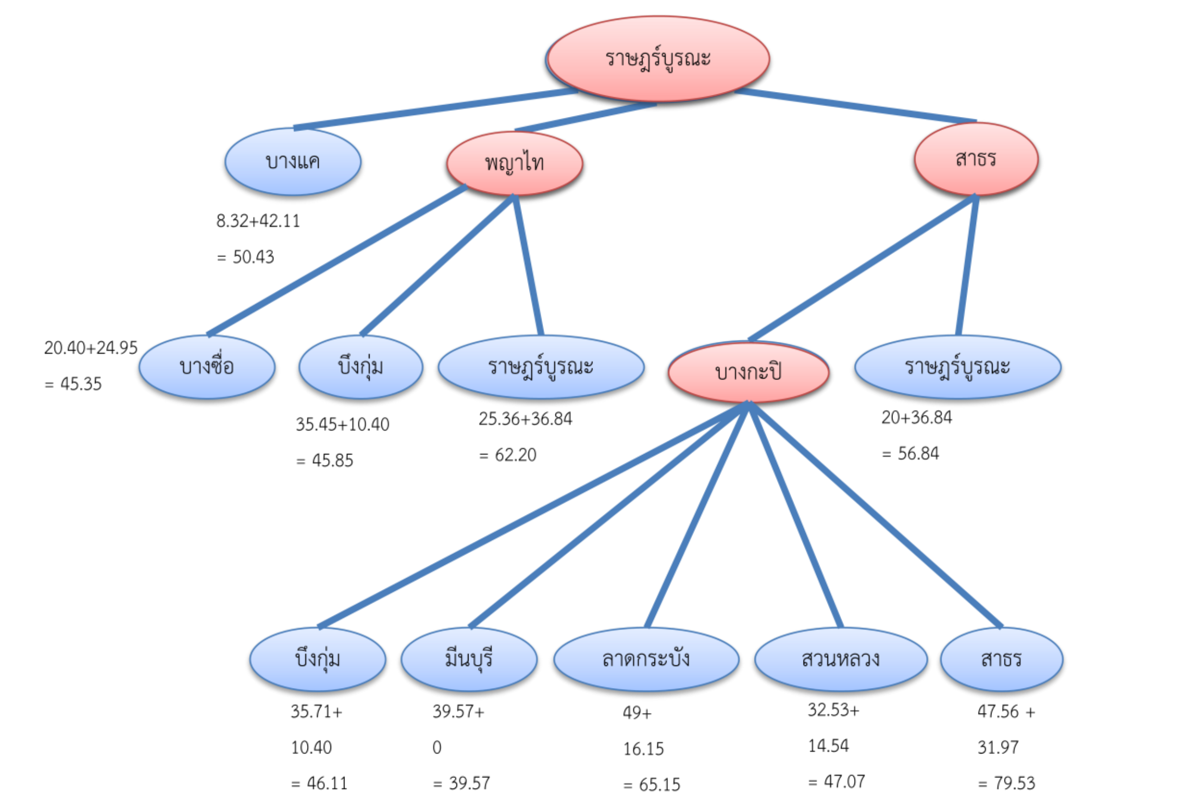
g(ราษฎร์บูรณะ) = 20 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรและกลับมาราษฎร์บูรณะ คือ 10 + 10 กิโลเมตร

ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้

f(บางกะปิ) = g(บางกะปิ) + h(บางกะปิ) = 28.78 + 10.79 = 39.57

f(ราษฎร์บูรณะ) = g(ราษฎร์บูรณะ) + h(ราษฎร์บูรณะ) = 20 + 36.84 = 56.84

โดยจะเห็นว่าค่าฟังก์ชัน Evaluation ของทุกสถานะสีน้ำเงิน เขตบางกะปิจะได้ค่าดีที่สุดให้เลือกเดินทางไปยัง เขตบางกะปิ



ตารางที่ 3-4 (ต่อ) ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค A* จากราษฎร์บูรณะไปยังเขตมีนบุรี

ขั้นตอนที่ 5: พิจารณาเส้นทางที่สามารถเดินทางได้จากบางกะปิ ซึ่งจะพบว่าสามารถเดินทางไป บึงกุ่ม มีนบุรี ลาดกระบัง สวนหลวง และ สาธรได้ โดยค่า g(n) ของแต่ละสถานะที่เกิดใหม่คือ g(บึงกุ่ม) = 35.71 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรและบางกะปิไปยังบึงกุ่ม คือ 10 + 18.78 + 6.93 เท่ากับ 35.71 กิโลเมตร g(มีนบุรี) = 39.57 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรและบางกะปิไปยังมีนบุรี คือ 10 + 18.78 + 10.79 เท่ากับ 39.57 กิโลเมตร g(ลาดกระบัง) = 49 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรและบางกะปิไปยังลาดกระบัง คือ 10 + 18.78 + 20.22 เท่ากับ 49 กิโลเมตร g(สวนหลวง) = 32.53 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรและบางกะปิไปยังสวนหลวง คือ 10 + 18.78 + 3.75 เท่ากับ 32.53 กิโลเมตร g(สาธร) = 47.56 เกิดจากระยะทางจริงจากราษฎร์บูรณะผ่านสาธรและบางกะปิไปยังสาธร คือ 10 + 18.78 + 18.78 เท่ากับ 47.56 กิโลเมตร ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้ f(บึงกุ่ม) = g(บึงกุ่ม) + h(บึงกุ่ม) = 35.71 + 10.40 = 46.11 f(มีนบุรี) = g(มีนบุรี) + h(มีนบุรี) = 39.57 + 0 = 39.57 f(ลาดกระบัง) = g(ลาดกระบัง) + h(ลาดกระบัง) = 49 + 16.15 = 65.15 f(สวนหลวง) = g(สวนหลวง) + h(สวนหลวง) = 32.53 + 14.54 = 47.07 f(สาธร) = g(สาธร) + h(สาธร) = 47.56 + 31.97 = 79.53 โดยจะเห็นว่าค่าฟังก์ชัน Evaluation ของสถานะสีน้ำเงินทั้งหมดของ มีนบุรีดีที่สุดให้เลือกเดินทางไปยัง เขตมีนบุรี และ เขตมีนบุรีคือสถานะเป้าหมายดังนั้นจึงค้นหาคำตอบพบและจบการค้นหา

จากเทคนิคการค้นหาด้วยวิธี A* นั้นจะพบว่า การเดินทางจากเขตราษฎร์บูรณะไปยังเขตมีนบุรีนั้นจะใช้เส้นทางดังนี้

ราษฎร์บูรณะ → สาธร → บางกะปิ → มีนบุรี

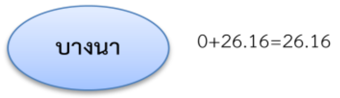
ซึ่งเมื่อพิจารณาระยะทางที่ใช้ในการเดินทางทั้งหมดจากระยะจริงในแผนที่ จะได้ระยะดังนี้

- ราษฎร์บูรณะ → สาธร ใช้ระยะทาง 10 กิโลเมตร
- สาธร → บางกะปิ ใช้ระยะทาง 18.78 กิโลเมตร
- บางกะปิ → มีนบุรี ใช้ระยะทาง 10.79 กิโลเมตร
- ดังนั้นระยะทางทั้งหมดคือ 10 + 18.78 + 10.79 = **39.57** กิโลเมตร

โดยคำตอบที่ได้นั้นให้ผลลัพธ์ที่ดีกว่าการค้นหาแบบ Greedy Best First Search ที่ให้ผลลัพธ์เป็นระยะทางทั้งหมด 45.85 กิโลเมตร

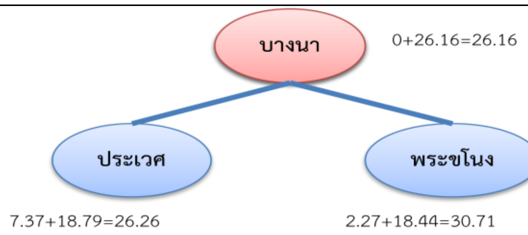
อีกทั้งการค้นหาแบบ A* สามารถแก้ไขปัญหาการวงเวียนไม่มีที่สิ้นสุดของการค้นหาแบบ Greedy Best First Search ได้อีกด้วย ดังเช่นตัวอย่างการหาเส้นทางการเดินทางจากเขตบางนาไปยังเขตมีนบุรี ที่วิธีการค้นหาแบบ Greedy Best First Search ไม่สามารถหาคำตอบได้ แต่วิธีการค้นหาแบบ A* สามารถหาคำตอบได้ ดังแสดงในตารางที่ 3-5

ตารางที่ 3-5 ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค A* จากบางนาไปยังมีนบุรี



ขั้นตอนที่ 1: เริ่มค้นหาที่สถานะเริ่มต้น คือ บางนา ที่มีค่า h(บางนา) = 26.16 และค่า g(บางนา) = 0 เนื่องด้วยเป็นสถานะแรกยังไม่มีค่าใช้จ่ายในการเดินทางเกิดขึ้น ดังนั้น

f(บางนา) = g(บางนา) + h(บางนา) = 0 + 26.16 = 26.16



ขั้นตอนที่ 2: พิจารณาเส้นทางที่สามารถเดินทางได้จากบางนา ซึ่งจะพบว่าสามารถเดินทางไป ประเวศ และพระขโนงได้ โดยค่า g(n) ของแต่ละสถานะที่เกิดใหม่คือ

g(ประเวศ) = 7.37 เนื่องจากระยะทางจริงที่ใช้ในการเดินทางจากบางนาไปยังประเวศ คือ 7.37 กิโลเมตร

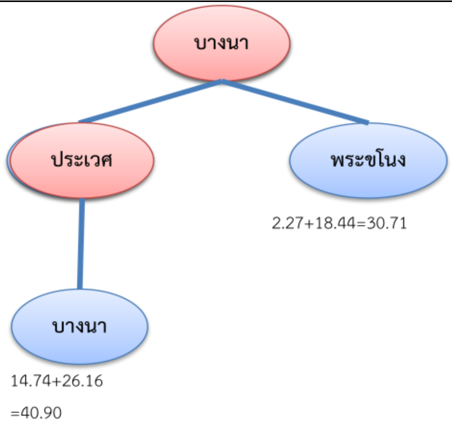
g(พระขโนง) = 2.27 เนื่องจากระยะทางจริงที่ใช้ในการเดินทางจากบางนาไปยังพระขโนง คือ 2.27 กิโลเมตร

ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้

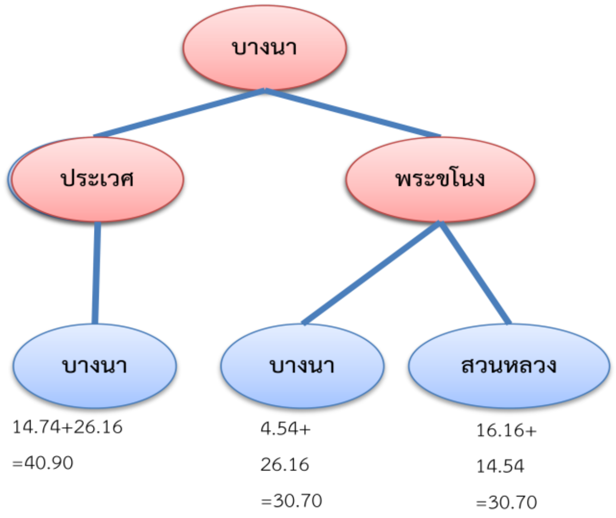
f(ประเวศ) = g(ประเวศ) + h(ประเวศ) = 7.37 + 18.79 = 26.26

f(พระขโนง) = g(พระขโนง) + h(พระขโนง) = 2.27 + 18.44 = 30.71

โดยจะเห็นว่าค่าฟังก์ชัน Evaluation ประเวศจะดีที่สุดให้เลือกเดินทางไปยัง เขตประเวศ



ตารางที่ 3-5 (ต่อ) ขั้นตอนการค้นหาเส้นทางด้วยเทคนิค A* จากบางนาไปยังมีนบุรี

ขั้นตอนที่ 3: พิจารณาเส้นทางที่สามารถเดินทางได้จากประเวศ ซึ่งจะพบว่าสามารถเดินทางกลับไปบางนาได้เพียงเขตเดียว โดยค่า g(n) ของสถานะที่เกิดใหม่คือ g(บางนา) = 14.74 เกิดจากระยะทางจริงในเดินทางจากบางนาผ่านประเวศและกลับบางนา คือ 7.37 + 7.37 กิโลเมตร ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้ f(บางนา) = g(บางนา) + h(บางนา) = 14.74 + 26.16 = 40.90 โดยจะเห็นว่าค่าฟังก์ชัน Evaluation ของสถานะสีน้ำเงินคือ บางนา และ พระขโนง นั้นพบว่าเขตพระขโนงจะดีที่สุดให้เลือกเดินทางไปยัง เขตพระขโนง

ขั้นตอนที่ 4: พิจารณาเส้นทางที่สามารถเดินทางได้จากพระขโนง ซึ่งจะพบว่าสามารถเดินทางไปบางนา และสวนหลวงได้ โดยค่า g(n) ของแต่ละสถานะที่เกิดใหม่คือ g(บางนา) = 4.54 ระยะทางจริงในเดินทางจากบางนาผ่านพระขโนงและกลับบางนา คือ 2.27 + 2.27 กิโลเมตร g(สวนหลวง) = 16.16 ระยะทางจริงในเดินทางจากบางนาผ่านพระขโนงและไปสวนหลวง คือ 2.27 + 13.89กิโลเมตร ดังนั้นเมื่อรวมกับค่าฮิวริสติกที่กำหนดในตารางจะได้ค่าฟังก์ชัน Evaluation ดังนี้ f(บางนา) = g(บางนา) + h(บางนา) = 4.54 + 26.16 = 30.70 f(สวนหลวง) = g(สวนหลวง) + h(สวนหลวง) = 16.16 + 14.54 = 30.70 โดยจะเห็นว่าค่าฟังก์ชัน Evaluation ของสถานะสีน้ำเงินคือ บางนา (ที่วิ่งผ่านประเวศ) บางนา (ที่วิ่งผ่านพระขโนง) และสวนหลวง นั้นพบว่า เขตบางนา (ที่ผ่านพระขโนง) และ เขตสวนหลวง ให้ค่าดีที่สุดเท่ากันดังนั้น จึงอยู่ที่อัลกอริทึมภายในจะเลือกลงซ้ายก่อน ขวาก่อน หรือ สุ่มเส้นทาง ไม่ว่าจะเลือกด้วยวิธีการไหนก็ตามจะเห็นได้ว่าการเวียนซ้ำระหว่างบางนาและประเวศจะหมดไป

ดังนั้นจึงสามารถสรุปคุณสมบัติของการค้นหาแบบ A* ได้ดังนี้

- Completeness สามารถค้นหาสถานะเป้าหมายพบทุกครั้ง
- Optimality เส้นทางไปสู่สถานะเป้าหมายเป็นเส้นทางที่ใช้ทรัพยากรน้อยที่สุด
- Time Complexity เพิ่มขึ้นหลายเท่าตัวตามขนาดของสถานะ (Exponential)
- Space Complexity เก็บสถานะทั้งหมดลงในหน่วยความจำ

3.5 การกำหนดฟังก์ชันฮิวริสติก

ลักษณะของฟังก์ชันฮิวริสตก จะเป็นสมการที่ง่ายเพื่อช่วยให้การคำนวณรวดเร็ว โดยผู้เขียนโปรแกรมจะเป็นผู้กำหนดฟังก์ชันที่เหมาะสมกับปัญหาที่พิจารณา ค่าที่ได้จากการคำนวณฟังก์ชันฮิวริสติกสามารถนำมาใช้เป็นตัวตัวสินใจในการเลือกเส้นทางไปสู่เป้าหมายได้ พิจารณาตัวอย่างปัญหาเกม 8-Puzzle ดังภาพที่ 3.5

7	2	4
5		6
8	3	1

(a) สถานะเริ่มต้น

	1	2
3	4	5
6	7	8

(b) สถานะเป้าหมาย

ภาพที่ 3-5 ตัวอย่างปัญหา 8-Puzzle

ฟังก์ชันฮิวริสติกที่นิยมใช้กับสำหรับแก้ปัญหา 8-puzzle มีอยู่ 2 ชนิดคือ

- h_1 แทนจำนวนตำแหน่งป้ายหมายเลขในสถานะปัจจุบันที่แตกต่างจากตำแหน่งของสถานะเป้าหมาย
- h_2 แทนผลรวมทั้งหมดของการย้ายตำแหน่ง ที่เรียกว่า “Manhattan Distance”

จากตัวอย่างฟังก์ชันฮิวริสติก h_1 จะพบว่าค่าของ h_1 คือ 8 เนื่องจากมีจำนวนตำแหน่งแผ่นป้ายหมายเลขของสถานะเริ่มต้นแตกต่างกับตำแหน่งแผ่นป้ายของสถานะเป้าหมายอยู่ทั้งหมด 8 แผ่น (ทุกแผ่นป้ายของสถานะเริ่มต้นไม่อยู่ตำแหน่งเดียวกันกับตำแหน่งของสถานะเป้าหมายเลย)

จากตัวอย่างฟังก์ชันฮิวริสติก h_2 จะคำนวณการเคลื่อนย้ายตำแหน่งของแต่ละฝ่าย แล้วนำมาบวกรวมกัน เช่น แผ่นป้ายหมายเลข “1” จากสถานะเริ่มต้นสามารถแสดงลำดับการเคลื่อนย้ายตำแหน่งตามการกระทำในเซตของ Successor Function เพื่อให้อยู่บนตำแหน่งตามสถานะเป้าหมายได้ คือ “up – up – left” ซึ่งจำนวนการย้ายตำแหน่งเป็น 3 ถ้าเป็นแผ่นป้ายหมายเลข “2” ลำดับการเคลื่อนย้ายจะเป็น “right” ซึ่งมีจำนวนการย้ายตำแหน่งเป็น 1 ดังนั้นค่า $h_2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$

หากนำเทคนิคการค้นหาและตัวอย่างฟังก์ชันฮิวริสติกทั้งสองชนิด (h_1 และ h_2) มาทดลองเพื่อเปรียบเทียบประสิทธิภาพโดยกำหนดดังนี้

- Blind Search จะใช้วิธีการค้นหาแบบ Iterative Deepening Search (IDS)
- Heuristic Search จะใช้วิธีการค้นหาแบบ A* ร่วมกันฟังก์ชัน h_1 (A* [h_1]) และ A* ร่วมกับฟังก์ชัน h_2 (A* [h_2])

เมื่อนำผลการทดลองมาเปรียบเทียบเรื่องทรัพยากรที่ใช้ในการค้นหาและการเกิด Branching Factor จะได้ผลลัพธ์ดังตารางที่ 3.6 และ 3.7 ตามลำดับ

ตารางที่ 3-6 เปรียบเทียบเรื่องทรัพยากรที่ใช้ในการค้นหาแต่ละวิธี

ทรัพยากรที่ใช้ในการค้นหา			
ระดับลิมิต (d)	**Blind Search**	**Heuristic Search**	
	IDS	A* [h_1]	A* [h_2]
2	10	6	6
4	112	13	12
6	680	20	18
8	6,384	39	25
10	47,127	93	39
12	3,664,035	227	73
14	-	539	113
16	-	1,301	211
18	-	3,056	363
20	-	7,276	676
22	-	18,094	1,219
24	-	39,135	1,641

ตารางที่ 3-7 เปรียบเทียบเรื่องการเกิด Branching Factor จากการค้นหาแต่ละวิธี

การเกิด Branching Factor			
ระดับลิมิต (d)	**Blind Search**	**Heuristic Search**	
	IDS	A* [h_1]	A* [h_2]
2	2.45	1.79	1.79
4	2.87	1.48	1.45
6	2.73	1.34	1.30
8	2.80	1.33	1.24
10	2.79	1.38	1.22
12	2.78	1.42	1.24
14	-	1.44	1.23
16	-	1.45	1.25
18	-	1.46	1.26
20	-	1.47	1.27
22	-	1.48	1.28
24	-	1.48	1.26

จากผลการทดลองดังกล่าว หากพิจารณาประสิทธิภาพของเทคนิคการค้นหาที่ใช้สำหรับปัญหาเกม 8-Puzzle จะพบว่าเทคนิคการค้นหาแบบ Heuristic จะใช้ทรัพยากรน้อยกว่าเทคนิคการค้นหาแบบ Blind Search อย่างชัดเจน ส่วนการเกิด Branching Factor หากพิจารณาที่ความลึกตั้งแต่ระดับลิมิตที่ 14 ขึ้นไป จะพบว่า การค้นหาแบบ IDS ไม่สามารถค้นหาต่อไปได้ เนื่องด้วยข้อจำกัดเกี่ยวกับการจัดเก็บข้อมูลที่มากเกินไป และหากพิจารณาประสิทธิภาพของฟังก์ชัน Heuristic ทั้งสองชนิด โดยใช้วิธีการค้นหาแบบ A* จะพบว่าการค้นหาในช่วงแรกนั้น ทั้งฟังก์ชัน h_1 และ h_2 มีการใช้ทรัพยากรและค่า Branching Factor ที่ใกล้เคียงกัน แต่เมื่อเริ่มทำการค้นหาในระดับลิมิตที่สูงขึ้น ฟังก์ชันฮิวริสติก h_2 จะมีประสิทธิภาพ มากกว่าฟังก์ชันฮิวริสติก h_1 ดังนั้นฟังก์ชันฮิวริสติก h_2 จึงเหมาะสมสำหรับการแก้ปัญหาเกม 8-Puzzle มากกว่า h_1

3.6 อัลกอริทึมแบบ Local Search

อัลกอริทึมการค้นหาวิธีต่างๆ ที่กล่าวถึงข้างต้น ถูกออกแบบให้มีหน้าที่ค้นหาในปริภูมิอย่างเป็นระบบ โดยจะต้องค้นหาทุกเส้นทางที่เป็นไปได้ และนำไปบันทึกลงในหน่วยความจำทั้งหมด เรียกเส้นทางที่สามารถไปถึงยังเป้าหมายว่า “เส้นทางการแก้ไขปัญหา (Path of Solution)” ในขณะที่หลายเส้นทาง ไม่สามารถพาไปถึงคำตอบได้ ทำให้การค้นหาบนปริภูมิทั้งหมดต้องใช้เวลาพื้นที่หน่วยความจำมาก ส่งผลให้

สิ้นเปลืองทรัพยากรเป็นอย่างมาก ข้อจำกัดดังกล่าวสามารถแก้ไขได้โดยใช้อัลกอริทึมที่เรียกว่า “อัลกอริทึมการค้นหาเฉพาะแห่ง (Local Search Algorithm)” โดยจะคำนึงถึง “สถานะปัจจุบัน” เท่านั้น วิธีการคือจะเริ่มค้นหาเฉพาะเส้นทางจากตำแหน่งสถานะปัจจุบันไปยังสถานะเป้าหมาย แทนการค้นหาจากสถานะเริ่มต้น

วัตถุประสงค์ของอัลกอรึทึมแบบ Local Search คือ ต้องการค้นพบสถานะที่ดีที่สุด ซึ่งหมายถึงสถานะเป้าหมายที่ต้องการ ดังนั้นจึงต้องนำฟังก์ชันวัตถุประสงค์ (Objective function) มาใช้เป็นมาตรฐานการวัดค่าของแต่ละสถานะในปริภูมิ หากสถานะมีค่า Heuristic ที่ดีค่าฟังก์ชันวัตถุประสงค์จะสูงขึ้นตามไปด้วย และหากพบสถานะที่มีค่าฟังก์ชันวัตถุประสงค์สูงสุดแสดงว่าค้นพบสถานะเป้าหมายที่ต้องการแล้ว

ข้อดีของ Local Search Algorithm คือ สามารถค้นหาสถานะเป้าหมายที่ต้องการได้อย่างมีประสิทธิภาพบนปริภูมิที่มีขนาดไม่จำกัด (Infinite State Space) ได้ เนื่องจากจะค้นหาเพียงบางส่วนของปริภูมิ และช่วยประหยัดทั้งเวลาและพื้นที่หน่วยความจำได้ โดยตัวอย่างอัลกอริทึมของ Local Search ได้แก่ Hill Climbing Search และ Simulated Annealing Search

3.6.1 วิธีการค้นหาแบบ Hill Climbing Search

Hill Climbing Search จะค้นหาโดยเลือกสถานะที่มีค่า Heuristic ที่ดีไปเรื่อยๆ จนกว่าจะพบสถานะเป้าหมายที่ต้องการเปรียบเสมือนกับการปีนเขา (Hill Climbing) นักปีนเขาจะเลือกหาเส้นทางเดินขึ้นไปจนกว่าจะพบยอดเขาสูงสุด แต่ข้อเสียของวิธีนี้ คือ ห้ามเดินย้อนกลับไปสู่สถานะก่อนหน้าและห้ามเลือกเส้นทางสถานะที่มีค่า Heuristic แย่ลงกว่าเดิม เปรียบเสมือนกับว่าต้องปีนเขาขึ้นไปให้ถึงยอดเขาเท่านั้นห้ามมีการถอยลงเด็ดขาด

ตารางที่ 3-6 แสดงขั้นตอนการนำวิธี Hill-Climbing Search มาใช้ในการแก้ไขปัญหาเกม 8-Puzzle โดยเริ่มจากสถานะปัจจุบันและสถานะเป้าหมายดังภาพที่ 3-6 ด้วยฮิวริสติกฟังก์ชัน h_1

4	5	7
6	3	2
	1	8

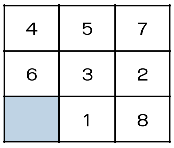
(a) สถานะเริ่มต้น

4	5	7
6	2	
1	3	8

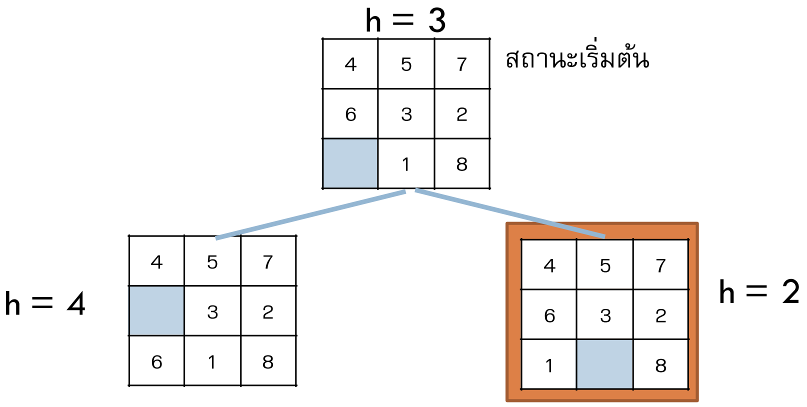
(b) สถานะเป้าหมาย

ภาพที่ 3-6 ตัวอย่างปัญหา 8-Puzzle เพื่อใช้ Hill-Climbing Search ในการแก้ปัญหา

ตารางที่ 3-6 ขั้นตอนการค้นหาเส้นทางด้วย Hill-Climbing Search ในการแก้ปัญหา 8-Puzzle



ขั้นตอนที่ 1: จำนวนแผ่นป้ายหมายเลขของสถานะปัจจุบันมีตำแหน่งจากสถานะเป้าหมาย h = 3

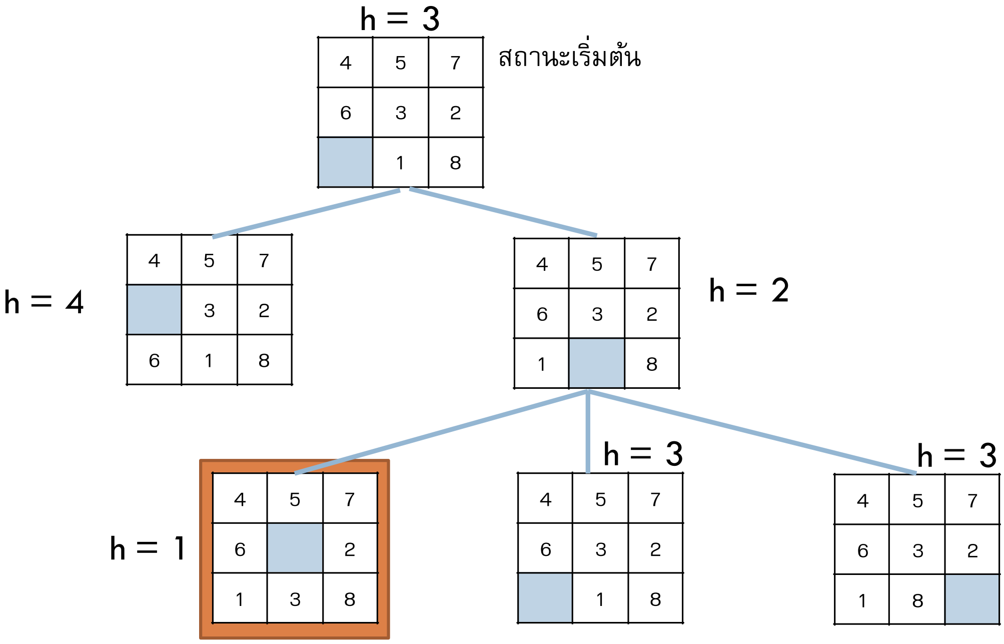


ขั้นตอนที่ 2: พิจารณาเส้นทางที่สามารถเดินทางได้จากสถานะเริ่มต้น ซึ่งจะพบว่าสามารถเลื่อนช่องว่างไปด้านบน และด้านขวาได้ เมื่อคำนวณค่าของฟังก์ชันฮิวริสติกของสถานะใหม่ที่เกิดขึ้นจะพบว่า

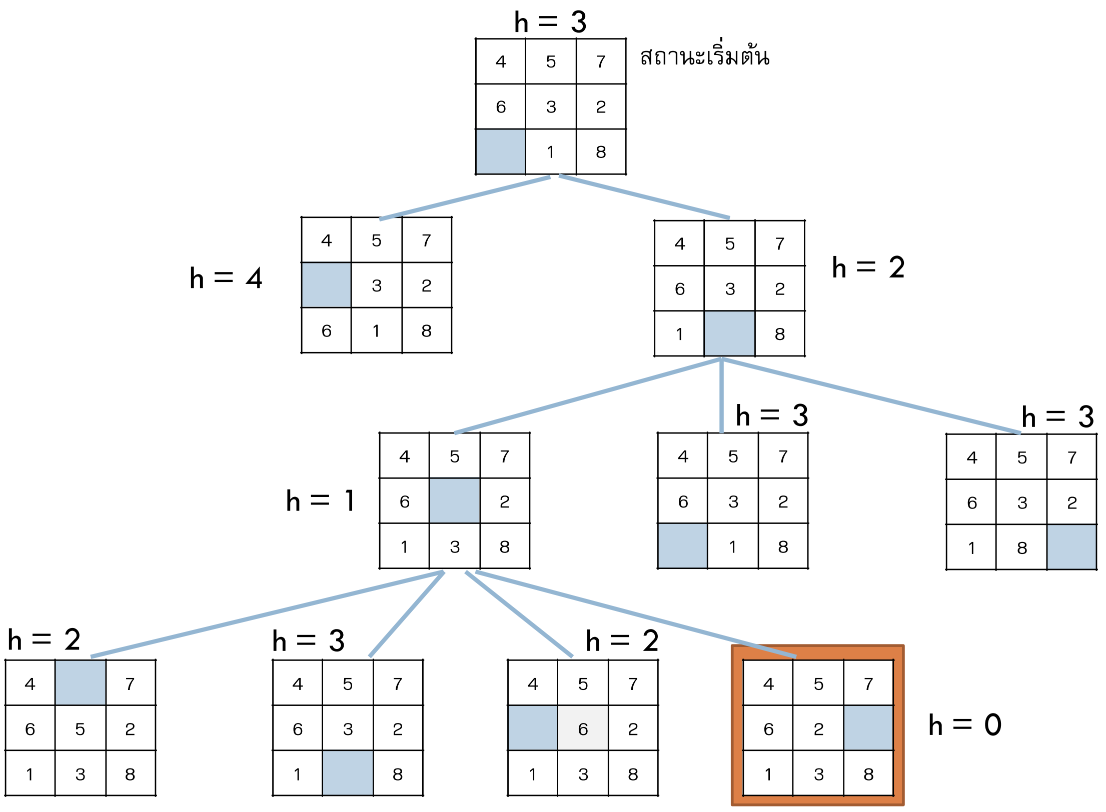
สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านบน จะมีค่า h = 4

สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านขวา จะมีค่า h = 2

ในที่นี้สามารถสรุปได้ว่าการเปลี่ยนสถานะจากสถานะเริ่มต้น โดยการเลื่อนช่องว่างไปด้านขวาให้ค่าฮิวริสติกที่ดีกว่า จึงเปลี่ยนสถานะปัจจุบันไปยังสถานะใหม่ที่เกิดจากการ**เลื่อนช่องว่างไปด้านขวา**



ตารางที่ 3-6 (ต่อ) ขั้นตอนการค้นหาเส้นทางด้วย Hill-Climbing Search ในการแก้ปัญหา 8-Puzzle

ขั้นตอนที่ 3: พิจารณาเส้นทางที่สามารถเดินทางได้จากสถานะปัจจุบัน ซึ่งจะพบว่าสามารถเลื่อนช่องว่างไปด้านบน ด้านซ้าย และด้านขวาได้ เมื่อคำนวณค่าของฟังก์ชันฮิวริสติกของสถานะใหม่ที่เกิดขึ้นจะพบว่า สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านบน จะมีค่า h = 1 สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านซ้าย จะมีค่า h = 3 สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านขวา จะมีค่า h = 3 ในที่นี้สามารถสรุปได้ว่าการเปลี่ยนสถานะจากสถานะปัจจุบัน โดยการเลื่อนช่องว่างไปด้านบนให้ค่าฮิวริสติกที่ดีกว่า จึงเปลี่ยนสถานะปัจจุบันไปยังสถานะใหม่ที่เกิดจากการ**เลื่อนช่องว่างไปด้านบน**

ขั้นตอนที่ 4: พิจารณาเส้นทางที่สามารถเดินทางได้จากสถานะปัจจุบัน ซึ่งจะพบว่าสามารถเลื่อนช่องว่างไปด้านบน ด้านล่าง ด้านซ้าย และด้านขวาได้ เมื่อคำนวณค่าของฟังก์ชันฮิวริสติกของสถานะใหม่ที่เกิดขึ้นจะพบว่า สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านบน จะมีค่า h = 2 สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านล่าง จะมีค่า h = 3 สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านซ้าย จะมีค่า h = 2 สถานะที่เกิดจากการเลื่อนช่องว่างไปด้านขวา จะมีค่า h = 0 ในที่นี้สามารถสรุปได้ว่าการเปลี่ยนสถานะจากสถานะปัจจุบัน โดยการเลื่อนช่องว่างไปด้านบนให้ค่าฮิวริสติกที่ดีกว่า จึงเปลี่ยนสถานะปัจจุบันไปยังสถานะใหม่ที่เกิดจากการ**เลื่อนช่องว่างไปด้านขวา ซึ่งเป็นสถานะเป้าหมาย**
สรุป: ดังนั้นจะเห็นได้ว่าการค้นหาสามารถค้นพบเส้นทางไปยังสถานะเป้าหมายด้วยการใช้ Successor Function ขวา -> บน -> ขวา

1	2	5
	7	4
8	6	3

(a) สถานะเริ่มต้น

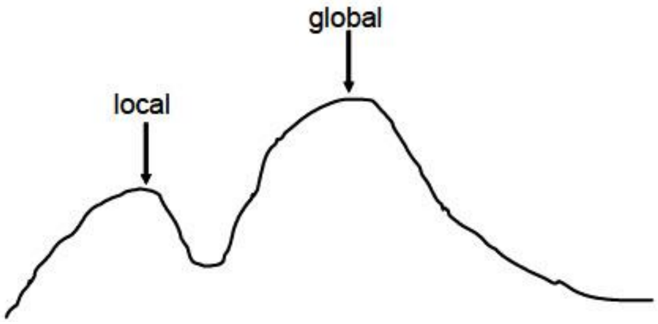
1	2	3
	7	4
8	6	5

(b) สถานะเป้าหมาย

ภาพที่ 3-7 ตัวอย่างปัญหา 8-Puzzle ที่ไม่สามารถแก้ปัญหาด้วย Hill-Climbing Search ได้

อย่างไรก็ตามก็แก้ปัญหาด้วยวิธี Hill-Climbing ไม่สามารถแก้ปัญหาได้ทุกปัญหาเนื่องด้วย บางครั้งการเลือกเส้นทางที่มีค่าฮิวริสติกดีนั้นไม่ใช่ว่าดีเสมอไป เช่นปัญหาของ 8-Puzzle ดังแสดงในภาพที่ 3-7 ถ้าใช้ฮิวริสติกฟังก์ชัน h_1 โดยพิจารณาที่สถานะเริ่มต้นค่า h = 2 จากสถานะเริ่มต้นสามารถสร้างสถานะใหม่ตาม Successor Function ได้ 3 สถานะคือ การเลื่อนช่องว่างไปด้านบน ด้านล่าง และด้านขวา โดยทั้งสามสถานะที่เกิดขึ้นใหม่นั้นเมื่อคำนวณค่าฮิวริสติกฟังก์ชันจะได้ h = 3 ซึ่งมีค่าที่แย่กว่าสถานะปัจจุบันทำให้ Hill-Climbing ไม่สามารถทำงานต่อได้และหยุดการค้นหาทันที

3.6.2 วิธีการค้นหาแบบ Simulated Annealing Search

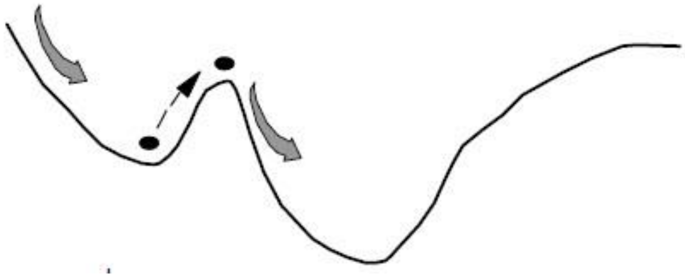


ภาพที่ 3-8 ตัวอย่างปัญหาที่เกิดกับอัลกอริทึม Hill-Climbing Search

พิจารณาตัวอย่างในภาพที่ 3-8 ซึ่งแสดงค่าฮิวริสติกด้วยความสูงจากฐานที่เกิดขึ้นระหว่างการค้นหาด้วยอัลกอริทึม Hill-Climbing ปัญหาที่เกิดขึ้นก็คืออัลกอริทึม Hill Climbing จะค้นพบสถานะที่มีค่าดีสุดเฉพาะที่ (local optimum) เท่านั้น ไม่สามารถค้นพบค่าดีสุดวงกว้าง (global optimum) ได้ เนื่องจากเมื่อการค้นหามาตกที่สถานะดีสุดเฉพาะที่แล้ว พบว่าสถานะใหม่ที่สร้างขึ้นจะมีค่าฮิวริสติกแย่ลงทั้งหมด ทำให้การค้นหาหยุดที่สถานะนั้น ปริภูมิสถานะจำนวนมากที่มีลักษณะของค่าฮิวริสติกแบบนี้ และ

ในปัญหาจำนวนมาก มักพบปริภูมิสถานะที่มีสถานะดีสุดเฉพาะที่มากกว่าหนึ่งสถานะ ทำให้การค้นหาประสบความลำบากในการค้นให้พบสถานะดีสุดวงกว้าง

ดังนั้นอัลกอริทึม Simulated annealing จึงได้ถูกออกแบบมาเพื่อให้สามารถหลุดออกได้จากสถานะดีสุดเฉพาะที่ โดยใช้แนวคิดอุณพหลศาสตร์ของกระบวนการอบเหนียว ซึ่งเป็นขั้นตอนการลดอุณหภูมิลงอย่างช้าๆ ระหว่างการหลอมเพื่อให้ได้โลหะที่อยู่ในสภาวะที่เหมาะที่สุด เป็นโลหะเหนียว ไม่เปราะ แนวคิดอธิบายให้เข้าใจได้ดีโดยดูจากภาพที่ 3-9



ภาพที่ 3-9 อัลกอริทึม Simulated annealing

เนื่องจากแนวคิดนี้เป็นการหาค่าต่ำสุด จึงใช้รูปที่กลับหัวกลับหาง ปัญหาของค่าดีสุดเฉพาะที่ (ในรูปนี้คือค่าต่ำสุดเฉพาะที่) ก็คือเมื่อการค้นหาแบบปีนเขา (ในกรณีนี้คือการค้นหาแบบลงเหว) มาตกที่สถานะนี้ การค้นหาก็จะหยุด วิธีที่จะแก้ปัญหาการตกที่สถานะนี้ได้ก็คือ ต้องดึงการค้นหาให้ขึ้นมาให้ได้เพื่อจะได้ค้นต่อไปจนพบค่าดีสุดวงกว้างอัลกอริทึมการอบเหนียวจำลองจะยอมให้การค้นหาวิ่งไปในทิศทางที่ไม่ดีได้ในช่วงเริ่มต้นของกระบวนการค้นหา เพื่อเป็นการสำรวจทั่วๆ แบบหยาบก่อน แล้วจึงค่อยๆ ค้นหาอย่างละเอียดเมื่อเวลาผ่านไป ด้วยแนวคิดเช่นนี้ทำให้เราคาดหวังว่า ผลลัพธ์สุดท้ายที่ได้จะไม่ขึ้นกับสถานะเริ่มต้นมากนัก เพื่อให้เข้าใจถึงอัลกอริทึมนี้ ขออธิบายการอบเหนียวของโลหะโดยย่อดังนี้

การอบเหนียวของโลหะเป็นการหลอมโลหะจนละลาย (ทำให้โลหะอยู่ในสถานะที่มีพลังงานสูง) แล้วค่อยๆ ลดอุณหภูมิลงทีละน้อยจนโลหะเปลี่ยนกลับมาอยู่ในสถานะของแข็งจุดมุ่งหมายคือพยายามทำให้โลหะกลับมาเป็นของแข็งในสถานะสุดท้ายที่มีพลังงานต่ำสุดซึ่งโดยทั่วไปตามธรรมชาติ สสารจะพยายามเปลี่ยนตัวเองจากสถานะที่มีพลังงานสูงไปสู่สถานะพลังงานต่ำ แต่ก็มีความน่าจะเป็นที่สสารเปลี่ยนจากพลังงานต่ำไปพลังงานสูงอยู่บ้างความน่าจะเป็นนี้สามารถคำนวณได้โดยสมการด้านล่าง

$$p = e^{-\Delta^{E/kT}}$$

โดยที่ ΔE เป็นระดับพลังงานที่เปลี่ยนไป (เป็นค่าบวก) *T* เป็นอุณหภูมิ และ *k* เป็นค่าคงที่ของโบลต์ซมันน์ (Boltzmann) การเปลี่ยนแปลงจากระดับพลังงานสูงไปต่ำนั้น มีความน่าจะเป็นที่จะเกิดในช่วงเริ่มต้นมากกว่าในช่วงปลายของกระบวนการอบเหนียวอัตราการลดอุณหภูมิในการอบเหนียวเรียกว่า *หมายกำหนดการอบเหนียว (annealing schedule)* ถ้าหมายกำหนดการอบเหนียวเร็ว กล่าวคือลดอุณหภูมิลงอย่างรวดเร็ว โลหะก็มีโอกาสเข้าสู่สถานะสุดท้ายที่มีพลังงานสูงอยู่ เราจึงต้องพิจารณาหมายกำหนดการอบเหนียวไม่ให้เร็วเกินไป แต่ถ้าช้าไปก็ทำให้เสียเวลาโดยไม่จำเป็นเช่นกันแนวคิดของการอบเหนียวจำลองก็ได้จากการล้อเลียนการอบเหนียวของโลหะ โดยเป็นการค้นหาแบบปีนเขา (ลงเหว) แบบหนึ่ง ซึ่งการค้นหาสามารถไปในทิศทางที่ไม่ดีได้โดยเฉพาะในช่วงต้นของกระบวนการค้นหา เพื่อสำรวจบริเวณที่น่าจะนำไปสู่คำตอบดีสุด เราได้สมการความน่าจะเป็นสำหรับการอบเหนียวจำลองดังนี้

$$p = e^{-\Delta^{E/T}}$$

โดยที่ ΔE เป็นค่าฮิวริสติกที่เปลี่ยนไป (เป็นค่าบวก) *T* เป็นอุณหภูมิ เนื่องจาก *k* เป็นค่าคงที่ จึงรวมเข้าไปใน *T* ได้ สมการนี้เมื่อนำไปใช้ร่วมกับการค้นหาแบบปีนเขา ก็จะช่วยให้กระบวนการค้นหาสามารถหลุดออกจากค่าดีสุดเฉพาะที่ได้ตรงกับความต้องการของเรา ตัวอย่างเช่นเมื่อเราอยู่ที่สถานะปัจจุบันมีค่าฮิวริสติกเท่ากับ A และเมื่อสร้างสถานะลูกที่มีค่าฮิวริสติกเท่ากับ B ซึ่งแย่ลง การค้นหาอาจไปยังสถานะลูกนี้ได้ โดยคำนวณค่าความน่าจะเป็น (*p*) ตามสมการ ด้านบน ได้เป็น

$$p = e^{-|A-B|/T}$$

ค่าที่ได้นี้จะอยู่ระหว่าง 0 ถึง 1 จากนั้นเราจะสุ่มตัวเลข(random(0,1)) ขึ้นมาหนึ่งตัว ถ้าค่า *p* มีค่ามากกว่า ก็จะรับสถานะลูกเป็นสถานะต่อไปได้ ค่า *T* เป็นพารามิเตอร์ของอัลกอริทึมนี้ที่เราสามารถปรับแต่งให้เหมาะสมสำหรับปัญหาหนึ่งๆ ที่เราสนใจ โดยช่วงเริ่มต้นกำหนดให้เป็นค่ามากแล้วค่อยๆ ลดลงเมื่อการค้นหาดำเนินต่อไป

เมื่อพิจารณาสมการนี้ เราจะได้คุณสมบัติที่เราต้องการดังนี้คือ เมื่อ *T* มาก (ในช่วงต้นของกระบวนการค้นหา) จะได้ *p* มีค่ามาก คือเรายอมให้การค้นหาไปในทิศทางที่ไม่ดีได้ง่ายหน่อยในช่วงต้นการค้นหา แต่เมื่อ *T* น้อย (ในช่วงปลายของกระบวนการค้นหา) จะได้ *p* มีค่าน้อยคือการค้นหาเริ่มเข้าสู่คำตอบก็ไม่ควรให้การค้นหากระโดดไปยังสถานะที่แย่ลงเมื่อพิจารณา ΔE จะพบว่า เมื่อ ΔE มีค่ามากจะได้ว่า *p* มีค่าน้อย กล่าวคือการก้าวกระโดดจากสถานะที่ดีไปยังสถานะที่ไม่ดีที่การก้าวกระโดดนั้นเป็นก้าวใหญ่ (มีการเปลี่ยนแปลงค่าฮิวริสติกมาก) จะเกิดขึ้นได้ยากกว่าการก้าวกระโดดที่เป็นก้าวเล็ก (มีการเปลี่ยนแปลงค่าฮิวริสติกน้อย) ซึ่งเป็นคุณสมบัติที่น่าพอใจ เนื่องจากเราเชื่อว่าสถานะที่เป็นคำตอบมักจะเป็นหลุมลึกๆ

ถ้าจะหลุดออกจากหลุมลึกได้ต้องก้าวใหญ่ ซึ่งไม่ควรให้เกิดขึ้นได้ง่าย ส่วนสถานะดีสุดเฉพาะที่มักเป็นหลุมตื้นๆ ดังนั้นจึงให้โอกาสหลุดลอดได้ง่ายหน่อย อัลกอริทึมการอบเหนียวจำลอง อัลกอริทึมจะมีลักษณะคล้ายกับอัลกอริทึมปีนเขา แต่สามารถเลือกสถานะที่มีค่าฮิวริสติกแย่ลงได้ ดังนั้นเมื่อสิ้นสุดกระบวนการค้นหาสถานะตัวสุดท้ายที่พิจารณาอยู่อาจไม่ใช่สถานะที่มีค่าฮิวริสติกดีสุด ดังนั้นเราจึงจำเป็นต้องจำสถานะที่มีค่าฮิวริสติกดีสุดที่ผ่านมาไว้ โดยเก็บค่าไว้ในตัวแปรชื่อ BEST-SO-FAR ในตอนเริ่มกระบวนการค้นหา จะกำหนดค่าคงที่ตัวหนึ่งเป็นอุณหภูมิการอบเหนียว (T) และลดอุณหภูมิลงตามความเหมาะสมในกรณีที่เราพิจารณาสถานะใหม่ (new state) ตัวหนึ่งอยู่ ถ้าพบว่าสถานะนี้มีค่าฮิวริสติกดีขึ้น ก็จะค้นหาต่อไปยังสถานะใหม่นี้ และปรับค่าตัวแปร BEST-SO-FAR แต่ถ้าหากว่าค่าฮิวริสติกแย่ลง เราจะคำนวณค่าความน่าจะเป็น ($e^{-\Delta E/T}$) ที่จะไปยังสถานะใหม่นี้ การประยุกต์ใช้อัลกอริทึมนี้ ผู้ใช้จำเป็นต้องกำหนดพารามิเตอร์ในอัลกอริทึมให้เหมาะสมสำหรับปัญหาที่กำลังพิจารณาอยู่ด้วย พารามิเตอร์นี้ได้แก่ ค่าอุณหภูมิเริ่มต้นและปริมาณการลดอุณหภูมิ

3.7 สรุป

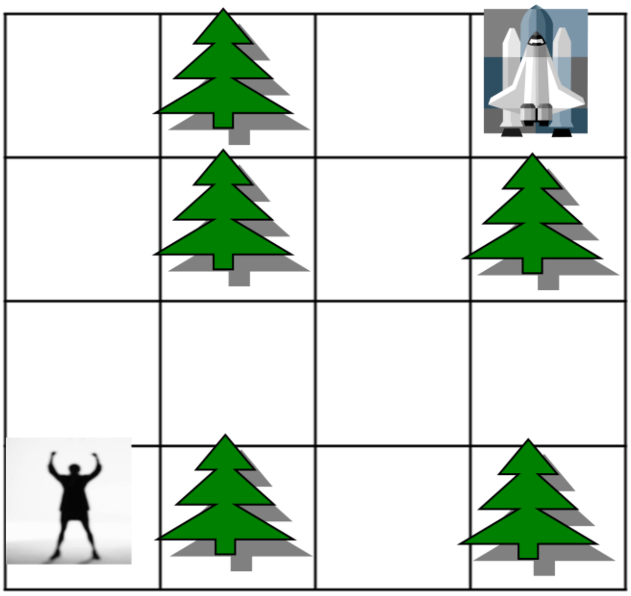
เทคนิคการค้นหาโดยมีเชาว์ปัญญาช่วยเป็นวิธีการค้นหาประเภทหนึ่งที่ใช้ค่าของ Heuristic ที่เกิดจากการคำนวณ Heuristic Function สำหรับช่วยในการตัดสินใจ โดยปกติค่า Heuristic นั้นจะถูกใช้เพื่อประเมินว่าสถานะใดๆ มีความใกล้เคียงหรือเหมาะสมในการพาไปสู่สถานะเป้าหมายมากน้อยเพียงใด เทคนิคการค้นหาแบบมีเชาว์ปัญญาช่วยนั้นแบ่งออกเป็น 2 หมวดหมู่ใหญ่ๆ คือ แบบค้นหาโดยรวม (Global Search) และแบบค้นหาภายในท้องถิ่น (Local Search)

การค้นหาประเภทค้นหาโดยรวม มีอัลกอริทึมพื้นฐานที่สำคัญอยู่ 2 อัลกอริทึม คือ Greedy Best First Search (GBFS) เป็นวิธีค้นหาที่ดูจากค่า Heuristic ตัวที่ดีที่สุดเป็นสำคัญและเลือกเส้นทางค้นหาจากสถานะที่ดีที่สุดในขณะนั้น ข้อเสียของเทคนิคประเภทนี้คือบางครั้งอาจจะไม่สามารถค้นหาเป้าหมายพบ ถ้า Heuristic Function ที่ใช้ไม่มีประสิทธิภาพเพียงพอ อีกอัลกอริทึมคือ A* ที่มีการคำนึงถึงทรัพยากรที่ใช้ไปแล้วตั้งแต่สถานะเริ่มต้นประกอบกับการใช้ค่า Heuristic ผลลัพธ์ที่ได้คือสามารถหาเส้นทางสู่สถานะเป้าหมายได้ดีขึ้น แต่อย่างไรก็ตามทั้ง 2 วิธีใช้ทรัพยากรในการค้นหาค่อนข้างมาก

การค้นหาประเภทค้นหาภายในท้องถิ่น เกิดขึ้นเพื่อแก้ปัญหาการใช้ทรัพยากรของการค้นหาประเภทค้นหาโดยรวม โดยมีอัลกอริทึมที่สำคัญคือ Hill-Climbing Search และ Simulated Annealing Search ซึ่งผลลัพธ์ที่ได้จากการค้นหาส่วนใหญ่จะไม่ใช่คำตอบที่ดีที่สุดของปัญหา แต่จะเป็นคำตอบที่ดีของบริเวณปริภูมิสถานะที่ทำการค้นหา

แบบฝึกหัด

1. อธิบายข้อแตกต่างระหว่างการค้นหาแบบ Greedy Best First Search และ A* Search
2. อธิบายลักษณะการค้นหาด้วยวิธี Hill-Climbing Search และปัญหาที่มีโอกาสเกิดขึ้นกับการค้นหาด้วยวิธีนี้
3. การค้นหาด้วยวิธี Simulated Annealing Search สามารถแก้ไขปัญหาที่เกิดจากการค้นหาด้วยวิธี Hill-Climbing Search ได้อย่างไร
4. จงเขียนปริภูมิสถานะของการค้นหาต่อไปนี้ด้วย วิธี GBFS และ A* โดยให้ใส่หมายเลขของโหนดที่ทำการหาค่าตามลำดับ พร้อมทั้งสรุปการใช้ successor function จากจุดเริ่มต้นจนถึงเป้าหมาย (ถ้าหาเป้าหมายได้) และหาค่า Total Path Cost ที่ใช้ (ถ้าหาเป้าหมายได้)



แผนปฏิบัติการสอนสัปดาห์ที่ 4

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง เทคนิคการค้นหาเมื่อมีคู่ปรปักษ์ **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. ส่วนประกอบพื้นฐานที่ต้องกำหนด	IS 4.1
2. วิธีการค้นหาแบบ Minimax	IS 4.2
3. วิธีการค้นหาแบบ Alpha-Beta Pruning	IS 4.3
4. การประยุกต์ใช้งานเทคนิคการค้นหาเมื่อมีคู่ปรปักษ์ในโลกความเป็นจริง	IS 4.4

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2-3-4**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 4
เทคนิคการค้นหาเมื่อมีคู่ปรปักษ์

ปัญหาของเกมที่มีการแข่งขันกันระหว่างผู้เล่น 2 คน จะไม่สามารถใช้เทคนิคการค้นหาปกติเพื่อแก้ไขปัญหาได้ เนื่องจากต้องมีการคำนึงถึงการเล่นของฝ่ายตรงข้ามด้วย ดังนั้นเพื่อแก้ปัญหาในลักษณะนี้จึงมีการประยุกต์ใช้เทคนิคการค้นหาเมื่อมีคู่ปรปักษ์ (Adversarial Search) โดยปกติเกมที่นำมาพิจารณานั้นผู้เล่นจะต้องทราบข้อมูลภายในเกมขณะนั้นทั้งหมด เช่น เกมหมากรุก และเกม tic-tac-toe เป็นต้น

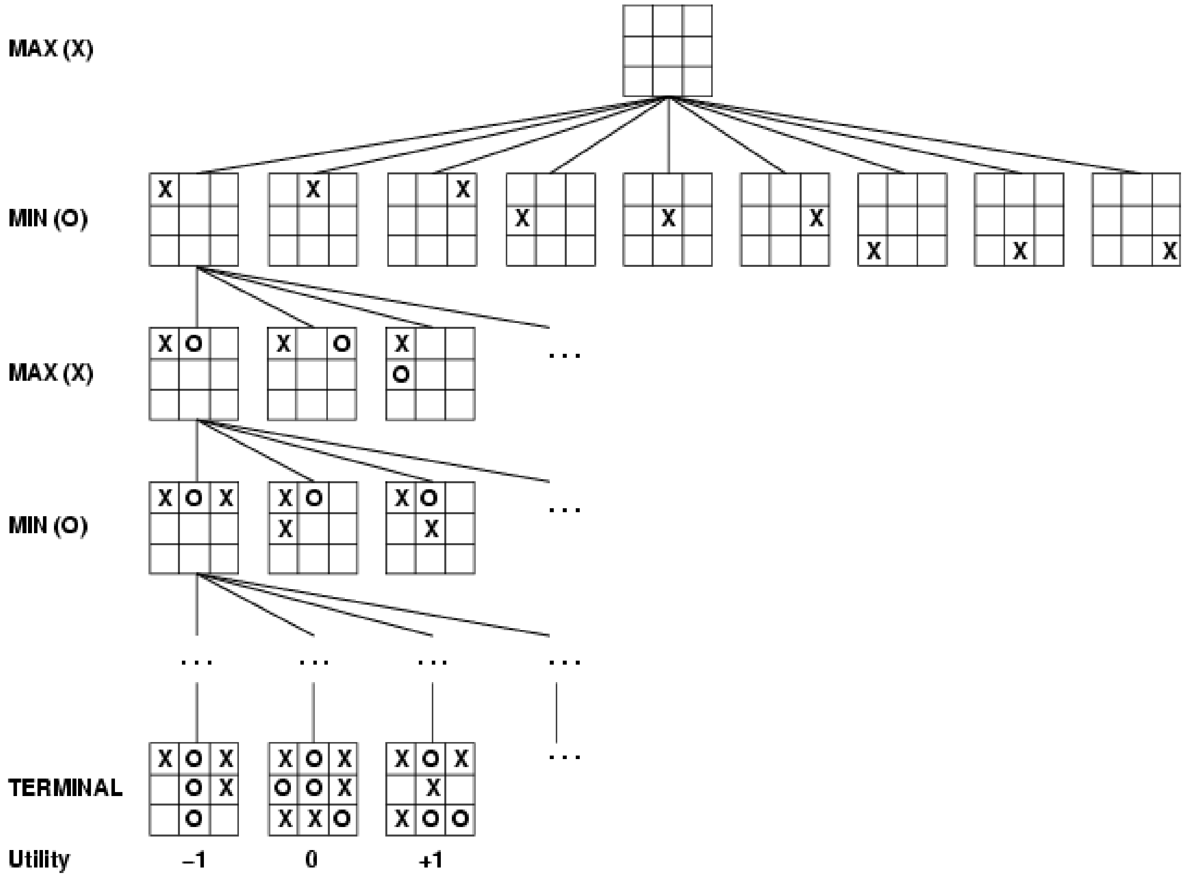
4.1 ส่วนประกอบพื้นฐานที่ต้องกำหนด

ส่วนประกอบพื้นฐานที่ต้องกำหนดใน Adversarial Search Techniques มีดังนี้

- **Initial state** กำหนดสถานะเริ่มต้น
- **Successor Function** กำหนดเซตของการกระทำทั้งหมดที่เป็นไปได้
- **Terminal Test** ตัวกำหนดการสิ้นสุด สถานะสิ้นสุดเรียกว่า *Terminal state*
- **Utility Function** ฟังก์ชันที่กำหนดค่าของ Terminal state เป็นตัวเลขบ่งบอกผลลัพธ์ของเกม เช่น ชนะ (+1) แพ้ (-1) เสมอ (0) ถ้าบางเกมมีการวัดผลที่ซับซ้อนอาจใช้ค่าได้

ตัวอย่างปัญหาเกม tic-tac-toe ที่แสดงส่วนประกอบพื้นฐานต่างๆ ดังภาพที่ 4-1 โดยผู้เล่นคนแรกคือผู้เล่นที่ลง “X” ซึ่งจะเห็นได้ว่าภายในปริภูมิสถานะจะต้องกำหนดส่วนประกอบพื้นฐาน ได้แก่ Initial State. Terminal State และ ค่า Utility ส่วนเส้นเชื่อมระหว่างสถานะจะเกิดจากการกระทำในเซตของ Successor Function

เทคนิควิธีการค้นหาคำตอบของเกมประเภทนี้มีเทคนิคที่นิยมกันอยู่ 2 เทคนิคคือ เทคนิควิธีการค้นหาแบบ Minimax และ เทคนิควิธีการค้นหาแบบ Alpha-Beta Pruning ซึ่งโดยปกติถ้าเป็นการแก้ไขปัญหาที่มีจำนวนเส้นทางสำรวจน้อย สามารถใช้วิธีการค้นหาแบบ Minimax ได้เลยเนื่องจากง่ายต่อการพัฒนาโปรแกรม แต่หากปัญหานั้นมีจำนวนเส้นทางสำรวจกว้างจะใช้วิธีการค้นหาแบบ Alpha-Beta Pruning (α-β Pruning) มาช่วยตัดบางเส้นทางที่ไม่จำเป็นออกไป ทำให้การค้นหาใช้ทรัพยากรน้อยลงและรวดเร็วมากขึ้น



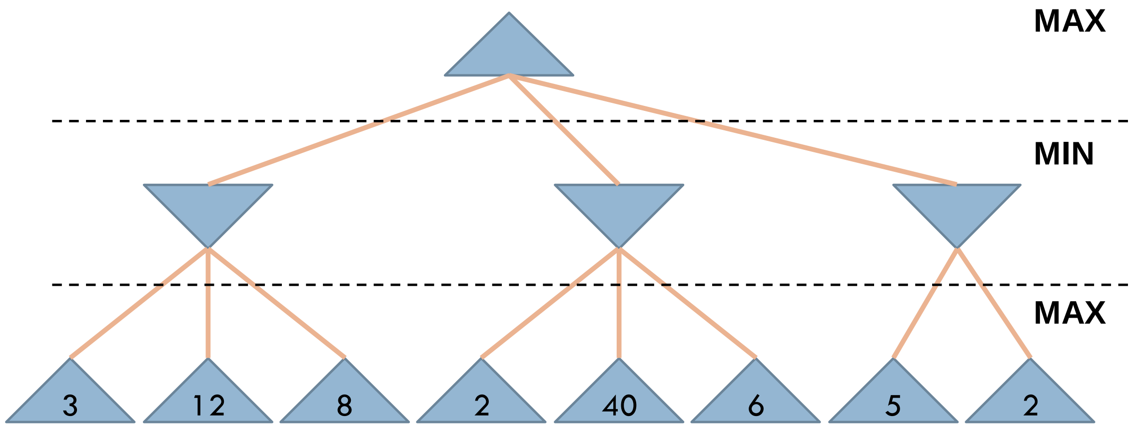
ภาพที่ 4-1 ปริภูมิสถานะของเกม tic-tac-toe ที่ผู้เล่น X และ O สลับกันลง

4.2 วิธีการค้นหาแบบ Minimax

Minimax จะมีลักษณะการค้นหาแบบ Depth-First Search โดยพิจารณาค่าต่างๆ จากโหนดใบไม้และเลือกค่าที่เหมาะสมขึ้นมาบนโหนดพ่อแม่ ทำเช่นนี้ไปจนกว่าโหนดรากจะได้รับค่า เกมที่นำมาพิจารณามักเป็นเกมที่เล่นกัน 2 คน กำหนดให้ผู้เล่นคนหนึ่งเป็น “MAX” (แทนผู้เล่นคือเรา) วัตถุประสงค์ของ MAX คือ การทำคะแนนให้มากที่สุดหรือหาทางชนะ ส่วนผู้เล่นอีกคนเป็น “MIN” วัตถุประสงค์ คือ พยายามทำให้ MAX ได้ค่าน้อยที่สุด หรือพยายามทำให้ MAX แพ้นั่นเอง โดยสัญลักษณ์ของโหนด MAX และ MIN จะมีลักษณะดังภาพที่ 4-2

(a) โหนด MAX (b) โหนด MIN

ภาพที่ 4-2 สัญลักษณ์ของโหนด MAX และโหนด MIN



ภาพที่ 4-3 ตัวอย่างต้นไม้ของ Minimax

ภาพที่ 4-3 แสดงตัวอย่างของปัญหาที่จะใช้เทคนิค Minimax มาพิจารณา โดยเส้นทางที่ผู้เล่นควรเลือกเดินนั้นจะเลือกตัวเลขที่ดีที่สุด (โหนด MAX จะเลือกตัวเลขมากที่สุด และโหนด MIN จะเลือกตัวเลขน้อยที่สุด) จากตัวอย่างเป็นโครงสร้างต้นไม้ที่มี Branching Factor เป็น 3 ส่วน โหนดใบไม้จะมีค่าต่างๆ กำหนดไว้ โดยค่าเหล่านี้มาจาก Utility Function ที่ผู้พัฒนาเกมกำหนดให้กับสถานะสิ้นสุด

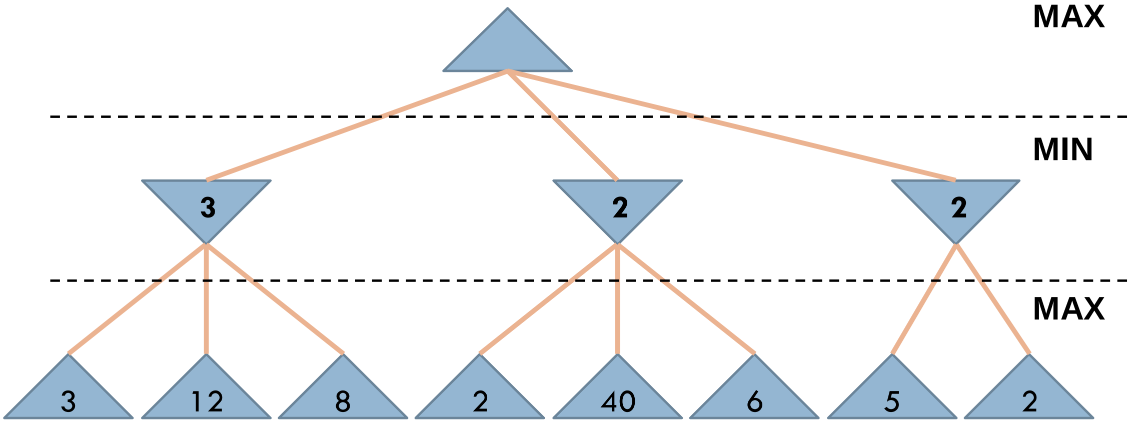
ซึ่งแต่ละโหนดจะมีค่า MinimaxValue ที่สามารถคำนวณได้จากฟังก์ชันดังนี้

$$\text{MinimaxValue}(n) = \begin{cases} \text{Utility}(n) & ;\ \text{ถ้า } n \text{ เป็นสถานะสิ้นสุด} \\ \max_{s \in Successors(n)} \text{MinimaxValue}(s) & ;\ \text{ถ้า } n \text{ เป็นโหนด MAX} \\ \min_{s \in Successors(n)} \text{MinimaxValue}(s) & ;\ \text{ถ้า } n \text{ เป็นโหนด MIN} \end{cases}$$

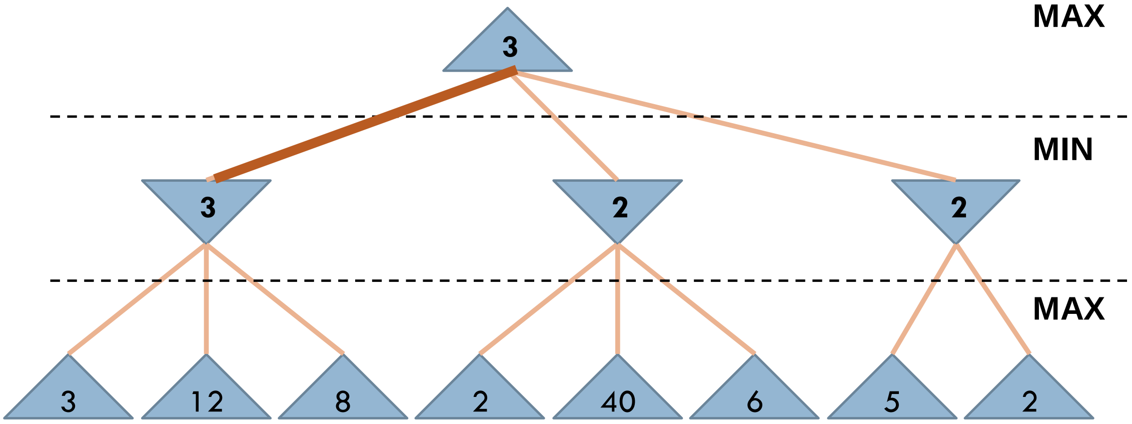
การพิจารณาจะใช้วิธีเลือกค่าจากโหนดใบไม้ โดยเริ่มจากต้นไม้ย่อย (Subtree) ข้างซ้ายก่อน

- โหนด MIN ตัวซ้ายสุดจะมีค่า MinimaxValue ซึ่งจะเลือกค่าที่น้อยสุดจากโหนดลูก คือ 3, 12, และ 8 ดังนั้นค่า MinimaxValue ของโหนด MIN คือ min{3, 12, 8} = 3
- โหนด MIN ตัวกลางจะมีค่า MinimaxValue ซึ่งจะเลือกค่าที่น้อยสุดจากโหนดลูก คือ 2, 40, และ 6 ดังนั้นค่า MinimaxValue ของโหนด MIN คือ min{2, 40, 6} = 2
- โหนด MIN ตัวขวาสุดจะมีค่า MinimaxValue ซึ่งจะเลือกค่าที่น้อยสุดจากโหนดลูก คือ 5 และ 2 ดังนั้นค่า MinimaxValue ของโหนด MIN คือ min{5, 2} = 2

ทำให้ค่าของต้นไม้ Minimax สามารถแสดงได้ดังภาพที่ 4-4



ภาพที่ 4-4 ตัวอย่างต้นไม้ของ Minimax หลังจากคำนวณค่าโหนด MIN

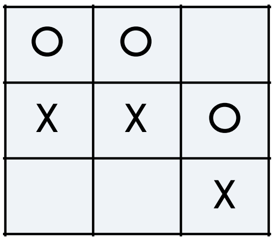


ภาพที่ 4-5 ตัวอย่างต้นไม้ของ Minimax หลังจากคำนวณค่าโหนด MAX

เมื่อได้ค่า MinimaxValue ของโหนด MIN ครบทุกโหนด ก็ทำการพิจารณาโหลด MAX ที่เป็นรากของต้นไม้ โดยค่า MinimaxValue ของ MAX คำนวณจาก max{3, 2, 2} = 3 โดยจะพบว่าเส้นทางที่จะพาให้ผู้เล่นไปยังโหลดเป้าหมายที่ดีที่สุดคือการเดินทางไปยังสถานะลูกด้านซ้าย ดังแสดงในภาพที่ 4-5

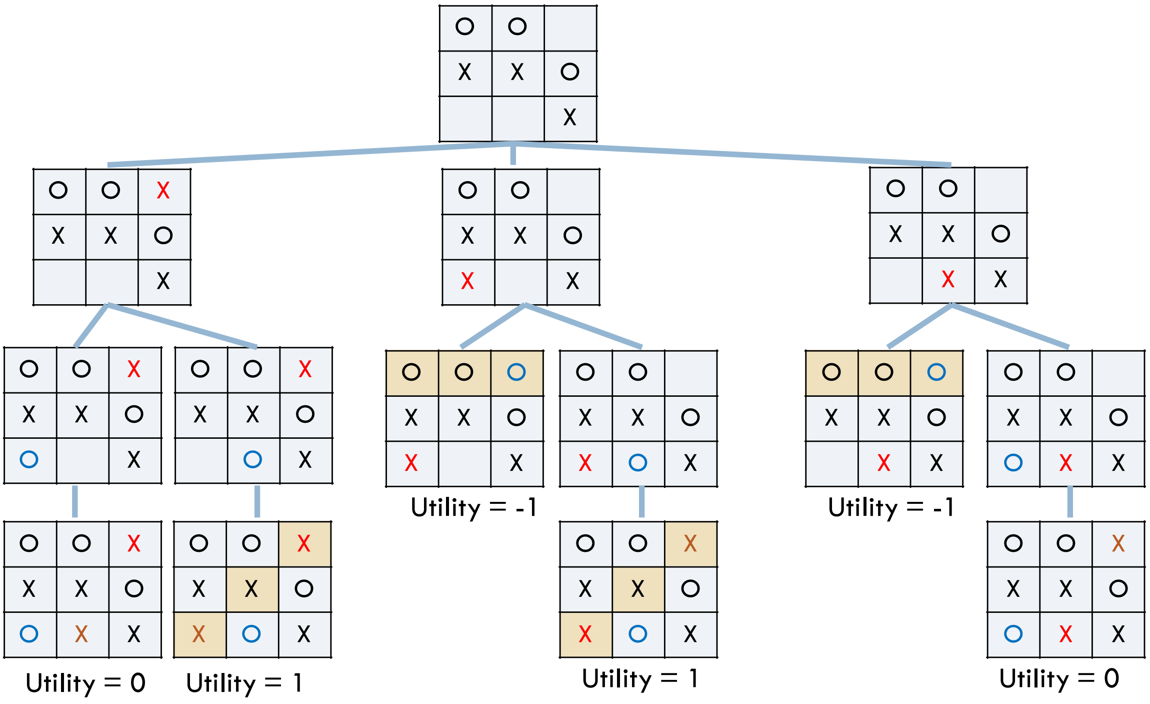
ตัวอย่างการใช้ Minimax แก้ปัญหาเกม tic-tac-toe

กำหนดสถานะปัจจุบันให้กับเกม tic-tac-toe ดังแสดงในภาพที่ 4-6 โดยจะเป็นตาที่ผู้เล่น X ต้องการลง กำหนดค่า Utility Function ให้ผู้เล่น X แพ้มีค่าเป็น -1, ถ้าผู้เล่น X ชนะมีค่าเป็น 1 และค่าเสมอกันมีค่าเป็น 0



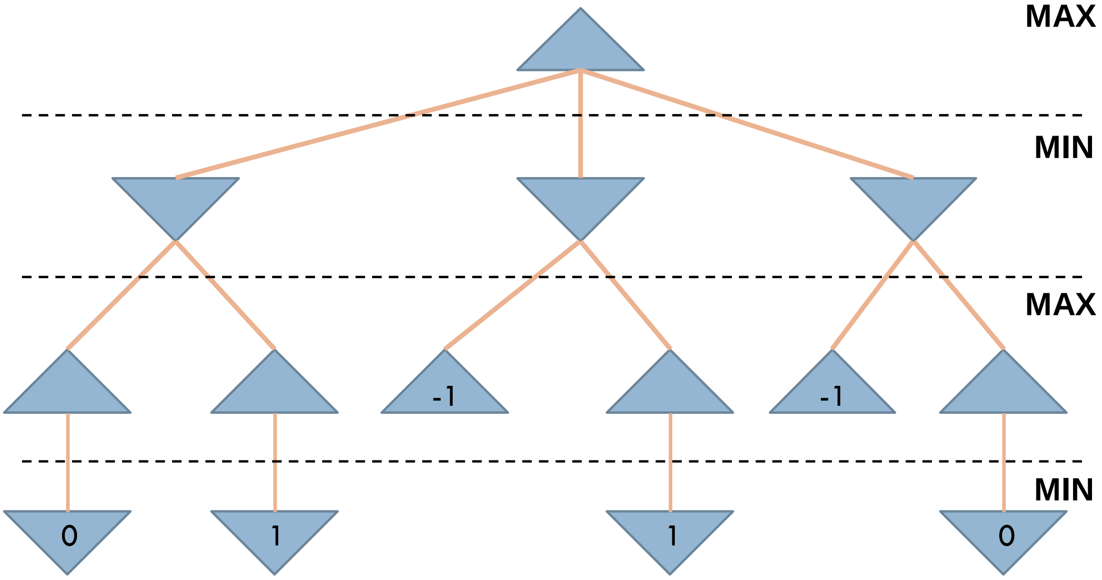
ภาพที่ 4-6 สถานะปัจจุบันของเกม tic-tac-toe

เมื่อนำสถานะปัจจุบันมาเขียนปริภูมิสถานะพร้อมกำหนดค่า Utility Function ให้กับสถานะสิ้นสุด (Terminal State) เรียบร้อยแล้วจะมีลักษณะดังแสดงในภาพที่ 4-7



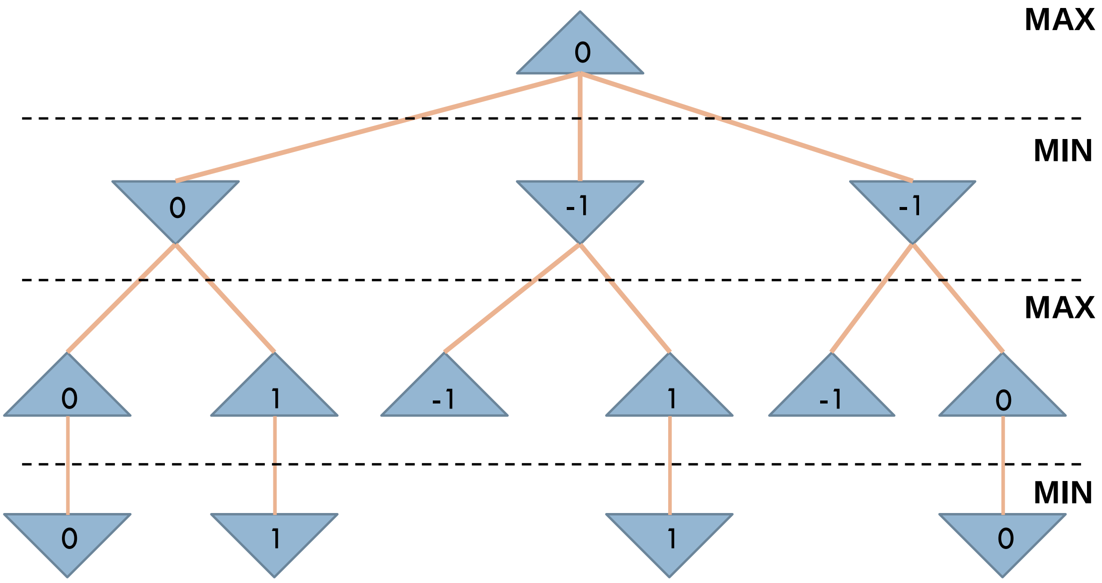
ภาพที่ 4-7 ปริภูมิสถานะของตัวอย่างเกม tic-tac-toe

จากภาพที่ 4-7 สามารถแปลงออกมาเป็นต้นไม้ Minimax โดยกำหนดโหนดรากให้เป็นโหนดในระดับชั้น MAX จะได้ต้นไม้ Minimax ดังแสดงในภาพที่ 4-8

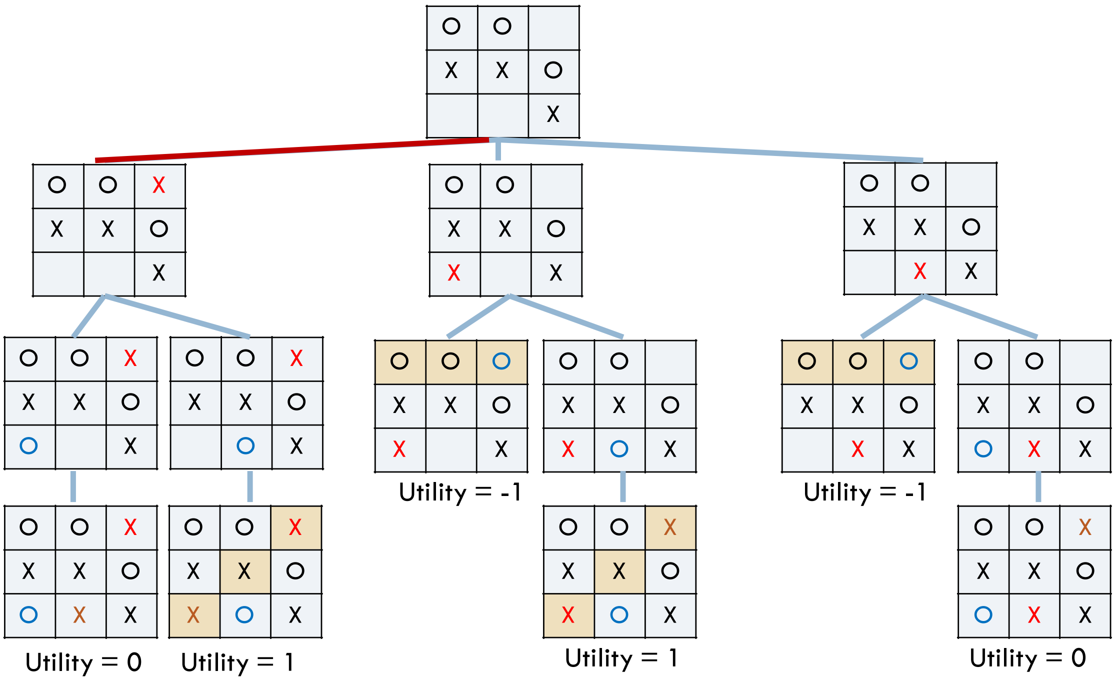


ภาพที่ 4-8 ต้นไม้ Minimax ของตัวอย่างเกม tic-tac-toe

เมื่อทำการคำนวณของ MinimaxValue ประจำโหนดต่างๆ ภายในต้นไม้ Minimax จะทำให้ได้ต้นไม้ดังภาพที่ 4-9 และจะพบว่าค่า MinimaxValue ของสถานะปัจจุบันคือ 0 และเส้นทางที่จะทำให้ได้ค่านี้คือการเลือกเดินทางไปทางโหนดด้านซ้าย ซึ่งหมายความว่าถ้าเลือกเดินทางด้านซ้ายสถานการแย่ที่สุดก็คือเสมอ แต่ถ้าหากเลือกเดินทางไปยังโหนดกลาง หรือโหนดด้านขวา สถานการแย่สุดก็คือแพ้



ภาพที่ 4-9 ต้นไม้ Minimax ที่สมบูรณ์ของตัวอย่างเกม tic-tac-toe



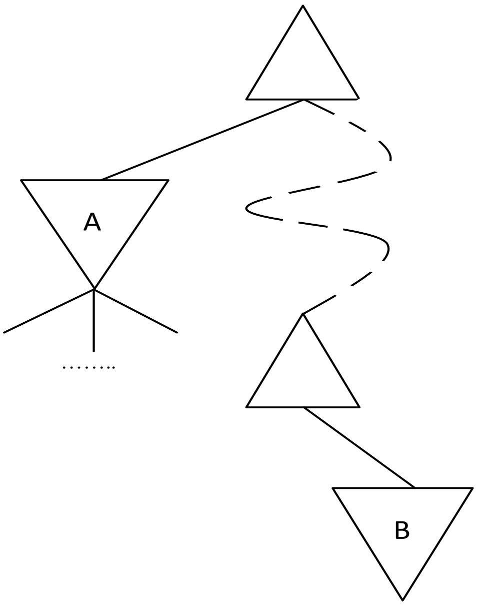
ภาพที่ 4-10 เส้นทางที่ควรเลือกภายในปริภูมิสถานะของตัวอย่างเกม tic-tac-toe

เมื่อแปลงต้นไม้ Minimax กลับมาเป็นปริภูมิสถานะเดิมของตัวอย่างเกม tic-tac-tae แสดงดังภาพที่ 4-10 ซึ่งจะทำให้ทราบว่าผู้เล่น X ควรจะลง X ที่มุมขวาบนของตารางเกม

4.3 วิธีการค้นหาแบบ Alpha-Beta Pruning

การค้นหาแบบ Minimax เป็นกระบวนการที่มุ่งหาเส้นทางจากทุกกิ่ง หากเกมมีพื้นที่สำรวจกว้าง จะทำให้เสียเวลาสำหรับการคำนวณและเสียพื้นที่หน่วยความจำเพิ่มขึ้น เพราะต้องเก็บทุกๆเส้นทางที่พิจารณา โดยอาจจะมีบางเส้นทางที่ไม่จำเป็นหรือไม่ได้นำมาใช้งาน จึงได้พัฒนาอัลกอริธึมเพื่อแก้ปัญหาดังกล่าวที่เรียกว่า “Alpha-Beta Pruning หรือ Alpha-Beta Cutoff ” โดยเพิ่มเทคนิคการตัดกิ่งหรือเส้นทางที่ไม่จำเป็นออกไป (Pruning/Cutoff) เพื่อประหยัดทรัพยากร เช่น พื้นที่หน่วยความจำที่ใช้และเวลาสำหรับการคำนวณ

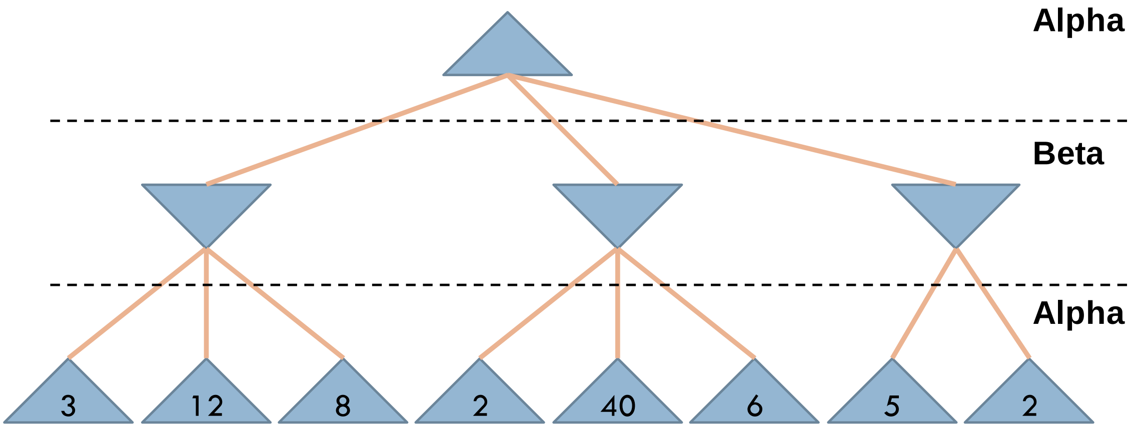
ภาพที่ 4-11 แสดงตัวอย่างหลักการของ ของ Alpha-Beta Pruning พบว่าหากโหนด A มีค่าดีกว่าโหนด B จะทำให้เส้นทางของโหนด B ไม่ถูกนำมาพิจารณา ดังนั้นจึงสามารถตัดกิ่งเส้นทางของโหนด B ออกได้ โดยไม่ส่งผลกระทบต่อผลผลลัพธ์สุดท้าย



ภาพที่ 4-11 หลักการของ Alpha-Beta Pruning

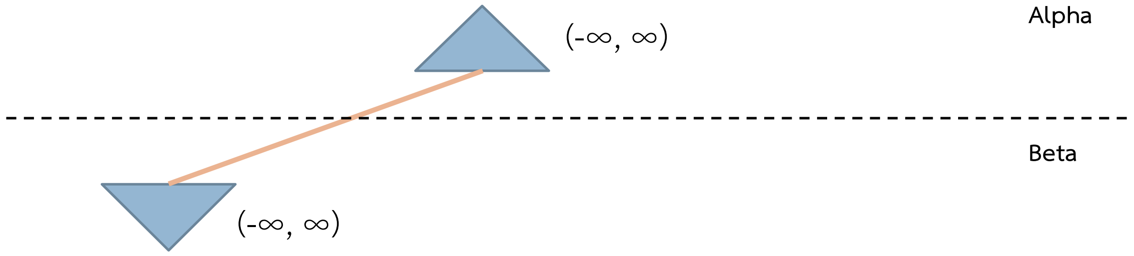
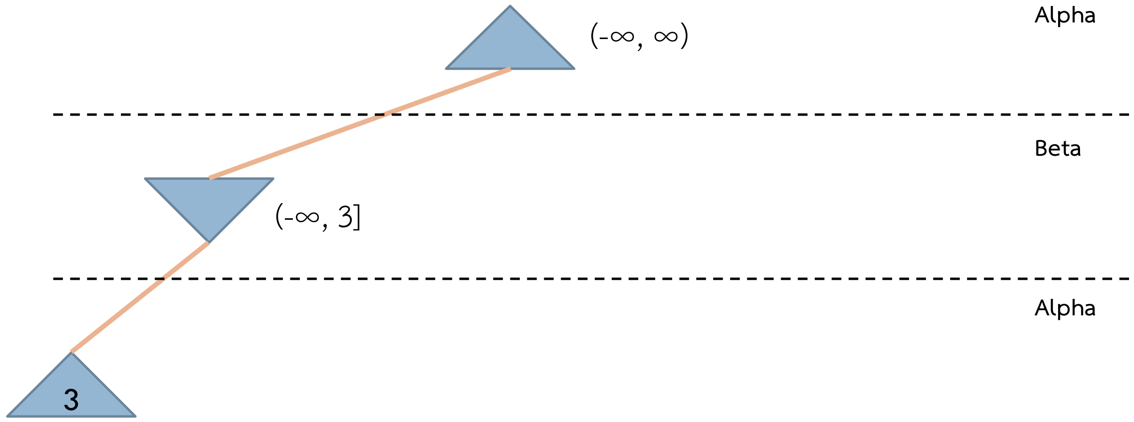
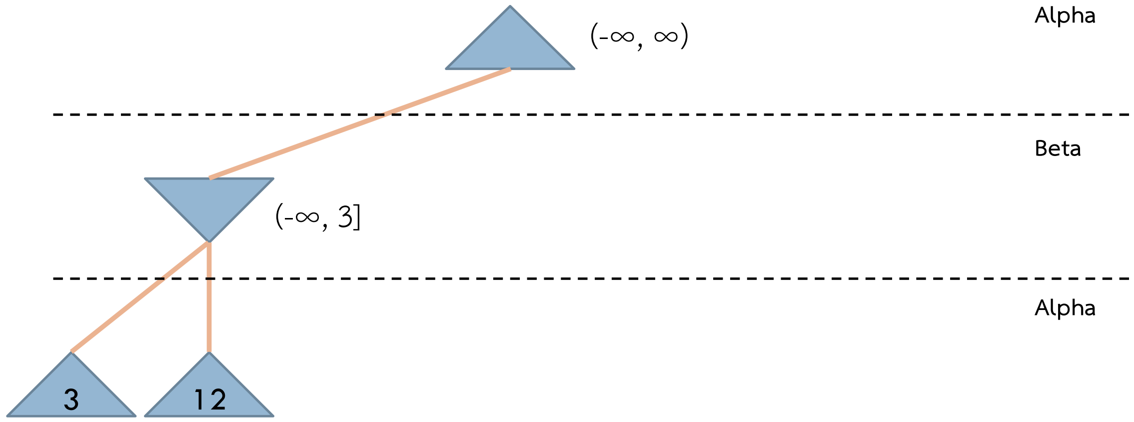
เทคนิคการตัดกิ่งสามารถนำ Alpha กับ Beta มาร่วมพิจารณาได้ โดยค่า Alpha (α) เป็นค่าดีที่สุดที่นำมาช่วยเลือกให้กับโหนด MAX ซึ่งค่าจะไม่สามารถลดลงได้ ส่วนค่า Beta (β) เป็นค่าดีที่สุดที่นำมาช่วยเลือกให้กับโหนด MIN ซึ่งค่าจะไม่สามารถเพิ่มขึ้นได้ วิธีพิจารณาการตัดกิ่งจะทำได้เมื่อพิสูจน์ว่าช่วงค่า Alpha จะต้องมากกว่าหรือเท่ากับ Beta จริง

การนำวิธี Alpha-Beta Pruning ไปประยุกต์โดยใช้ตัวอย่างเดียวกับการค้นหาด้วยวิธี Minimax (ภาพที่ 4-3) แต่คำที่ใช้เรียกชื่อระดับชั้นจะมีการเปลี่ยนแปลงจาก MAX และ MIN เป็น Alpha และ Beta โดย Alpha เลือกค่าที่ดีที่สุดให้กับโหนด MAX และค่า Beta สามารถเลือกค่าที่ดีที่สุดให้กับโหนด MIN ดังภาพที่ 4-12 และมีขั้นตอนการค้นหาดังแสดงในตารางที่ 4-1

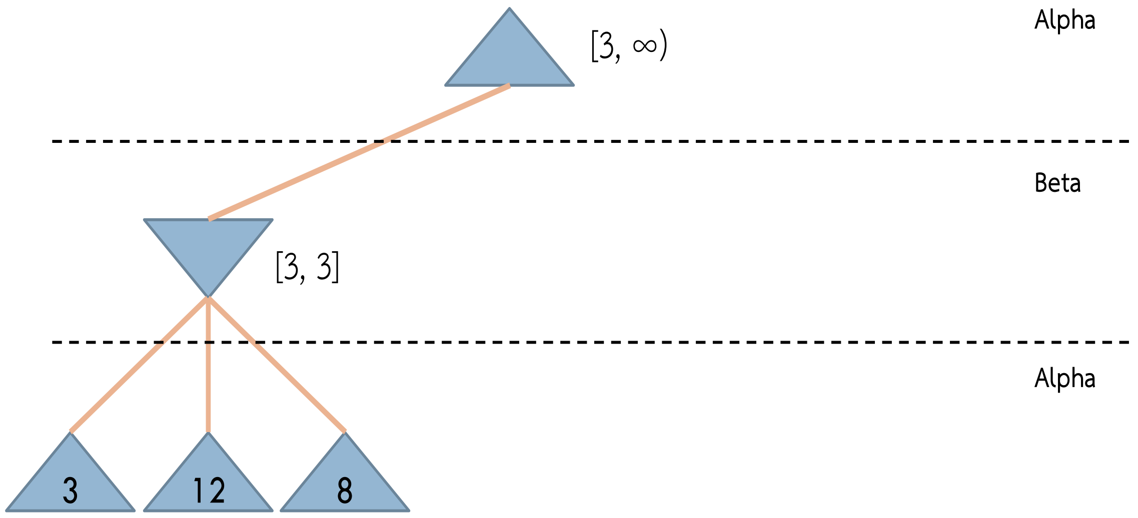


ภาพที่ 4-12 ตัวอย่างต้นไม้ Alpha-Beta

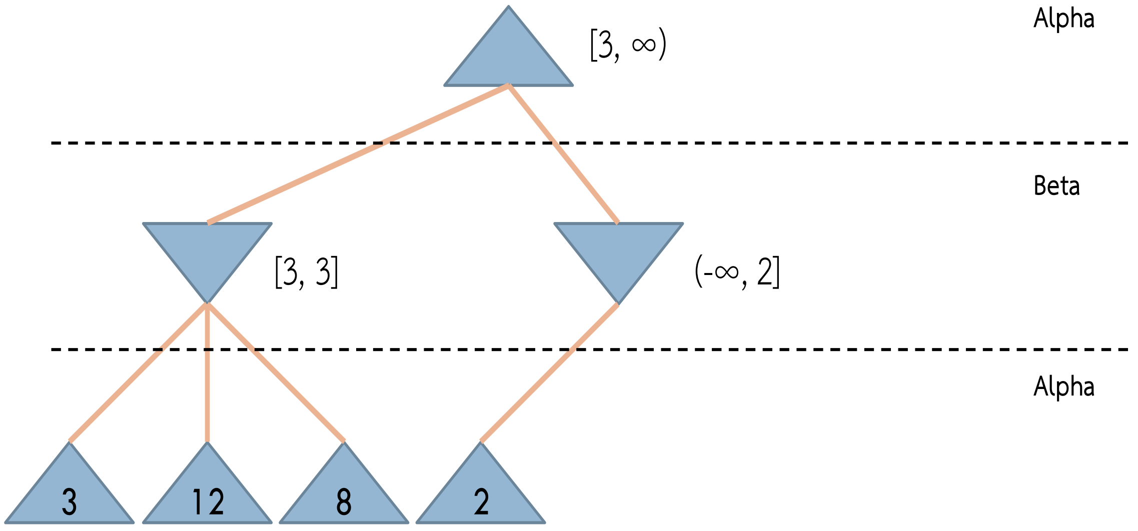
ตารางที่ 4-1 ขั้นตอนการค้นหาด้วยวิธี Alpha-Beta Pruning

ขั้นตอนที่ 1 กำหนดค่าเริ่มต้นให้กับโหนดรากที่อยู่ในระดับชั้นของ Alpha เป็น (-∞, ∞)

ขั้นตอนที่ 2 สำรวจสถานะลูกแบบลงลึกทางด้านซ้ายก่อน โดยโหนดที่สร้างใหม่ก็มีค่าประจำโหนดเป็น (-∞, ∞)

ขั้นตอนที่ 3 สำรวจสถานะลูกด้านซ้ายก่อน เริ่มจากโหนดซ้ายสุดที่มีค่าเป็น 3 โดยโหนด MIN ต้องการค่าที่น้อยที่สุดทำให้พิจารณาได้ว่าโหนด MIN มีค่าอย่างมากที่สุดเป็น 3 ดังนั้นช่วงค่า Beta คือ (-∞,3]

ขั้นตอนที่ 4 สำรวจสถานะลูกถัดไป ซึ่งมีค่าประจำโหนดเป็น 12 จึงไม่มีการเปลี่ยนแปลงค่า Beta

ตารางที่ 4-1(ต่อ) ขั้นตอนการค้นหาด้วยวิธี Alpha-Beta Pruning

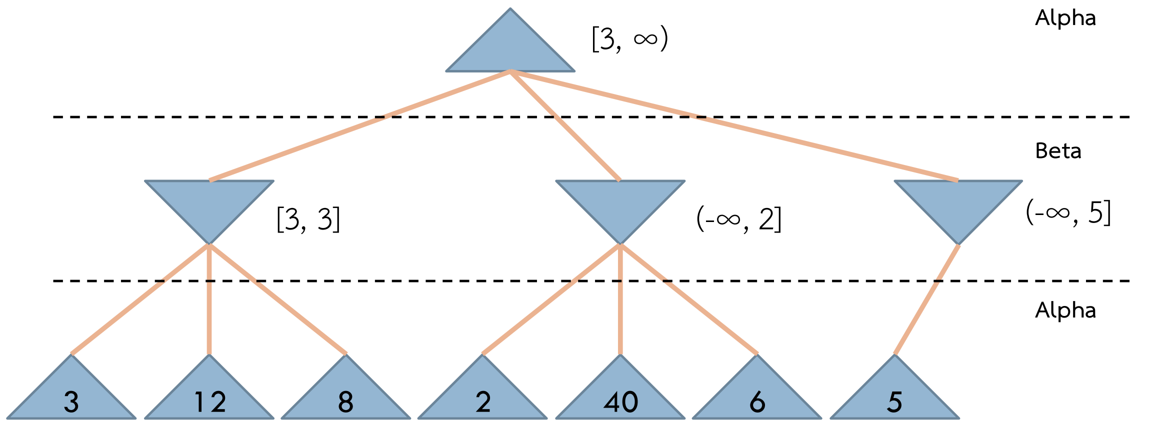


ขั้นตอนที่ 5 พิจารณาค่าสุดท้ายของต้นไม้ย่อยชุดนี้คือ 8 จะพบว่ามีค่ามากกว่า 3 อีกเช่นกันจึงใช้ไม่ได้ ทำให้ Beta เลือกค่า 3 ลงในโหนด MIN เนื่องจากมีค่าน้อยสุด ดังนั้นช่วงของค่า Beta จะได้เป็น [3,3] และช่วงของค่า Alpha สำหรับโหนด MAX ด้านบน คือ [3 , +∞) เนื่องจากค่าโหนด MAX ต้องการค่าที่มากที่สุดอย่างต่ำต้องที่เป็น 3

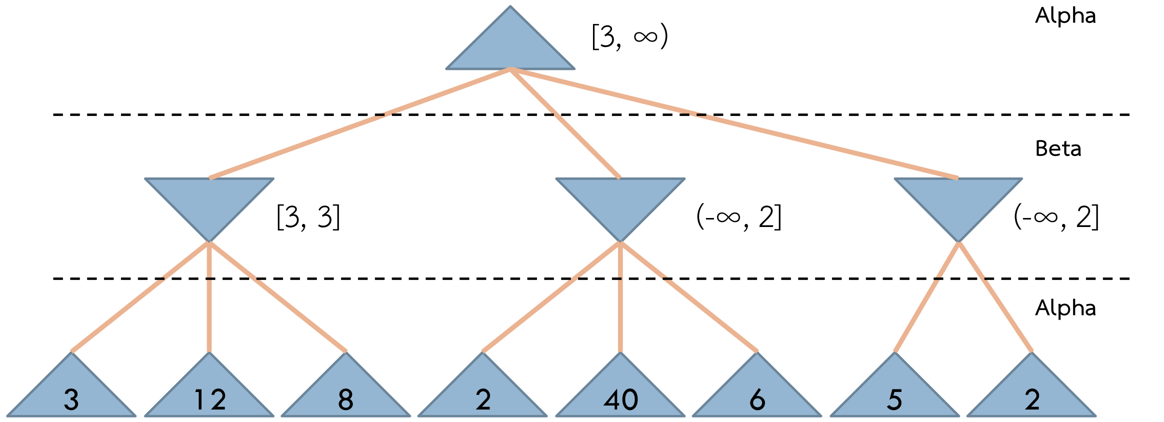


ขั้นตอนที่ 6 พิจารณากลุ่มต้นไม้ย่อยชุดถัดไปเริ่มจากโหนดแรกมีค่าเป็น 2 กล่าวได้ว่าค่าของโหนด MIN ในต้นไม้ย่อยชุดนี้จะมีช่วงของค่า Beta อย่างมากที่สุด เป็น 2 แต่ช่วงของค่า Alpha ของโหนด MAX ด้านบนจะมีค่าอย่างต่ำเป็น 3 ทำให้ค่า Alpha มากว่า Beta อย่างแน่นอน จึงไม่จำเป็นต้องเลือกค่าถัดไปในต้นไม้ย่อยชุดนี้ สามารถตัดกิ่งได้

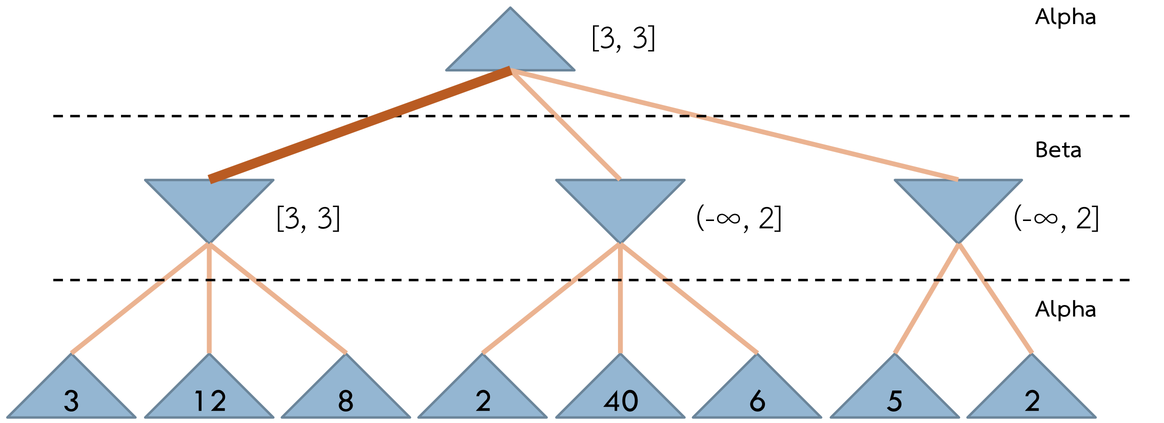
ตารางที่ 4-1(ต่อ) ขั้นตอนการค้นหาด้วยวิธี Alpha-Beta Pruning



ขั้นตอนที่ 7 พิจารณากลุ่มต้นไม้ย่อยชุดถัดไปเริ่มจากโหนดแรกมีค่าเป็น 5 โดยโหนด MIN ต้องการค่าที่น้อยที่สุดทำให้พิจารณาได้ว่าโหนด MIN มีค่าอย่างมากที่สุดเป็น 5 ดังนั้นช่วงค่า Beta คือ (-∞,5]



ขั้นตอนที่ 8 พิจารณาโหนดถัดไปมีค่าเป็น 2 กล่าวได้ว่าค่าของโหนด MIN ในต้นไม้ย่อย ชุดนี้จะมีช่วงของค่า Beta อย่างมากที่สุด เป็น 2 แต่ช่วงของค่า Alpha ของโหนด MAX ด้านบนจะมีค่าอย่างต่ำเป็น 3 ทำให้ค่า Alpha มากกว่า Beta อย่างแน่นอน จึงไม่จำเป็นต้องเลือกค่าถัดไปในต้นไม้ย่อยชุดนี้ สามารถตัดกิ่งได้ (แต่ไม่มีกิ่งเหลือพอดี)



ขั้นตอนที่ 9 ค่า Alpha ของโหนดบนสุดจึงมีค่าเป็นมากสุดเป็น 3 และเลือกเดินทางไปยังด้านซ้าย

4.4 การประยุกต์ใช้งานเทคนิคการค้นหาเมื่อมีคู่ปรปักษ์ในโลกความเป็นจริง

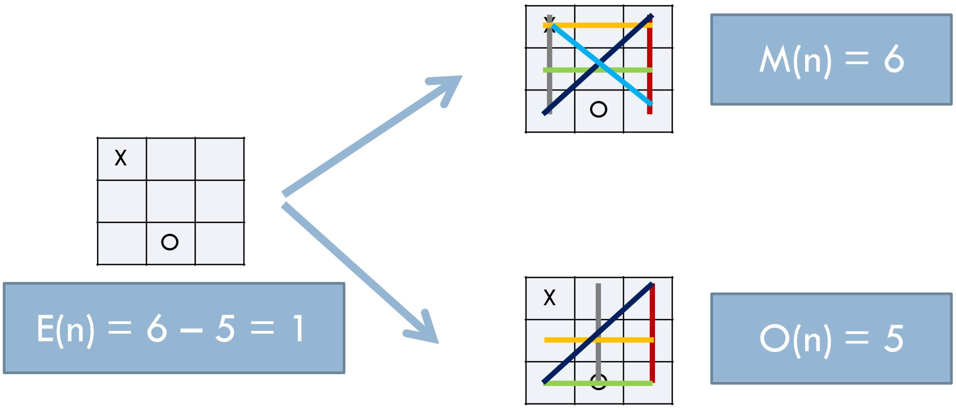
พิจารณาจากเกม tic-tac-toe อีกครั้ง จากตัวอย่างที่ผ่านมาการกำหนดค่า Utility Function แบบมี 3 ค่าคือ -1 สำหรับผู้เล่นแพ้, 0 สำหรับการเสมอกัน และ 1 สำหรับผู้เล่นชนะ อาจจะไม่สะดวกมากนัก เนื่องจากความลึกสุดสูงของปริภูมิสถานะคือ 9 ชั้น และมีจำนวนสถานะสูงสุดขอบเขตบน (รวมข้อมูลที่ซ้ำ และไม่หยุดเมื่อมีการแพ้ชนะ) อยู่ทั้งหมด 9! = 362,880 สถานะ ซึ่งการที่จะค้นหาไปจนรู้ค่า Utility Function ของ Terminal State นั้นจะสิ้นเปลืองทั้งเวลาและหน่วยความจำ โดยเฉพาะอย่างยิ่งถ้าปัญหาเกมประเภทอื่นที่มีความลึกและความซับซ้อนของปัญหามากขึ้นก็จะยิ่งแย่ลงไปอีก

ดังนั้นจึงมีการนำ Heuristic Function เข้ามาช่วยในการคำนวณค่า Utility Function ของทุกสถานะแทน โดยจะใช้ค่า Heuristic Function ที่คำนวณได้นั้นมาทำหน้าที่แทนค่า Utility Function เลย โดยกำหนดจำนวนชั้นในการค้นหาเพื่อเพิ่มความรวดเร็วและลดหน่วยความจำที่ใช้ในการค้นหา

4.4.1 Heuristic Function สำหรับ tic-tac-toe

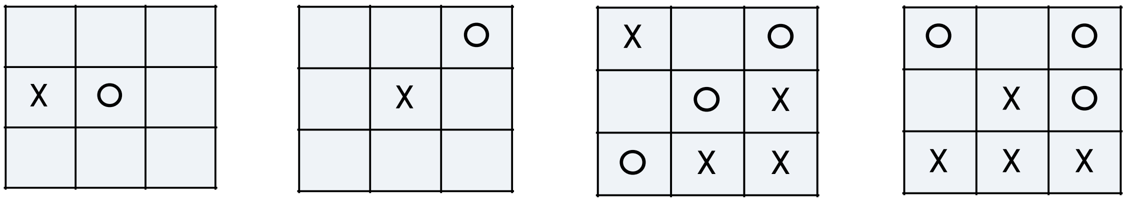
Heuristic Function ที่น่าสนใจคือ E(n) = M(n) – O(n) โดยกำหนด

- E(n) คือ ค่าประเมินสำหรับสถานะ n
- M(n) คือ จำนวนเส้นทางที่ผู้เล่นจะชนะ (กำหนดให้ผู้เล่นเป็น X)
- O(n) คือ จำนวนเส้นทางที่ฝ่ายตรงข้ามจะชนะ (กำหนดให้ฝ่ายตรงข้ามเป็น O)



ภาพที่ 4-13 การคำนวณค่า E(n) ของสถานะหนึ่งระหว่างการเล่น tic-tac-toe

จากภาพที่ 4-13 จะเห็นได้ว่าจากสถานะทางด้านซ้ายเมื่อต้องการหาค่า M(n) ซึ่งก็คือจำนวนเส้นทางทั้งหมดที่ผู้เล่น X ยังมีโอกาสชนะได้โดยไม่ถูก O กั้นพบว่ามี 6 เส้นทาง ในขณะที่ O(n) คือจำนวนเส้นทางที่ O ยังมีโอกาสชนะได้โดยไม่ถูก X กั้นพบว่ามี 5 เส้นทาง ดังนั้นค่า E(n) ของสถานะนี้ ก็คือ M(n) – O(n) = 6 – 5 = 1 (ยิ่งค่า E(n) มากยิ่งดีแสดงว่า X มีช่องทางชนะมากกว่า O)



(a) E(n) = 4 – 6 = -2

(b) E(n) = 5 – 4 = 1

(c) E(n) = -∞

(d) E(n) = ∞

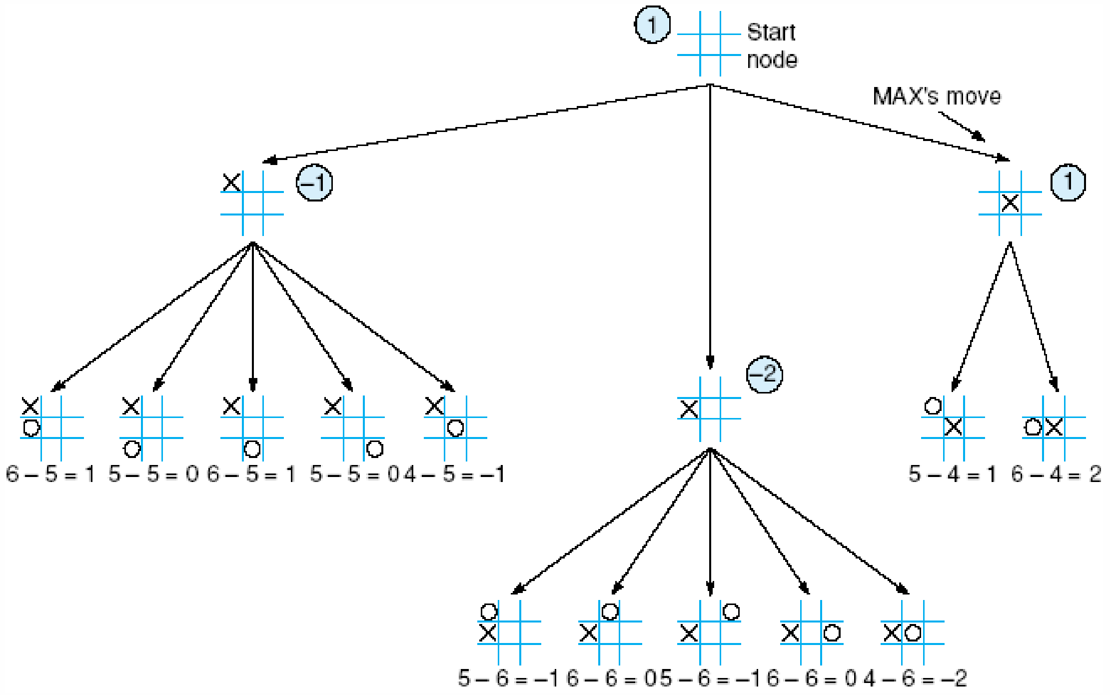
ภาพที่ 4-14 ตัวอย่างการคำนวณค่า E(n)

กรณีที่มีการแพ้ชนะเกิดขึ้นแล้วสามารถใส่ค่า E(n) ได้ตรงๆ เลยคือ

- ถ้า O ชนะจะให้ค่า E(n) = -∞
- ถ้า X ชนะจะให้ค่า E(n) = ∞

ภาพที่ 4-14 แสดงตัวอย่างการคำนวณค่า E(n) ของสถานะ 4 สถานะของเกม tic-tac-toe โดยภาพที่ 4-14(a) นั้นผู้เล่น X จะอยู่ในสถานะการณ์ที่เสียเปรียบ ภาพที่ 4-14(b) ผู้เล่น X อยู่ในสถานะการณ์ที่ได้เปรียบ ภาพที่ 4-14(c) ผู้เล่น X ได้แพ้ผู้เล่น O ไปแล้ว และภาพที่ 4-15(d) ผู้เล่น X ชนะเกม

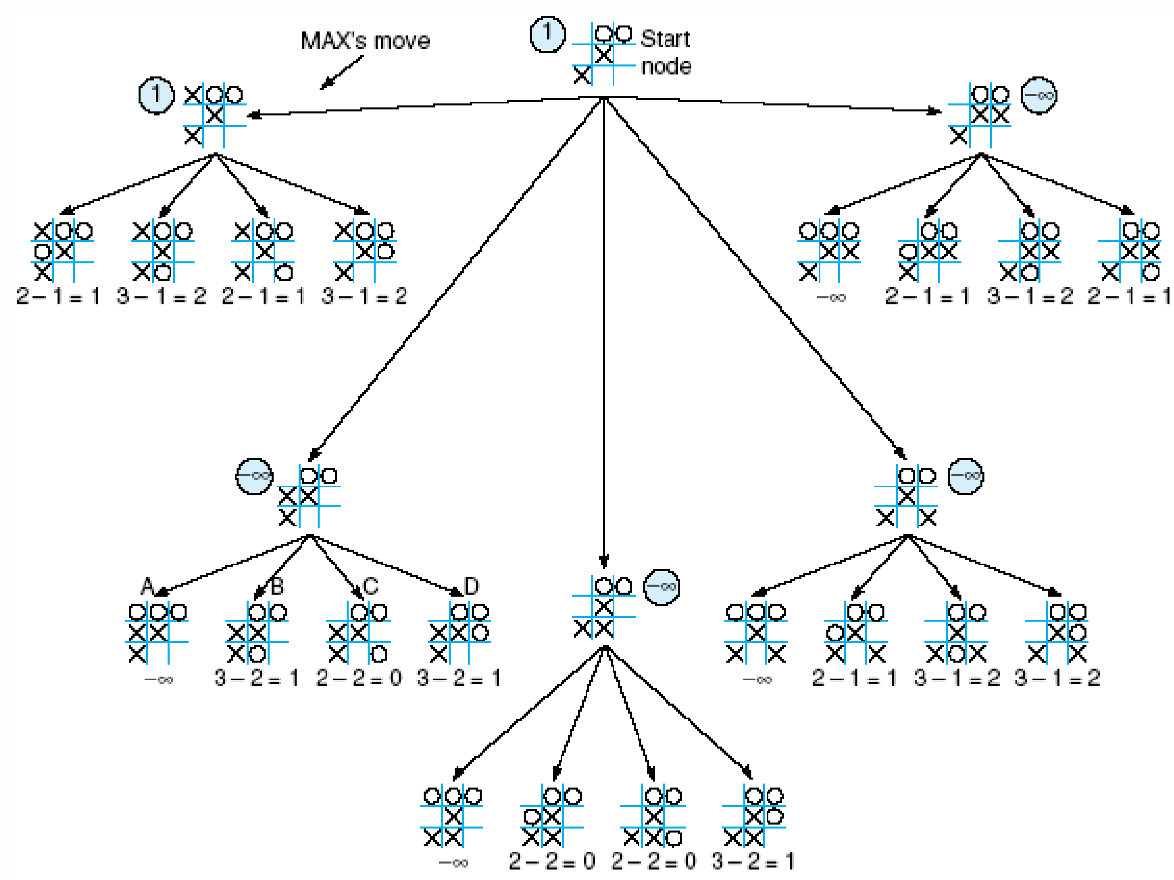
4.4.2 ตัวอย่างการค้นหาเส้นทางในเกม tic-tac-toe



ภาพที่ 4-15 การเลือกเส้นทางในเกม tic-tac-toe ที่สถานะเริ่มต้นด้วยการใช้ Heuristic Function

ภาพที่ 4-15 แสดงการคำนวณค่า Utility Function ด้วยการใช้ค่าประเมินที่คำนวณจาก Heuristic Function โดยกำหนดการลงค้นหาเพียงแค่ 3 ชั้นเท่านั้น สำหรับเทคนิคที่ใช้ในการค้นหาเพื่อให้เข้าใจได้ง่ายมากยิ่งขึ้น จึงขอใช้เทคนิคที่ใช้ในการค้นหาใช้แบบ Minimax โดยจะสังเกตได้ว่าเมื่อคำนวณค่า MinimaxValue ของโหนด MIN และ MAX แล้ว ค่า MinimaxValue ของโหนดรากคือค่า 1 ที่ได้จาก การเดินทางไปยังสถานะทางด้านขวาสุด ผู้เล่นจึงควรตัดสินใจเลือกไปทางขวา (MAX's move)

ในแต่ละครั้งที่ X ต้องการจะเดินก็จะมาสร้างต้นไม้ใหม่อีกครั้งที่สถานะปัจจุบันเป็นราก และสร้างสถานะลูกลึกลงไปแค่ 3 ชั้นดังที่ทำในขั้นตอนแรก และจะวนทำเช่นนี้ไปเรื่อยๆ จนกว่าจะจบเกม แสดงดังภาพที่ 4-16



ภาพที่ 4-16 การเลือกเส้นทางในเกม tic-tac-toe ที่สถานะสุดท้ายด้วยการใช้ Heuristic Function

จะเห็นได้ว่าเมื่อมีการใช้ Heuristic Function ควบคู่ไปกับการจำกัดระดับชั้นใน การค้นหาจะทำให้วิธีการค้นหาแบบมีคู่ปรปักษ์นั้นรวดเร็วมากขึ้นในขณะที่ให้ทรัพยากรน้อยลง อย่างไร ก็ตาม การที่จะให้ได้ผลลัพธ์ที่ดีขึ้นนั้นจะขึ้นอยู่กับองค์ประกอบ 2 อย่าง คือ Heuristic Function ที่มีประสิทธิภาพ และจำนวนชั้นที่ใช้ในการค้นหา (ถ้าจำนวนชั้นเยอะผลลัพธ์ที่ได้จะดีขึ้น แต่จะสูญเสียความเร็ว และใช้ทรัพยากรมากขึ้น)

4.5 สรุป

เทคนิคการค้นหาเมื่อมีคู่ปรปักษ์ (Adversarial Search) เป็นเทคนิคการค้นหาทางเลือกที่เหมาะสม สำหรับปัญหาของเกมที่มีการแข่งขันกันระหว่างผู้เล่น 2 คนที่ต้องมีการคำนึงถึงการเล่นของฝ่ายตรงข้าม โดยเกมที่นำมาพิจารณานั้นผู้เล่นจะต้องทราบข้อมูลภายในเกมขณะนั้นทั้งหมด เช่น เกมหมากรุก และเกม tic-tac-toe เป็นต้น อัลกอริทึมพื้นฐานที่ใช้สำหรับเทคนิคนี้มีด้วยกัน 2 อัลกอริทึมคือ Minimax และ Alpha-Beta Prunning

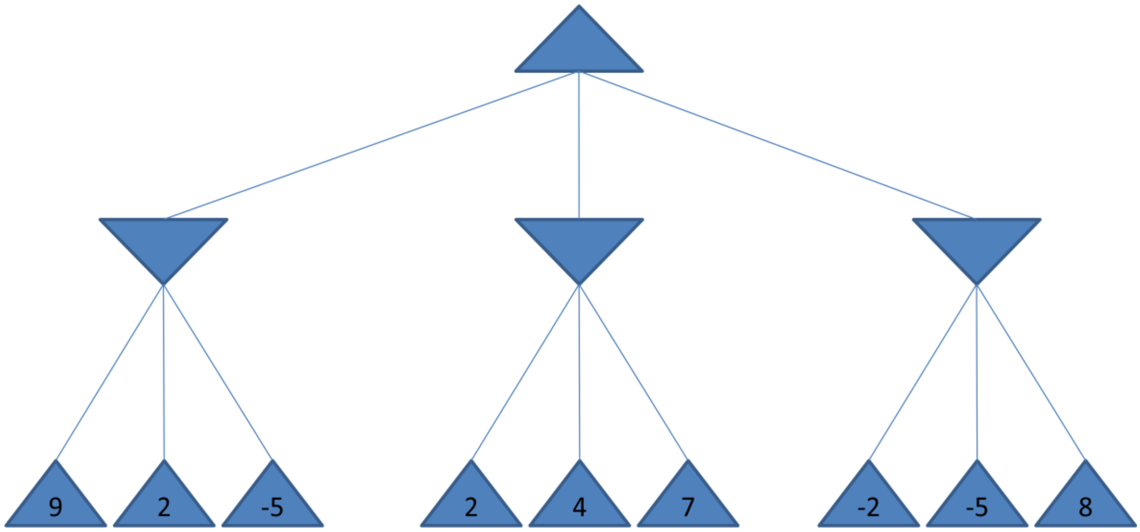
อัลกอริทึม Minimax เป็นอัลกอรีทึมที่ง่ายที่สุดคือ ภายในต้นไม้ Minimax จะประกอบด้วยโหนด MIN และ โหนด MAX สลับกันคนละชั้น เริ่มจากชั้นที่เป็นราก (ชั้นของผู้เล่น) คือโหนด MAX หมายถึงผู้เล่นพยายามที่จะเลือกทางที่ดีที่สุดสำหรับตนเอง ในขณะที่สถานะที่ฝ่ายตรงข้ามเล่นจะเป็นชั้นของโหนด MIN หมายถึงฝ่ายตรงข้ามพยายามจะทำให้ผู้เล่นได้คะแนนน้อยที่สุด

อัลกอริทึม Alpha-Beta Prunning เป็นการนำอัลกอริทึมแบบ Minimax มาปรับปรุงให้ดีขึ้น โดยสามารถตัดกิ่งของต้นไม้บางกิ่งที่รู้ว่าจะไม่สามารถทำให้ผลลัพธ์เปลี่ยนแปลงได้ออกไปจากการคำนวณ ซึ่งจะทำให้การค้นหาแบบนี้ใช้ทรัพยากรที่น้อยกว่าแบบ Minimax

อัลกอริทึมทั้ง 2 แบบถ้ามีการใช้งานตรงๆ กับปัญหาที่มีความซับซ้อนจะไม่เหมาะสม เนื่องจากจะใช้ทรัพยากรที่มาก และเวลาในการค้นหาที่นาน จึงควรมีการใช้งานร่วมกับ Heuristic Function และกำจัดระดับชั้นในการค้นหา เพื่อให้ใช้ทรัพยากรน้อยลงและมีความเร็วในการค้นหามากขึ้น

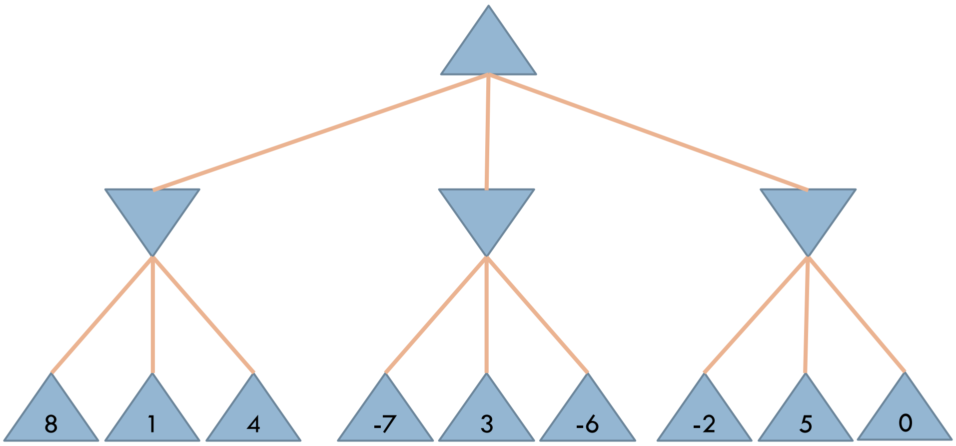
แบบฝึกหัด

1. จงเติมค่า MinimaxValue ของต้นไม้ Minimax ให้สมบูรณ์พร้อมทั้งระบุเส้นทางที่ควรเลือกเดิน

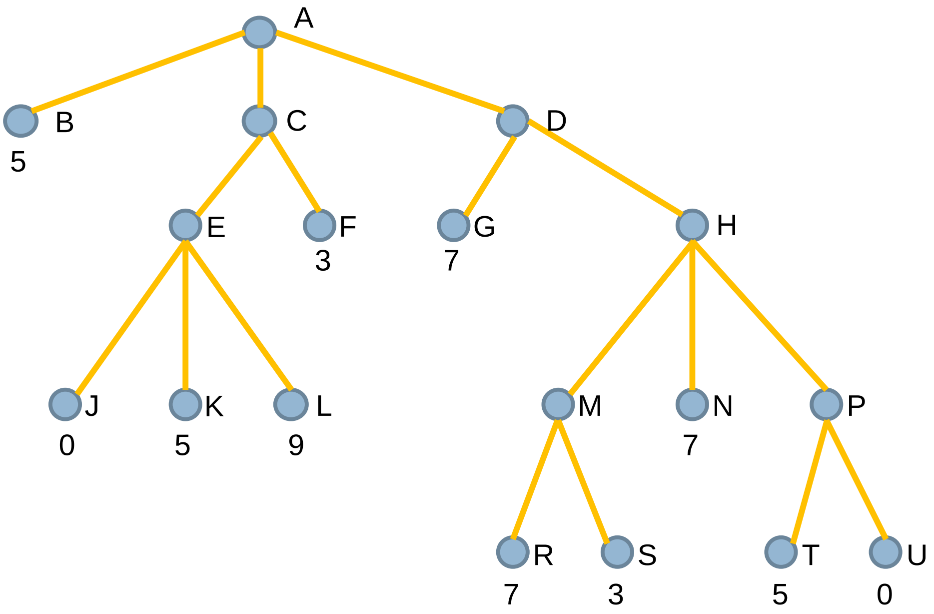


2. กำหนดให้สถานะเริ่มต้นมีเหรียญอยู่ 5 เหรียญ มีผู้เล่น 2 คน ผลัดกันหยิบเหรียญ แต่ละตาผู้เล่นจะต้องหยิบ 1 เหรียญ หรือ 2 เหรียญ คนที่หยิบเหรียญสุดท้ายคนนั้นแพ้ ให้ผู้เล่นเป็นคนเริ่มต้นหยิบเหรียญคนแรก กำหนด Utility function มีค่า -1 ถ้าผู้เล่นแพ้ และมีค่า 1 ถ้าผู้เล่นชนะ จงเขียนปริภูมิสถานะ, หาค่า minimax, และเลือกทางเดินที่จะชนะ

3. จงแสดงเส้นทางการค้นหาด้วยวิธี Alpha-Beta Prunning



4. จงเขียนต้นไม้ Minimax พร้อมทั้งแสดงวิธีค้นหาแบบ Alpha-Beta Prunning จากซ้ายไปขวาและจากขวาไปซ้าย



แผนปฏิบัติการสอนสัปดาห์ที่ 5

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง เทคนิคการแก้ปัญหาแบบมีเงื่อนไข **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. องค์ประกอบของปัญหา CSP	IS 5.1
2. ชนิดของตัวแปรและเงื่อนไขในปัญหา CSP	IS 5.2
3. กราฟ Constraint	IS 5.3
4. เทคนิคการค้นหาสำหรับปัญหา CSP	IS 5.4
5. การปรับปรุงประสิทธิภาพในการค้นหา	IS 5.5

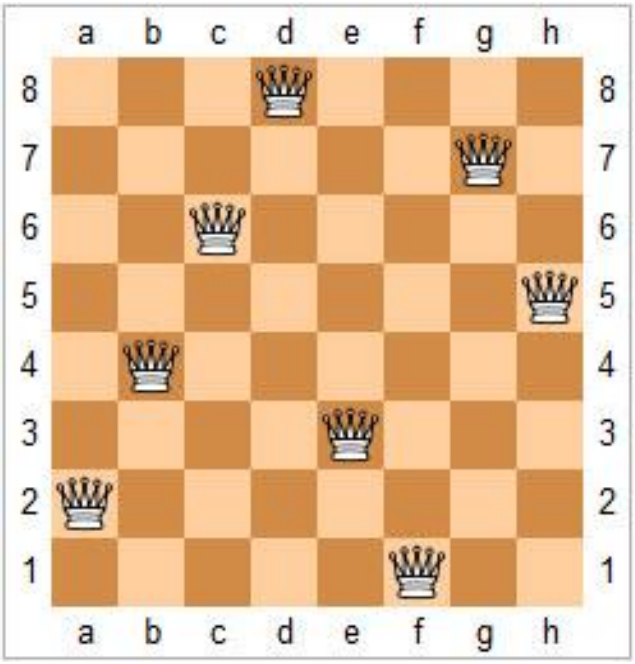
ข. การจัดการเรียนการสอน

เวลา – นาที		**0 — 60 — 120 — 180**
วัตถุประสงค์		**1-2-3-4-5**
การนำเข้าสู่บทเรียน		
ให้เนื้อหา		
สรุปเนื้อหา		
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**
วิธีการสอน:	บรรยาย	
	อธิบาย	
	ถาม – ตอบ	
	ปฏิบัติด้วยตนเอง	
สื่อการสอน:	กระดานดำ	
	คอมพิวเตอร์	
	โปรเจคเตอร์	

บทที่ 5
เทคนิคการแก้ปัญหาแบบมีเงื่อนไข

เทคนิคการแก้ปัญหาแบบมีเงื่อนไข (Constraint Satisfaction Problems: CSP) คือ การกำหนดเงื่อนไขเพื่อช่วยในการแก้ปัญหา โดยการกำหนดตัวแปร (Variables) โดเมน (Domains) และข้อจำกัด (Constraint) ตัวอย่างของปัญหาที่สามารถแก้ได้ด้วย CSP เช่น Sudoku, 8 Queens Puzzle, Einstein's Puzzle ส่วนใหญ่จะเป็นปัญหาที่มีขนาดไม่ใหญ่มากประมาณว่าสั่งแล้วคำตอบต้องออกมาภายในไม่กี่วินาที คือ เอาตัวเลขทั้งหมดมาบวก ลบ คูณ หาร กันให้ครบทุกรูปแบบแล้วค่อยตรวจดูว่าคำตอบถูกหรือไม่ ซึ่งเป็นการศึกษาเกี่ยวกับการแก้ปัญหารูปแบบหนึ่งที่เราต้องการหาคำตอบ และคำตอบที่ได้จะต้องเป็นคำตอบที่อยู่ภายใต้เงื่อนไขที่กำหนดเอาไว้ เช่น จะทำอย่างไรให้เติมสีลงไปในแต่ละส่วนของรูปภาพ โดยที่แต่ละส่วนที่อยู่ติดกันจะต้องไม่มีสีเหมือนกัน ดังแสดงในภาพที่ 5-1

ภาพที่ 5-1 การเติมสี 4 สีในช่องต่างโดยช่องติดกันห้ามมีสีเดียวกัน (ที่มาจาก : http://en.wikipedia.org)



ภาพที่ 5-2 เป็นภาพของ Queen 8*8 (ที่มาจาก : http://en.wikipedia.org)

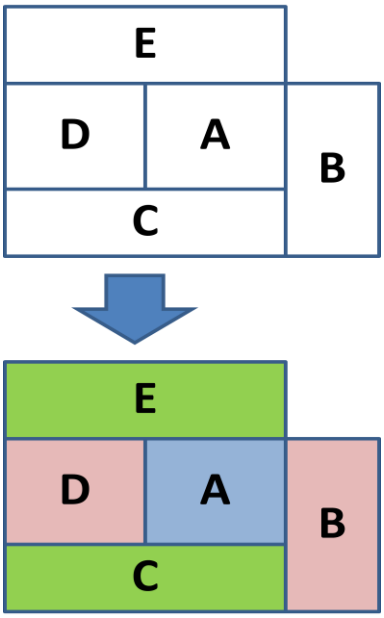
หรือปัญหาการวาง Queen ของหมากรุกแสดงดังภาพที่ 5-2 ที่สามารถเดินได้ 8 ทิศทางและเดินในแต่ละทิศทางไปกี่ช่องก็ได้ จำนวน 8 ตัวลงไปในตาราง 8x8 อย่างไรโดยที่ไม่กินกันเอง

5.1 องค์ประกอบของปัญหา Constraint Satisfaction Problem (CSP)

การแก้ไขปัญหาประเภท CSP จำเป็นต้องมีข้อมูลเบื้องต้น หรือองค์ประกอบที่สำคัญทั้งหมด 4 ส่วนคือ

- ชุดของตัวแปร (Variables) เช่น X_1 , X_2 , … , X_n โดยที่แต่ละตัวแปร X_i มีค่าที่เป็นไปได้อยู่ในโดเมน D_i
- โดเมน (Domain) D_i เป็นขอบเขตของค่าที่เป็นไปได้สำหรับตัวแปร
- ชุดของเงื่อนไข (Constraints) เช่น C_1 , C_2 , … , C_m เงื่อนไขจะเป็น Subset บาง Subset ของตัวแปร และในเงื่อนไขจะระบุค่าที่อนุญาตให้ตัวแปรที่อยู่ใน Subset นั้นเป็นไปได้
- คำตอบ (Solution) จะมีได้ต่อเมื่อมีการกำหนดค่าให้กับตัวแปรทุกตัวและค่าของตัวแปรแต่ละตัวไม่ขัดกับ Constraint ที่กำหนด

5.1.1 ตัวอย่างปัญหาการทำ Map Coloring

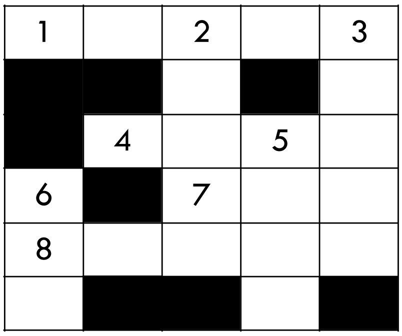


ภาพที่ 5-3 ตัวอย่างปัญหา Map Coloring

ปัญหาประเภทนี้คือมีลักษณะของช่องให้ลงสี จากตัวอย่างในภาพที่ 5-3 กำหนดให้มีสีทั้งหมด 3 สีคือ สีแดง (Red: R), สีเขียน (Green: G) และสีน้ำเงิน (Blue: B) และมีข้อกำหนดคือ ห้ามช่องที่ติดกันมีการลงสีเดียวกัน เมื่อทราบถึงขอบเขตและข้อกำหนดต่างๆ แล้วจะสามารถนำมาเขียนองค์ประกอบของปัญหา CSP ได้ดังนี้

- ตัวแปรมี 5 ตัวแปรที่เป็นช่องที่ต้องการลงสี คือ A, B, C, D และ E
- โดเมนค่าของตัวแปรคือ 3 สีที่จะลงในช่องได้ คือ R, G และ B
- ข้อกำหนด (Constraints)
    - ค่าของสีในช่อง A ต้องไม่เท่ากับสีในช่อง B: A != B
    - ค่าของสีในช่อง A ต้องไม่เท่ากับสีในช่อง C: A != C
    - ค่าของสีในช่อง A ต้องไม่เท่ากับสีในช่อง D: A != D
    - ค่าของสีในช่อง A ต้องไม่เท่ากับสีในช่อง E: A != E
    - ค่าของสีในช่อง B ต้องไม่เท่ากับสีในช่อง C: B != C
    - ค่าของสีในช่อง C ต้องไม่เท่ากับสีในช่อง D: C != D
    - ค่าของสีในช่อง D ต้องไม่เท่ากับสีในช่อง E: D != E

5.1.2 ตัวอย่างปัญหาเกม Crossword Puzzle



AFT	HOSES	LINE
ALE	KEEL	SAILS
EEL	KNOT	SHEET
HEEL	LASER	STEER
HIKE	LEE	TIE

ภาพที่ 5-4 ตัวอย่างปัญหาเกม Crossword Puzzle

ปัญหาประเภทนี้คือนำคำศัพท์มาใส่ในช่องว่างทางแนวตั้งหรือแนวนอนที่มีจำนวนช่องยาวเท่ากับความยาวของคำศัพท์ และช่องว่างที่เป็นจุดเชื่อมระหว่าง 2 คำศัพท์จะต้องเป็นตัวอักษรเดียวกัน สามารถนำมาเขียนองค์ประกอบของปัญหา CSP ได้ดังนี้

- ตัวแปรมี 8 ตัว (8 คำ ที่ต้องเติมคำศัพท์) คือ 1, 2, 3, 4, 5, 6, 7 และ 8
- โดเมนคือคำศัพท์ที่จะกำหนดค่าให้กับตัวแปร มีทั้งหมด 15 คำศัพท์ คือ
  AFT, HOSES, LINE, ALE, KEEL, SAILS, EEL, KNOT, SHEET, HEEL, LASER, STERR, HIKE, LEE, และ TIE
- ข้อกำหนด (Constraints) :
  - อักษรตัวที่ 3 ของตัวแปร 1 ต้องเท่ากับอักษรตัวที่ 1 ของตัวแปร 2: 1[3] = 2[1]
  - อักษรตัวที่ 5 ของตัวแปร 1 ต้องเท่ากับอักษรตัวที่ 1 ของตัวแปร 3: 1[5] = 3[1]
  - อักษรตัวที่ 2 ของตัวแปร 4 ต้องเท่ากับอักษรตัวที่ 3 ของตัวแปร 2: 4[2] = 2[3]
  - อักษรตัวที่ 3 ของตัวแปร 4 ต้องเท่ากับอักษรตัวที่ 1 ของตัวแปร 5: 4[3] = 5[1]
  - อักษรตัวที่ 4 ของตัวแปร 4 ต้องเท่ากับอักษรตัวที่ 3 ของตัวแปร 3: 4[4] = 3[3]
  - อักษรตัวที่ 1 ของตัวแปร 7 ต้องเท่ากับอักษรตัวที่ 4 ของตัวแปร 2: 7[1] = 2[4]
  - อักษรตัวที่ 2 ของตัวแปร 7 ต้องเท่ากับอักษรตัวที่ 2 ของตัวแปร 5: 7[2] = 5[2]
  - อักษรตัวที่ 3 ของตัวแปร 7 ต้องเท่ากับอักษรตัวที่ 4 ของตัวแปร 3: 7[3] = 3[4]
  - อักษรตัวที่ 1 ของตัวแปร 8 ต้องเท่ากับอักษรตัวที่ 2 ของตัวแปร 6: 8[1] = 6[2]
  - อักษรตัวที่ 3 ของตัวแปร 8 ต้องเท่ากับอักษรตัวที่ 5 ของตัวแปร 2: 8[3] = 2[5]
  - อักษรตัวที่ 4 ของตัวแปร 8 ต้องเท่ากับอักษรตัวที่ 3 ของตัวแปร 5: 8[4] = 5[3]
  - อักษรตัวที่ 5 ของตัวแปร 8 ต้องเท่ากับอักษรตัวที่ 5 ของตัวแปร 3: 8[5] = 3[5]

5.1.3 ตัวอย่างปัญหาเกม Sudoku

			1		4			
1	9		6	7				5
	6		5	2			1	
	2				6		8	
		5					6	
	4	3		8		7		
		2			5		4	
		9	3	4	2		7	
			8				9	

2	5	7	1	9	4	6	3	8
1	9	8	6	7	3	4	2	5
3	6	4	5	2	8	9	1	7
7	2	1	4	5	6	3	8	9
9	8	5	7	3	1	2	6	4
6	4	3	2	8	9	7	5	1
8	7	2	9	6	5	1	4	3
5	1	9	3	4	2	8	7	6
4	3	6	8	1	7	5	9	2

ภาพที่5-5 ตัวอย่างปัญหาเกม Sudoku

Sudoku เป็นเกมที่ต้องใส่ตัวเลข 1 – 9 ให้เต็มตารางโดยมีข้อกำหนดอยู่ว่าในแต่ละหลัก และแต่ละแถวห้ามมีตัวเลขซ้ำกัน รวมถึงภายในกล่อง 3x3 ช่องยังห้ามมีตัวเลขซ้ำกันอีกด้วย จากปัญหาดังกล่าวสามารถนำมาเขียนองค์ประกอบของปัญหา CSP ได้ดังนี้

- ตัวแปรมีทั้งหมด 81 ตัว
  - $X_{11}, X_{12}, ..., X_{19}$ สำหรับช่องว่างแถวที่ 1 หลักที่ 1 จนถึง หลักที่ 9
  - $X_{21}, X_{22}, .., X_{29}$ สำหรับช่องว่างแถวที่ 2 หลักที่ 1 จนถึง หลักที่ 9
  - ... สำหรับช่องว่างแถวที่ 3, 4, 5, 6, 7, 8 หลักที่ 1 จนถึง หลักที่ 9
  - $X_{91}, X_{92}, ..., X_{99}$ สำหรับช่องว่างแถวที่ 9 หลักที่ 1 จนถึง หลักที่ 9
- โดเมนคือค่าที่สามารถกำหนดค่าลงตัวแปรได้ ก็คือค่า 1, 2, 3, 4, 5, 6, 7, 8, และ 9
- ข้อกำหนด (Constraints) ประกอบด้วย 3 ส่วนหลักๆ คือ
  - ข้อกำหนดในแต่ละแถว: ค่าของหลักที่ 1 ถึง 9 ของแต่ละแถวต้องไม่เท่ากัน
    - ตั้งแต่แถวที่ 1: $X_{11} != X_{12} != X_{13} != X_{14} != X_{15} != X_{16} != X_{17} != X_{18} != X_{19}$
    - จนถึงแถวที่ 9: $X_{91} != X_{92} != X_{93} != X_{94} != X_{95} != X_{96} != X_{97} != X_{98} != X_{99}$
  - ข้อกำหนดในแต่ละหลัก: ค่าของแถวที่ 1 ถึง 9 ของแต่ละหลักต้องไม่เท่ากัน
    - ตั้งแต่หลักที่ 1: $X_{11} != X_{21} != X_{31} != X_{41} != X_{51} != X_{61} != X_{71} != X_{81} != X_{91}$
    - จนถึงหลักที่ 9: $X_{19} != X_{29} != X_{39} != X_{49} != X_{59} != X_{69} != X_{79} != X_{89} != X_{99}$
  - ข้อกำหนดในแต่ละกล่อง 3x3 ที่ค่าภายในต้องไม่เท่ากัน
    - ตั้งแต่กล่องบนซ้าย: $X_{11} != X_{12} != X_{13} != X_{21} != X_{22} != X_{23} != X_{31} != X_{32} != X_{33}$
    - จนถึงกล่องล่างขวา: $X_{77} != X_{78} != X_{79} != X_{87} != X_{88} != X_{89} != X_{97} != X_{98} != X_{99}$

5.2 ชนิดของตัวแปรและเงื่อนไขในปัญหา CSP

คำตอบของปัญหา CSP จะได้มาจากการกำหนดค่าให้กับแต่ละตัวแปร โดยที่ไม่ขัดกับเงื่อนไข ซึ่งสิ่งที่เราต้องการอาจเป็นคำตอบแค่ 1 คำตอบที่ไม่ขัดกับเงื่อนไข คำตอบทั้งหมดที่เป็นไปได้หรือคำตอบที่ดีที่สุด (ในการหาคำตอบที่ดีทีสุดนี้เราจะเรียกว่า Constraint Optimization Problem)

5.2.1 ชนิดของตัวแปรใน CSP

ตัวแปรในปัญหา CSP นั้นสามารถเก็บค่าได้หลายประเภท ซึ่งปัญหาส่วนใหญ่จะมีค่าที่ต้องการจัดเก็บในตัวแปรแบ่งออกเป็น 3 ประเภทดังนี้

- ชนิดตัวแปรแบบ Discrete และมีขอบเขตของโดเมนที่แน่นอน เป็นรูปแบบที่ง่ายที่สุด
- ชนิดตัวแปรแบบ Discrete และไม่มีขอบเขตของโดเมน เช่น โดเมนเป็นค่าของจำนวนเต็ม
- ชนิดของตัวแปรเป็นแบบ Continuous เช่น จำนวนจริง (ในการแก้ปัญหาตัวแปรในลักษณะนี้ ส่วนใหญ่จะใช้เทคนิคจำพวก Linear Programming ในการแก้ปัญหา)

5.2.2 ชนิดของเงื่อนไขใน CSP

เงื่อนไขในปัญหา CSP นั้นมีหลายรูปแบบแต่ส่วนใหญ่จะแบ่งออกเป็น 4 ประเภทดังนี้

- Unary Constraints จะมีตัวแปรในแต่ละเงื่อนไขเพียงตัวแปรเดียว เช่น A ≠ ‘Green’
- Binary Constraints จะมีตัวแปรในแต่ละเงื่อนไขสองตัวแปร เช่น A ≠ B
- Higher-order Constraints จะมีตัวแปรในแต่ละเงื่อนไขตั้งแต่สามตัวแปรขึ้นไป (ในกรณีนี้มักจะเปลี่ยนเงื่อนไขให้เป็น Binary Constraints เพื่อให้ง่ายขึ้น)
- Preference (Soft Constraints) : เป็นเงื่อนไขที่เพิ่มขึ้นมาจากเงื่อนไขปกติ เช่น สีแดงดีกว่าสีเขียว หรือการทำให้ในภาพมีพื้นที่ที่เป็นสีแดงเยอะจะดีกว่า (จำนวนของพื้นที่สีแดงจะต้องเยอะที่สุด) โดยเงื่อนไขอื่นยังคงเดิม คือ สีเดียวกันห้ามอยู่ติดกัน เงื่อนไขแบบนี้จะเป็นเงื่อนไขในการแก้ปัญหาแบบ Constrained Optimization Problems

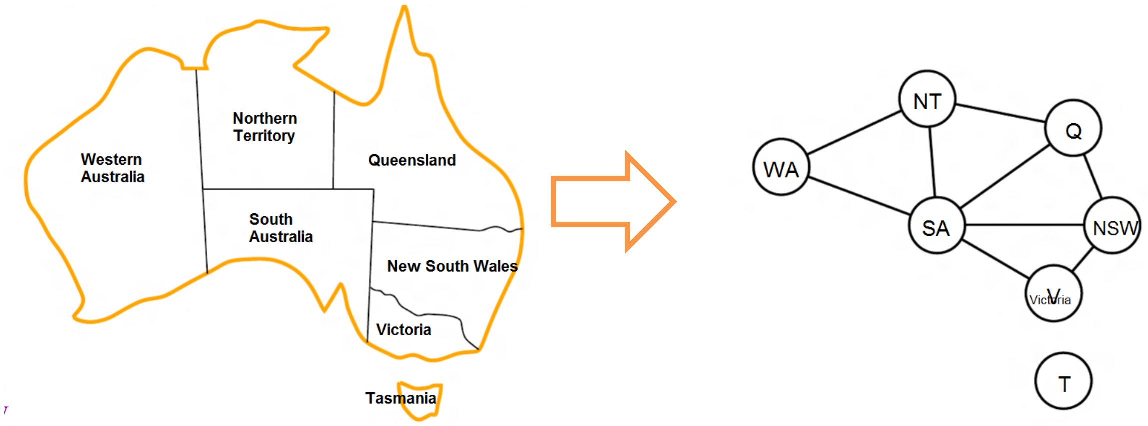
ปัญหาของ Constraint Satisfaction Problems นั้นเราสามารถแทนอยู่ในรูปของโครงสร้างข้อมูลกราฟแบบไม่มีทิศทาง (Undirected Graph) ที่เรียกว่า Constraint Graph ได้ เพื่อให้ง่ายในการออกแบบและพัฒนา โดยที่โหนดของกราฟ คือ ตัวแปร และเส้นเชื่อม (Edge) คือ Constraint โดยปกติ Constraint Graph จะแสดงเงื่อนไขแบบ Unary และ Binary Constraint เท่านั้น แต่ถ้าเป็น Higher-order Constraints จะต้องแปลงให้อยู่ในรูปแบบของ Binary Constraint ก่อน

5.3 กราฟ Constraint

เพื่อให้ปัญหา CSP ที่ต้องการแก้ไขนั้น สามารถมองข้อกำหนด (Constraint) ต่างๆ ได้ง่ายขึ้น จึงมีการแทนตัวปัญหาให้อยู่ในรูปของกราฟแบบไม่มีทิศทาง ที่เรียกว่า Constraint Graph โดยภายในกราฟจะประกอบด้วย 2 ส่วนคือ

- โหนด (node) แทนตัวแปร
- เส้นเชื่อม (edge) แทนข้อจำกัดหรือเงื่อนไข (binary constraint)

ตัวอย่างการทำ Constraint Graph ของปัญหา Map Coloring



ภาพที่5-6 การทำ Constraint Graph ของตัวอย่างปัญหา Map Coloring

จากตัวอย่างปัญหา Map Coloring สามารถแปลงปัญหาให้อยู่ในรูปของ Constraint Graph ได้ดังแสดงในภาพที่ 5-6 โดยที่แต่ช่องที่จะลงสี จะแทนด้วยโหนด เช่น Western Australia จะกลายเป็นโหนด WA จากนั้นลากเส้นเชื่อมระหว่างโหนดเพื่อแสดงความสัมพันธ์หรือข้อกำหนด โดยจากภาพจะเห็นได้ว่า จะมี Binary Constraint อยู่ 9 เงื่อนไขคือ

- WA ! = NT
- WA ! = SA
- NT ! = Q
- NT ! = SA
- Q ! = SA
- Q ! = NSW
- SA ! = NSW
- SA ! = V
- NSW ! = V

5.4 เทคนิคการค้นหาสำหรับปัญหา CSP

เทคนิคที่ใช้ในการแก้ปัญหา CSP มีหลายเทคนิค อย่างไรก็ตามเทคนิคพื้นฐานมีทั้งหมด 3เทคนิค

1. Systematic Search Algorithm
2. Consistency Techniques
3. Forward Checking

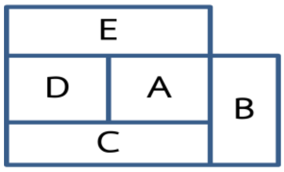
5.4.1 Systematic Search Algorithm

เป็นเทคนิคการค้นหาที่มีรูปแบบตายตัว คือ ค้นหาทุกคำตอบที่เป็นไปได้ วิธีนี้จะรับประกันนี้ว่าจะเจอคำตอบแน่นอนหากคำตอบนั้นมีอยู่ หรือใช้ในการพิสูจน์ว่าปัญหานี้ไม่มีคำตอบ แต่ปัญหาหลักคือวิธีนี้จะใช้เวลานานในการค้นหาคำตอบ ตัวอย่าง Systematic Search Algorithm เช่น Generate and Test (GT) และ Backtracking (BT)

5.4.1.1 Generate and Test (GT)

เป็นเทคนิคแบบ Brute Force คือ สุ่มหรือวนสร้างผลลัพธ์ขึ้นมา จากนั้นเอามาตรวจสอบกับข้อกำหนดที่ได้ตั้งไว้

ตัวอย่างปัญหา Map Coloring



ภาพที่ 5-7 ตัวอย่างปัญหา Map Coloring

กรณีปัญหาจากภาพที่ 5-7 เริ่มต้นจากการนำข้อในโดเมนมาใส่เป็นคำตอบให้กับตัวแปรแต่ละตัว จากนั้นนำค่าในตัวแปรที่ได้มาตรวจสอบความถูกต้องว่าขัดกับข้อกำหนดหรือไม่ โดยเริ่มจากกำหนดให้ตัวแปรมีค่าเป็น “R” ทั้งหมด เมื่อผิดข้อกำหนดก็ไล่เป็นเป็น “G” และ “B” ตามลำดับ ดังขั้นตอนด้านล่าง

A = ‘R’, B = ‘R’, C = ‘R’, D = ‘R’, E = ‘R’ ⇒ ผิดข้อกำหนด A != B

A = ‘R’, B = ‘R’, C = ‘R’, D = ‘R’, E = ‘G’ ⇒ ผิดข้อกำหนด A != B

A = ‘R’, B = ‘R’, C = ‘R’, D = ‘R’, E = ‘B’ ⇒ ผิดข้อกำหนด A != B

A = ‘R’, B = ‘R’, C = ‘R’, D = ‘G’, E = ‘R’ ⇒ ผิดข้อกำหนด A != B

……

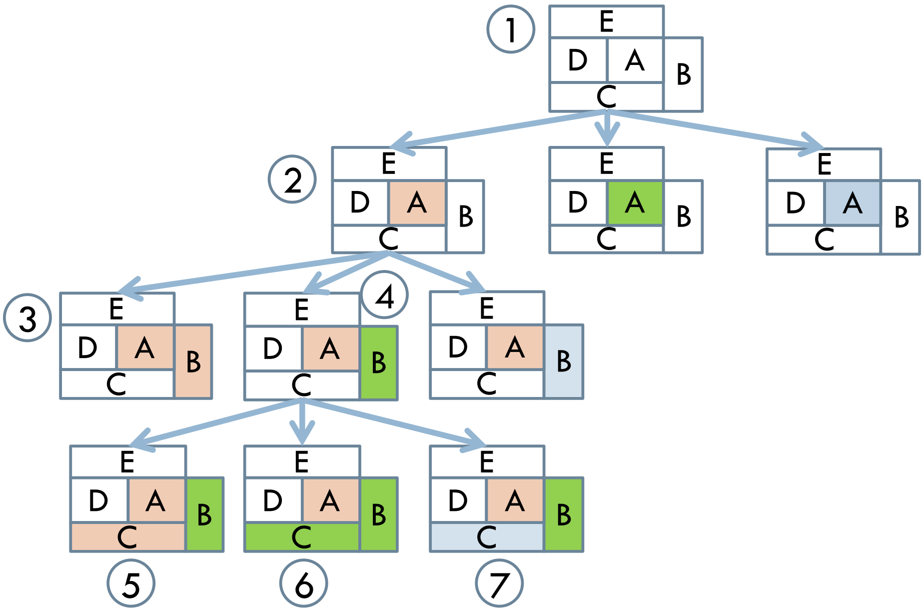
ข้อเสีย วิธีสร้างและทดสอบไม่ได้มีประสิทธิภาพเพราะการสุ่มหรือวนค่าเพื่อสร้างผลลัพธ์แล้วนำมาตรวจสอบกับข้อกำหนดทำให้ใช้เวลาในการค้นหานานมาก

5.4.1.2 Backtracking (BT)

เป็นเทคนิคที่ใช้การแก้ปัญหาแบบปริภูมิสถานะมาแก้ปัญหา CSP โดยส่วนใหญ่แล้วจะใช้การค้นหาแบบแนวลึกก่อน โดยให้

- สถานะเริ่มต้น คือ ทุกตัวแปรไม่มีการกำหนดค่า
- กฎในการเป็นสถานะ คือ กำหนดค่าที่อยู่ใน Domain ลงไปให้แต่ละตัวแปร
- สถานะเป้าหมาย คือ ทุกตัวแปรถูกกำหนดค่าโดยไม่ขัดกับเงื่อนไข
- ตรวจสอบข้อกำหนดของแต่ละตัวแปรก่อนกำหนดค่า จากนั้นลงไปหาค่าสำหรับตัวแปรถัดไป ถ้าพบทางตันจะย้อนกลับมาใส่ค่าใหม่

ตัวอย่างปัญหา Map Coloring



ภาพที่ 5-8 ตัวอย่างการค้นหาเส้นทางแบบ Backtracking เพื่อแก้ปัญหา Map Coloring

ภาพที่ 5-8 แสดงตัวอย่างการค้นหาเส้นทางแบบ Backtracking โดยมีขั้นตอนแรกคือหมายเลข (1) ตัวแปรต่างๆ ยังไม่มีการกำหนด จากนั้นในขั้นตอนที่ 2 เริ่มกำหนดค่าให้กับตัวแปร A เป็นค่า “R” พร้อมตรวจสอบความถูกต้องของข้อกำหนด และในขั้นตอนที่ 3 ก็ทำการกำหนดค่า “R” ให้กับตัวแปร B เมื่อตรวจสอบข้อกำหนดจะพบว่าขัดกับข้อกำหนดที่ว่า สีของ A ต้องไม่เท่ากับสีของ B ดังนั้นจึงหยุดการค้นหาสำหรับกิ่งนี้แล้วย้อนกลับไปค้นหาในขั้นตอนที่ 4 คือเปลี่ยนค่าของ B เป็น “G” จากนั้นก็ตรวจสอบความถูกต้องเมื่อไม่ขัดกับข้อกำหนดก็สำรวจลงลึกต่อไปในขั้นตอนที่ 5 คือกำหนดค่าของตัวแปร C เป็น “R” เมื่อตรวจสอบแล้วพบว่าขัดกับข้อกำหนดก็ย้อนกลับมากำหนดค่า C ใหม่เป็น “B” ในขั้นตอนที่ 6 ซึ่งก็ยังขัดกับข้อกำหนดอยู่ จึงย้อนกลับมาเปลี่ยนค่า C ใหม่ให้เป็น “B” ตามขั้นตอนที่ 7 และสำรวจลงลึกแบบนี้ไปเรื่อยๆ จนกว่าจะได้คำตอบที่ไม่ขัดกับข้อกำหนดทุกข้อ

ข้อเสียของ Backtracking

ในการแก้ปัญหาจะ ทดลองค่าที่ผิดซ้ำแล้วซ้ำเล่า โดยไม่พิจารณาถึงเหตุผลว่าทำไมถึงผิดเงื่อนไข และไม่สามารถตรวจจับความขัดแย้งกับเงื่อนไขได้ก่อนความขัดแย้งนั้นเกิดขึ้น ทำให้เสียเวลาไปในเส้นทางที่ไม่ถูกต้อง

5.4.2 Consistency Techniques

เป็นเทคนิคที่คิดขึ้นมาเพื่อแก้ไขปัญหาการตรวจสอบความขัดแย้งกับเงื่อนไขได้ช้าของ Backtracking โดยการตรวจจับความขัดแย้งและตัดโดเมนของตัวแปรที่ทำให้ขัดแย้งออกก่อนที่จะหาคำตอบ ทำให้ปริภูมิสถานะมีขนาดเล็กลง เช่น

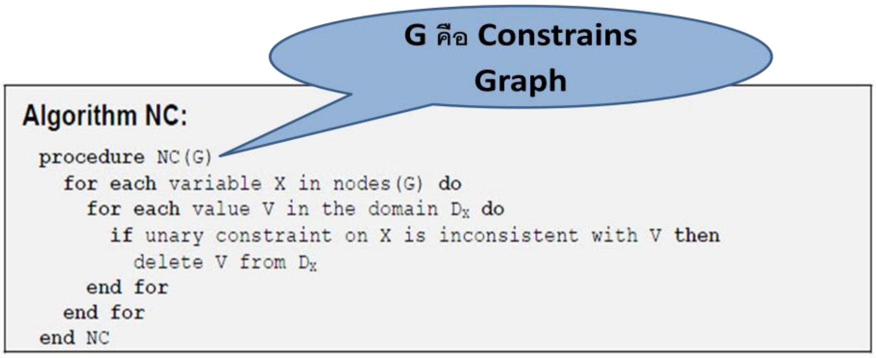
- โดเมนของตัวแปร A คือ D_A = { 3 , 4 , 5 , 6 , 7 }
- โดเมนของตัวแปร B คือ D_B = { 1 , 2 , 3 , 4 , 5 }
- สมมติเงื่อนไข คือ A < B จะเห็นว่า มีบางตัวใน D_A และ D_B ที่ขัดแย้งกับเงื่อนไข ซึ่งสามารถตัดออกไปได้เลย ดังนั้นจะเหลือค่าในโดเมน D_A = {3 , 4} และ D_B = {4 , 5}

ขั้นตอนของ Consistency Techniques จะมีอยู่ 2 ขั้นตอน คือ

- Node Consistency
- Arc Consistency

5.4.2.1 Node Consistency

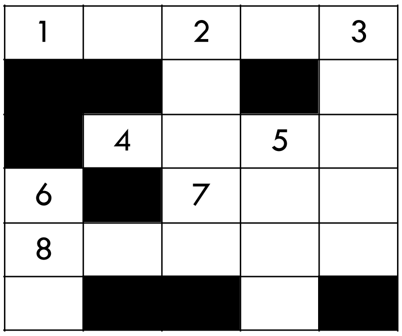
เป็นวิธีการตัดค่าในโดเมนที่ไม่เหมาะสมกับตัวแปรนั้นทิ้ง นั่นคือพิจารณาตัด Unary Constraint ที่ไม่เหมาะสมทิ้งไป ตัวแปรจะแทนด้วยโหนดใน Constraints Graph ดังนั้นการที่ทำให้ตัวแปรมีเฉพาะค่าที่เหมาะสมเท่านั้นในโดเมน จะเรียกว่าการทำให้โหนดนั้นเป็น Node Consistent นิยามได้ว่า โหนดที่แทนตัวแปร X ใน Constraints Graph จะเป็น Node Consistent ได้ก็ต่อเมื่อ ทุกค่า v ในโดเมน D_x ของ X ไม่ขัดแย้งกับ Unary Constraint ของ X และ CSP จะเป็น Node Consistent ได้ก็ต่อเมื่อ ทุกโหนดใน Constraints Graph เป็น Node Consistent โดยอัลกอริทึมแสดงในภาพที่ 5-9




```
procedure NC(G)
  for each variable X in nodes(G) do
    for each value V in the domain Dx do
      if unary constraint on X is inconsistent with V then
        delete V from Dx
    end for
  end for
end NC
```

ภาพที่ 5-9 อัลกอริทึมการทำ Node Consistance

ตัวอย่าง การทำ Node Consistency ของปัญหา Crossword Puzzle



AFT	HOSES	LINE
ALE	KEEL	SAILS
EEL	KNOT	SHEET
HEEL	LASER	STEER
HIKE	LEE	TIE

ภาพที่ 5-10 ตัวอย่างปัญหาเกม Crossword Puzzle

จากภาพที่ 5-10 เป็นปัญหา Crossword Puzzle ที่มีตัวแปรทั้งหมด 8 ตัว และค่าในโดเมนทั้งหมด 15 ค่า โดยมีรายละเอียดดังนี้

ตัวแปร : 8 ตัว (8 คำ) คือ 1, 2, 3, 4, 5, 6, 7 และ 8

โดเมน : { 'HOSES', 'LASER', 'SAILS', 'SHEET', 'STEER', 'TIE', 'HEEL , 'HIKE', 'KEEL', 'KNOT', 'LINE , 'AFT', 'ALE ', 'EEL', 'LEE' }

เมื่อพิจารณาถึงความยาวของคำพบว่า ความยาวของคำศัพท์แบ่งออกเป็น 3 กลุ่มคือ กลุ่มที่มีความยาว 3 ตัวอักษร, 4 ตัวอักษร และ 5 ตัวอักษร ดังนี้

- กลุ่มคำศัพท์ที่มีความยาว 3 ตัวอักษร: D_3 = {'AFT', 'ALE', 'EEL', 'LEE', 'TIE'}
- กลุ่มคำศัพท์ที่มีความยาว 4 ตัวอักษร: D_4 = {'LINE', 'KEEL', 'KNOT', 'HEEL', 'HIKE'}
- กลุ่มคำศัพท์ที่มีความยาว 5 ตัวอักษร: D_5 = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}

เมื่อพิจารณาถึงตัวแปรที่จะกำหนดค่าจะพบว่า

- ตัวแปร 6 และ 7 ค่าที่กำหนดจะต้องเป็นค่าของโดเมน D_3 เท่านั้น
- ตัวแปร 4 และ 5 ค่าที่กำหนดจะต้องเป็นค่าของโดเมน D_4 เท่านั้น
- ตัวแปร 1, 2, 3, และ 8 ค่าที่กำหนดจะต้องเป็นค่าของโดเมน D_5 เท่านั้น

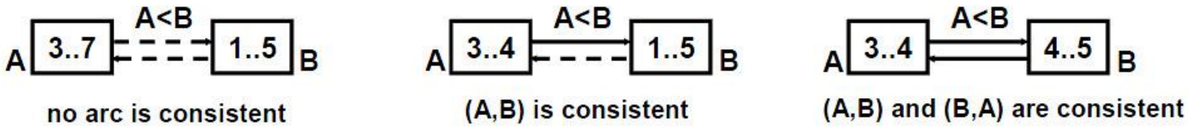
ซึ่งเมื่อทราบแบบนี้แล้ว ทำให้ขอบเขตการค้นหาคำตอบของแต่ละตัวแปรเล็กลง

5.4.2.2 Arc Consistency

เมื่อทำ Node Consistency เสร็จแล้ว แสดงว่าทุกตัวแปรมีโดเมนที่ไม่ขัดแย้งกับ Unary Constraints ดังนั้น เราไม่จำเป็นต้องพิจารณา Unary Constraints อีกต่อไป และจะมาพิจารณา Binary Constraints แทน ซึ่ง Binary Constraints จะถูกแทนด้วย Arc ใน Constraints Graph ดังนั้นจึงเรียกว่า Arc Consistency

นิยามได้ว่า Arc (V_i , V_j) จะเป็น Arc Consistent ได้ก็ต่อเมื่อทุกค่า x ในโดเมนของ V_i จะต้องมีบางค่า y ในโดเมนของ V_j ที่เมื่อกำหนดค่าให้กับ V_i และ V_j แล้วไม่ขัดกับข้อกำหนดอยู่ เช่น หากกำหนด V_i = x และ V_j = y แล้วจะไม่ขัดกับข้อกำหนด และ CSP จะเป็น Arc Consistent ก็ต่อเมื่อ ทุก Arc (V_i , V_j) เป็น Arc Consistent เมื่อเจอค่าที่ทำให้ขัดแย้งกับเงื่อนไข จะตัดค่านั้นออกไปจากโดเมนของตัวแปรนั้นๆ

แนวคิดของ Arc Consistency นั้นอยู่ในรูปแบบของกราฟที่มีทิศทาง เพราะฉะนั้นหาก Arc (V_i , V_j) เป็น Arc Consistent แต่ Arc (V_j , V_i) อาจจะไม่ใช่ก็ได้ ดังนั้นต้องตรวจสอบทั้ง Arc (V_i , V_j) และ Arc (V_j , V_i) ดังแสดงในภาพที่ 5-11 โดยมีอัลกอริทึมดังภาพที่ 5-12

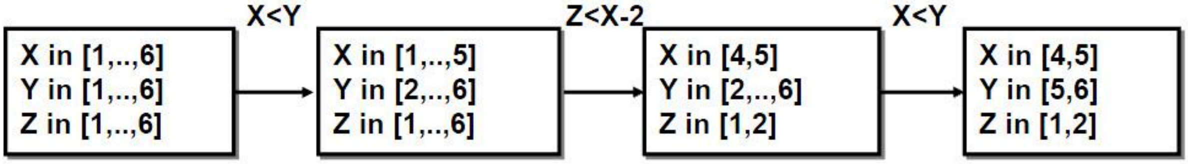


ภาพที่ 5-11 การทำให้ CSP เป็น Arc Consistent

```
procedure   REVISE(Di, Dj)
   DELETE = false
   for each X in Di do
        if ไม่มีค่า Y ใน Dj ที่ทำให้ (X,Y) consistent then
             delete X from Di
             DELETE = true
     end if
   end for
   return  DELETE;
end REVISE;
```

ภาพที่ 5-12 อัลกอริทึม REVISE สำหรับตัดค่าในโดเมนของตัวแปรออก

การทำให้ CSP เป็น Arc Consistent โดยการใช้ Revise เพียงอย่างเดียวนั้นไม่สามารถทำได้ เพราะเมื่อมีการตัดค่าในโดเมนของตัวแปรหนึ่งแล้วอาจทำให้ ค่าในโดเมนของตัวแปรที่ได้ Revise ไปแล้วก่อนหน้าขัดแย้งกับเงื่อนไขได้ เช่น ตัวแปร X, Y, Z มีโดเมน {1, 2, 3, 4, 5 ,6} โดยมีเงื่อนไข X < Y และ Z < X-2 ดังภาพที่ 5-13



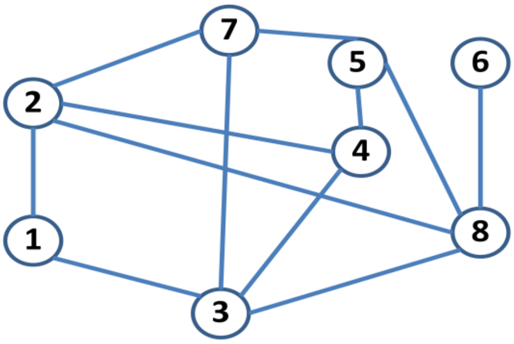
ภาพที่ 5-13 การทำให้ CSP เป็น Arc Consistent โดยการใช้ Revise

วิธีแก้ปัญหา คือ ให้ Revise จนกว่าทุกตัวแปรจะไม่มีการเปลี่ยนแปลงโดเมน โดยใช้อัลกอริทึมที่ชื่อว่า AC-1 แสดงดังภาพที่ 5-14

```
procedure   AC-1
   Q = { (Di, Dj)  ใน arcs(G) }
    repeat
        CHANGE = false
        for  each (Di, Dj) in Q do
            CHANGE = REVISE(Di, Dj)   or CHANGE
        end for
    until not(CHANGE)
end AC-1
```

ภาพที่ 5-14 อัลกอริทึม AC-1

ตัวอย่างการนำอัลกอริทึม AC-1 มาตัดค่าในโดเมนของปัญหา Crossword Puzzle สิ่งแรกที่ต้องพิจารณาก็คือ Constraint Graph เนื่องจากจะทำให้เห็น คู่ของค่าในโดเมนที่มีข้อกำหนดซึ่งและกันได้สะดวกขึ้น จากปัญหาในภาพที่ 5-10 สามารถนำมาเขียน Constraint Graph ได้ดังภาพที่ 5-15



ภาพที่ 5-15 Constraint Graph ของปัญหาเกม Crossword Puzzle

จากการทำ Node Consistency ของปัญหาเกม Crossword Puzzle ด้วย พบว่าโดเมนคำตอบของแต่ละตัวแปร ที่จะเหลือคือ

- โดเมนค่าของตัวแปร 1 (D1) = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}
- โดเมนค่าของตัวแปร 2 (D2) = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}
- โดเมนค่าของตัวแปร 3 (D3) = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}
- โดเมนค่าของตัวแปร 4 (D4) = {'LINE', 'KEEL', 'KNOT', 'HEEL', 'HIKE'}
- โดเมนค่าของตัวแปร 5 (D5) = {'LINE', 'KEEL', 'KNOT', 'HEEL', 'HIKE'}
- โดเมนค่าของตัวแปร 6 (D6) = {'AFT', 'ALE', 'EEL', 'LEE', 'TIE'}
- โดเมนค่าของตัวแปร 7 (D7) = {'AFT', 'ALE', 'EEL', 'LEE', 'TIE'}
- โดเมนค่าของตัวแปร 8 (D8) = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}

จากอัลกอริทึม AC-1 กำหนดค่า Q คือคู่ความสัมพันธ์แบบ Binary Constraint ของปัญหาซึ่งจาก Constraint Graph จะพบว่ามี จำนวน edge) ทั้งหมด 12 เส้น แต่ความสัมพันธ์ มี 2 ทิศทาง สรุปจึงมีทั้งหมด 24 คู่ความสัมพันธ์ ดังนั้นค่า Q คือ

Q = { (D1, D2), (D1, D3),
(D2, D1), (D2, D4), (D2, D7), (D2, D8),
(D3, D1), (D3, D4), (D3, D7), (D3, D8),
(D4, D2), (D4, D3), (D4, D5),
(D5, D4), (D5, D7), (D5, D8),
(D6, D8),
(D7, D2), (D7, D3), (D7, D5),
(D8, D2), (D8, D3), (D8, D5), (D8, D6) }

เมื่อได้ค่า Q แล้วก็ทำการตรวจสอบด้วยอัลกอริทึม REVISE ทีละคู่ความสัมพันธ์ใน Q โดยคู่แรกที่ทำการเปรียบเทียบคือคู่โดเมน D1 และ D2 (D1, D2) โดยมีข้อกำหนดในปัญหาที่มีความสัมพันธ์กันระหว่างตัวแปร D1 และ D2 คือ 1[3] = 2[1]

- D1 = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}
- D2 = {'HOSES', 'SAILS', 'SHEET', 'LASER', 'STEER'}

นำค่าแรกใน D1 คือ "HOSES" แล้วเปรียบเทียบตัวอักษรที่ 3 ซึ่งก็คือ "S" ว่ามีค่าใดใน D2 ที่ตัวอักษรแรกเป็น "S" หรือไม่ (ตามข้อกำหนด 1[3] = 2[1]) ถ้าหากว่ามีจะไม่สามารถ ตัดคำนี้ออกจากโดเมนได้ ซึ่งจะพบว่าในโดเมน D2 มีคำศัพท์ว่า "SAILS" ที่ตัวอักษรตัวแรกขึ้นต้นด้วย "S" ดังนั้นจึงไม่สามารถตัดคำศัพท์ "HOSES" ออกจากโดเมน D1 ได้

จากนั้นนำค่าที่ 2 ในโดเมน D1 คือ "LASER" แล้วเปรียบเทียบตัวอักษรที่ 3 ซึ่งก็คือ "S" ว่ามีค่าใดใน D2 ที่ตัวอักษรแรกเป็น "S" หรือไม่ (ตามข้อกำหนด 1[3] = 2[1]) ซึ่งจะพบว่าในโดเมน D2 มีคำศัพท์ว่า "SAILS" ที่ตัวอักษรตัวแรกขึ้นต้นด้วย "S" ดังนั้นจึงไม่สามารถตัดคำศัพท์ "LASER" ออกจากโดเมน D1 ได้

จากนั้นนำค่าที่ 3 ในโดเมน D1 คือ "SAILS" แล้วเปรียบเทียบตัวอักษรที่ 3 ซึ่งก็คือ "I" ว่ามีค่าใดใน D2 ที่ตัวอักษรแรกเป็น "I" หรือไม่ (ตามข้อกำหนด 1[3] = 2[1]) ซึ่งจะพบว่าในโดเมน D2 ไม่มีคำศัพท์ที่ขึ้นต้นด้วย "I" ดังนั้นจึงสามารถตัดคำศัพท์ "SAILS" ออกจากโดเมน D1 ได้

ให้ทำแบบนี้ต่อไปจนครบค่าทุกค่าในโดเมน D1 เมื่อทำเสร็จแล้วก็วนรอบทำการเปรียบเทียบแบบนี้ให้ครบทุกคู่ความสัมพันธ์ทั้งหมดที่อยู่ใน Q ถ้ามีการเปลี่ยนแปลงค่าในโดเมน เพียงแค่ค่าเดียว ก็จะทำการวนซ้ำเปรียบเทียบใหม่ทั้งหมดอีกทั้ง โดยจากตัวอย่างจะพบว่าเมื่อทำการตัดค่าในโดเมนรอบแรกของอัลกอริทึม AC-1 จะสามารถตัดค่าในโดเมนได้จนเหลือค่าในโดเมนแต่ละตัวดังนี้

D1 = {'HOSES', ~~'LASER'~~, ~~'SAILS'~~, ~~'SHEET'~~, ~~STEER'~~}

D2 = {~~'HOSES'~~, ~~'LASER'~~, 'SAILS', ~~'SHEET'~~, ~~STEER'~~}

D3 = {~~'HOSES'~~, ~~'LASER'~~, ~~'SAILS'~~, 'SHEET', STEER'}

D4 = {~~'HEEL'~~, 'HIKE', ~~'KEEL'~~, ~~'KNOT'~~, ~~'LINE'~~}

D5 = {~~'HEEL'~~, ~~'HIKE'~~, 'KEEL', ~~'KNOT'~~, ~~'LINE'~~}

D6 = {~~'AFT~~, 'ALE', ~~'EEL'~~, ~~'LEE'~~, ~~'TIE'~~}

D7 = {~~'AFT~~, ~~'ALE'~~, 'EEL', ~~'LEE'~~, ~~'TIE'~~}

D8 = {~~'HOSES'~~, 'LASER', ~~'SAILS'~~, ~~'SHEET'~~, ~~STEER'~~}

เมื่อวนการทำงานรอบที่ 2 ของอัลกอริทึม AC-1 พบว่าสามารถตัดค่าในโดเมน D3 คำว่า “SHEET” เพิ่มได้อีกคำและไม่สามารถตัดค่าใดๆ ต่อไปได้อีก อัลกอริทึมจึงสิ้นสุดการทำงานลง เนื่องจากปัญหานี้เป็นปัญหาที่ง่ายทำให้สามารถได้คำตอบของปัญหาเลยจากการใช้งานอัลกอริทึม AC-1 โดยคำตอบของปัญหาที่ได้คือ

D1 = {‘HOSES’, ‘LASER’, ‘SAILS’, ‘SHEET’, STEER’} => ตัวแปร 1 = ‘HOSES’

D2 = {‘HOSES’, ‘LASER’, ‘SAILS’, ‘SHEET’, STEER’} => ตัวแปร 2 = ‘SAILS’

D3 = {‘HOSES’, ‘LASER’, ‘SAILS’, ‘SHEET’, STEER’} => ตัวแปร 3 = ‘STEER’

D4 = {‘HEEL’, ‘HIKE’, ‘KEEL’, ‘KNOT’, ‘LINE’} => ตัวแปร 4 = ‘HIKE’

D5 = {‘HEEL’, ‘HIKE’, ‘KEEL’, ‘KNOT’, ‘LINE’} => ตัวแปร 5 = ‘KEEL’

D6 = {‘AFT, ‘ALE’, ‘EEL’, ‘LEE’, ‘TIE’} => ตัวแปร 6 = ‘ALE’

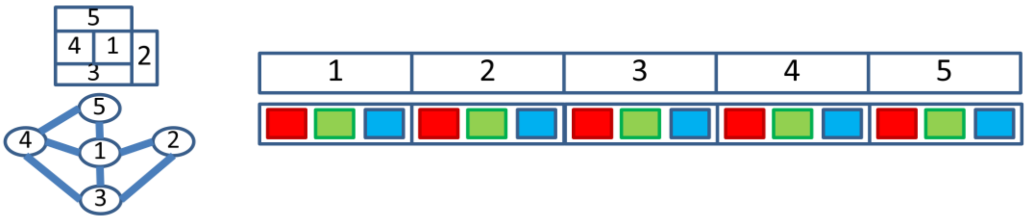
D7 = {‘AFT, ‘ALE’, ‘EEL’, ‘LEE’, ‘TIE’} => ตัวแปร 7 = ‘LEE’

D8 = {‘HOSES’, ‘LASER’, ‘SAILS’, ‘SHEET’, STEER’} => ตัวแปร 8 = ‘STEER’

ข้อเสียของอัลกอริทึม AC-1 คือ ประสิทธิภาพยังไม่ดีเนื่องจากการเปลี่ยนแปลงของคู่หนึ่งในกราฟจะมีผลให้ต้องเริ่มตรวจสอบ arc consistency ใหม่ทั้งหมด (ทุกคู่ของ edge ในกราฟ) ถ้า REVISE(Di, Dj) มีการลดค่าใน Di ก็ควรตรวจสอบเพิ่มแค่คู่ของ (Di, Dm) โดย Dm คือโดเมนใดๆ ที่มีความสัมพันธ์กับ Di โดยไม่จำเป็นต้องตรวจสอบ (Dm, Di) จึงมีการพัฒนาอัลกอริทึมใหม่มีชื่อว่า AC-3 โดยเป็นการลดคู่ตรวจสอบลงจากอัลกอริทึม AC-1 ดังภาพที่ 5-16

```
procedure  AC-3
   Q = { (Di, Dj)  ใน arcs(G) }
    while not Q empty
        select and delete  (Dk, Dm) from Q;
        if REVISE(Dk, Dm) then
           Q = Q union {(Dx, Dk) ใน arc(G)}
        end if
    end while
end AC-3
```

ภาพที่ 5-16 อัลกอริทึม AC-3




```
procedure AC-3
  Q = { (Di, Dj) ใน arcs(G) }
   while not Q empty
      select and delete (Dk, Dm) from Q;
      if REVISE(Dk, Dm) then
         Q = Q union {(Dx, Dk) ใน arc(G)}
      end if
   end while
end AC-3
```

Q = {(D1, D2), (D1, D3), (D1, D4), (D1, D5),
(D2, D1), (D2, D3), (D3, D1), (D3, D2),
(D3, D4), (D4, D1), (D4, D3), (D4, D5),
(D5, D1), (D5, D4) }

D1 = {'R', 'G', 'B'}
D2 = {'R', 'G', 'B'}
D3 = {'R', 'G', 'B'}
D4 = {'R', 'G', 'B'}
D5 = {'R', 'G', 'B'}

ข้อกำหนด (constraint)
1 != 2, 1 != 3,
1 != 4, 1 != 5,
2 != 3, 3 != 4,
4 != 5

ภาพที่ 5-17 รายละเอียดปัญหา Map Coloring ด้วยอัลกอริทึม AC-3

อย่างไรก็ตามทั้งอัลกอริทึม AC-1 และ AC-3 ไม่สามารถช่วยในการตัดค่าในโดเมนของตัวแปรแต่ละตัวได้เสมอ เช่น ปัญหาของ Map Coloring ดังแสดงในภาพที่ 5-17 ที่ไม่สามารถตัดค่าในโดเมนคำตอบใดๆ ได้เลย

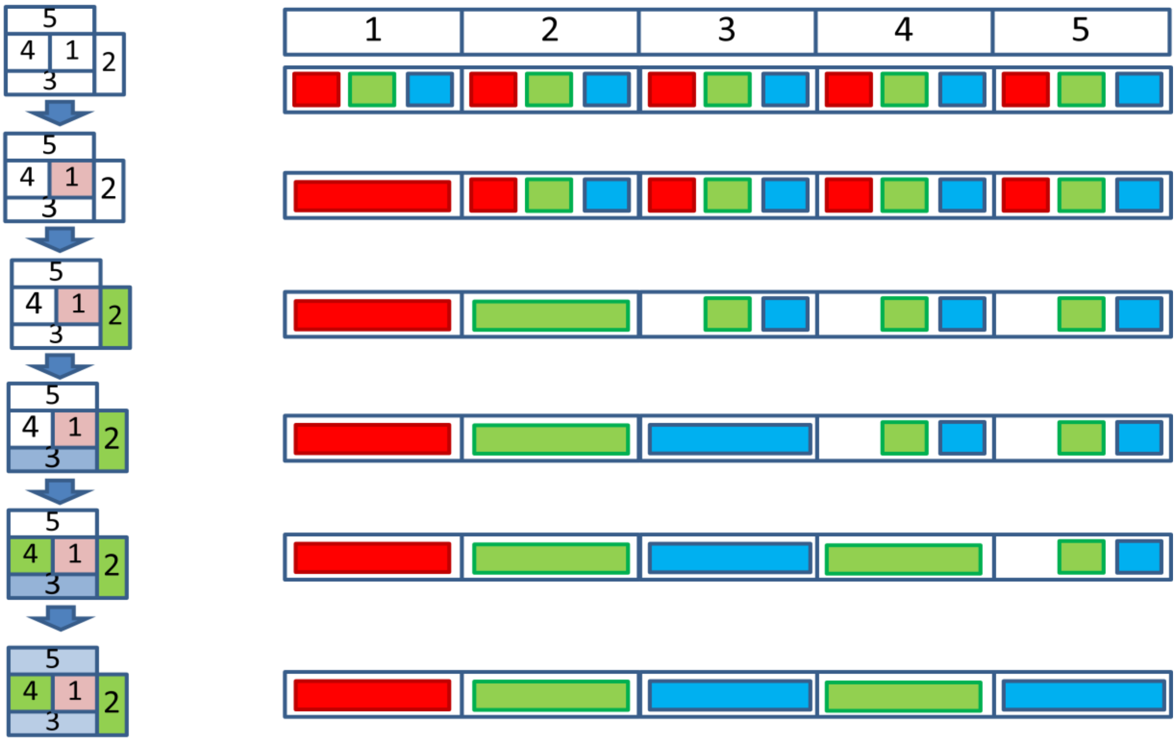
5.4.3 Forward Checking

เป็นกระบวนการค้นหาที่มีพื้นฐานมาจากการค้นหาแบบปริภูมิสถานะ แต่จะมีการนำข้อดีของ Consistency Techniques คือ ตัดค่าข้อมูลในโดเมนออกก่อนทำการค้นหาเพื่อลดขนาดข้อมูลในการค้นหา และจะหยุดทำเมื่อไม่มีค่าที่เหมาะสมสำหรับตัวแปร

อัลกอริทึม Forward Checking คือ

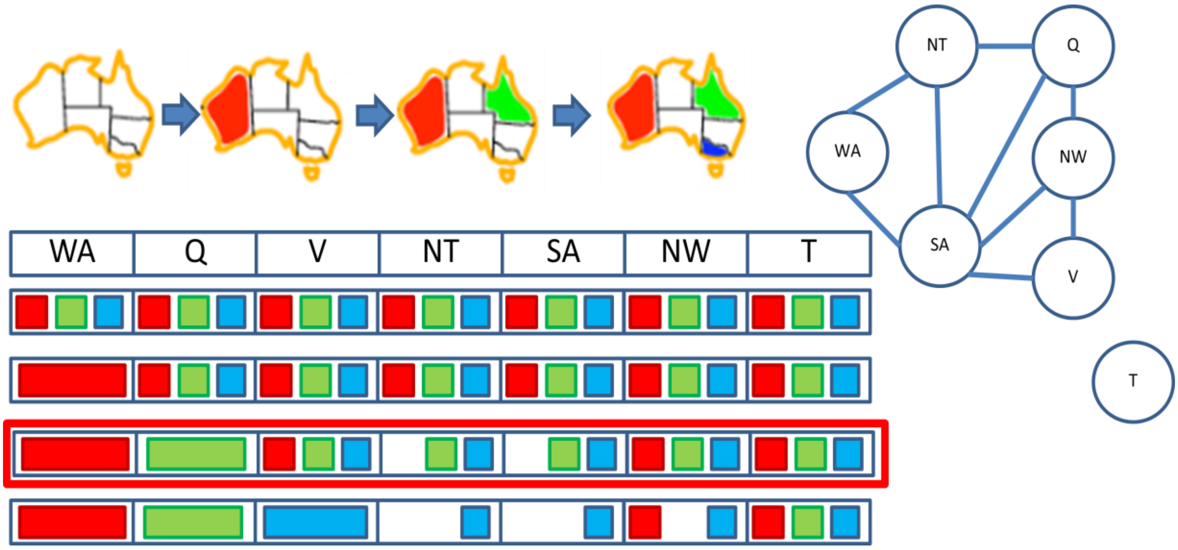
- หลังจากที่ตัวแปร X ถูกใส่ค่า v
- ให้ตรวจสอบค่าของตัวแปร Y ที่เชื่อมต่อกับ X
- ลบค่าในโดเมนของตัวแปร Y ที่ขัดกับค่าของ v

ภาพที่ 5-18 แสดงขั้นตอนการใช้ Forward Checking ในการแก้ปัญหาตัวอย่างปัญหา Map Coloring แบบมีตัวแปร 5 ตัว แต่ละตัวแปรมีโดเมนคำตอบที่เป็นไปได้คือ "R", "G", และ "B" ขั้นตอนแรกเป็นการสุ่มให้ตัวแปร 1 มีค่า "R" ซึ่งจะทำให้สามารถตัด "R" ออกจากคำตอบของตัวแปร 3, 4 และ 5 ได้เนื่องจาก จากความสัมพันธ์ตัวแปรทั้ง 3 ตัวนี้ห้ามมีค่าเดียวกับตัวแปร 1 จากนั้นทำซ้ำสุ่มเลือกตัวแปรและตัดค่าของโดเมนอื่นไปเรื่อยๆ จนได้คำตอบ



ภาพที่ 5-18 ขั้นตอนการแก้ปัญหา Map Coloring ด้วยวิธี Forward Checking

อย่างไรก็ตามอัลกอริทึม Forward Checking เพียงอย่างเดียวนั้น ยังไม่มีประสิทธิภาพเพียงพอที่จะช่วยตรวจสอบข้อกำหนดในการแก้ปัญหาก่อนที่จะพบความขัดแย้งกัน ตัวอย่างปัญหา Map Coloring อีกปัญหาดังภาพที่ 5-19



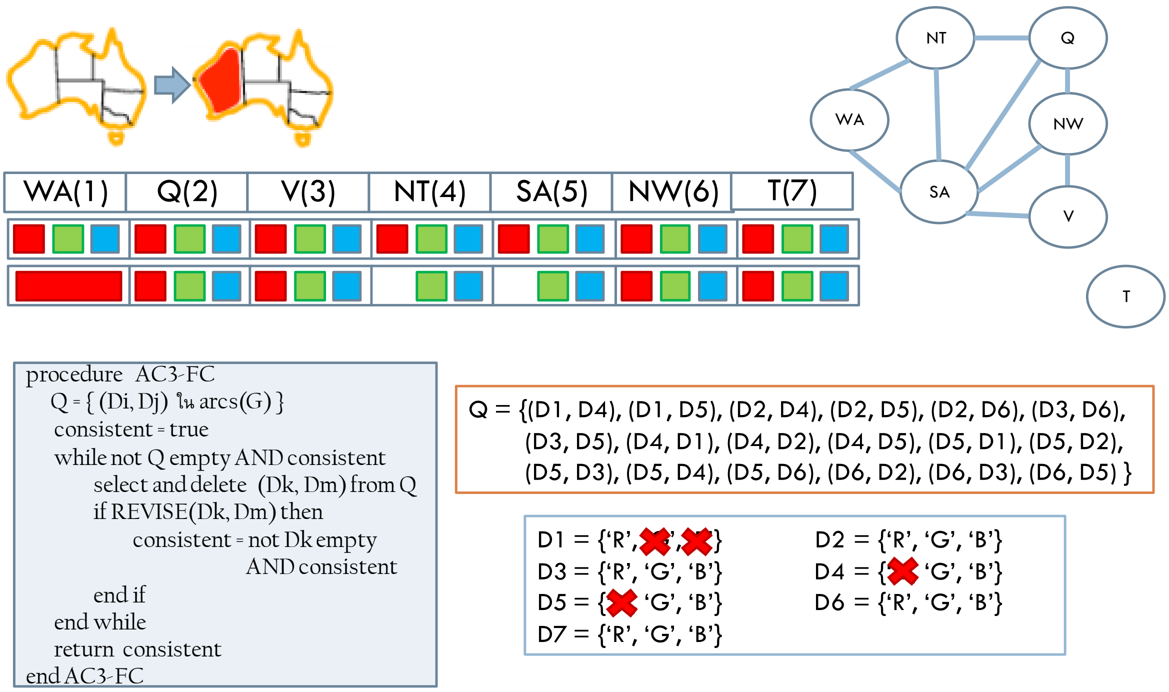
ภาพที่ 5-19 ขั้นตอนการแก้ปัญหา Map Coloring ของออสเตรเลียด้วยวิธี forward checking

```
procedure   AC3-FC
   Q = { (Di, Dj)  ใน arcs(G) }
    consistent = true
    while not Q empty AND consistent
         select and delete  (Dk, Dm) from Q
         if REVISE(Dk, Dm) then
              consistent = not Dk empty AND  consistent
         end if
    end while
    return  consistent
end AC3-FC
```

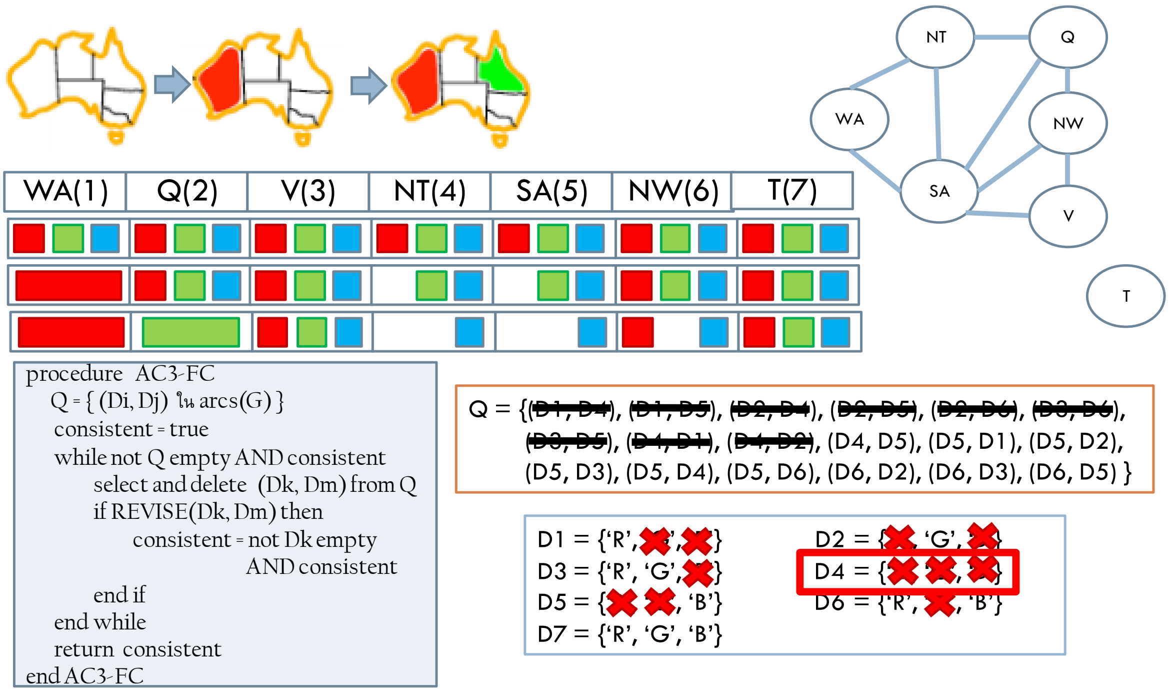
ภาพที่ 5-20 อัลกอริทึม AC3-FC

จากภาพที่ 5-19 จะเห็นได้ว่า อัลกอริทึมได้สุ่มให้ค่า WA เป็นสีแดง จากนั้นในขั้นตอนที่ 2 ทำการตัดสีแดงออกจากโดเมนคำตอบของตัวแปร NT และ SA พร้อมกับเลือกสีเขียนเป็นคำตอบของ Q และในขั้นตอนที่ 3 ทำการตัดสีเขียวออกจากโดเมนคำตอบของตัวแปร NT, SA และ NW ซึ่งจริงๆแล้วในขั้นตอนนี้จะพบว่าโดเมนคำตอบของ NT และ SA เหลือแค่สีน้ำเงินเท่านั้น แต่ทว่าจากข้อกำหนด NT และ SA ห้ามมีสีเดียวกัน อย่างไรก็ตามอัลกอริทึม Forward Checking เพียงอย่างเดียวนั้นไม่ได้ทำการตรวจสอบข้อกำหนดนี้ และยังทำงานต่อโดยเลือกสีน้ำเงินให้กับตัวแปร V ซึ่งจะส่งผลต่อไปให้ สีน้ำเงินถูกลบออกจากโดเมนคำตอบของตัวแปร NW และ SA โดยจะทำให้ SA ไม่มีคำตอบเหลืออยู่ในโดเมน ซึ่งขั้นตอนนี้อัลกอริทึม Forward Checking ถึงจะรู้ว่าสุ่มค่าเริ่มต้นให้กับตัวแปรผิด และทำการสุ่มค่าอื่น แทน ดังนั้นเพื่อแก้ไขปัญหานี้ให้อัลกอริทึม Forward Checking สามารถตรวจสอบข้อกำหนดก่อนทำงานขั้นตอนถัดไป จึงมีพัฒนาอัลกอริทึมที่นำอัลกอริทึม Forward Checking และอัลกอริทึม AC-3 มาประยุกต์ร่วมกัน กลายเป็นอัลกอริทึม AC3-FC ดังแสดงในภาพที่ 5-20

การแก้ปัญหา Map coloring ของออสเตรเลียที่ใช้อัลกอริทึม AC-FC3 โดยภาพที่ 5-21 แสดงขั้นตอนแรกของอัลกอริทึม โดยเริ่มจากการทำ Forward Checking โดยการสุ่มค่าเริ่มต้นให้กับ ตัวแปร WA เป็นสีแดงจะสามารถทำให้ตัดค่าในโดเมนคำตอบของ WA ได้ 2 ค่าก็คือ สีเขียวและสีน้ำเงิน และส่งผลให้ตัวแปร NT และ SA ซึ่งมีข้อกำหนดห้ามเป็นสีเดียวกันกับ WA ตัดค่าในโดเมนคำตอบ ที่เป็นสีแดงออกได้ จากนั้นก่อนจะทำการสุ่มค่าให้กับตัวแปรถัดไปเพื่อทำ Forward Checking อีกครั้ง ให้ทำอัลกอริทึม AC-3 ก่อน โดยจากตัวอย่างนั้นการทำ AC-3 ไม่สามารถตัดค่าในโดเมนคำตอบใดๆ ออกไปได้ จึงจบขั้นตอนแรกของอัลกอริทึม AC3-FC เพียงเท่านี้และวนซ้ำทำ Forward Checking กับตัวแปรถัดไป ซึ่งก็คือตัวแปร Q ดังแสดงในภาพที่ 5-22



ภาพที่ 5-21 ขั้นตอนแรกของอัลกอริทึม AC3-FC เพื่อแก้ปัญหา Map Coloring



ภาพที่ 5-22 ขั้นตอนที่ 2 ของอัลกอริทึม AC3-FC เพื่อแก้ปัญหา Map Coloring

จากภาพที่ 5-22 จะเห็นได้ว่าเมื่อเลือกค่าให้กับตัวแปร Q เป็นสีเขียว ด้วยอัลกอริทึม Forward Checking จะทำให้สามารถตัดค่าสีแดงและสีน้ำเงินออกจากโดเมนคำตอบของตัวแปร Q ได้ซึ่งส่งผลให้สามารถตัดสีเขียวออกจากตัวแปร NT, SA และ NW ที่ขัดกับข้อกำหนดออกได้ด้วย จากนั้นใช้อัลกอริทึม AC-3 เพื่อตรวจสอบความถูกต้องของคำตอบที่เหลืออยู่ในแต่ละโดเมน พบว่าในคู่ความสัมพันธ์ (D4, D5) จะสามารถตัดสีน้ำเงินออกจากโดเมนคำตอบของ NT ได้ ทำให้ NT ไม่เหลือคำตอบในโดเมน ซึ่ง ณ ขั้นตอนนี้จะทำให้อัลกอริทึม AC3-FC ทราบว่าการสุ่มค่าคำตอบให้กับตัวแปรนั้นผิด และสามารถทราบก่อนการใช้อัลกอริทึม Forward Checking เพียงอย่างเดียว

5.5 การปรับปรุงประสิทธิภาพในการค้นหา

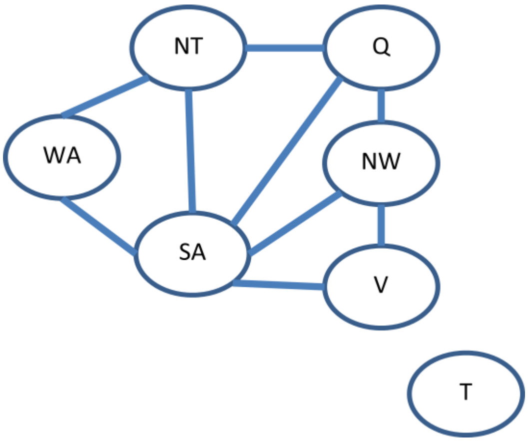
เมื่อ Forward Checking ไม่ได้เป็นอัลกอริทึมที่ใช้แก้ปัญหาโดยตรง แต่เป็นเพียงแค่ตัดตัวเลือกออก จากการทดลองและการวิเคราะห์ของนักวิจัยหลายๆ คนพบว่าสาเหตุของปัญหาที่เกิดขึ้นเกิดจากลำดับของการกำหนดค่าลงไปในตัวแปร และลำดับค่าที่กำหนดลงไปในแต่ละตัวแปรซึ่งมีผลอย่างมากในการแก้ปัญหา และนักวิจัยได้กำหนดหลักการในการกำหนดค่าตัวแปรไว้ดังนี้

- วิธีการเลือก**ตัวแปร**ที่จะใช้ในการกำหนดค่าที่นิยมใช้อาศัยหลักการของ “First-fail” (จะไปสู่จุดหมายได้จะต้องลองไปทางที่มีโอกาสล้มเหลวมากที่สุดก่อน) หรือเรียกว่า Minimum Remaining Value (MRV) Heuristic ซึ่งก็คือ **เลือกตัวแปรที่เหลือค่าในโดเมนน้อยที่สุดขึ้นมาทำงานก่อน** ในกรณีที่มี**ตัวแปร**ที่มีค่าในโดเมนเท่ากัน จะนับ **degree heuristic** คือเลือกตัวแปรที่มี constraint มากที่สุดขึ้นมาใช้งานก่อน
- เมื่อทำการเลือก**ตัวแปร**ที่จะทำการค้นหาได้แล้ว ต่อไปก็คือ**เลือกค่า**ที่จะให้ตัวแปรนั้นใช้ จะอาศัยหลักการที่เรียกว่า Least Constraint Value คือเลือกค่าที่ให้กระทบกับเพื่อนบ้านน้อยที่สุด

เมื่อทำการปรับปรุงประสิทธิภาพในการค้นหานี้ มาใช้ในการแก้ปัญหา Map Coloring ของออสเตรเลียอีกครั้ง จะพบว่าการเลือกตัวแปรแรกที่ควรจะสุ่มใส่ค่าคือเลือกจากตัวที่ค่าในโดเมนเหลือน้อยที่สุด โดยในขั้นตอนแรกนั้น ทุกตัวแปรมีค่าที่เหลือในโดเมน 3 ค่าเท่ากันหมด จึงต้องพิจารณาจากตัวแปรไหนที่มี Constraint มากที่สุด จาก Constraint Graph ในภาพที่ 5-23 พบว่าแต่ละตัวแปรมีจำนวน Constraint ดังนี้

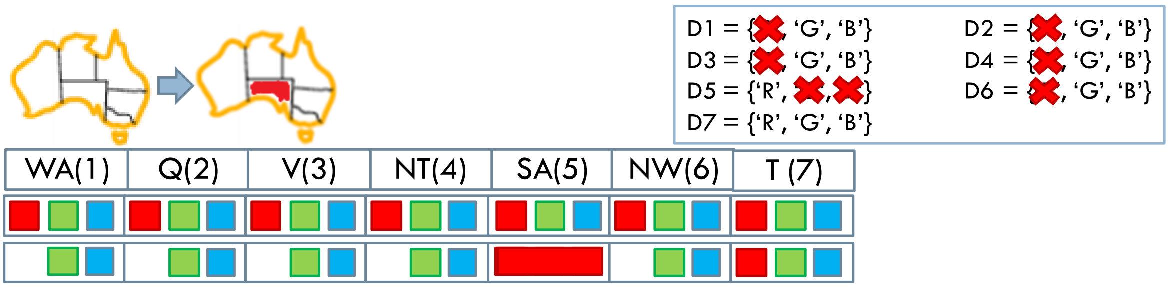
WA = 2, NT = 3, **SA = 5**, Q = 3, NW = 3, V = 2, และ T = 0

โดยตัวแปร SA มีจำนวน Constraint มากที่สุดคือ 5 Constraints ดังนั้น Forward Checking จึงควรเลือก SA ขึ้นมากำหนดค่าเป็นตัวแรก ต่อมาคือค่าที่เลือก สีแดง, สีเขียว หรือ สีน้ำเงินที่เลือกแล้วจะกระทบกับเพื่อนบ้านน้อยที่สุด ซึ่งพบว่าทั้ง 3 สีจะส่งผลกระทบต่อเพื่อนบ้านเท่ากันดังนั้นเลือกสีอะไรก็ได้



ภาพที่ 5-23 Constraint Graph ของปัญหา Map Coloring ออสเตรเลีย

เมื่อทำการเลือกตัวแปร SA เพื่อกำหนดค่าสีแดง และทำอัลกอริทึม Forward Checking จะพบว่าสามารถตัดสีแดงออกจากโดเมนคำตอบของตัวแปร WA, NT, Q, NW และ V ได้ ดังภาพที่ 5-24 หลังจากนั้นก็ใช้อัลกอริทึม AC-3 ซึ่งไม่สามารถตัดค่าในโดเมนของตัวแปรใดๆ ได้

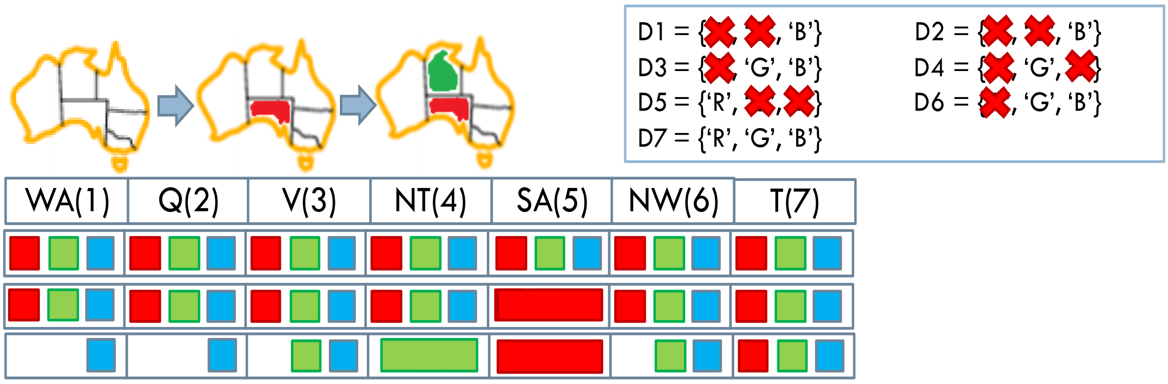


ภาพที่ 5-24 การทำอัลกอริทึม AC3-FC แบบปรับปรุงประสิทธิภาพในขั้นตอนที่ 1

ขั้นตอนที่ 2 ก็เริ่มต้น Forward Checking ใหม่โดยเลือกตัวแปรขึ้นมากำหนดค่าจากตัวแปร 6 ตัวที่เหลือ ซึ่งเมื่อพิจารณาจากจำนวนของ Constraint พบว่า

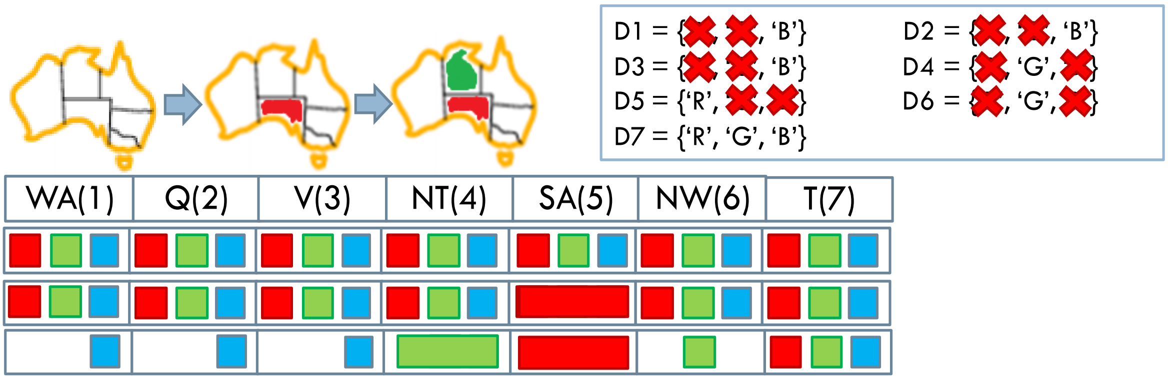
WA = 2, **NT = 3**, **Q = 3**, NW = 3, V = 2, และ T = 0

โดยตัวแปร NT และ Q มีจำนวน Constraint มากที่สุดคือ 4 Constraints ดังนั้น Forward Checking จึงควรเลือก NT หรือ Q เพื่อกำหนดค่าเป็นตัวถัดไป ต่อมาคือค่าที่เลือกสีที่เหลือคือ สีเขียว หรือสีน้ำเงินที่เลือกแล้วจะกระทบกับเพื่อนบ้านน้อยที่สุด ซึ่งพบว่าทั้ง 2 สีจะส่งผลกระทบต่อเพื่อนบ้านเท่ากัน ดังนั้นเลือกสีอะไรก็ได้ จากตัวอย่างในภาพที่ 5-25 จึงทำการเลือกสีเขียวเพื่อกำหนดค่าให้กับตัวแปร NT



ภาพที่ 5-25 การทำอัลกอริทึม AC3-FC ในส่วนของ FC แบบปรับปรุงประสิทธิภาพในขั้นตอนที่ 2

เมื่อกำหนดค่าสีเขียวให้กับตัวแปร NT เรียบร้อยแล้ว การทำ Forward Checking จะทำให้สามารถตัดค่าสีเขียวออกจากโดเมนคำตอบของตัวแปร WA และ Q ได้ ทำให้ทั้ง WA และ Q เหลือค่าในโดเมนคำตอบเพียงแค่สีน้ำเงินเท่านั้น จากนั้นทำอัลกอริทึม AC-3 พบว่าคู่ (D6, D2) สามารถตัดค่าสีน้ำเงินออกจากโดเมน D6 ได้เนื่องด้วย ถ้า NW ได้คำตอบเป็นสีน้ำเงินจะทำให้ขัดกับข้อกำหนด และเมื่อวนอีกรอบจะพบว่ามีคู่ (D3, D6) ที่สามารถตัดค่าสีเขียวออกจากโดเมนคำตอบของ V ได้ ดังนั้นเมื่อทำ AC-3 เสร็จค่าในโดเมนทั้ง 7 โดเมนจะเหลือค่าดังภาพที่ 5-26



ภาพที่ 5-26 การทำอัลกอริทึม AC3-FC ในส่วนของ AC-3 แบบปรับปรุงประสิทธิภาพในขั้นตอนที่ 2

พบว่าหลังจากทำ AC3-FC ในขั้นตอนที่ 2 แล้วจะได้คำตอบในโดเมนของตัวแปรแต่ละตัวเหลือเพียง 1 ค่าเท่านั้น และไม่ขัดต่อข้อกำหนด จึงสามารถสรุปเป็นคำตอบได้เลย หากสังเกตุที่ตัวแปร T ถึงแม้จะเหลือค่าในโดเมนถึง 3 ค่า แต่อย่างไรก็ตามเนื่องจากตัวแปร T ไม่มีความสัมพันธ์หรือข้อกำหนดร่วมกับตัวแปรอื่นเลย ดังนั้นจึงสามารถเลือกสีใดก็ได้เป็นค่าให้กับตัวแปร T ตัวอย่างผลลัพธ์จะแสดงได้ดังภาพที่ 5-27

ภาพที่ 5-27 ผลลัพธ์ของการใช้ AC3-FC เพื่อแก้ปัญหา Map Coloring ออสเตรเลีย

5.6 สรุป

ปัญหาประเภทมีข้อกำหนดหรือเงื่อนไขที่การแก้ปัญหาจะต้องได้คำตอบที่ไม่ขัดกับข้อกำหนดหรือเงื่อนไขใดๆ เรียกว่า Constraint Satisfaction Problems (CSP) โดยวิธีการแก้ไขปัญหาแบบง่ายที่สุดคือ Generate and Test แต่ทว่าการแก้ปัญหาด้วยเทคนิคนี้จะเสียเวลาในการแก้ไขปัญหาเป็นอย่างยิ่งเนื่องด้วย จะต้องค้นหาคำตอบจากค่าทุกค่าในโดเมนคำตอบของทุกตัวแปร ดังนั้นจึงมีการพัฒนาอัลกอริทึม Backtracking ขึ้นมาเพื่อตัดกิ่งของต้นไม้ที่จะไม่พาไปสู่คำตอบออกไปจากการค้นหา จึงทำให้การค้นหามีประสิทธิภาพมากขึ้นกว่าอัลกอริทึมแบบ Generate and Test แต่อย่างไรก็ตามการค้นหาแบบ Backtracking ไม่มีการตรวจสอบความสัมพันธ์ของข้อกำหนดก่อนการค้นหาทำให้ประสิทธิภาพยังไม่ดีเท่าที่ควร เพื่อพัฒนาให้อัลกอริทึมแบบ Backtracking มีประสิทธิภาพมากขึ้นจึงมีการพัฒนาอัลกอริทึม REVISE, AC-1 และ AC-3 ขึ้นมาเพื่อตรวจสอบความสัมพันธ์และตัดค่าในโดเมนคำตอบที่ขัดกับข้อกำหนดออก จึงทำให้การค้นหามีประสิทธิภาพมากขึ้น มากไปกว่านั้นยังมีการนำอัลกอริทึม Forward Checking ที่ช่วยในการตัดค่าในโดเมนคำตอบของตัวแปรอื่นออก หลังจากมีการสุ่มคำตอบในตัวแปรบางตัวเพื่อลดค่าในโดเมนคำตอบลงอีก ทำให้การค้นหารวดเร็วมากขึ้น ท้ายที่สุดเพื่อให้การค้นหามีประสิทธิภาพและรวดเร็วจึงได้ทำอัลกอริทึม Forward Checking มารวมกับอัลกอริทึม AC-3 กลายเป็น AC3-FC

การเลือกตัวแปรและการกำหนดค่าเริ่มต้นให้กับตัวแปรใดๆ ก่อนเรียกใช้งานอัลกอริทึม Forward Checking ภายในอัลกอริทึม AC3-FC นั้น จากการวิจัยพบว่า การเลือกตัวแปรนั้นมีส่วนสำคัญที่จะทำให้การค้นหาพบคำตอบรวดเร็วขึ้น โดยมีหลักการเลือกตัวแปรคือ เลือกตัวแปรที่มีค่าในโดเมนคำตอบน้อยที่สุดขึ้นมาก่อน แต่ถ้าทุกตัวแปรมีค่าในโดเมนคำตอบเหลือเท่าๆ กัน ให้เลือกตัวแปรที่มีข้อกำหนดเกี่ยวพันกับตัวแปรอื่นๆ มากที่สุดขึ้นมาก่อน จากนั้นการเลือกคำตอบให้กับตัวแปรก็คือเลือกคำตอบที่เมื่อเลือกแล้วจะกระทบกับตัวแปรอื่นน้อยที่สุด

แบบฝึกหัดท้ายบท

จากปัญหาการจัดตารางห้องเรียน โดยต้องจัดวิชาเรียนที่มีจำนวนทั้งหมด 4 วิชา C1, C2, C3, C4 มีรายละเอียดดังนี้

Class	Time
C1	8.00 – 10.30
C2	9.00 – 11.30
C3	10.00 - 12.30
C4	11.0 – 13.30

กำหนดให้มีจำนวนห้องเรียนทั้งหมด 3 ห้อง R1, R2, R3 และมีข้อกำหนดดังต่อไปนี้

- แต่ละวิชาจะต้องมีห้องเรียน 1 ห้องใน R1, R2 หรือ R3
- R3 เล็กเกิน ไม่สามารถใช้สอนวิชา C3 ได้
- R2 และ R3 เล็กเกิน ไม่สามารถสอนวิชา C4 ได้

1. จงเขียนค่าในโดเมนของแต่ละตัวแปรที่ใช้หลักการ node consistency แล้ว
2. เขียนข้อกำหนด (Constraint) ให้อยู่ในรูปของตัวแปร
3. เขียน Constraint Graph
4. เขียนขั้นตอนการแก้ปัญหา CSP

แผนปฏิบัติการสอนสัปดาห์ที่ 6

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี
เรื่อง ตรรกศาสตร์ที่ว่าด้วยประพจน์และการอนุมาน **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. พื้นฐานของตรรกศาสตร์	IS 6.1
2. ตรรกศาสตร์ที่ว่าด้วยประพจน์	IS 6.2
3. คำนิยามเกี่ยวกับค่าความเป็นจริง	IS 6.3
4. กฎของการแทนที่	IS 6.4
5. กฎของการอนุมาน	IS 6.5
6. การนิรนัยหาข้อสรุป	IS 6.6

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		1-2-3-4-5-6			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 6
ตรรกศาสตร์ที่ว่าด้วยประพจน์และการอนุมาน

การที่เครื่องจักรหรือสิ่งใดๆ จะมีความฉลาดปกติจะมีองค์ประกอบอยู่ 4 อย่างคือ สามารถรับข้อมูลจากสิ่งแวดล้อมได้ (Perceiving) สามารถนำข้อมูลมาเป็นความรู้ (Knowledge Representation) การนำความรู้มาหาเหตุผล (Reasoning) และ การกระทำตามเหตุผล (Acting)

การรับข้อมูลจากสิ่งแวดล้อมอาจเป็นการรับข้อมูลจากเซ็นเซอร์ เสียง ภาพ หรือ คีย์บอร์ด จากนั้นจึงนำข้อมูลที่รับเข้ามาเปลี่ยนเป็นความรู้ที่เรียกว่าการแทนองค์ความรู้ ศาสตร์ที่สำคัญที่ใช้ในการแทนองค์ความรู้ก็คือ ตรรกศาสตร์ (Logic) หรือเรียกว่า การแทนองค์ความรู้เชิงตรรกะ (Logic Knowledge Representation)

6.1 พื้นฐานของตรรกศาสตร์

ตรรกะ (Logic) คือ ศาสตร์ที่ว่าด้วยการหาเหตุและผลด้วยวิธีการต่างๆ อย่างมีรูปแบบและระบบที่ชัดเจน โดยการพิสูจน์จากข้อเท็จจริงที่กำหนด ตรรกศาสตร์แบ่งออกเป็น 2 ประเภท

- **ตรรกศาสตร์แบบดั้งเดิม** เป็นตรรกศาสตร์แรกเริ่มที่พัฒนามาจากหลักการและกระบวนการทางเหตุผลของอริสโตเติล
    - **ตรรกนิรนัย (Deductive Logic)** เป็นการหาความจริงจากส่วนมากไปหาส่วนน้อย
    - **ตรรกอุปนัย (Inductive Logic)** เป็นการหาความจริงจากส่วนน้อยไปหาส่วนมาก
- **ตรรกสัญลักษณ์ (Symbolic Logic)** เป็นตรรกศาสตร์ที่ใช้วิธีการทางคณิตศาสตร์เข้ามาพิสูจน์ข้อเท็จจริง มีการใช้สัญลักษณ์แทนการใช้เวลาที่มีความกำกวม

ในทางปัญญาประดิษฐ์จะใช้ตรรกสัญลักษณ์ในการแทนองค์ความรู้ โดยตรรกสัญลักษณ์สามารถแบ่งได้ 2 ประเภท คือ

- **ตรรกะที่ว่าด้วยประพจน์ (Propositional Logic)** เป็นตรรกศาสตร์ที่ว่าด้วยการทดสอบประโยคหรือเนื้อหาที่เรียกว่า **ประพจน์** เพื่อหาข้อเท็จจริงหรือความสมเหตุสมผลของประพจน์ลักษณะต่างๆ ทั้งที่เป็นประพจน์เชิงเดี่ยว และ ประพจน์เชิงซ้อน
- **ตรรกะที่ว่าด้วยภาคขยาย (Predicate Logic)** เป็นตรรกศาสตร์ที่ทดสอบประโยคกล่าวอ้างที่เกี่ยวข้องกับประโยคทั่วไป และ ประพจน์เชิงเดี่ยว เพื่อหาความสมเหตุสมผลของประโยคกล่าวอ้าง

ประพจน์ (Proposition) เป็นส่วนที่ใช้พิสูจน์ เพื่อบ่งชี้ความจริงตามหลักเหตุผล มี 2 ชนิด คือ

- **ประพจน์เชิงเดี่ยว (Single Proposition)**

  เป็นประโยคหรือเนื้อหาทางตรรกะที่มีเพียงใจความเดียว มีประธานและภาคแสดงเพียงตัวเดียว และไม่สามารถแบ่งย่อยเนื้อหาได้อีก ประพจน์เชิงเดียวมีทั้งหมด 4 รูปแบบคือ

  - **ประพจน์เชิงเดี่ยวแบบยืนยัน** แสดงถึงข้อความทั้งหมดที่สามารถยืนยันได้ ไม่มีข้อความแสดงการปฏิเสธ เช่น นักเรียน**ทุกๆคน**อยากเรียนจบ
  - **ประพจน์เชิงเดี่ยวแบบปฏิเสธ** แสดงถึงข้อความที่มีคำคัดค้านหรือปฏิเสธ เช่น **ไม่มี**นักเรียนคนไหนอยากสอบตก
  - **ประพจน์เชิงเดี่ยวแบบยืนยันบางส่วน** แสดงข้อความยืนยันบางส่วน เช่น นักเรียน**ส่วนใหญ่**เรียน 4 ปีก็จบการศึกษา
  - **ประพจน์เชิงเดี่ยวแบบปฏิเสธบางส่วน** แสดงข้อความปฏิเสธบางส่วน เช่น คนไทย**ส่วนใหญ่ไม่**ยากจน

- **ประพจน์เชิงซ้อน (Compound Proposition)**

  เป็นการนำเอาประพจน์เชิงเดี่ยวหลายประโยคมารวมกันด้วยคำเชื่อมประโยค โดยทั่วไปจะมีตัวเชื่อมอยู่ 4 รูปแบบดังนี้

  - **ประโยคความรวม** เป็นประโยคตรรกะที่เกิดจากคำเชื่อม “และ”, “แต่”, “แม้”, “เมื่อ” ในตรรกะศาสตร์จะใช้ตัว AND ($\wedge$) เช่น ฉันชอบกินข้าวสวย**แต่**เธอชอบกินข้าวเหนียว
  - **ประโยคความเลือก** คือ ประโยคตรรกะที่เกิดจากคำเชื่อม “หรือ” ($\vee$) เช่น พรุ่งนี้เป็นวันพุธ**หรือ**วันพฤหัสบดี
  - **ประโยคมีเงื่อนไข** เป็นประโยคตรรกะที่เกิดจากคำเชื่อม “ถ้า…แล้ว” โดยประพจน์หนึ่งจะเป็นเงื่อนไข อีกตัวจะเป็นผลสรุป ($\Longrightarrow$) เช่น **ถ้า**นักศึกษาทำข้อสอบได้คะแนนเต็ม**แล้ว**จะได้เกรด A
  - **ประโยคสมภาค** คือ ประโยคตรรกะที่เกิดจากคำเชื่อม “..ก็ต่อเมื่อ..” ($\Longleftrightarrow$) เช่น สมชายเป็นคนดี**ก็ต่อเมื่อ**สมชายไม่ทำชั่ว

ประพจน์จะเป็นประโยคที่มีเหตุผลและพิจารณาได้ว่าเป็นจริงหรือเท็จ จึงมีกฎ 3 ข้อดังนี้

- ประพจน์ต้องเป็นจริงหรือเท็จเท่านั้น เช่น กรุงเทพเป็นเมืองหลวงของประเทศไทย
- ประพจน์จะเป็นจริงหรือเท็จพร้อมกันไม่ได้ เช่น กรุงเทพเป็นเมืองหลวงของประเทศไทยตั้งอยู่ที่ภาคเหนือ
- ประพจน์เชิงซ้อน จะหาค่าความจริงโดยรวม เช่น ประเทศไทยอยู่ในทวีปเอเซียมีกรุงเทพเป็นเมืองหลวง

6.2 ตรรกศาสตร์ที่ว่าด้วยประพจน์ (Propositional Logic)

ภาษาทุกภาษาในโลกนี้จะประกอบด้วยองค์ประกอบ 2 ส่วนคือ ไวยกรณ์ของภาษา (Syntax) และความหมายของประโยค (Semantics) ตรรกศาสตร์ที่ว่าด้วยประพจน์ก็ถือว่าเป็นภาษาหนึ่งภาษาเช่นกัน

6.2.1 ไวยกรณ์ของภาษาตรรกศาสตร์ที่ว่าด้วยประพจน์

ไวยกรณ์ของภาษาประกอบด้วยคำนิยามและคำอธิบายต่างๆ ดังนี้

- **Atomic sentences** ประกอบไปด้วย 1 proposition symbol
- แต่ละ proposition symbol จะให้ค่า จริง หรือ เท็จ
- **Symbol** จะใช้ภาษาอังกฤษตัวพิมพ์ใหญ่ เช่น P, Q, R แล้วแต่จะตั้ง
- มี 2 symbol ที่สงวนค่าความเป็นจริงไว้ คือ
    - **True** จะเป็นจริงเสมอ
    - **False** จะเป็นเท็จเสมอ
- **Complex sentences** เกิดจากการนำ sentences มาเชื่อมต่อกันซึ่งมีตัวเชื่อมอยู่ 5 ตัว
    - Not $\neg$
    - And $\wedge$
    - Or $\vee$
    - Implies $\Rightarrow$
    - If and only if $\Leftrightarrow$

ซึ่งสามารถนำไวยกรณ์ของภาษาตรรกศาสตร์ที่ว่าด้วยประพจน์มาเขียนให้อยู่ในรูปของโครงสร้างภาษาแบบ Backus-Naur Form (BNF) ได้ดังแสดงในภาพที่ 6-1

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \textbf{True} \mid \textbf{False} \mid \mathit{Symbol} \\
\mathit{Symbol} &\rightarrow \textbf{P} \mid \textbf{Q} \mid \textbf{R} \mid \ldots \\
\mathit{ComplexSentence} &\rightarrow \neg\, \mathit{Sentence} \\
&\mid\ (\, \mathit{Sentence} \wedge \mathit{Sentence}\,) \\
&\mid\ (\, \mathit{Sentence} \vee \mathit{Sentence}\,) \\
&\mid\ (\, \mathit{Sentence} \Rightarrow \mathit{Sentence}\,) \\
&\mid\ (\, \mathit{Sentence} \Leftrightarrow \mathit{Sentence}\,)
\end{aligned}
$$

ภาพที่ 6-1 โครงสร้างของภาษาตรรกศาสตร์ที่ว่าด้วยประพจน์ในรูปแบบ BNF

6.2.2 ความหมายของประโยคในภาษาตรรกศาสตร์ที่ว่าด้วยประพจน์

ความหมาย (Semantics) ของ Propositional Logic มีเพื่อใช้กำหนดค่าความเป็นจริงให้กับ ประโยค (Sentence)

- **สำหรับ** Atomic Sentence
  - True คือจริงเสมอ
  - False คือเท็จเสมอ
  - Symbol ขึ้นอยู่กับค่าความจริงที่กำหนด
- **สำหรับ** Complex Sentence ให้ถือตามตารางความจริง ดังภาพที่ 6-2

P	Q	¬P	P ∧ Q	P ∨ Q	P ⇒ Q	P ⇔ Q
True	True	False	True	True	True	True
True	False	False	False	True	False	False
False	True	True	False	True	True	False
False	False	True	False	False	True	True

ภาพที่ 6-2 ตารางความเป็นจริงของ Complex Sentence

ตัวอย่างที่ 1 ถ้าฝนตกแล้วจะอยู่บ้าน

- P แทน ฝนตก
- Q แทน อยู่บ้าน
- รูปประโยคจะเป็น P ⇒ Q

ดังนั้นถ้า P เป็นจริง คือฝนตก ประโยคจะให้ค่าเป็นจริง เมื่อ Q เป็นจริง ก็คือ อยู่บ้าน

ตัวอย่างที่ 2 นักศึกษาจะสอบผ่านก็ต่อเมื่อตั้งใจเรียน

- P แทน นักศึกษาจะสอบผ่าน
- Q แทน นักศึกษาตั้งใจเรียน
- รูปประโยคจะเป็น P ⇔ Q

ดังนั้นถ้า Q เป็นจริง คือนักศึกษาตั้งใจเรียน ประโยคจะให้ค่าเป็นจริง เมื่อ P เป็นจริงก็คือนักศึกษาจะสอบผ่าน

6.3 คำนิยามเกี่ยวกับค่าความเป็นจริง

ตำนิยามที่ควรรู้เกี่ยวกับค่าความเป็นจริงของตรรกศาสตร์ว่าด้วยประพจน์ที่สำคัญๆ มีดังนี้

- Tautology เป็นประโยคที่ให้ความเป็นจริงในทุกกรณี เช่น

$$R \Rightarrow ((P \Rightarrow Q) \lor \neg(R \Rightarrow Q))$$

เมื่อเขียนตารางความเป็นจริงเพื่อดูค่าความเป็นจริงของประโยคในทุกกรณี ดังภาพที่ 6-3 โดยจะเห็นได้ว่าผลลัพธ์ของประโยคจะเป็นจริงในทุกกรณี

P	Q	R	$P \Rightarrow Q$	$R \Rightarrow Q$	$\neg(R \Rightarrow Q)$	$(P \Rightarrow Q) \lor \neg(R \Rightarrow Q)$	$R \Rightarrow ((P \Rightarrow Q) \lor \neg(R \Rightarrow Q))$
T	T	T	T	T	F	T	**T**
T	T	F	T	T	F	T	**T**
T	F	T	F	F	T	T	**T**
T	F	F	F	T	F	F	**T**
F	T	T	T	T	F	T	**T**
F	T	F	T	T	F	T	**T**
F	F	T	T	F	T	T	**T**
F	F	F	T	T	F	T	**T**

ภาพที่ 6-3 ตารางความเป็นจริงของประโยคที่เป็น Tautology

- Self-contradiction เป็นประโยคที่ให้ความเป็นเท็จในทุกกรณี เช่น

$$\neg(P \Rightarrow Q) \land \neg(Q \Rightarrow P)$$

เมื่อเขียนตารางความเป็นจริงเพื่อดูค่าความเป็นจริงของประโยคในทุกกรณี ดังภาพที่ 6-4 โดยจะเห็นได้ว่าผลลัพธ์ของประโยคจะเป็นเท็จในทุกกรณี

P	Q	$P \Rightarrow Q$	$\neg(P \Rightarrow Q)$	$Q \Rightarrow P$	$\neg(Q \Rightarrow P)$	$\neg(P \Rightarrow Q) \land \neg(Q \Rightarrow P)$
T	T	T	F	T	F	**F**
T	F	F	T	T	F	**F**
F	T	T	F	F	T	**F**
F	F	T	F	T	F	**F**

ภาพที่ 6-4 ตารางความเป็นจริงของประโยคที่เป็น Self-contradiction

- Contingent เป็นประโยคที่สามารถมีทั้งค่าจริงและเท็จ

เมื่อมีประโยคมากกว่า 1 ประโยค เราสามารถเรียกประโยคทั้ง 2 ประโยคนั้น

- Consistent ก็ต่อเมื่อประโยคเหล่านั้นมีโอกาศที่จะเป็นจริงในกรณีเดียวกัน เช่น

$$(P \vee Q) \text{ และ } \neg(P \Leftrightarrow \neg Q)$$

เมื่อเขียนตารางความเป็นจริงเพื่อดูค่าความเป็นจริงของทั้ง 2 ประโยคในทุกกรณี ดังภาพที่ 6-5 โดยจะเห็นได้ว่าค่าความจริงของประโยคทั้ง 2 มีค่าจริงในกรณีแรกทั้ง 2 ประโยค จึงทำให้ทั้งสองประโยคนี้ Consistent ซึ่งกันและกัน

P	Q	$P \vee Q$	$\neg Q$	$P \Leftrightarrow \neg Q$	$\neg(P \Leftrightarrow \neg Q)$
T	T	**T**	F	F	**T**
T	F	**T**	T	T	**F**
F	T	**T**	F	T	**F**
F	F	**F**	T	F	**T**

consistent

ภาพที่ 6-5 ตารางความเป็นจริงของสองประโยคที่ Consistent กัน

- Inconsistent ก็ต่อเมื่อประโยคเหล่านั้นไม่ Consistent กัน เช่น

$$(P \Rightarrow Q) \wedge P \text{ และ } \neg(Q \vee \neg P)$$

ซึ่งมีตารางความเป็นจริงของประโยคทั้ง 2 ประโยคในทุกกรณี ดังภาพที่ 6-6

P	Q	$P \Rightarrow Q$	$(P \Rightarrow Q) \wedge P$	$\neg P$	$Q \vee \neg P$	$\neg(Q \vee \neg P)$
T	T	T	**T**	F	T	**F**
T	F	F	**F**	F	F	**T**
F	T	T	**F**	T	T	**F**
F	F	T	**F**	T	T	**F**

ภาพที่ 6-6 ตารางความเป็นจริงของสองประโยคที่ Inconsistent กัน

- Logically equivalent ก็ต่อเมื่อค่าความเป็นจริงของทั้ง 2 ประโยคเหมือนกันในทุกกรณี เช่น $\neg P \Rightarrow \neg Q$ และ $\neg(Q \wedge \neg P)$ ที่มีค่าความจริงแสดงดังภาพที่ 6-7

P	Q	$\neg P$	$\neg Q$	$\neg P \Rightarrow \neg Q$	$Q \wedge \neg P$	$\neg(Q \wedge \neg P)$
T	T	F	F	**T**	F	**T**
T	F	F	T	**T**	F	**T**
F	T	T	F	**F**	T	**F**
F	F	T	T	**T**	F	**T**

ภาพที่ 6-7 ตารางความเป็นจริงของสองประโยคที่ Logically equivalent กัน

6.4 กฎของการแทนที่ (Rules of Replacement)

ประโยคทางตรรกศาสตร์สามารถแทนที่กันได้ถ้าทั้ง 2 ประโยคนั้น Logically Equivalent ซึ่งกันและกัน ดังนั้นจึงมีกฎการแทนที่ขึ้นเพื่อแปลงประโยคทางตรรกศาสตร์ให้อยู่ในรูปแบบที่เหมาะสมขึ้น โดยกฎการแทนที่ทั้งหมด แสดงดังภาพที่ 6-8

ชื่อกฎ	Logically equivalent
Double negation (DN)	$\neg\neg P \equiv P$
Commutativity (Com)	$P \vee Q \equiv Q \vee P$ $P \wedge Q \equiv Q \wedge P$
Associativity (Assoc)	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$ $(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$
Tautology (Taut)	$P \vee P \equiv P$, $P \wedge P \equiv P$
Demorgan's Law (DM)	$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$ $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
Transposition (Trans)	$P \Rightarrow Q \equiv \neg Q \Rightarrow \neg P$
Material Implication (Impl)	$P \Rightarrow Q \equiv \neg P \vee Q$
Exportation (Exp)	$P \Rightarrow (Q \Rightarrow R) \equiv (P \wedge Q) \Rightarrow R$
Distribution (Dist)	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$ $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
Material Equivalent (Equiv)	$P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$ $\equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$

ภาพที่ 6-8 รายการกฎของการแทนที่

การเปรียบเทียบความเท่ากันทางตรรกศาสตร์ (Logically Equivalent) นั้นสามารถทำได้ด้วยการใช้ตารางความเป็นจริง เช่น $\neg(P \Rightarrow (Q \wedge R)) \equiv \neg[(P \Rightarrow Q) \wedge (P \Rightarrow R)]$ ดังภาพที่ 6-9

P	Q	R	$(Q \wedge R)$	$P \Rightarrow (Q \wedge R)$	$\neg(P \Rightarrow (Q \wedge R))$	$(P \Rightarrow Q)$	$(P \Rightarrow R)$	$(P \Rightarrow Q) \wedge (P \Rightarrow R)$	$\neg[(P \Rightarrow Q) \wedge (P \Rightarrow R)]$
T	T	T	T	T	**F**	T	T	T	**F**
T	T	F	F	F	**T**	T	F	F	**T**
T	F	T	F	F	**T**	F	T	F	**T**
T	F	F	F	F	**T**	F	F	F	**T**
F	T	T	T	T	**F**	T	T	T	**F**
F	T	F	F	T	**F**	T	T	T	**F**
F	F	T	F	T	**F**	T	T	T	**F**
F	F	F	F	T	**F**	T	T	T	**F**

ภาพที่ 6-9 ตารางความเป็นจริงเพื่อเปรียบเทียบความเท่ากันทางตรรกศาสตร์

แทนที่จะเปรียบเทียบความเท่ากันทางตรรกศาสตร์ด้วยตารางความเป็นจริงนั้น ยังสามารถพิสูจน์ได้จากการแทนที่กฎด้วย เนื่องจากเมื่อมีจำนวนสัญลักษณ์มากขึ้น เช่น 10 สัญลักษณ์ จะทำให้กรณีที่ต้องเขียนในตารางความเป็นจริงต้องเขียนถึง 2^{10} กรณี (1024 กรณี) ซึ่งจะลำบากและยุ่งยากมากขึ้น ดังนั้นการแทนที่ด้วยกฎจึงนำมาเติมเต็มในส่วนนี้ จากตัวอย่างสมการเดียวกันกับพิสูจน์ด้วยตารางความเป็นจริงในภาพที่ 6-9 สามารถนำมาใช้วิธีแทนที่กฎได้ดังนี้

$\neg(P \Rightarrow (Q \land R))$	$\equiv \neg(\neg P \lor (Q \land R))$	Impl
	$\equiv P \land \neg(Q \land R)$	DM
	$\equiv P \land (\neg Q \lor \neg R)$	DM
	$\equiv (P \land \neg Q) \lor (P \land \neg R)$	Dist
	$\equiv (\neg(\neg P) \land \neg Q) \lor (\neg(\neg P) \land \neg R)$	DN
	$\equiv \neg(\neg P \lor Q) \lor \neg(\neg P \lor R)$	DM
	$\equiv \neg(\neg P \lor Q) \lor \neg(\neg P \lor R)$	DM
	$\equiv \neg[(\neg P \lor Q) \land (\neg P \lor R)]$	DM
	$\equiv \neg[(P \Rightarrow Q) \land (P \Rightarrow R)]$	Impl

6.5 กฎของการอนุมาน (Rules of Inference)

การอนุมานด้วยวิธีการให้เหตุผลจะต้องมีการตรวจสอบความสมเหตุสมผล กฎของการอนุมานเชิงตรรกศาสตร์ส่วนใหญ่ประกอบด้วย 9 กฎ

6.5.1 Modus Ponen (MP) หรือ (⇒-elimination)

เป็นกฎที่ใช้เพื่อกำจัดตัว $\Rightarrow$ ถ้า $P \Rightarrow Q$ เป็นจริง และ P เป็นจริง จะอนุมานได้ว่า Q เป็นจริง ดังแสดงในภาพที่ 6-10

$$\frac{\begin{array}{c} P \Rightarrow Q \\ P \end{array}}{Q}$$

ภาพที่ 6-10 กฎการอนุมาน Modus Ponen

6.5.2 Modus Tollens (MT) หรือ ($\Rightarrow$-elimination)

เป็นกฎที่ใช้เพื่อกำจัดตัว $\Rightarrow$ ถ้า P $\Rightarrow$ Q เป็นจริง และ $\neg$Q เป็นจริง จะอนุมานได้ว่า $\neg$P เป็นจริง ดังแสดงในภาพที่ 6-11

$$\begin{array}{l} P \Rightarrow Q \\ \underline{\neg Q} \\ \neg P \end{array}$$

ภาพที่ 6-11 กฎการอนุมาน Modus Tollens

6.5.3 Disjunctive Syllogism (DS) หรือ ($\vee$-elimination)

เป็นกฎที่ใช้เพื่อกำจัดตัว $\vee$ ถ้า P $\vee$ Q เป็นจริง และ $\neg$P เป็นจริง จะอนุมานได้ว่า Q เป็นจริง หรือถ้า P $\vee$ Q เป็นจริง และ $\neg$Q เป็นจริง จะอนุมานได้ว่า P เป็นจริง ดังแสดงในภาพที่ 6-12

$$\begin{array}{l} P \vee Q \\ \underline{\neg P} \\ Q \end{array} \qquad \begin{array}{l} P \vee Q \\ \underline{\neg Q} \\ P \end{array}$$

ภาพที่ 6-12 กฎการอนุมาน Disjunctive Syllogism

6.5.4 Addition (Add) หรือ ($\vee$-introduction)

เป็นกฎที่ใช้เพื่อเพิ่มตัว $\vee$ ถ้า P เป็นจริงแล้ว P $\vee$ Q ก็ต้องเป็นจริงด้วย หรือ ถ้า Q เป็นจริงแล้ว P $\vee$ Q ก็ต้องเป็นจริงด้วย ดังแสดงในภาพที่ 6-13

$$\begin{array}{c} \underline{P} \\ P \vee Q \end{array} \qquad \begin{array}{c} \underline{Q} \\ P \vee Q \end{array}$$

ภาพที่ 6-13 กฎการอนุมาน Addition

6.5.5 Simplification (Simp) หรือ (∧-elimination)

เป็นกฎที่ใช้เพื่อกำจัดตัว ∧ ถ้า P ∧ Q เป็นจริงแล้ว P ต้องเป็นจริง หรือถ้า P ∧ Q เป็นจริงแล้ว Q ต้องเป็นจริงเสมอ ดังแสดงในภาพที่ 6-14

$$\frac{P \wedge Q}{P} \qquad \frac{P \wedge Q}{Q}$$

ภาพที่ 6-14 กฎการอนุมาน Simplification

6.5.6 Conjunction (Conj) หรือ (∧-introduction)

เป็นกฎที่ใช้เพื่อเพิ่มตัว ∧ ถ้า P เป็นจริง และ Q เป็นจริงแล้ว P ∧ Q จะเป็นจริง ดังแสดงในภาพที่ 6-15

$$\frac{\begin{array}{c} P \\ Q \end{array}}{P \wedge Q}$$

ภาพที่ 6-15 กฎการอนุมาน Conjunction

6.5.7 Hypothetical syllogism (HS) หรือ (Chain Reasoning)

ถ้า P ⇒ Q เป็นจริงและ Q ⇒ R เป็นจริงแล้ว P ⇒ R จะเป็นจริง ดังภาพที่ 6-16

$$\frac{\begin{array}{c} P \Rightarrow Q \\ Q \Rightarrow R \end{array}}{P \Rightarrow R}$$

ภาพที่ 6-16 กฎการอนุมาน Hypothetical syllogism

6.5.8 Constructive Dilemma (CD)

ถ้า $(P \Rightarrow Q) \wedge (R \Rightarrow S)$ เป็นจริงและ $P \vee R$ เป็นจริงแล้ว $Q \vee S$ จะเป็นจริง ดังภาพที่ 6-17

$$\frac{\begin{array}{l}(P \Rightarrow Q) \wedge (R \Rightarrow S) \\ P \vee R\end{array}}{Q \vee S}$$

ภาพที่ 6-17 กฎการอนุมาน Constructive Dilemma

6.5.9 Absorption (Abs)

ถ้า $(P \Rightarrow Q)$ เป็นจริงแล้ว $P \Rightarrow (P \wedge Q)$ จะเป็นจริง ดังภาพที่ 6-18

$$\frac{(P \Rightarrow Q)}{P \Rightarrow (P \wedge Q)}$$

ภาพที่ 6-18 กฎการอนุมาน Absorption

6.6 การนิรนัยหาข้อสรุป

การนิรนัยหาข้อสรุปแบ่งออกเป็นวิธีใหญ่ๆ 2 วิธีคือ การนิรนัยทางตรง (Direct deduction) และการนิรนัยทางอ้อม (Indirect deduction)

6.6.1 Direct deduction

เป็นการนิรนัยแบบตรงเพื่อหาข้อสรุปจากเซ็ตของสมมุติฐาน ที่ประกอบไปด้วยลำดับของประโยคสมมุติฐาน (premise) ประโยคที่มาจากกฎของการอนุมาน (rules of inference) ประโยคที่มาจากกฎของการแทนที่ (rules of replacement)

ตัวอย่างที่ 1: การใช้ direct deduction เพื่อพิสูจน์ความจริงของข้อสรุป M จากสมมุติฐานคือ $C \vee D$, $C \Rightarrow O$, $D \Rightarrow M$, และ $\neg O$ มีขั้นตอนดังนี้

วิธีที่ 1

1.	$C \vee D$	premise
2.	$C \Rightarrow O$	premise
3.	$D \Rightarrow M$	premise
4.	$\neg O$	premise
5.	$\neg C$	2,4 MT
6.	D	1,5 DS
7.	M	3,6 MP

วิธีที่ 2

1.	$C \vee D$	premise
2.	$C \Rightarrow O$	premise
3.	$D \Rightarrow M$	premise
4.	$\neg O$	premise
5.	$(C \Rightarrow O) \wedge (D \Rightarrow M)$	2,3 Conj
6.	$O \vee M$	1,5 CD
7.	M	4,6 DS

ทั้ง 2 วิธีข้างต้นสามารถสรุป M ว่าเป็นจริงได้ทั้งสองวิธีจึงสรุปได้ว่า จากสมมุติฐานที่ให้มา ข้อสรุป M เป็นข้อสรุปที่จริง

ตัวอย่างที่ 2: พิสูจน์ $T \Rightarrow U$ จากสมมุติฐาน $P \Leftrightarrow Q$, $(S \vee T) \Rightarrow Q$, และ $\neg P \vee (\neg T \wedge R)$ มีขั้นตอนการพิสูจน์ดังนี้

1.	$P \Leftrightarrow Q$	premise
2.	$(S \vee T) \Rightarrow Q$	premise
3.	$\neg P \vee (\neg T \wedge R)$	premise
4.	$(P \Rightarrow Q) \wedge (Q \Rightarrow P)$	1 Equiv
5.	$(Q \Rightarrow P)$	4 Simp
6.	$(S \vee T) \Rightarrow P$	2,4 HS

7. $P \Rightarrow (\neg T \wedge R)$ 3 Impl
8. $(S \vee T) \Rightarrow (\neg T \wedge R)$ 6,7 HS
9. $\neg(S \vee T) \vee (\neg T \wedge R)$ 8 Impl
10. $(\neg S \wedge \neg T) \vee (\neg T \wedge R)$ 9 DM
11. $[(\neg S \wedge \neg T) \vee \neg T] \wedge [(\neg S \wedge \neg T) \vee R]$ 10 Dist
12. $(\neg S \wedge \neg T) \vee \neg T$ 11 Simp
13. $(\neg S \vee \neg T) \wedge (\neg T \vee \neg T)$ 12 Dist
14. $\neg T \vee \neg T$ 13 Simp
15. $\neg T$ 14 Taut
16. $\neg T \vee U$ 15 Add
17. $T \Rightarrow U$ 16 Impl

6.6.2 Indirect deduction

การนิรนัยทางอ้อมหรือ Indirect deduction มี2 วิธีคือ

- **Conditional proof:** สมมุติค่าความเป็นจริงให้ 1 ตัวประเพื่อแก้ปัญหา
- **Indirect proof:** กำหนดข้อสรุปเป็นเท็จ แล้วถ้ามีพิสูจน์หักล้างได้ จะทำให้ข้อสรุปนั้นเป็นจริง

ตัวอย่างที่ 3: พิสูจน์ข้อสรุป $\neg P$ จากสมมุติฐานคือ $P \Rightarrow Q$ และ $P \Rightarrow (Q \Rightarrow \neg P)$ ด้วยวิธี Indirect proof

1. $P \Rightarrow Q$ premise
2. $P \Rightarrow (Q \Rightarrow \neg P)$ premise
3. P assumption
4. Q 1,3 MP
5. $(Q \Rightarrow \neg P)$ 2,3 MP
6. $\neg P$ 5,4 MP
7. $P \wedge \neg P$ 3,6 Conj
8. False 7 IP

เมื่อพิสูจน์ได้ False หมายความว่าสมมุติฐานที่กำหนด P ผิด ซึ่งก็คือ P ไม่เป็นไร เพราะฉนั้นจึงสรุปได้ว่า $\neg P$ เป็นจริง

ตัวอย่างที่ 4: จงพิสูจน์ว่าคำกล่าวต่อไปนี้ถูกต้อง ถ้าอุณหภูมิและความดันคงที่ฝนจะไม่ตก ขณะนี้อุณหภูมิคงที่ ดังนั้นถ้าฝนตกแล้วหมายความว่าความดันไม่คงที่

วิธีพิสูจน์ กำหนด

- A แทน อุณหภูมิคงที่
- B แทน ความดันคงที่
- C แทน ฝนตก

แทนประโยคด้วย propositional logic

- Premise : ถ้าอุณหภูมิและความดันคงที่ฝนจะไม่ตก $(A \land B) \Rightarrow \neg C$
- Premise : ขณะนี้อุณหภูมิคงที่ A
- Conclusion : ถ้าฝนตกแล้วหมายความว่าความดันไม่คงที่ $C \Rightarrow \neg B$

ขั้นตอนการพิสูจน์

1. $(A \land B) \Rightarrow \neg C$	Premise
2. A	Premise
3. $\neg(C \Rightarrow \neg B)$	Assumption
4. $\neg(A \land B) \lor \neg C$	1 Impl
5. $\neg A \lor \neg B \lor \neg C$	4 DM
6. $\neg B \lor \neg C$	2,5 DS
7. $\neg C \lor \neg B$	6 Comm
8. $C \Rightarrow \neg B$	7 Impl
9. $\neg(C \Rightarrow \neg B) \land C \Rightarrow \neg B$	3,8 Conj
10. False	9 IP

6.7 สรุป

ตรรกะ คือ ศาสตร์ที่ว่าด้วยการหาเหตุและผลด้วยวิธีการต่างๆ อย่างมีรูปแบบและระบบที่ชัดเจน ซึ่งอ้างอิงตามหลักของเหตุและผล เป็นศาสตร์หนึ่งที่มีความสำคัญและเป็นพื้นฐานของการสร้างองค์ความรู้ให้แก่ปัญญาประดิษฐ์ เพื่อให้สามารถตอบสนองต่อการเรียนรู้และตัดสินใจอย่างมีเหตุและผลเลียนแบบหลักการคิดของมนุษย์ โดยส่วนใหญ่นิยมใช้ ตรรกสัญลักษณ์ ที่สามารถเขียนตัวเชื่อมประโยค รวมทั้งมีการใช้กฎของการแทนที่ และกฎของการอนุมาน เพื่อสรุปหาความเป็นจริงของข้อสรุปได้

แบบฝึกหัด

1. กำหนดให้ p แทนประโยค "ฉันซื้อฉลากกินแบ่ง" และ q แทนประโยค "ฉันถูกรางวัล" จงเขียนรูปประโยคภาษาไทยเมื่อกำหนดตรรกศาสตร์ที่ว่าด้วยประพจน์ดังนี้

   1.1 $\neg p$

   1.2 $p \vee q$

   1.3 $p \wedge q$

   1.4 $p \Rightarrow q$

   1.5 $\neg p \Rightarrow \neg q$

   1.6 $\neg p \vee (p \wedge q)$

2. จงเขียนตารางความเป็นจริงจากประพจน์ต่อไปนี้และบอกด้วยว่าประพจน์นี้เป็น Tautology, Self-contradiction หรือ Contingent

   2.1 $p \wedge \neg p$

   2.2 $p \vee \neg p$

   2.3 $(p \vee \neg q) \Rightarrow q$

   2.4 $(p \vee q) \Rightarrow (p \wedge q)$

   2.5 $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$

   2.6 $(p \Rightarrow q) \Rightarrow (q \Rightarrow p)$

3. จงเปรียบเทียบความเท่ากันทางตรรกศาสตร์ของประพจน์

   $\neg (P \wedge (Q \vee R \vee S))$ และ $\neg P \vee (\neg Q \wedge \neg R \wedge \neg S)$

   ด้วยการใช้กฎการแทนที่

4. จากประโยคคำพูดต่อไปนี้

   รูปภาพมีอยู่ 2 ประเภทคือแบบกลมและแบบสี่เหลี่ยม รูปภาพจะเป็นภาพสีหรือภาพขาวดำ สมชายเจอรูปภาพเมื่อวานนี้ ถ้ารูปภาพเป็นแบบสี่เหลี่ยมแล้วจะเป็นภาพขาวดำ ถ้ารูปภาพเป็นแบบกลมจะเป็นภาพดิจิตอลแบบมีสี ถ้ารูปภาพเป็นดิจิตอลหรือแบบขาวดำแล้วมันคือรูปคน ถ้าเป็นรูปภาพคนแล้วมันคือภาพของเพื่อน

   จงพิสูจน์ว่ารูปภาพที่สมชายเจอเป็นรูปภาพของเพื่อนด้วยวิธี Indirect Proof

แผนปฏิบัติการสอนสัปดาห์ที่ 7

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง ตรรกศาสตร์ที่ว่าด้วยภาคขยายและการอนุมาน **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. องค์ประกอบพื้นฐานตรรกศาสตร์ว่าด้วยภาคขยาย	IS 7.1
2. ตัวเชื่อมและตัวบ่งปริมาณของตรรกศาสตร์ที่ว่าด้วยภาคขยาย	IS 7.2
3. ความสัมพันธ์ของตัวบ่งปริมาณ	IS 7.3
4. กฎของการอนุมาน	IS 7.4
5. การแทนค่าและการทำให้เท่ากัน	IS 7.5
6. รีโซลูชัน	IS 7.6
7. ตัวอย่างการหาสรุปด้วยวิธีรีโซลูชัน	IS 7.7

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2-3-4-5-6-7**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 7
ตรรกศาสตร์ที่ว่าด้วยภาคขยายและการอนุมาน

จากบทที่แล้วเป็นการเรียนรู้เกี่ยวกับตรรกศาสตร์ที่ว่าด้วยประพจน์ (Propositional Logic) ซึ่งเป็นการแทนประโยคด้วยสัญลักษณ์ เช่น สมชายเป็นคน แทนด้วยตรรกสัญลักษณ์ P, สมปองเป็นคน แทนด้วยตรรกสัญลักษณ์ Q, หรือ สมหญิงเป็นคน แทนด้วยตรรกสัญลักษณ์ R ซึ่งจะเห็นได้ว่าจะต้องใช้สัญลักษณ์ 1 ตัวเพื่อแทนประโยคหนึ่งประโยค ไม่สามารถจะเข้าถึงค่าที่มีคุณลักษณ์เหมือนกันเฉพาะตัวได้ ดังนั้น ตรรกศาสตร์ที่ว่าด้วยภาคขยายหรือตรรกะเพรดิเคต (Predicate calculus หรือ Predicate Logic) เป็นตรรกศาสตร์ระดับสูงกว่าตรรกศาสตร์ที่ว่าด้วยประพจน์ (ส่วนใหญ่จะใช้ First-Order Logic) ที่สามารถทำให้ประโยคที่มีคุณลักษณะเหมือนกันเข้าถึงค่าเฉพาะแต่ละตัวได้ เช่น การกำหนด Predicate ชื่อ Human( ) แทนการเป็นคน จากนั้นสามารถแทนประโยค สมชายเป็นคน, สมปองเป็นคน, และ สมหญิงเป็นคน ได้ด้วยตรรกสัญลักษณ์ Human(สมชาย), Human(สมปอง) และ Human(สมหญิง) ตามลำดับ

7.1 องค์ประกอบพื้นฐานตรรกศาสตร์ว่าด้วยภาคขยาย

ตรรกะเพรดิเคตเป็นภาษาหนึ่งซึ่งมีวากยสัมพันธ์ (syntax) และความหมาย (semantics) ของภาษาที่กำหนดขึ้นอย่างชัดเจน เราเรียกสูตรที่ถูกต้องตามวากยสัมพันธ์ของภาษานี้ว่าสูตรรูปดี (well form formular) ภาษานี้ประกอบด้วยสัญลักษณ์พื้นฐานดังนี้ คือ

- สัญลักษณ์แสดงเพรดิเคต (predicate symbol) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไป และขึ้นต้นด้วยตัวอักษรใหญ่ เช่น P, Q, R เป็นต้น
- สัญลักษณ์แสดงตัวแปร (variable symbol) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไป และขึ้นต้นด้วยตัวอักษรเล็ก เช่น x, y, z เป็นต้น
- สัญลักษณ์แสดงฟังก์ชัน (function symbol) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไป และขึ้นต้นด้วยตัวอักษรเล็ก เช่น f, g, h เป็นต้น
- สัญลักษณ์แสดงค่าคงที่ (constant symbol) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไป และขึ้นต้นด้วยตัวอักษรใหญ่ เช่น A, B, C เป็นต้น
- เครื่องหมายวงเล็บ เช่น [ ] , { } , ( ) เป็นต้น เครื่องหมายวงเล็บทั้ง 3 ประเภทนั้นไม่มีความแตกต่างกันทางความหมาย

เพรดิเคตเป็นสัญลักษณ์ที่ใช้แสดงความสัมพันธ์ในโดเมนที่กล่าวถึง เช่น

FATHER (SOMCHAI , SOMSRI)

สูตรข้างต้นเรียกว่า สูตรอะตอม (Atomic formula) ซึ่งเป็นสูตรรูปดีที่เล็กที่สุดที่ถูกต้องตามวากยสัมพันธ์ของภาษานี้ สูตรอะตอมประกอบด้วยสัญลักษณ์เพรดิเคตในที่นี้คือ ‘ซึ่งแสดงความสัมพันธ์ ‘พ่อ’ ความสัมพันธ์นี้จะรับอาร์กิวเมนต์ (argument) 2 ตัว คือ ‘SOMCHAI’ และ ‘SOMSRI’ ซึ่งอาร์กิวเมนต์ทั้งสองนี้เป็นค่าคงที่ สูตรอะตอมที่ถูกต้องตามวากยสัมพันธ์จะต้องประกอบด้วยเพรดิเคตและอาร์กิวเมนต์ตั้งแต่ 0 ตัวขึ้นไป โดยที่อาร์กิวเมนต์แต่ละตัวขั้นด้วยเครื่องหมาย ‘ , ’ และอาร์กิวเมนต์ทั้งหมดต้องถูกคลุมด้วยเครื่องหมายวงเล็บ ลำดับของอาร์กิวเมนต์มีความสำคัญ ผู้เขียนต้องกำหนดลำดับเอง ในตัวอย่างนี้เราต้องการให้มีความหมายว่า คน 2 คน ในโดเมนนี้ที่ชื่อ ‘SOMCHAI’ และ ‘SOMSRI’ มีความสัมพันธ์กันโดยที่ ‘SOMCHAI’ เป็นพ่อของ ‘SOMSRI’

ภาษานี้ใช้ตัวแปรเพื่อให้แทนถึงค่าคงที่ใด ๆ ได้ และสามารถใช้ฟังก์ชันเพื่อระบุหาพจน์ (term) หนึ่งจากพจน์อื่น ๆ ได้ ดังแสดงในตัวอย่างด้านล่างนี้ตามลำดับ

FATHER (x, y)

x และ y แทนตัวแปร ซึ่งสูตรนี้แสดงว่า x เป็นพ่อของ y

HAS – MONEY (SOMCHAI , salary (SOMCHAI))

สูตรนี้แสดงความสัมพันธ์ ‘HAS – MONEY’ และมีฟังก์ชัน ‘salary’ ที่ระบุหาพจน์จาก ‘SOMCHAI’ ซึ่งพจน์นี้อาจเป็นค่างคงที่ค่าหนึ่ง เช่น 10850 เป็นต้น

การแปลความหมาย (interpretation) คือการกำหนดค่าให้กับเพรดิเคต ค่าคงที่ ตัวแปร ฟังก์ชันในโดเมนนั้น ซึ่งการกำหนดค่าเหล่านี้จะเป็นค่านิยามความหมายของสูตรอะตอมเช่นการแปลความหมายของสูตรที่ (3) คือการกำหนดให้ ‘SOMCHAI’ มีค่าคือ คนที่ชื่อสมชาย, กำหนดให้ ‘salary(SOMCHAI)’ มีค่าเท่ากับ 10850 บาท และกำหนดให้ ‘HAS – MONEY’ มีค่าเป็นการที่สมชายมีเงินเท่ากับ 10850 บาท เป็นต้น เมื่อเรานิยามการแปลความหมายแล้ว เราสามารถบอกได้ว่าสูตรอะตอมตัวหนึ่ง ๆ มีค่าเป็น T (จริง) ถ้าความสัมพันธ์ที่แสดงในสูตรนั้นเป็นจริงในโดเมนที่กล่าวถึง และจะมีค่าเป็น F (เท็จ) ถ้าไม่ใช่

ตัวเชื่อมและตัวบ่งปริมาณ

7.2 ตัวเชื่อมและตัวบ่งปริมาณของตรรกศาสตร์ที่ว่าด้วยภาคขยาย

ภาษานี้มีตัวเชื่อม (connective) และตัวบ่งปริมาณ (quantifier) เพื่อใช้เขียนความสัมพันธ์ได้ซับซ้อนยิ่งขึ้น ตรงกับความต้องการของผู้ใช้มากขึ้น

7.2.1 ตัวเชื่อมภาษา

ตัวเชื่อมที่มีในภาษานี้เหมือนกับตัวเชื่อมในตรรกศาสตร์แบบประพจน์ ซึ่งได้แก่

- และ แทนด้วยเครื่องหมาย $\wedge$
- หรือ แทนด้วยเครื่องหมาย $\vee$
- ถ้าแล้ว แทนด้วยเครื่องหมาย $\Rightarrow$
- นิเสธ แทนด้วยเครื่องหมาย $\neg$ หรือ ~

ตัวเชื่อมเหล่านี้ทำหน้าที่เชื่อมสูตรอะตอมหลายตัวเข้าด้วยกัน เพื่อสร้างเป็นสูตรใหม่ที่ซับซ้อนยิ่งขึ้น ตัวอย่างเช่น ถ้าต้องการเขียนสูตรแทนประโยค ‘John lives in a yellow house.’ ก็อาจเขียนได้โดยใช้ตัวเชื่อม $\wedge$ ได้ดังนี้

LIVE(JOHN , HOUSE) $\wedge$ COLOR(HOUSE, YELLOW)

ตัวเชื่อม $\Rightarrow$ ใช้เขียนประโยคเงื่อนไข (if... then ... เป็นต้น) เช่น ถ้าต้องการเขียนสูตรแทนประโยค ‘if the car belongs to John then it is green,’ ก็อาจเขียนได้โดยใช้ตัวเชื่อม $\Rightarrow$ ดังนี้

OWNS (JOHN, CAR) $\Rightarrow$ COLOR (CAR, GREEN)

ตัวเชื่อม $\neg$ ใช้เปลี่ยนค่าความจริงของสูตรแทนประโยค ‘John did not write computer chess.’ ก็เขียนได้ดังนี้

$\neg$WRITE (JOHN , COMPUTER-CHESS)

เมื่อกำหนดการแปลความหมายของสูตรอะตอมแล้ว ค่าความจริงของสูตรที่ประกอบด้วยตัวเชื่อมสามารถหาได้โดยใช้ตารางค่าความจริงเช่นเดียวกันกับตรรกศาสตร์ว่าด้วยประพจน์

7.2.2 ตัวบ่งปริมาณ

นอกจากตัวเชื่อมแล้ว ภาษานี้ยังสามารถรองรับการบ่งบอกปริมาณในการเขียนสูตรได้ ตัวบ่งปริมาณที่มีให้ในภาษานี้ ได้แก่

- ตัวบ่งปริมาณเอกภพ (universal quantifier) แทนด้วยเครื่องหมาย $\forall$
- ตัวบ่งปริมาณมีอยู่ (existential quantifier) แทนด้วยเครื่องหมาย $\exists$

ตัวอย่างการใช้ตัวบ่งปริมาณ เช่นต้องการเขียนประโยค ‘All elephants are gray.’ ก็สามารถแสดงได้โดยสูตรด้านล่างนี้

$$\forall x\ (ELEPHANT\ (x) \Longrightarrow COLOR\ (x\ ,\ GRAY))$$

หรือในกรณีของประโยค ‘There is a person who wrote computer-chess.’ เขียนแทนด้วยสูตรด้านล่างนี้

$$\exists x\ (WRITE(x,\ COMPUTER\text{-}CHESS))$$

ปัญหาหนึ่งที่เกิดขึ้นในกรณีที่มีตัวบ่งปริมาณปรากฏในสูตรก็คือ เราอาจค่าความจริงของสูตรนั้นไม่ได้ เช่นกำหนดให้สูตรเป็น ‘$\forall x\ (P(x))$’ และเมื่อให้การแปลความหมายของ ‘P’ และให้โดเมนของ x เป็นโดเมนอนันต์ เช่น x แทนเลขจำนวนจริง บางครั้งเราอาจหาค่าความจริงของสูตรนี้ไม่ได้ เนื่องจากเราไม่สามารถนำเลขจำนวนจริงมาทดสอบสูตรนี้ได้จนหมด แต่ถ้าหากว่ามีจำนวนจริงตัวใดตัวหนึ่งที่ทำให้สูตรเป็นเท็จก็จะสรุปได้ว่าสูตรเป็นเท็จแต่ถ้าหากว่าตราบเท่าที่จำนวนจริงที่นำมาทดสอบยังให้ค่าเป็นจริงอยู่ก็จะยังคงสรุปไม่ได้

ตรรกะว่าด้วยส่วนขยายอันดับที่หนึ่ง หรือตรระกะเพรดิเคตอันดับที่หนึ่ง (first-order predicate logic) คือตรรกะเพรดิเคตที่ไม่มีตัวบ่งปริมาณของสัญลักษณ์แสดงเพรดิเคตหรือของสัญลักษณ์แสดงฟังก์ชัน ในบทนี้เราจะสนใจเฉพาะตรรกะเพรดิเคตอันดับที่หนึ่งเท่านั้น ไม่สนใจตรรกะเพรดิเคตอันดับสูง (high order predicate logic) เนื่องจากตรรกะเพรดิเคตอันดับที่หนึ่งก็สามารถใช้เขียนความสัมพันธ์แทนความรู้ต่างๆ ได้อย่างกว้างขวางมาก และการคำนวณเชิงตรรกะของการแทนความรู้ประเภทนี้ก็ไม่สูงเกินไปนัก ตัวอย่างของสูตรที่เป็นตรรกะเพรดิเคตอันดับที่หนึ่ง ได้แก่ประโยค $\forall x\ (ELEPHANT\ (x) \Longrightarrow COLOR\ (x\ ,\ GRAY))$ และประโยค $\exists x\ (WRITE(x,\ COMPUTER\text{-}CHESS))$ ที่ได้กล่าวถึงข้างต้นเป็นต้น สำหรับตัวอย่างประโยคที่ไม่เป็นตรรกะเพรดิเคตอันดับที่หนึ่งก็เช่น $\forall x \forall y\ (y(x) \Rightarrow COLOR(x,\ GRAY))$ เป็นต้น สังเกตว่าในกรณีนี้ ‘y’ เป็นตัวแปรแสดงเพรดิเคต ซึ่งในกรณีเช่นนี้การแปลความหมายของสูตรนี้จะมีความยุ่งยากซับซ้อนมากขึ้น คือเราต้องหาทุกความสัมพันธ์‘y’ เพื่อนำมาตรวจสอบหาค่าความจริงของสูตร‘$y(x) \Rightarrow COLOR(x,GRAY)$’

7.3 ความสัมพันธ์ของตัวบ่งปริมาณ

การใช้ตัวบ่งปริมาณนั้นมีข้อควรระวังเนื่องด้วยถ้าใช้งานผิดจะทำให้ความหมายของประโยคทางตรรกศาสตร์เปลี่ยนไปคือถ้ามีการใช้ตัวบ่งปริมาณ$\forall$ ประโยคหลักที่ตามมาจะใช้ตัวเชื่อม $\Rightarrow$ แต่ถ้าใช้ตัวบ่งปริมาณ $\exists$ ตัวเชื่อมในประโยคหลักจะใช้ $\wedge$

ตัวอย่างประโยค 1: *นักศึกษาที่เรียน ECT ทุกคนจะเก่ง*

- $\forall$x(At(x, ECT) $\wedge$ smart(x)) ผิด
- $\forall$x(At(x, ECT) $\Longrightarrow$ smart(x)) **ถูก**

ตัวอย่างประโยค 2: *นักศึกษาบางคนที่เรียน ECT จะเก่ง*

- $\exists$x(At(x, ECT) $\Longrightarrow$ smart(x)) ผิด
- $\exists$x(At(x, ECT) $\wedge$ smart(x)) **ถูก**

การใช้ตัวบ่งปริมาณสามารถเปลี่ยนรูปประโยคไปมาระหว่างตัว $\forall$ และ $\exists$ ได้ด้วยกฎความสัมพันธ์ดังนี้ไปนี้

$$\neg(\forall x)A(x) \equiv (\exists x)\neg A(x)$$
$$\neg(\exists x)A(x) \equiv (\forall x)\neg A(x)$$
$$(\forall x)A(x) \wedge B \equiv (\forall x)(A(x) \wedge B)$$
$$(\forall x)A(x) \vee B \equiv (\forall x)(A(x) \vee B)$$
$$(\exists x)A(x) \wedge B \equiv (\exists x)(A(x) \wedge B)$$
$$(\exists x)A(x) \vee B \equiv (\exists x)(A(x) \vee B)$$
$$(\forall x)A(x) \wedge (\forall y)B(y) \equiv (\forall x)(\forall y)(A(x) \wedge B(y))$$
$$(\forall x)A(x) \wedge (\exists y)B(y) \equiv (\forall x)(\exists y)(A(x) \wedge B(y))$$
$$(\exists x)A(x) \wedge (\exists y)B(y) \equiv (\exists x)(\forall y)(A(x) \wedge B(y))$$
$$(\exists x)A(x) \wedge (\exists y)B(y) \equiv (\exists x)(\exists y)(A(x) \wedge B(y))$$

$\forall x \forall y$ มีความหมายเหมือนกับ $\forall y \forall x$ และสามารถเขียนเป็น $\forall xy$

$\exists x\ \exists y$ มีความหมายเหมือนกับ $\exists y\ \exists x$ และสามารถเขียนเป็น $\exists xy$

แต่ข้อควรระวัง $\exists x\ \forall y$ ไม่เหมือนกับ $\forall y\ \exists x$

7.4 กฎของการอนุมาน

เมื่อสังเกตถึงปรากฏการณ์ สิ่งต่างๆ หรือความสัมพันธ์ต่างๆ ในโดเมนที่สนใจและนำมาเขียนให้อยู่ในรูปของตรรกะเพรดิเคตได้แล้ว จะทำให้สามารถมองได้ว่าสิ่งที่เขียนในรูปของตรรกะเพรดิเคตก็คือ ความรู้ที่ทราบในโดเมนนั้นๆ ขั้นตอนต่อไปซึ่งเป็นข้อดีของการแทนความรู้ก็คือ การอนุมาน (inference) เพื่อหาข้อเท็จจริงใหม่ๆ หรือผลสรุปที่แฝงอยู่ในความรู้นั้นได้ เราเรียกกฎที่ใช้อนุมานเพื่อหาความรู้ใหม่ว่า กฎการอนุมาน (rule of inference) ซึ่งจะทำหน้าที่สร้างสูตรใหม่จากสูตรหลายๆ ตัวที่มีอยู่ กฎการอนุมานมีอยู่หลากหลายชนิด ตัวอย่างของกฎการอนุมาน เช่น

กฎโมดัสโพเน็นส์ (Modus Ponens):

$$\frac{\begin{array}{l} W1 \Rightarrow W2 \\ W1 \end{array}}{W2}$$

กฎเจาะจงตัวแปรเอกภพ (Universal Specialization):

$$\frac{\forall x\ (W(x))}{W(A)}$$

กฎข้อแรกหมายความว่า ถ้า ‘W1 ⇒ W2’ และ ‘W1’ เป็นจริงแล้ว สามารถสรุปได้ว่า ‘W2’ จะเป็นจริงด้วย ส่วนกฎข้อที่สองหมายความว่า ถ้า ‘∀x (W(x))’ เป็นจริงแล้ว สามารถสรุปได้ว่า ‘W(A)’ จะเป็นจริงด้วย เมื่อ ‘A’ เป็นค่าคงที่ในโดเมนที่กล่าวถึง โดยเรียกสูตรใหม่ที่เกิดขึ้นเรียกว่า ทฤษฎี (theorem) และลำดับของกฎการอนุมานที่ใช้ในการสร้างทฤษฎีว่าการพิสูจน์ (proof) ของทฤษฎีนั้นตัวอย่างเช่นจากสูตร 2 ตัวคือ ‘∀x (W1(x) ⇒ W2(x))’ และ ‘W1(A)’ ทำให้สามารถอนุมานได้ว่า ‘W2(A)’ เป็นจริงถ้าสูตร 2 ตัวบนเป็นจริง ‘W2(A)’ เป็นทฤษฎี ส่วนการพิสูจน์ก็คือลำดับของกฎการอนุมานด้านล่างนี้

ใช้กฎเจาะจงตัวแปรเอกภพ
ได้ว่า

$$\frac{\forall x\ (W1(x) \Rightarrow W2(x))}{W1(A) \Rightarrow W2(A)}$$

จากนั้นใช้กฎโมดัสโพเน็นส์
ได้ว่า

$$\frac{\begin{array}{l} W1(A) \Rightarrow W2(A) \\ W1(A) \end{array}}{W2(A)}$$

ปัญหาหนึ่งของการพิสูจน์ก็ดังเช่นที่แสดงในตัวอย่างข้างต้นนี้คือ จะรู้ได้อย่างไรว่าเวลาที่ใช้กฎเจาะจงตัวแปรเอกภพ จะต้องแทนตัวแปรด้วยค่าคงที่ตัวใด ในตัวอย่างข้างต้นได้แทนตัวแปร x ด้วยค่าคงที่ A และทำให้สามารถอนุมานต่อได้ เพราะจะได้ว่า ‘W1(A)’ ที่ด้านซ้ายมือในสูตร ‘W1(A) ⇒ W2(A)’ เท่ากันกับ ‘W1(A)’ ที่มีอยู่ แต่ถ้าแทน x ด้วยค่าคงที่อื่น เช่น B ก็จะไม่สามารถอนุมานต่อได้ ปัญหานี้เป็นปัญหาที่สำคัญที่ต้องพิจารณาอย่างละเอียดถี่ถ้วน ดังนั้นส่วนต่อไปที่จะศึกษาก็คือการแทนค่าและการทำให้เท่ากัน

7.5 การแทนค่าและการทำให้เท่ากัน

การแทนค่า (Substitution) คือการแทนพจน์ (term) ให้กับตัวแปร พจน์หมายรวมถึงค่าคงที่ ฟังก์ชัน และตัวแปร สูตรที่ได้จากการทำการแทนค่าพจน์ในตัวแปรของสูตรใดๆ เรียกว่าตัวอย่างการแทนค่า (substitution instance) ของสูตรนั้นๆ เช่นตัวอย่างการแทนค่าของสูตร

P(x, f(y), B)

ได้แก่

P(z, f(w),B)

และ

P(C, f(A), B)

ซึ่งสามารถเขียนการแทนค่าอยู่ในรูปของเซตคู่ลำดับ

$$s = \{t_1/v_1, t_2/v_2,..., t_n/v_n\}$$

โดยที่ คู่ลำดับ t_i/v_i หมายถึงพจน์ t_i ถูกแทนค่าให้กับตัวแปร v_i เมื่อกำหนดให้

s1 = {z/x, w/y}

เมื่อแทนค่า s1 ลงในสูตร P(x, f(y), B) ก็จะได้ P(z, f(w), B) ในขณะเดียวกันถ้ากำหนดให้

s2 = {C/x, A/y}

เมื่อแทนค่า s2 ลงในสูตร P(x, f(y), B) ก็จะได้ P(C, f(A), B) อีกทั้งสามารถเขียนสูตรที่ได้จากการแทนค่า s กับสูตร E ด้วย Es จากตัวอย่างข้างต้นจะได้ว่า

P(z, f(w), B) = P(x, f(y), B)s1 และ P(C, f(A), B) = P(x, f(y), B)s2

จุดประสงค์หนึ่งของการแทนความรู้ในรูปของตรรกะเพรดิเคตก็คือ เมื่อเขียนความรู้อยู่ในรูปตรรกะเพรดิเคตแล้ว ทำให้สามารถอนุมานหาความรู้ใหม่ๆ ที่แฝงอยู่ในความรู้ที่มีอยู่ได้ และดังเช่นที่แสดงในกฎการอนุมานว่า ปัญหาหนึ่งที่พบในการอนุมานหาความรู้ใหม่ก็คือ การแทนค่าให้กับตัวแปรว่าจะต้องใช้ค่าคงที่ใดแทนค่าให้กับตัวแปรใดเพื่อที่จะทำให้พจน์บางตัวเท่ากัน ซึ่งส่งผลให้การอนุมานสำเร็จ เราได้กล่าวถึงการแทนค่าไปแล้ว ส่วนต่อไปที่ก็จะกล่าวคือ การทำให้เท่ากัน (unification)

สูตร E1 และ E2 สามารถทำให้เท่ากัน (unify) ถ้ามีการแทนค่า s ที่ทำให้ E1s = E2s และในกรณีนี้เรียก s ว่าเป็นตัวทำให้เท่ากัน (unifier) ของ E1 และ E2 ตัวอย่างเช่น P[x, f(y), B] และP[x, f(B), B] จะสามารถทำให้เท่ากันได้โดยมีตัวทำให้เท่ากันคือ s = {A/x, B/y} และผลของการทำให้เท่ากันก็จะได้สูตรคือ P[A, f(B), B] สูตรสองตัวใดๆ มักจะมีตัวทำให้เท่ากันมากกว่าหนึ่งตัว แต่ตัวที่ต้องให้ความสนใจคือตัวทำให้เท่ากันที่ใช้การแทนค่าไม่มากเกินความจำเป็น โดยเรียกตัวทำให้เท่ากันแบบนี้ว่า ตัวทำให้เท่ากันกว้างสุด – เอ็มจียู (most general unifier – mgu) ซึ่งสามารถนิยามได้ดังนี้

g เป็นตัวทำให้เท่ากันกว้างสุดของ E1 และ E2 ก็ต่อเมื่อ ถ้ามี s ที่เป็นตัวทำให้เท่ากันอื่นของ E1 และ E2 แล้ว จะต้องมีตัวทำให้เท่ากัน s' ที่ทำให้ E1s = E1gs' และ E2s = E2gs'

จากตัวอย่างด้านบน เอ็มจียูของ P[x, f(y), B] และ P[x, f(B), B] คือ {B/y} ซึ่งจะเห็นได้ว่าใช้การแทนค่าไม่มากเกินไป ต่างจาก s = {A/x, B/y} ด้านบนที่มีการแทนค่าเกินความจำเป็นคือ ‘A/x ’ อัลกอริทึมสำหรับหาเอ็มจียูมีดังนี้

```
Algorithm: Unify(L1, L2)
1. IF L1 or L2 are both variables or constants THEN
        IF L1 = L2 THEN return NIL
        ELSE IF L1 is a variable THEN
                IF L1 occurs in L2 THEN return {FAIL} ELSE
                        return {L2/L1}
        ELSE IF L2 is a variable THEN
                IF L2 occurs in L1 THEN return {FAIL} ELSE
                        return {L1/L2}
        ELSE return {FAIL}
2. IF the predicate or function symbols of L1 and L2 are not identical
   THEN return {FAIL}
3. IF L1 and L2 have a different number of arguments THEN return {FAIL}
4. SUBST := NIL
5. FOR i := 1 TO number of arguments in L1 DO
        5.1 S := Unify(i^th argument of L1, i^th argument of L2)
        5.2 IF S contains FAIL THEN return {FAIL}
        5.3 IF S <> NIL THEN
             5.3.1 Apply S to the remainder of both L1 and L2
             5.3.2 SUBST := SUBST ∪ S
6. return SUBST
```

ตัวอย่าง เช่นกำหนดให้ L1 = P(A, x, h(g(z))) และ L2 = P(z, h(y), h(y)) และต้องการทำให้ L1 เท่ากับ L2 โดยเรียกอัลกอริทึม Unify(L1, L2) โดยจะมีขั้นตอนการทำดังนี้

- ขั้นตอนที่ 1 ตรวจสอบว่า L1 และ L2 เป็นค่าตัวแปรหรือค่าคงที่หรือไม่ พบว่า L1 และ L2 ไม่ได้เป็นทั้งตัวแปรหรือค่าคงที่จึงข้ามไปยังขั้นตอนถัดไป
- ขั้นตอนที่ 2 ตรวจสอบว่า L1 และ L2 มีเพรดิเคตตัวเดียวกันหรือไม่ พบว่าทั้งสองตัวใช้เพรดิเดตตัวเดียวกันคือ P ดังนั้นจึงข้ามไปยังขั้นตอนถัดไป
- ขั้นตอนที่ 3 ตรวจสอบว่า L1 และ L2 มีจำนวนอาร์กิวเมนต์เท่ากันหรือไม่ พบว่าทั้งสองตัวมีจำนวนอาร์กิวเมนต์ที่เท่ากันคือ 3 ดังนั้นจึงข้ามไปยังขั้นตอนถัดไป
- ขั้นตอนที่ 4 กำหนดค่าเริ่มต้นให้กับผลลัพธ์ของการแทนค่าที่จะทำให้ L1 เท่ากับL2 โดยเริ่มต้นให้เท่ากับเซตว่าง (แทนด้วย NIL ในอัลกอริทึม)
- ขั้นตอนที่ 5 เป็นขั้นตอนที่พยายามทำอาร์กิวเมนต์ในตำแหน่งที่ตรงกันของ L1 และ L2 ให้เท่ากัน โดยเริ่มจากอาร์กิวเมนต์ตัวที่ 1 ถึงตัวสุดท้าย
    - เริ่มจากอาร์กิวเมนต์ตัวที่ 1 โดยเรียกอัลกอริทึม S = Unify(A, z) ในการเรียกครั้งนี้จะเป็นการเรียกซ้ำ (recursive) ซึ่งจะตรวจสอบด้วยขั้นตอนที่ 1 พบว่า z เป็นตัวแปรและไม่ปรากฏใน A จึงคืนค่า {A/z} ที่จุดนี้จะได้ S = {A/z}
    - เนื่องจาก S ไม่ประกอบด้วย FAIL และไม่เท่ากับ NIL จึงทำขั้นตอนที่ 5.3.1
    - โดยขั้นตอนที่ 5.3.1 นี้นำการแทนค่าตาม S ไปกระทำกับ L1 และ L2 ดั้งเดิม และได้ค่าตามนี้คือ

      L1 = P(A, x, h(g(A)))

      L2 = P(A, h(y), h(y))

      สังเกตว่าอาร์กิวเมนต์ในตำแหน่งที่ 3 ของ L1 ซึ่งเดิมมี z อยู่ด้วยได้ถูกแทนค่าด้วย A เนื่องจากว่าตัวแปรตัวเดียวต้องถูกแทนค่าด้วยพจน์เดียวกันเสมอ
    - เมื่อทำขั้นตอนที่ 5.3.2 จะได้ว่า SUBST = NIL ∪ {A/z} = {A/z}
    - จากนั้นจะเริ่มพิจารณาอาร์กิวเมนต์ตัวที่ 2 ของ L1 และ L2 ให้เท่ากัน โดยเรียกอัลกอริทึม S = Unify(x, h(y)) ในการเรียกครั้งนี้จะตรวจสอบด้วยขั้นตอนที่ 1 พบว่า x เป็นตัวแปรและไม่ปรากฏใน h(y) ทำให้ได้ S = {h(y)/x}
    - เนื่องจาก S ไม่ประกอบด้วย FAIL และไม่เท่ากับ NIL จึงทำขั้นตอนที่ 5.3.1
    - โดยขั้นตอนที่ 5.3.1 ได้นำการแทนค่าตาม S ไปกระทำกับ L1 และ L2 ดั้งเดิม และได้ค่าตามนี้คือ

      L1 = P(A, h(y), h(g(A)))

      L2 = P(A, h(y), h(y))

- เมื่อทำขั้นตอนที่ 5.3.2 จะได้ว่า SUBST = {A/z} ∪ {h(y)/x} = {A/z, h(y)/x}
- จากนั้นเริ่มทำอาร์กิวเมนต์ตัวที่ 3 ของ L1 และ L2 ให้เท่ากัน โดยการเรียกอัลกอริทึม S =Unify(h(g(A)), h(y)) ในการเรียกครั้งนี้เนื่องจาก h ของ h(g(A)) และ h(y) เป็นฟังก์ชันซึ่งตรวจสอบด้วยขั้นตอนที่ 2 และพบว่ามีจำนวนอาร์กิวเมนต์เท่ากันเท่ากับ 1 จึงเรียกอัลกอริทึม Unify(g(A), y) ซึ่งเมื่อทำสำเร็จจะคืนค่าเท่ากับ {g(A)/y} ณ จุดนี้เราได้ S ของอาร์กิวเมนต์ตัวที่ 3 ของ L1 และ L2 เท่ากับ {g(A)/y} เนื่องจาก S ไม่ประกอบด้วย FAIL และไม่เท่ากับ NIL เราจึงทำขั้นตอนที่ 5.3.1 โดยนำการแทนค่าตาม S ไปกระทำกับ L1 และ L2 ดั้งเดิมและได้ค่าตามนี้คือ

  L1 = P(A, h(g(A)), h(g(A)))

  L2 = P(A, h(g(A)), h(g(A)))
- เมื่อทำขั้นตอนที่ 5.3.2 จะได้ว่า

  SUBST = {A/z, h(y)/x} ∪ {g(A)/y} = {A/z, h(y)/x, g(A)/y}

  ซึ่งเป็นผลลัพธ์สุดท้ายของการทำให้เท่ากันนี้

7.6 รีโซลูชัน

วิธีการอนุมานทางตรรกเพรดิเคตได้กล่าวไปแล้วสองวิธีคือกฎโมดัสโพเน็นส์และกฎเจาะจงตัวแปรเอกภพ ซึ่งกฎทั้งสองข้อนี้มีข้อจำกัดอยู่ เช่นกฎโมดัสโพเน็นส์จะใช้ได้กับสูตรที่ต้องอยู่ในรูปแบบที่กำหนดเท่านั้นคือ ตัวแรกอยู่ในรูป W1 ⇒ W2 และตัวที่สองเป็น W1 ซึ่งเหมือนกันกับด้านซ้ายมือของสูตรแรก หรือกฎเจาะจงตัวแปรเอกภพก็ใช้ได้กับสูตรที่มีตัวแปรแล้วแทนที่ด้วยค่าคงที่เท่านั้น การใช้งานจึงจำกัดเช่นกัน ในขณะที่ความรู้ที่เขียนอยู่ในตรรกะเพรดิเคตโดยทั่วไปมีหลากหลายรูปแบบ ทำให้การใช้กฎดังกล่าวไม่กว้างขวางเพียงพอหรือมีโอกาสพบรูปแบบที่กฎใช้ได้น้อยมาก หัวข้อนี้จะกล่าวถึงกฎการอนุมานอีกวิธีหนึ่งซึ่งสามารถใช้ได้อย่างกว้างขวางเรียกว่ารีโซลูชัน (resolution) ที่สามารถใช้กับสูตรทุกตัวที่เป็นอนุประโยค (clause)

อนุประโยค คือสูตรอะตอมหลายตัวมาเชื่อมกันด้วยเครื่องหมาย ∨ และสูตรอะตอมบางตัวอาจมีเครื่องหมายนิเสธอยู่บางตัวอาจไม่มี ตัวอย่างของอนุประโยคก็เช่น P(x) ∨ Q(x, y) ∨ ¬R(A) เป็นต้น ในทางปฏิบัติสูตรที่เขียนแทนความรู้อาจไม่อยู่ในรูปของอนุประโยค ดังนั้นก่อนที่จะใช้เทคนิครีโซลูชันเพื่ออนุมานได้นั้น จำเป็นต้องแปลงสูตรให้อยู่ในรูปของอนุประโยคก่อน โดยใช้ขั้นตอนพร้อมทั้งยกตัวอย่างการแปลงสูตรให้เป็นอนุประโยคดังต่อไปนี้

$$(\forall x) \{ P(x) \Rightarrow \{ (\forall y)[P(y) \Rightarrow P(f(x, y))] \land \neg(\forall y)[Q(x, y) \Rightarrow P(y)] \} \}$$

1. กำจัดเครื่องหมาย ⇒ : เปลี่ยนรูปของ X ⇒ Y ให้เป็น ¬X ∨ Y

$(\forall x) \{ \neg P(x) \lor \{(\forall y)[\neg P(y) \lor P(f(x, y))] \land \neg(\forall y)[\neg Q(x, y) \lor P(y)] \} \}$

2. ลดขอบเขตของเครื่องหมายนิเสธ: นิเสธแต่ละตัวจะมีขอบเขตไม่เท่ากัน ถ้าพบนิเสธที่คลุมบริเวณกว้าง ก็ให้ลดขอบเขตให้แคบสุดโดยกระจายนิเสธเข้าไปข้างในบริเวณที่มันคลุมอยู่ และเปลี่ยนเครื่องหมายอื่นๆ เป็นตรงข้ามให้หมด เช่นนิเสธของ '∧' เป็น '∨' และนิเสธของ '∀' เป็น '∃' ฯลฯ

$(\forall x)\{ \neg P(x) \lor \{(\forall y)[\neg P(y) \lor P(f(x, y))] \land (\exists y) [Q(x, y) \land \neg P(y)] \} \}$

3. ทำตัวแปรเป็นมาตรฐาน: เปลี่ยนชื่อตัวแปรตามขอบเขตของตัวบ่งปริมาณ เช่นถ้าเราพบว่ามี y ซ้ำกันสองที่ ก็ให้เปลี่ยนชื่อตัวใดตัวหนึ่ง และความหมายของสูตรที่ได้จะไม่เปลี่ยนไปจากเดิม

$(\forall x)\{ \neg P(x) \lor \{(\forall y)[\neg P(y) \lor P(f(x, y))] \land (\exists w)[Q(x, w) \land \neg P(w)] \} \}$

4. กำจัดตัวบ่งปริมาณมีอยู่: แทนค่าตัวแปรด้วยฟังก์ชันสคอเล็ม(Skolem function) ซึ่งเป็นฟังก์ชันที่แทนค่าตัวแปรตัวหนึ่งด้วยฟังก์ชันของตัวแปรอื่นๆ ที่ตัวแปรตัวนั้นขึ้นอยู่กับมัน ในกรณีของตัวอย่างสูตรในขั้นตอนที่ 3 ที่ตำแหน่งของ $(\forall x)\{\ldots(\exists w)[Q(x, w) \land \neg P(w)]$ ซึ่งจะอ่านได้ว่าสำหรับ x ทุกตัวจะมี w บางตัวที่ทำให้Q(x, w) และ ¬P(w) เป็นจริง แสดงว่าถ้าเลือก x มาหนึ่งตัวจะต้องมี w 1 ตัวที่ทำให้สูตรเป็นจริง หมายความว่า w ขึ้นกับ x หรือเป็นฟังก์ชันของ x ซึ่งก็คือ w = g(x) เมื่อ g เป็นฟังก์ชันสคอเล็ม

$(\forall x)\{ \neg P(x) \lor \{(\forall y)[\neg P(y) \lor P(f(x, y))] \land [Q(x, g(x)) \land \neg P(g(x))] \} \}$

5. แปลงให้อยู่ในรูปแบบพรีเน็กซ์ (prenex form): ย้ายตัวบ่งปริมาณเอกภพทุกตัวมาอยู่หน้าสุด และรูปแบบที่ได้ใหม่นี้เรียกว่ารูปแบบพรีเน็กซ์

$(\forall x)(\forall y)\{ \neg P(x) \lor \{[\neg P(y) \lor P(f(x, y))] \land [Q(x, g(x)) \land \neg P(g(x))]\}\}$

6. จัดรูปของพรีเน็กซ์ใหม่ให้อยู่ในรูปทั่วไปแบบและ (conjuctive normal form): รูปที่สูตรทุกตัวเชื่อมกันด้วยเครื่องหมาย '∧' แต่ภายในสูตรมีแต่เครื่องหมาย '∨' โดยใช้ความสมมูลของสูตรต่อไปนี้ P ∨ (Q ∧ R) = (P ∨ Q) ∧ (P ∨ R)

$(\forall x)(\forall y)\{[\neg P(x) \lor \neg P(y) \lor P(f(x, y))] \land [\neg P(x) \lor (Q(x, g(x)) \land \neg P(g(x)))] \}$

$(\forall x)(\forall y)\{[\neg P(x) \lor \neg P(y) \lor P(f(x, y))] \land [\neg P(x) \lor Q(x, g(x))] \land [\neg P(x) \lor \neg P(g(x))] \}$

7. ละเครื่องหมายตัวบ่งปริมาณเอกภพ: เนื่องจากว่า ณ จุดนี้ตัวแปรทุกตัวจะมีตัวบ่งปริมาณเป็นแบบเอกภพเท่านั้น

$$[\neg P(x) \lor \neg P(y) \lor P(f(x, y))] \quad \land \quad [\neg P(x) \lor Q(x, g(x))] \quad \land \quad [\neg P(x) \lor \neg P(g(x))]$$

8. ที่จุดนี้เราจะได้อนุประโยคตั้งแต่ 1 ประโยคขึ้นไป โดยที่แต่ละประโยคเชื่อมกันด้วยเครื่องหมาย $\land$ นำอนุประโยคเหล่านั้นมาเขียนเรียงกัน

(1) $\neg P(x) \lor \neg P(y) \lor P(f(x, y))$

(2) $\neg P(x) \lor Q(x, g(x))$

(3) $\neg P(x) \lor \neg P(g(x))$

9. เปลี่ยนชื่อตัวแปร: เปลี่ยนชื่อตัวแปรโดยตัวแปรเดียวกันที่ปรากฏในหลายอนุประโยคให้เขียนใหม่ด้วยตัวแปรคนละตัว

(1) $\neg P(x1) \lor \neg P(y) \lor P(f(x1, y))$

(2) $\neg P(x2) \lor Q(x2, g(x2))$

(3) $\neg P(x3) \lor \neg P(g(x3))$

7.6.1 รีโซลูชันของอนุประโยคพื้นฐาน

ก่อนที่จะอธิบายถึงรีโซลูชันของอนุประโยคทั่วไป ขอกล่าวถึงรีโซลูชันของอนุประโยคพื้นฐาน (ground clause) ก่อน ซึ่งจะง่ายกว่าของอนุประโยคทั่วไป อนุประโยคพื้นฐานคืออนุประโยคที่ไม่มีตัวแปร การทำรีโซลูชันสำหรับอนุประโยคพื้นฐานจะรับอินพุตเป็นอนุประโยคพ่อแม่ (parent clause) 2 ประโยค และจะให้อนุประโยคเป็นเอาต์พุต 1 ประโยคสามารถแสดงได้ดังต่อไปนี้

$$\begin{array}{l} P1 \lor P2 \lor \cdots \lor Pn \\ \neg P1 \lor Q2 \lor \cdots \lor Qm \\ \hline P2 \lor \cdots \lor Pn \lor Q2 \lor \cdots \lor Qm \end{array}$$

เราเรียกอนุประโยค 2 ประโยคบนว่า อนุประโยคพ่อแม่ (parent clause) ส่วนอนุประโยคที่เป็นผลลัพธ์เรียกว่า รีโซเวนท์ (resolvent) ซึ่งหมายความว่าถ้าเรามีอนุประโยค 2 ประโยคและพบว่าอนุประโยคทั้งสองมีสัญพจน์อยู่ 1 ตัวที่เหมือนกันทุกประการยกเว้นเครื่องหมายนิเสธที่ต่างกัน

(สัญพจน์แรกไม่มีนิเสธแต่สัญพจน์ที่สองมีนิเสธ (P1 กับ ¬P1) หรือกลับกัน) ผลลัพธ์ที่ได้จะเป็นอนุประโยคที่นำสัญพจน์อื่นๆ ทั้งหมดของทั้งสองอนุประโยคมารวมกัน ยกเว้นสัญพจน์ที่เหมือนกัน (ต่างเฉพาะเครื่องหมาย) นั้น ผลที่ได้จากการทำ รีโซลูชันจะเป็นจริงถ้าอนุประโยคพ่อแม่ทั้งสองเป็นจริง ตัวอย่างของการใช้รีโซลูชันแสดงดังด้านล่างนี้

อนุประโยคพ่อแม่	รีโซเวนท์
P และ ¬P ∨ Q	Q
P ∨ Q และ ¬P ∨ Q	Q
P ∨ Q และ ¬P ∨ ¬Q	Q ∨ ¬Q และ P ∨ ¬P
¬P และ P	NIL
¬P ∨ Q และ ¬Q ∨ R	¬P ∨ R

ดังจะเห็นได้จากตัวอย่างทั้งห้าด้านบน รีโซลูชันสามารถใช้ได้อย่างกว้างขวางมาก ดังเช่น ตัวอย่างแรกในตารางก็เป็นกรณีหนึ่งของรีโซลูชัน ซึ่งเท่ากับกฎโมดัสโพเน็นท์ สำหรับรีโซลูชันทั่วไปที่กระทำกับอนุประโยคที่มีตัวแปรด้วยนั้นขั้นตอนจะซับซ้อนกว่าเดิม ซึ่งจะต้องใช้การทำให้เท่ากันร่วมด้วยเพื่อทำให้อนุประโยคพ่อแม่ประกอบด้วยสัญพจน์เติมเต็ม (complimentary literals) สัญพจน์เติมเต็มคือสัญพจน์ที่เหมือนกันทุกประการเว้นแต่ว่าตัวหนึ่งมีเครื่องหมายนิเสธส่วนอีกตัวไม่มี เช่น ¬Q(z) กับ Q(z) เป็นต้น ในการอธิบายการทำรีโซลูชันทั่วไป จะเขียนแทนอนุประโยคด้วยเซตของสัญพจน์ เช่น ¬P(z, f(A)) ∨ ¬Q(z) เขียนแทนด้วย {¬P(z, f(A)), ¬Q(z)} เป็นต้น ขั้นตอนของรีโซลูชันทั่วไปทำตามขั้นตอนต่อไปนี้

- กำหนดให้อนุประโยคพ่อแม่เป็น {Li} และ {Mi}
- เลือก {li} และ {mi} ที่เป็นเซตย่อยของ {Li} และ {Mi} ตามลำดับ ซึ่งมี s ที่เป็นเอ็มจียูของ {li} และ {¬mi} (หมายความว่าเซตย่อยทั้งสองจะต้องสามารถทำให้เป็นสัญพจน์เติมเต็มได้)
- จะได้ว่ารีโซเวนท์ของอนุประโยคพ่อแม่ {Li} กับ {Mi} คือ {{Li} – {li}}s ∪ {{Mi} – {mi}}s โดยทั่วไปรีโซเวนท์จากรีโซลูชันทั่วไปอาจมีได้มากกว่าหนึ่งประโยคขึ้นกับการเลือกเซตย่อยที่จะมาทำให้เป็นสัญพจน์เติมเต็มตัวอย่างของการทำรีโซลูชันทั่วไปแสดงดังด้านล่างนี้

- กำหนดให้ {Li} = {P[x, f(A)], P[x, f(y)], Q(y)} และ {Mi} = {¬P[z, f(A)], ¬Q(z)} (กล่าวคือต้องการทำรีโซลูชันระหว่างอนุประโยค P[x, f(A)] ∨ P[x, f(y)] ∨ Q(y) กับ ¬P[z, f(A)] ∨ ¬Q(z))
- เลือก {li} = {[p(x, f(A)]} และ {mi} = {¬P[z, f(A)]} โดยที่ s = {z/x}
- ได้ว่ารีโซเวนท์เท่ากับ ({P[x, f(A)], P[x, f(y)], Q(y)} – {[p(x ,f(A)]}){z/x} ∪ ({¬P[z, f(A)], ¬Q(z)} – {¬P[z, f(A)]}){z/x} = {P[z, f(y)], Q(y), ¬Q(z)}
- แต่ถ้าเลือก {li} = {[p(x, f(A)], P[x, f(y)]} และ {mi} = {¬P[z, f(A)]} จะได้รีโซเวนท์{Q(A), ¬Q(z)}

7.6.2 การปฏิเสธแบบรีโซลูชัน

จากที่ได้แสดงให้เห็นข้างต้นว่าการทำรีโซลูชันจะทำให้สามารถหาความรู้ที่แฝงอยู่ในความรู้ที่มีอยู่ได้ และในหลายๆกรณีเมื่อมีความรู้ที่แสดงอยู่ในรูปตรรกะเพรดิเคตและต้องการพิสูจน์อนุประโยคตัวใหม่ใดๆ ว่าเป็นผลสรุปของความรู้ที่มีอยู่หรือไม่ วิธีการพิสูจน์ก็อาจทำโดยการเลือกอนุประโยคพ่อแม่ 2 ประโยคแล้วหารีโซเวนท์ ถ้ารีโซเวนท์เป็นความรู้ใหม่ที่เราต้องการพิสูจน์ก็แสดงว่าเราพิสูจน์สำเร็จ ถ้าไม่ใช่เราก็อาจเลือกอนุประโยคพ่อแม่ 2 ประโยคอื่นๆ แล้วลองทำรีโซลูชันดู หรืออาจนำรีโซเวนท์ที่ได้ก่อนหน้านี้มาจับคู่กับอนุประโยคอื่นเพื่อเป็นอนุประโยคพ่อแม่แล้วทำรีโซลูชันต่อไป อย่างไรก็ดีการพิสูจน์แบบนี้อาจไม่มีเป้าหมายที่ชัดเจน ทำให้การพิสูจน์เสียเวลาในการคำนวณมาก เรามีวิธีการซึ่งเน้นที่เป้าหมายในการพิสูจน์เพื่อทำให้การพิสูจน์ทำได้ดีขึ้น เราเรียกวิธีการที่จะนำเสนอนี้ว่าการปฏิเสธแบบรีโซลูชัน (resolution refutation) ก่อนอื่นขอกล่าวถึงความขัดแย้ง (contradictory) ที่ใช้ในการพิสูจน์โดยการปฏิเสธแบบรีโซลูชันดังนี้สัญพจน์ 2 ตัวใดๆ จะขัดแย้งกัน ถ้าตัวหนึ่งสามารถทำให้เท่ากับนิเสธของอีกตัวหนึ่งได้เช่น MAN(x) กับ ~MAN(Spot) ขัดแย้งกัน เนื่องจาก MAN(x) ทำให้เท่ากับ MAN(Spot) ได้ วิธีการปฏิเสธแบบรีโซลูชันคือ การที่จะพิสูจน์ว่าสูตร W เป็นผลสรุปของเซตของสูตร K ทำได้โดยการพิสูจน์ว่า K ∪ {~W} ขัดแย้งกัน ตัวอย่างเช่น

กำหนดให้ K = {MAN(Marcus), ~MAN(x) ∨ MORTAL(x)}
กำหนดให้ W = MORTAL(Marcus)
จะได้ว่า K ∪ {~W} = { MAN(Marcus),
~MAN(x) ∨ MORTAL(x),
~MORTAL(Marcus) }

จากการทำรีโซลูชันกับอนุประโยค 2 ประโยคล่าง จะได้ ~MAN(Marcus) ซึ่งขัดแย้งกับอนุประโยคที่ 1 ‘MAN(Marcus)’ ซึ่งแสดงว่า MORTAL(Marcus) เป็นผลสรุปของ K

การพิสูจน์ในลักษณะนี้จะมีเป้าหมายในการพิสูจน์ กล่าวคือ ~W จะเป็นตัวที่ถูกเลือกเพื่อไปการทำรีโซลูชันกับอนุประโยคอื่น แล้วนำรีโซเวนท์ที่ได้ไปกระทำรีโซลูชันกับความขัดแย้งอนุประโยคอื่นๆ ต่อไปตามลำดับจนกระทั่งพบความขัดแย้ง โดยอัลกอริทึมการปฏิเสธแบบรีโซลูชันมีขั้นตอนดังนี้

```
Algorithm: Resolution Refutation
     1. Convert all the statements of F to clause form.
     2. Negate P and convert the result to clause form. Add
        it to the set of clauses obtained in 1.
     3. Let Clauses be the set of clauses. /* F ∪ {~P} */
     4. UNTIL (NIL is a member of Clauses) or
        (Clauses do not change) DO
            4.1 Select Ci and Cj which are resolvable.
            4.2 Calculate the resolvent of Ci and Cj, and call it Rij.
            4.3 Clauses := Clauses ∪ {Rij}
```

จากอัลกอริทึมด้านบนนี้เริ่มต้นด้วยการแปลงสูตรใดๆ ให้อยู่ในรูปของอนุประโยค แล้วเปลี่ยน P ซึ่งเป็นสูตรที่เราต้องการพิสูจน์ให้อยู่ในรูปของนิเสธแล้วเพิ่มเข้าไปใน F จากนั้นจึงทำรีโซลูชันในขั้นตอนที่ 4 จนกระทั่งพบความขัดแย้ง หรือ Clauses ไม่เปลี่ยนแปลง ในกรณีที่พบความขัดแย้งอัลกอริทึมจะหยุดและได้ว่า P เป็นผลสรุปของ F เมื่อสังเกตในขั้นตอน 4.3 จะเห็นได้ว่า Clauses จะขยายตัวเรื่อยๆ ทุกครั้งที่ได้รีโซเวนท์ตัวใหม่ อย่างไรก็ดีหากพบว่าไม่มีรีโซเวนท์ใหม่เกิดขึ้นอีกหรือไม่สามารถทำรีโซลูชันได้อีก Clauses จะหยุดขยายตัว ซึ่งเป็นกรณีอีกหนึ่งกรณีที่อัลกอริทึมนี้จะหยุด และในกรณีนี้ได้ว่า P ไม่ใช่

ผลสรุปของ F อย่างไรก็ดีในบางครั้งเราอาจพบกรณีที่ Clauses ยังคงขยายตัวต่อเรื่อยๆ ซึ่งในกรณีเช่นนี้เราไม่สามารถสรุปอะไรได้ กล่าวคือ P อาจเป็นผลสรุปของ F หรือไม่ก็ได้

7.7 ตัวอย่างการหาสรุปด้วยวิธีรีโซลูชัน

จากประโยคต่อไปนี้จะพิสูจน์ว่าสมชายมีความสุข

คนที่สอบวิชาประวัติศาสตร์ผ่านและถูกลอตเตอรี่จะมีความสุข แต่ถ้าใครที่เรียนอยู่หรือโชคดีจะผ่านวิชาที่เรียนทุกวิชา สมชายไม่ได้เรียนแต่สมชายโชคดี ใครที่โชคดีจะถูกลอตเตอรี่

ขั้นตอนแรกสุดคือต้องแปลงประโยคให้อยู่รูปแบบตรรกะเพรดิเคตก่อนซึ่งสามารถแบ่งออกได้เป็น 4 อนุประโยค ดังนี้

- คนที่สอบวิชาประวัติศาสตร์ผ่านและถูกลอตเตอรี่จะมีความสุข

  ∀x((Pass(x, HISTORY) ∧ Win(x, LOTTERY)) ⇒ Happy(x))

- ใครที่เรียนอยู่หรือโชคดีจะผ่านวิชาที่เรียนทุกวิชา

  ∀x∀y ((Study(x) ∨ Lucky(x)) ⇒ Pass(x, y))

- สมชายไม่ได้เรียนแต่สมชายโชคดี

  ¬Study(สมชาย) ∧ Lucky(สมชาย)

- ใครที่โชคดีจะถูกลอตเตอรี่

  ∀x(Lucky(x) ⇒ Win(x, LOTTERY))

เมื่อได้ประโยคในรูปแบบตรรกะเพรดิเคตแล้ว ก็เริ่มต้นขั้นตอนให้การแปลงรูปของประโยคเหล่านี้ให้อยู่ในรูปแบบของ conjuntive normal form ตามขั้นตอนต่อไปนี้

- ขั้นตอนที่ 1 : กำจัดเครื่องหมาย ⇒
  - ∀x(¬(Pass(x, HISTORY) ∧ Win(x, LOTTERY)) ∨ Happy(x))
  - ∀x∀y (¬(Study(x) ∨ Lucky(x)) ∨ Pass(x, y))
  - ¬Study(สมชาย) ∧ Lucky(สมชาย)
  - ∀x(¬Lucky(x) ∨ Win(x, LOTTERY))
- ขั้นตอนที่ 2 : ลดขอบเขตของเครื่องหมายนิเสธ
  - ∀x(¬Pass(x, HISTORY) ∨ ¬Win(x, LOTTERY) ∨ Happy(x))
  - ∀x∀y (¬Study(x) ∧ ¬Lucky(x)) ∨ Pass(x, y))
  - ¬Study(สมชาย) ∧ Lucky(สมชาย)
  - ∀x(¬Lucky(x) ∨ Win(x, LOTTERY))
- ขั้นตอนที่ 3 : ทำตัวแปรเป็นมาตรฐาน

  -- ไม่มีประโยคไหนที่ต้องทำ –
- ขั้นตอนที่ 4 : กำจัดตัวบ่งปริมาณมีอยู่

  -- ไม่มีตัวบ่งปริมาณมีอยู่ จึงไม่ต้องทำ –

- ขั้นตอนที่ 5 : แปลงให้อยู่ในรูปแบบพรีเน็กซ์ (prenex form)

  -- ทุกประโยคอยู่ในรูปแบบพรีเน็กซ์หมดแล้ว –

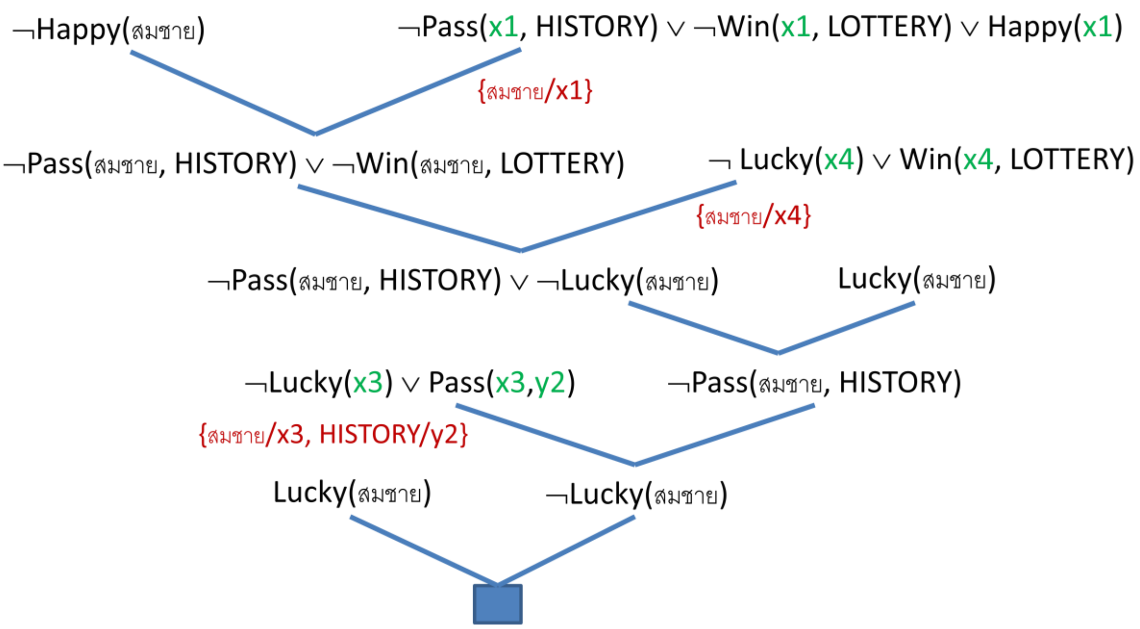
- ขั้นตอนที่ 6 : จัดรูปของพรีเน็กซ์ใหม่ให้อยู่ในรูปทั่วไปแบบและ
  - ∀x(¬Pass(x, HISTORY) ∨ ¬Win(x, LOTTERY) ∨ Happy(x))
  - ∀x∀y [(¬Study(x) ∨ Pass(x, y)) ∧ (¬Lucky(x) ∨ Pass(x, y))]
  - ¬Study(สมชาย) ∧ Lucky(สมชาย)
  - ∀x(¬Lucky(x) ∨ Win(x, LOTTERY))

- ขั้นตอนที่ 7 : ละเครื่องหมายตัวบ่งปริมาณเอกภพ
  - ¬Pass(x, HISTORY) ∨ ¬Win(x, LOTTERY) ∨ Happy(x)
  - (¬Study(x) ∨ Pass(x, y)) ∧ (¬Lucky(x) ∨ Pass(x, y))
  - ¬Study(สมชาย) ∧ Lucky(สมชาย)
  - (¬Lucky(x) ∨ Win(x, LOTTERY)

- ขั้นตอนที่ 8 : แยกประโยคออกเป็นประโยคย่อย
  - ¬Pass(x, HISTORY) ∨ ¬Win(x, LOTTERY) ∨ Happy(x)
  - (¬Study(x) ∨ Pass(x, y))
  - (¬Lucky(x) ∨ Pass(x, y))
  - ¬Study(สมชาย)
  - Lucky(สมชาย)
  - (¬Lucky(x) ∨ Win(x, LOTTERY)

- ขั้นตอนที่ 9 : เปลี่ยนชื่อตัวแปร
  - ¬Pass(x1, HISTORY) ∨ ¬Win(x1, LOTTERY) ∨ Happy(x1)
  - (¬Study(x2) ∨ Pass(x2, y1))
  - (¬Lucky(x3) ∨ Pass(x3, y2))
  - ¬Study(สมชาย)
  - Lucky(สมชาย)
  - (¬Lucky(x4) ∨ Win(x4, LOTTERY)

เมื่อแปลงประโยคความรู้ที่มีอยู่ให้อยู่ในรูปแบบของ Conjunctive normal form เรียบร้อยแล้วก็ถึงขั้นตอนการพิสูจน์ โจทย์ต้องการทราบว่า สมชายมีความสุข ‘HAPPY(สมชาย)’ ดังนั้นเมื่อใช้วิธีปฏิเสธแบบรีโซลูชันก็จะเป็นการพิสูจน์ประโยค ‘¬Happy(สมชาย)’ ให้เป็นเท็จ ซึ่งสามารถพิสูจน์ได้เป็นขั้นตอนดังภาพที่ 7-1



ภาพที่ 7-1 การพิสูจน์ด้วยวิธีปฏิเสธแบบรีโซลูชัน

จากภาพจะเห็นได้ว่าการพิสูจน์นำทางมาสู่เซตว่าง ซึ่งหมายความว่าสมมุติฐานที่ว่า สมชายไม่มีความสุข ‘¬Happy(สมชาย)’ นั้นเป็นเท็จ ทำให้สามารถตอบได้ว่า สมชายมีความสุข ‘HAPPY(สมชาย)’ เป็นจริง

7.8 สรุป

ตรรกศาสตร์ที่ว่าด้วยภาคขยายหรือตรรกะเพรดิเคต เป็นกระบวนการตรรกที่มีความซับซ้อนที่กล่าวถึงการอนุมาน โดยหาค่าความจริงด้วยหลักการทางตรรกะในการวิเคราะห์และแทนรูปประโยคในตัวภาษา องค์ประกอบพื้นฐานของตรรกะเพรดิเดต ได้แก่ ตัวอักษร (Alphabet), ส่วนแสดงความสัมพันธ์ (Predicate), ตัวเชื่อม (Connective), ตัวบ่งปริมาณ (Quantifier) และภาษา (Language) สำหรับข้อดีของตรรกะเพรดิเดตก็คือ มีหลักในการเปลี่ยนรูปแบบของภาษาให้อยู่ในรูปแบบที่สามารถนำไปคำนวณาค่าความเท็จจริงได้ง่าย อีกทั้งมีการบ่งบอกถึงปริมาณของเนื้อหาได้ด้วย ทำให้มีประสิทธิภาพสูงในการหาค่าความจริง

แบบฝึกหัด

1. จากข้อมูลต่อไปนี้

- Marcus was a man.
- Marcus was a Pompeian.
- All Pompeian were Romans.
- Caesar was a ruler.
- All Romans were either loyal to Caesar or hated him
- Everyone is loyal to someone
- People only try to assassinate rulers they aren't loyal to.
- Marcus tried to assassinate Caesar.
- All men are people.

จงพิสูจน์ว่า Marcus hated Caesar.

2. จากข้อมูลต่อไปนี้

- Marcus was a man
- Marcus was a Pompeian
- Marcus was born in 40 AD.
- All men are mortal
- All Pompeians died when the volcano erupted in 79 AD.
- No mortal lives longer than 150 years
- It's now 2011
- Alive means not dead
- If someone dies, then he's dead at all later times.

จงพิสูจน์ Marcus is not alive.

แผนปฏิบัติการสอนสัปดาห์ที่ 8

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง ตรรกศาสตร์กำกวม **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. แนวคิดพื้นฐานเกี่ยวกับฟัซซีลอจิก	IS 8.1
2. เซตแบบฉบับ	IS 8.2
3. ฟัซซีเซต	IS 8.3
4. ฟังก์ชันความเป็นสมาชิก	IS 8.4
5. ตัวแปรภาษา	IS 8.5
6. กฎฟัซซี	IS 8.6
7. ระบบกฎแบบฟัซซี	IS 8.7

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2-3-4-5-6-7**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

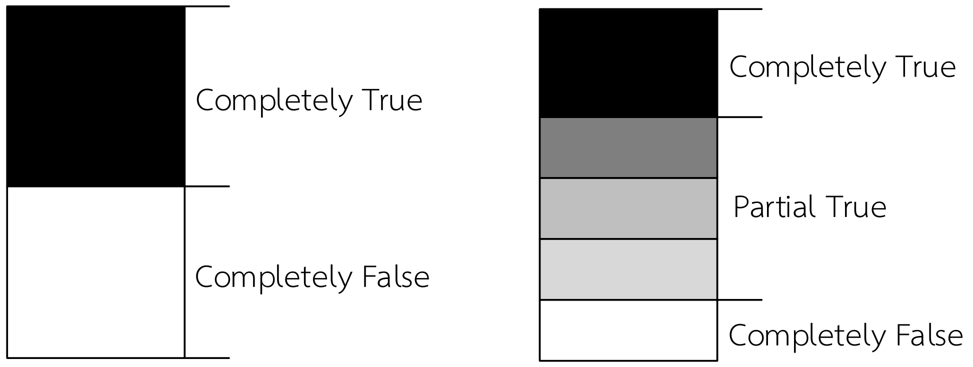
บทที่ 8
ตรรกศาสตร์กำกวม

ตรรกศาสตร์กำกวม หรือฟัซซีลอจิก (Fuzzy Logic) เป็นศาสตร์ด้านคอมพิวเตอร์ที่ทำงานโดยอาศัยฟัซซีลอจิกที่คิดค้นโดย L. A. Zadeh ในปี ค.ศ. 1965 ฟัซซีลอจิกเป็นตรรกะที่อยู่บนพื้นฐานความเป็นจริงที่ว่า ทุกสิ่งบนโลกแห่งความเป็นจริงไม่ใช่มีเฉพาะสิ่งมีความแน่นอนเท่านั้น แต่มีหลายสิ่งหลายเหตุการณ์ที่เกิดขึ้นอย่างไม่เที่ยงและไม่แน่นอน (Uncertain) อาจเป็นสิ่งที่คลุมเครือ (Fuzzy) ไม่ใช่ชัดเจน (Exact) เช่น เซตของอายุคน อาจแบ่งเป็น วัยทารก วัยเด็ก วัยรุ่น วัยกลางคน และวัยชรา จะเห็นได้ว่าในแต่ละช่วงอายุคนไม่สามารถระบุได้แน่ชัดว่าวัยทารกกับวัยเด็กแยกจากกันแน่ชัดช่วงใด วัยทารกอาจถูกตีความว่าเป็นอายุระหว่าง 0 ถึง 1 ปี บางคนอาจตีความว่าวัยทารกอยู่ในช่วงอายุ 0 ถึง 2 ปี สิ่งเหล่าเป็นตัวอย่างของความไม่แน่นอน ซึ่งเป็นลักษณะทางธรรมชาติที่เกิดขึ้นทั่วไป เซตของเหตุการณ์ที่ไม่แน่นอนเช่นนี้เรียกว่า ฟัซซีเซต (Fuzzy Set)

จากแนวความคิดของ Zadeh เกี่ยวกับความไม่แน่นอนได้มีการขยายแนวคิดเพื่อนำไปประยุกต์ใช้ในด้านต่าง ๆ มากมายจนนับไม่ถ้วน ได้มีนักวิจัยได้คิดค้นทฤษฎีเสริมกับแนวคิดเดิมจนทำให้ฟัซซีเซตโดดเด่นในวงการคอมพิวเตอร์ ถึงแม้ว่าฟัซซีเซตจะนำเสนอจากคนอเมริกันแต่ประเทศอเมริกาก็ไม่ได้นำไปประยุกต์ใช้อย่างจริงจังในช่วงต้น ๆ แต่ประเทศญี่ปุ่นเล็งเห็นคุณค่าของศาสตร์ด้านนี้ได้เป็นผู้บุกเบิกฟัซซีเซตทางการค้า โดยได้นำไปประยุกต์ใช้ในเครื่องใช้ไฟฟ้ามากมาย เช่น เครื่องปรับอากาศ เครื่องซักผ้า หม้อหุงข้าว และอื่น ๆ ในยุคปัจจุบันประเทศสหรัฐอเมริกาได้ในความสำคัญกับศาสตร์นี้มากขึ้น โดยได้มีการทุ่มงบประมาณให้กับการวิจัยมากขึ้นและ ฟัซซีลอจิกถูกนำไปประยุกต์ใช้งานต่าง ๆ มากมาย ตัวอย่างเช่น ในโครงการอวกาศ NASA และโครงการด้านการทหาร

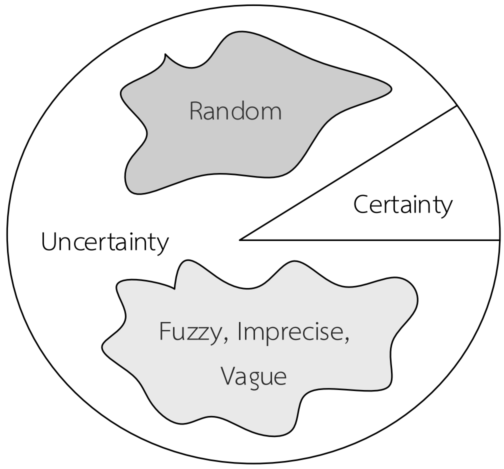
8.1 แนวคิดพื้นฐานเกี่ยวกับฟัซซีลอจิก

ฟัซซีลอจิกเป็นเครื่องมือที่ช่วยในการตัดสินใจภายในใต้ความไม่แน่นอนของข้อมูลโดยยอมให้มีความยืดหยุ่นได้ ใช้หลักเหตุผลที่คล้ายการเลียนแบบวิธีความคิดที่ซับซ้อนของมนุษย์ ฟัซซีลอจิกมีลักษณะที่พิเศษกว่าตรรกะแบบจริงเท็จ (Boolean Logic) เนื่องจากเป็นแนวคิดที่มีการต่อขยายในส่วนของความจริง (Partial True) โดยค่าความจริงจะอยู่ในช่วงระหว่างจริง (Completely True) กับเท็จ (Completely False) ส่วนตรรกศาสตร์เดิมจะมีค่าเป็นจริงกับเท็จเท่านั้น แสดงดังภาพที่ 8-1



ภาพที่ 8-1 ตรรกะแบบบูลีน และ ตรรกะแบบฟัซซี

ความเป็นฟัซซี (Fuzziness) มีชื่อเรียกว่า มัลติวาลานซ์ (Multivalance) ซึ่งมีค่าที่ความเป็นสมาชิกมากกว่า 2 ค่า และแตกต่างกับไบวาลานซ์ (Bivalance) ที่มีความเป็นสมาชิกเพียง 2 ค่า ฟัซซีเซตเป็นเครื่องมือทางคณิตศาสตร์ที่สื่อถึง “ความไม่แน่นอน (Uncertainty)” เช่น หากกำหนดว่า คนที่อ้วนคือคนที่มีน้ำหนักมากกว่า 75 กิโลกรัม คอมพิวเตอร์จะให้ผลว่าคนที่มีน้ำหนัก 74.50 กิโลกรัม ไม่เป็นคนที่อ้วน แต่สำหรับฟัซซีลอจิกโดยตามทฤษฎีของฟัซซีเซตจะใช้ลักษณะความหมายตัวแปร (Linguistic) มากกว่าปริมาณ (Quantitative) ของตัวแปร เช่น การหาความหมายของ “คนที่อ้วน” แต่ละคนจะไม่สามารถนิยามค่าความอ้วนที่ตรงกันและระบุเป็นหนึ่งเดียว (Identical) สำหรับคนที่อ้วนได้ เช่น นาย ก. จะให้ความหมายของ “คนอ้วน” หมายถึงคนที่มีน้ำหนักมากกว่า 70 กิโลกรัม นาย ข. ให้ความหมายว่าเป็นคนที่มีน้ำหนักมากกว่า 75 กิโลกรัม ซึ่งทั้งสองคนต่างแสดงความหมายของคำว่า “คนที่อ้วน” โดยเปรียบเทียบในมุมมองของตัวเองตามน้ำหนักของ และถึงแม้ว่าบุคคลที่มีน้ำหนัก 75 กิโลกรัม แต่หากพิจารณาจากกลุ่มคนที่มีน้ำหนักเฉลี่ย 90 กิโลกรัม บุคคลนี้ก็จะไม่จัดอยู่ในกลุ่มคนที่อ้วน แสดงให้เห็นว่า ความอ้วนไม่ได้มีลักษณะความไม่แน่นอนแบบสุ่ม



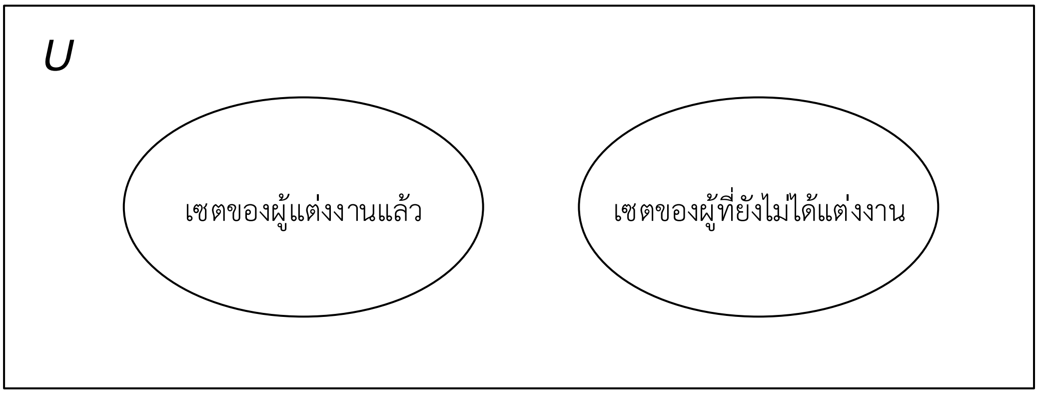
ภาพที่ 8-2 ความไม่แน่นอน Uncertainty

ภาพที่ 8-2 เป็นการแสดงให้เห็นว่าแนวทางในการตัดสินใจของปัญหาทั้งหมดมีเพียงส่วนน้อยที่เป็นสิ่งที่แน่นอน (Certainty) ที่เหลือคือสิ่งที่ไม่แน่นอนซึ่งประกอบด้วยความไม่แน่นอนที่มีลักษณะแบบสุ่มและความไม่แน่นอนที่มีลักษณะเป็นฟัซซี หรือคลุมเครือ ซึ่งมีมากกว่าร้อยละ 40 เพราะปัญหาส่วนมากเกี่ยวข้องกับการตัดสินใจของมนุษย์ซึ่งจะตัดสินใจตามพื้นฐานความคิดของตนเป็นหลัก

ฟัซซีจะสร้างวิธีทางคณิตศาสตร์ที่แสดงถึงความคลุมเครือ ความไม่แน่นอนของระบบที่เกี่ยวข้องกับความคิดความรู้สึกของมนุษย์ เมื่อพิจารณาส่วนประกอบต่าง ๆ ในความไม่แน่นอนเพื่อกำหนดเงื่อนไขในการตัดสินใจ (Decision Making) โดยอาศัยเซตของความไม่เป็นสมาชิก (Set Membership)

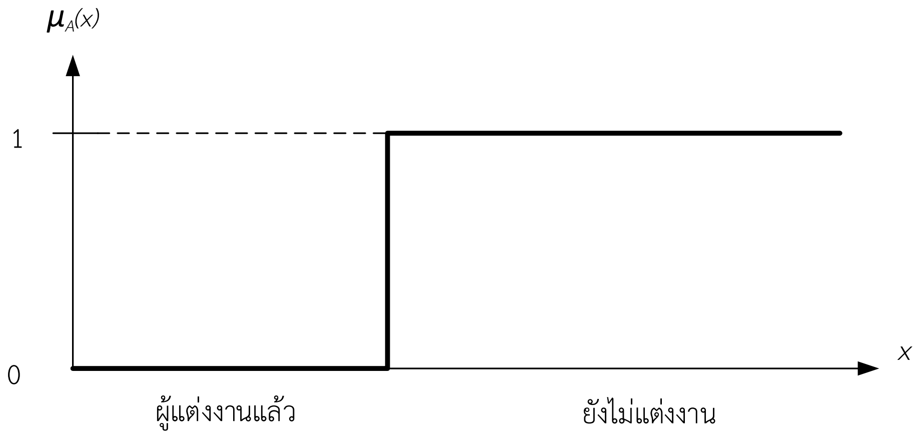
8.2 เซตแบบฉบับ

ในเซตแบบฉบับ (Classical Set) หรือเซตทวินัย (Crisp Set) เป็นเซตที่มีค่าความเป็นสมาชิกเป็น 0 หรือ 1 {0, 1} เท่านั้น เซตในทฤษฎีเซตแบบฉบับจะมีขอบเขตแบบแข็ง (Sharp Boundary) ซึ่งเป็นขอบเขตที่ตัดขาดจากกันแบบทันทีทันใด เซตแบบฉบับมีการกำหนดค่าความเป็นสมาชิกตามแนวคิดเลขฐานสอง โดยที่ตัวแปรหนึ่ง ๆ จะมีค่าความเป็นสมาชิกเพียงสองค่า คือ 0 ไม่เป็นสมาชิก และ 1 เป็นสมาชิก ตัวอย่างเช่น เซตของคู่แต่งงาน จะสามารถบอกได้ว่าอย่างแน่ชัดว่าเป็นกลุ่มผู้แต่งงานหรือไม่แต่งงาน



ภาพที่ 8-3 ตัวอย่างเซตแบบต้นฉบับ

ภาพที่ 8-3 แสดงตัวอย่างของเซตย่อยสองเซต คือเซตของผู้ที่แต่งงานและเซตของผู้ที่ไม่แต่งงานจะเห็นได้ว่าคนหนึ่งคนจะเป็นสมาชิกภาพได้เพียงเซตเดียวเท่านั้น แต่งงานหรือไม่แต่งงาน



ภาพที่ 8-4 ฟังก์ชันความเป็นสมาชิกในผู้ที่ไม่แต่งงาน

ภาพที่ 8-4 แสดงฟังก์ชันความเป็นสมาชิกของเซตผู้ที่ไม่แต่งงาน จากภาพจะเห็นได้ว่า ผู้ที่แต่งงานแล้วจะมีค่าความเป็นสมาชิกในเซตของผู้ไม่แต่งงานเป็น 0 ส่วนผู้ที่ไม่แต่งงานมีค่าความเป็นสมาชิกภาพของเซตผู้ที่ไม่แต่งงานเป็น 1 ค่าความเป็นสมาชิกของทั้งสองเซตจะตัดขาดจากกันอย่างทันทีทันใด รูปแบบคณิตศาสตร์ของเซตแบบฉบับมีรูปดังนี้

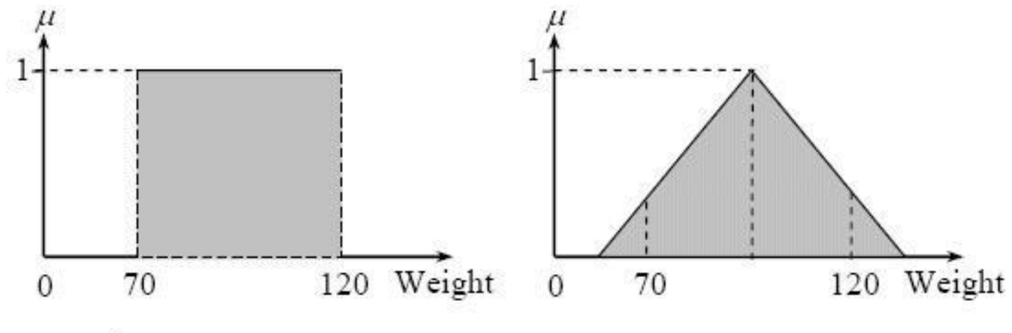
$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \in A \end{cases}$$

เมื่อ *A* เป็นเซตแบบฉบับหรือเซตแบบทวินัย *x* เป็นสมาชิกในเซต μ_A เป็นค่าความเป็นสมาชิกในเซต และ $\mu_A(x)$ เป็นฟังก์ชันความเป็นสมาชิกในเซต *A*

8.3 ฟัซซีเซต

ฟัซซีเซต เป็นเซตที่มีขอบเขตที่ราบเรียบ ทฤษฎีฟัซซีเซตจะครอบคลุมทฤษฎีเซตแบบฉบับ โดยฟัซซีเซตยอมให้มีค่าความเป็นสมาชิกของเซตระหว่าง 0 และ 1 ในโลกแห่งความเป็นจริงเซตไม่ใช่มีเฉพาะเซตแบบฉบับเท่านั้น จะมีเซตแบบฟัซซีด้วย ฟัซซีเซตจะมีขอบเขตแบบฟัซซีไม่ใช่เปลี่ยนแปลงทันทีทันใดจากขาวเป็นดำ ตัวอย่างเช่น เซตของคู่แต่งงานที่มีความสุข จะเห็นได้ว่าสมาชิกในเซตนี้จะไม่มีเฉพาะคู่แต่งงานที่มีความสุขระดับเดียวกันหมด บางคู่จะมีความสุขมาก บางคู่มีความสุขน้อย แตกต่างกันไป การใช้เซตแบบดั้งเดิมจึงไม่เหมาะสม

ยกตัวอย่างเกี่ยวกับความอ้วน นิยามคำว่าคนอ้วนในเซตทวินัยอาจกำหนดเป็นคนที่มีน้ำหนักตั้งแต่ 70 ถึง 120 กิโลกรัม โดยนิยามแบบฟัซซีเซตอาจกำหนดเป็นคนที่มีความอ้วนประมาณ 80 กิโลกรัม ซึ่งเป็นการให้นิยามที่ไม่แสดงถึงขอบเขตที่แน่นอน ดังแสดงในภาพที่ 8-5



ภาพที่ 8-5 การกำหนดค่าความเป็นสมาชิกของเซตทวินัยและเซตแบบฟัซซี

8.3.1 นิยามของฟัซซีเซต

กำหนดให้ *X* เป็นเซตที่ไม่ว่าง ฟัซซีเซต *A* สามารถแสดงลักษณะเฉพาะได้จากฟังก์ชันความเป็นสมาชิก

$$\mu_A(x): X \rightarrow [0,1]$$

เมื่อ μ_A สามารถตีความเป็นค่าของความเป็นสมาชิกภาพของตัวประกอบ *x* ในฟัซซีเซต *A* สำหรับแต่ละฟัซซีเซต (อ่านว่า “*x* เป็นสมาชิกของ *X*”) สามารถเขียนเป็นเซตของคู่ลำดับ (Tuples)

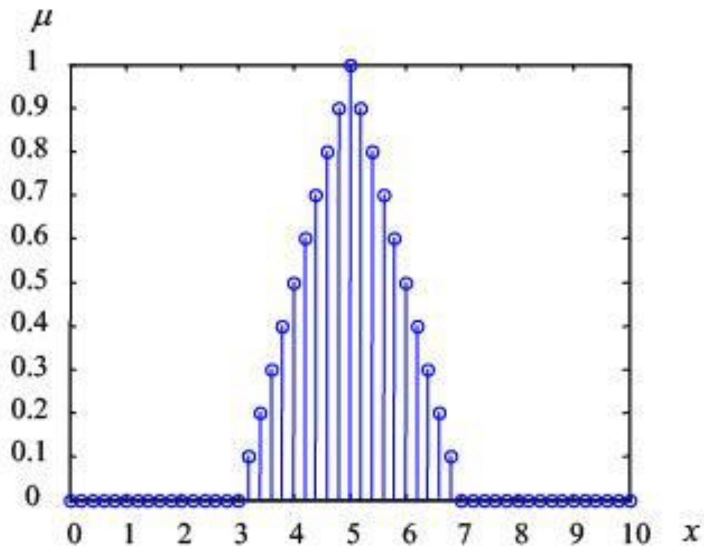
$$A = \{(x, \mu_A(x)) | x \in X\}$$

เมื่อ *A* หมายถึงสมาชิกของฟัซซีเซต (Set Membership), μ_A หมายถึง ฟังก์ชันความเป็นสมาชิก (Membership Function) บางครั้งแทนด้วย *A(x)* และ *X* หมายถึงเอกภพสัมพัทธ์ (Universe) หรือประชากร

ถ้า X = {x1, x2, x3, … ,xn} เป็นเซตจำกัด และ *A* เป็นฟัซซีเซตใน *X* ซึ่งเป็นชนิดวิยุต (Discrete) และจำกัด สัญกรณ์ (Notation) ของฟัซซีเซต เขียนได้เป็น

$$A = \left\{\frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots + \frac{\mu_A(x_n)}{x_n}\right\} = \left\{\sum_{i=0}^{n} \frac{\mu_A(x_i)}{x_i}\right\}$$

และสามารถแสดงฟังก์ชันความเป็นสมาชิกของเซตฟัซซีแบบวิยุต A ได้ดังภาพที่ 8-6

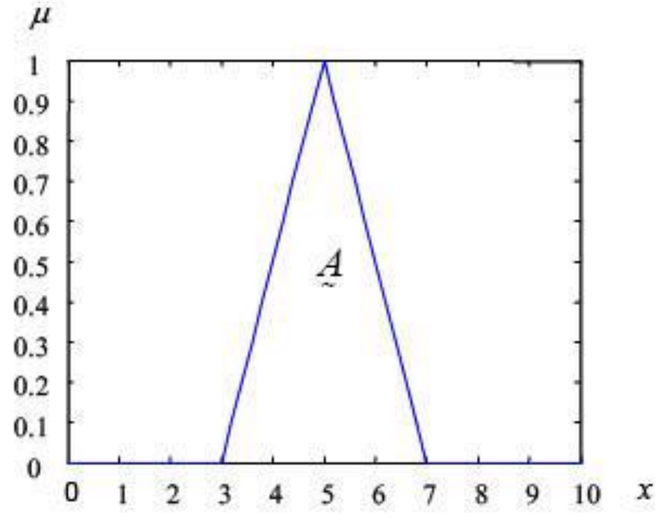


ภาพที่ 8-6 ฟังก์ชันความเป็นสมาชิกของเซตฟัซซีแบบวิยุต *A*

ถ้าเอกภพสัมพัทธ์ *X* เป็นค่าต่อเนื่อง (Continuous) สัญกรณ์ (Notation) ของฟัซซีเซต *A* เขียนได้เป็น

$$A = \left\{ \int \frac{\mu_{A(x)}}{x} \right\}$$

และสามารถแสดงฟังก์ชันความเป็นสมาชิกของเซตฟัซซีแบบต่อเนื่อง *A* ได้ดังภาพที่ 8-7



ภาพที่ 8-7 ฟังก์ชันความเป็นสมาชิกของเซตฟัซซีแบบต่อเนื่อง *A*

ทฤษฎีฟัซซีเซตสามารถแก้ปัญญาข้อจำกัดของเซตแบบดั้งเดิมได้ โดยฟัซซีเซตยอมให้มีค่าหรือดีกรีของความเป็นสมาชิก (Degree of Membership) ซึ่งแสดงด้วยค่าตัวเลขระหว่าง 0 และ 1 หรือเขียนเป็นสัญลักษณ์ [0, 1] โดย

- 0 หมายถึง ไม่เป็นสมาชิกในเซต และ 1 หมายถึง เป็นสมาชิกในเซต
- ค่าระหว่าง 0 กับ 1 เป็นสมาชิกบางส่วนในเซต

การทำเช่นนี้ ทำให้เกิดความราบเรียบในการเปลี่ยนจากพื้นที่นอกเซตไปอยู่ในเซตของสมาชิกต่างๆ โดยมีฟังก์ชันสมาชิก (Membership Function) เป็นฟังก์ชันจัดเทียบ (Mapping Function) วัตถุในโดเมนใด ๆ ให้เป็นค่าความเป็นสมาชิกในฟัซซีเซต

ความเป็นสมาชิกสำหรับ ฟัซซีเซต มีจำนวนระดับความเป็นสมาชิกเป็นอนันต์ คือค่าต่อเนื่องในช่วงตั้งแต่ 0 ถึง 1 ซึ่งครอบคลุมการกำหนดสมาชิกแบบฉบับ และเซตแบบฉบับหรือเซตทวินัย (crisp set) จะกำหนดตามดังสมการต่อไปนี้

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \in A \end{cases}$$

เมื่อ A เป็นเซตแบบฉบับ หรือ เซตแบบทวินัย, x เป็นสมาชิกในเซต, μ_A เป็นค่าความเป็นสมาชิกในเซต และ $\mu_A(x)$ เป็นฟังก์ชันความเป็นสมาชิกในเซต A

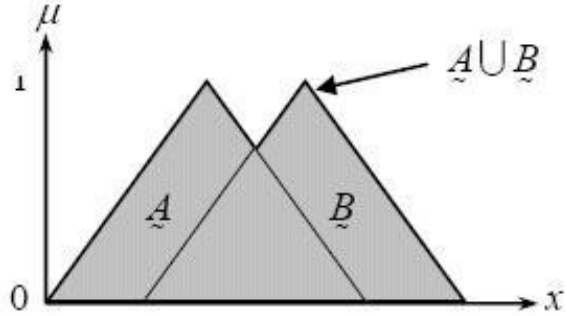
8.3.2 การดำเนินการทางฟัซซีเซต

การดำเนินการของฟัซซีเซตมีคุณสมบัติเหมือนกับเซตโดยทั่วไป มีการดำเนินการ (operation) อยู่ 3 อย่าง คือ Union, Intersection และ Complement

8.3.2.1 ยูเนียน (Union) จะเป็น OR operation ในสมการ แสดงตัวอย่างดังภาพที่ 8-8

$$\mu_{\underset{\sim}{A}\cup\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) \vee \mu_{\underset{\sim}{B}}(x)$$

$$= \max\left(\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right)$$

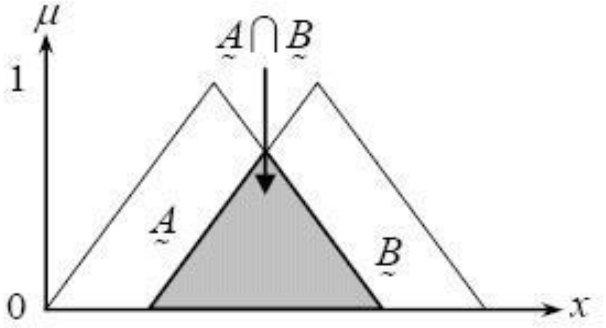


ภาพที่ 8-8 ยูเนียนของฟัซซีเซต A และ B

8.3.2.2 อินเตอร์เซกชัน (Intersection) จะเป็น AND operation ในสมการ แสดงตัวอย่างดังภาพที่ 8-9

$$\mu_{\underset{\sim}{A}\cap\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) \wedge \mu_{\underset{\sim}{B}}(x)$$

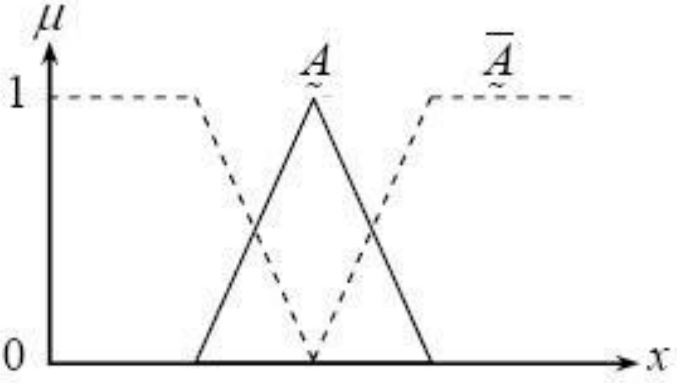
$$= \min\left(\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right)$$



ภาพที่ 8-9 Intersection ของฟัซซีเซต A และ B

8.3.2.3 คอมพลีเมนต์ (Complement) แสดงตัวอย่างดังภาพที่ 8-10

$$\mu_{\overline{\underset{\sim}{A}}}(x) = 1 - \mu_{\underset{\sim}{A}}(x)$$



ภาพที่ 8-10 Complement ของฟัซซีเซต *A*

8.3.3 คุณสมบัติของเซตฟัซซี

เซตฟัซซีมีคุณสมบัติตามเซตแบบฉบับ ได้แก่

- Commutativity $A \cup B = B \cup A$
  $A \cap B = B \cap A$
- Associativity $A \cup (B \cup C) = (A \cup B) \cup C$
  $A \cap (B \cap C) = (A \cap B) \cap C$
- Distributivity $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- Idemptency $A \cup A = A$ และ $A \cap A = A$
- Identity $A \cup 0 = A$ และ $A \cap X = A$
  $A \cap 0 = 0$ และ $A \cup X = X$
- Transitivity ถ้า $A \subseteq B$ และ $B \subseteq C$ แล้ว $A \subseteq C$
- Involution $\bar{\bar{A}} = A$

8.4 ฟังก์ชันความเป็นสมาชิก

ฟังก์ชันความเป็นสมาชิก (Membership function) เป็นฟังก์ชันที่มีการกำหนดระดับความเป็นสมาชิกของตัวแปรที่ต้องการใช้งาน โดยเริ่มจากการแทนที่กับตัวแทนที่มีความไม่ชัดเจน ไม่แน่นอน และคลุมเครือ ดังนั้นส่วนที่สำคัญต่อคุณสมบัติหรือการดำเนินการของฟัซซี เพราะรูปร่างของฟังก์ชันความเป็นสมาชิกมีความสำคัญต่อกระบวนการคิดและแก้ไขปัญหา โดยฟังก์ชันความเป็นสมาชิกจะไม่สมมาตรกันหรือสมมาตรกันทุกประการก็ได้

8.4.1 ชนิดของฟังก์ชันความเป็นสมาชิก

ชนิดของฟังก์ชันความเป็นสมาชิกที่ใช้งานทั่วไปมีหลายชนิด แต่ในที่นี้จะกล่าวถึงเพียงบาง 6 ชนิดดังนี้

8.4.1.1 ฟังก์ชันสามเหลี่ยม (triangular membership function) ฟังก์ชันสามเหลี่ยมมีทั้งหมด 3 พารามิเตอร์คือ {*a*, *b*, *c*}

$$\boldsymbol{Triangular(x\!:a,b,c)} = \begin{cases} \mathbf{0} & \boldsymbol{x < a} \\ \boldsymbol{(x-a)/(b-a)} & \boldsymbol{a \le x \le b} \\ \boldsymbol{(c-x)/(c-b)} & \boldsymbol{b \le x \le c} \\ \mathbf{0} & \boldsymbol{x > c} \end{cases}$$

8.4.1.2 ฟังก์ชันสี่เหลี่ยมคางหมู (trapezoidal membership function) ฟังก์ชันสี่เหลี่ยมคางหมูมีทั้งหมด 4 พารามิเตอร์คือ {*a*, *b*, *c*, *d*}

$$\boldsymbol{Trapezoidal(x\!:a,b,c)} = \begin{cases} \mathbf{0} & \boldsymbol{x < a} \\ \boldsymbol{(x-a)/(b-a)} & \boldsymbol{a \le x < b} \\ \mathbf{1} & \boldsymbol{b \le x < c} \\ \boldsymbol{(d-x)/(d-c)} & \boldsymbol{c \le x < d} \\ \mathbf{0} & \boldsymbol{x > c} \end{cases}$$

8.4.1.3 ฟังก์ชันเกาส์เซียน (Gaussian membership function) ฟังก์ชันเกาส์เซียนมีทั้งหมด 2 พารามิเตอร์คือ {*m*, σ} ซึ่ง *m* หมายถึงค่าเฉลี่ย และ σ หมายถึง ค่าเบี่ยงเบนมาตรฐาน

$$\boldsymbol{Guassian(x\!:m,\sigma) = exp - \frac{(x-m^2)}{\sigma^2}}$$

8.4.1.4 ฟังก์ชันระฆังคว่ำ (Bell-shaped membership function) ฟังก์ชันรูประฆังคว่ำมีพารามิเตอร์ทั้งหมด 3 ค่าคือ {*a*, *b*, *c*}

$$\boldsymbol{Bell - Shaped(x\!:a,b,c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}}$$

8.4.1.5 ฟังก์ชันตัวเอส (Smooth Membership Function) ฟังก์ชันรูปตัวเอสมีพารามิเตอร์ทั้งหมด 2 ค่าคือ {*a*, *b*}

$$\boldsymbol{s(x\!:a,b,c)} = \begin{cases} \mathbf{0} & \boldsymbol{x < a} \\ \mathbf{2}\left(\dfrac{\boldsymbol{x-b}}{\boldsymbol{b-a}}\right)^2 & \boldsymbol{a \le x \le \dfrac{a+b}{2}} \\ \mathbf{1-2}\left(\dfrac{\boldsymbol{x-b}}{\boldsymbol{b-a}}\right)^2 & \boldsymbol{\dfrac{a+b}{2} \le x < c} \\ \mathbf{1} & \boldsymbol{x \le b} \end{cases}$$

8.4.1.6 ฟังก์ชันตัวแซด (Z-membership function) ฟังก์ชันรูปตัวเอสมีพารามิเตอร์ทั้งหมด 2 ค่าคือ {*a*, *b*}

$$\boldsymbol{Z(x\!:a,b)} = \begin{cases} \mathbf{1} & \boldsymbol{x <} a \\ \mathbf{1-2}\left(\dfrac{\boldsymbol{x-b}}{\boldsymbol{b-a}}\right)^2 & \boldsymbol{a \le x \le \dfrac{a+b}{2}} \\ \mathbf{2}\left(\dfrac{\boldsymbol{x-b}}{\boldsymbol{b-a}}\right)^2 & \boldsymbol{\dfrac{a+b}{2} \le x <} c \\ \mathbf{0} & \boldsymbol{x \ge b} \end{cases}$$

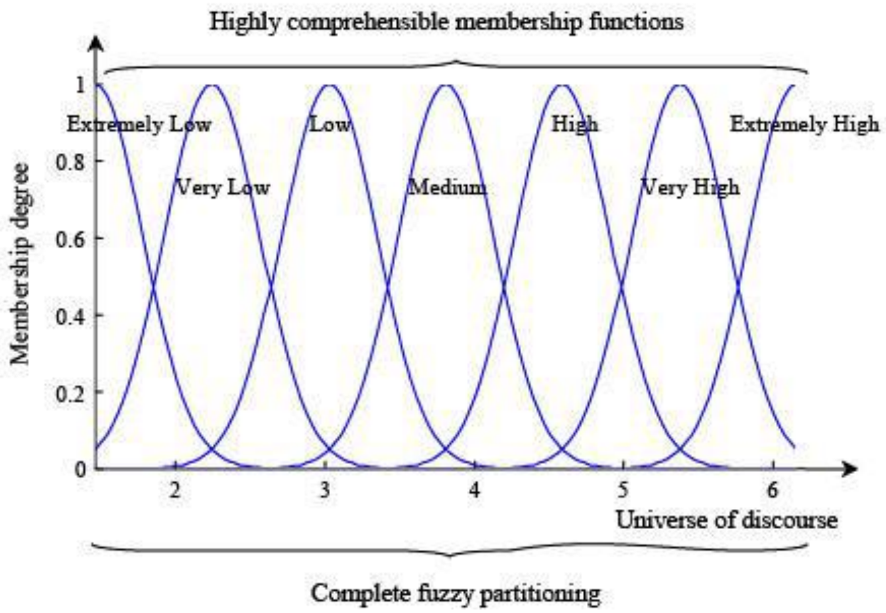
การเลือกฟังก์ชันของความเป็นสมาชิก จะต้องเลือกตามความเหมาะสมความครอบคลุมของข้อมูลที่จะรับเข้ามา โดยสามารถที่ทับซ้อนกันเพื่อให้การดำเนินงานราบเรียบ ซึ่งมีความเป็นสมาชิกหลายค่าได้ และฟังก์ชันความเป็นสมาชิกเปลี่ยนแปลงแก้ไขให้เหมาะกับงานที่กำลังปฏิบัติงานหรือตามความต้องการ

8.5 ตัวแปรภาษา (Linguistic Variable)

เซตแบบฟัซซีสามารถประยุกต์ใช้ในการอธิบายค่าของตัวแปรเช่นเดียวกับเซตแบบดั้งเดิม เช่น ประโยค “อุณหภูมิในห้อง*เย็น*” คำว่า “เย็น” เป็นคำที่ใช้แสดงปริมาณอุณหภูมิในทางรูปนัย สามารถเขียนได้เป็น *ปริมาณอุณหภูมิในห้องเย็น* หรือ *Temperature Quantity is Cold* ตัวแปร Temperature Quantity เป็นตัวแปรภาษา ซึ่งเป็นแนวคิดที่สำคัญมากในตรรกะแบบฟัซซี ตัวแปรภาษาช่วยกำหนดค่าของสิ่งที่จะอธิบายทั้งในรูปคุณภาพ โดยใช้พจน์ภาษา (linguistic term) และในรูปปริมาณ โดยใช้ฟังก์ชันความเป็นสมาชิก (membership function) ซึ่งแสดงความของของเซตแบบฟัซซี พจน์ภาษาใช้สำหรับการแสดง

แนวคิดและองค์ความรู้ในการสื่อสารของมนุษย์ ส่วนฟังก์ชันความเป็นสมาชิกมีประโยชน์ในการจัดการกับอินพุตที่เป็นข้อมูลเชิงตัวเลข

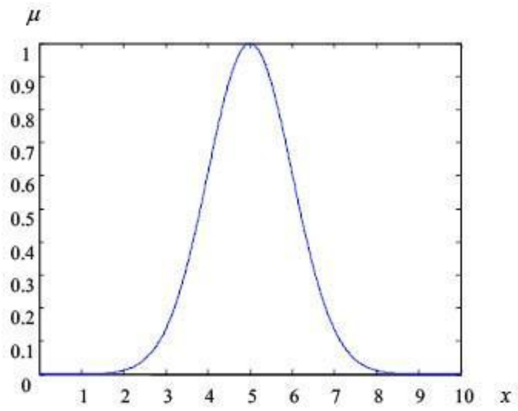
ตัวแปรภาษาเป็นการประกอบกัน (composition) ของตัวแปรสัญลักษณ์ (symbolic variable) และตัวแปรเชิงเลข (numerical variable) ตัวอย่างตัวแปรสัญลักษณ์ เช่น “รูปร่าง เป็น ทรงกระบอก” (Shape = Cylinder) คำว่า “รูปร่าง” เป็นตัวแปรที่บอกถึงรูปร่างของวัตถุ ตัวอย่างตัวแปรเชิงเลข เช่น “ความสูงเท่ากับ 4 ฟุต” (Height = 4) ตัวแปรเชิงเลขจะมีใช้กันในสาขางานด้านวิทยาศาสตร์ วิศวกรรมศาสตร์ คณิตศาสตร์ การแพทย์ และอื่น ๆ ส่วนตัวแปรสัญลักษณ์มีความสำคัญในวิทยาการเกี่ยวกับปัญญาประดิษฐ์และการตัดสินใจ การใช้ตัวแปรภาษาเป็นการรวมตัวแปรเชิงเลขกับตัวแปรสัญลักษณ์เข้าด้วยกัน ภาพที่ 8-11 แสดงตัวอย่างเซตตัวแปรภาษาของเซตฟัซซี ได้แก่ Extremely Low, Very Low, Low, Medium, High, Very High และ Extremely High



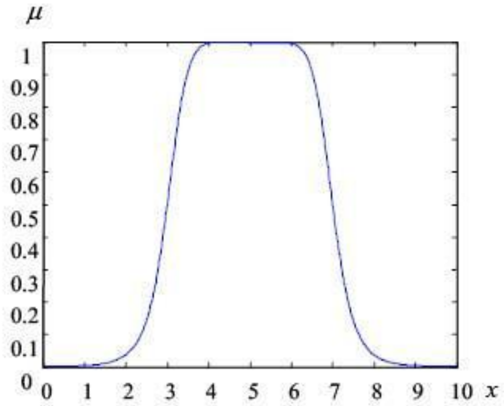
ภาพที่ 8-11 ตัวอย่างตัวแปรภาษา

8.6 กฎฟัซซี (Fuzzy Rules)

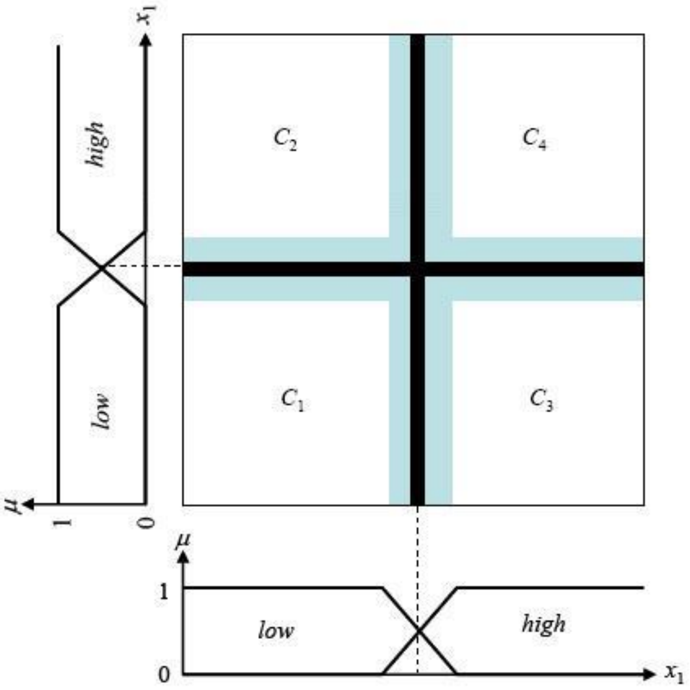
วิทยาการเกี่ยวกับฟัซซีลอจิกมีจำนวนมาก แต่ที่นิยมและการประยุกต์ใช้งานมากที่สุดเห็นจะได้แก่ กฎฟัซซีแบบถ้า-แล้ว (fuzzy if-then rule) ตัวอย่างการใช้กฎในการแยกกลุ่มดังแสดงในภาพที่ 8-12 และภาพที่8-13 สำหรับปริภูมิรูปแบบ (pattern space) การจัดกลุ่มด้วยกฎฟัซซีแสดงได้ดังภาพที่ 8-14



ภาพที่ 8-12 ฟังก์ชันเกาส์เซียน



ภาพที่ 8-13 ฟังก์ชันระฆังคว่ำ



ภาพที่ 8-14 ตัวอย่างปริภูมิรูปแบบการจัดกลุ่มด้วยกฎฟัซซี

จากภาพที่ 8-14 สามารถเขียนเป็นกฎในรูปประโยคภาษาได้ดังนี้

กฎข้อ 1: ถ้า x_1 มีค่า *low* และ x_2 มีค่า *low* แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_1

กฎข้อ 2: ถ้า x_1 มีค่า *low* และ x_2 มีค่า *high* แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_2

กฎข้อ 3: ถ้า x_1 มีค่า *high* และ x_2 มีค่า *low* แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_3

กฎข้อ 4: ถ้า x_1 มีค่า *high* และ x_2 มีค่า *high* แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_4

เมื่อ x_1 เป็นตัวแปรภาษาในมิติที่ 1, x_2 เป็นตัวแปรภาษาในมิติที่ 2, *low* และ *high* เป็นพจน์ภาษา (linguistic terms), ข้อมูล (x_1, x_2) เป็นคู่ลำดับของวัตถุที่ต้องการจัดกลุ่ม และ C_1, C_2, C_3 และ C_4 เป็นกลุ่มข้อมูล 1, 2, 3 และ 4

สมมุติให้กฎข้อ l, l = 1, 2, ..., L เป็นลำดับของกฎ ให้ข้อมูลเป็น $\mathbf{x} = [x_1, x_2, ..., x_n]$ เมื่อ n เป็นจำนวนมิติของข้อมูล ให้ $A_{l,i}$ เป็นพจน์ภาษาในกฎข้อที่ l มิติที่ i และให้กลุ่มข้อมูลเป็น C_k, k = 1, 2, ..., K รูปแบบทั่วไปของกฎฟัซซีสามารถเขียนได้ดังนี้

กฎข้อ 1: ถ้า x_1 มีค่า $A_{1,1}$ และ x_2 มีค่า $A_{1,2}$ และ ... และ x_n มีค่า $A_{1,n}$ แล้ว ข้อมูล $\mathbf{x}$ เป็นกลุ่ม C_1

กฎข้อ 2: ถ้า x_1 มีค่า $A_{2,1}$ และ x_2 มีค่า $A_{2,2}$ และ ... และ x_n มีค่า $A_{2,n}$ แล้ว ข้อมูล $\mathbf{x}$ เป็นกลุ่ม C_2

...

กฎข้อ l: ถ้า x_1 มีค่า $A_{l,1}$ และ x_2 มีค่า $A_{l,2}$ และ ... และ x_n มีค่า $A_{l,n}$ แล้ว ข้อมูล $\mathbf{x}$ เป็นกลุ่ม C_k

8.7 ระบบกฎแบบฟัซซี

ในระบบปัญญาประดิษฐ์มีวิธีการหลายวิธีในการที่จะแสดงองค์ความรู้ของมนุษย์ในรูปแบบต่าง ๆ เช่น ตรรกะ (logic) เฟรม (frames) โครงข่ายความหมาย (semantic nets) ภววิทยา (ontology) และกฎ (rules) ซึ่งแบบหลังสุดเป็นวิธีหนึ่งที่นิยมใช้ในระบบฟัซซี

8.7.1 รูปแบบกฎฟัซซี

ในระบบฟัซซีองค์ความรู้สามารถแสดงในรูปประโยค

ถ้า ข้อตั้ง (ข้อนำ) **ดังนั้น** ข้อยุติ (ข้อตาม)

IF premise (antecedent), THEN conclusion (consequent)

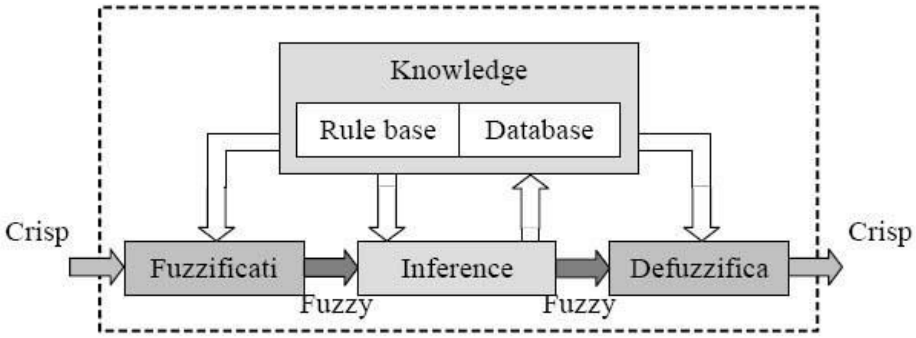
ข้อความข้างต้นเป็นที่รู้จักกันดีในนาม “รูปแบบฐานกฎถ้า-แล้ว” (IF-THEN rule-based form) หรือ รูปแบบนิรนัย (deductive form) ในรูปแบบการแสดงอนุมาน หากเราทราบความจริง (ข้อตั้ง ข้อสมมุติฐาน หรือข้อนำ) แล้วเราสามารถอนุมาน หรือหาข้อสรุปความจริงอีกอย่างหนึ่งที่เรียกว่าข้อยุติหรือข้อตาม การแสดงรูปแบบองค์ความรู้นี้ เรียกว่า *องค์ความรู้ตื้น* (shallow knowledge) ซึ่งค่อนข้างมีความเหมาะสมในบริบทของภาษา เนื่องจากเป็นการแสดงประสบการณ์ของมนุษย์และองค์ความรู้เชิงฮิวริสติก (heuristics) ในรูปแบบประโยคภาษามนุษย์ที่ใช้ในการสื่อสารทั่วไป แต่ไม่เป็นรูปแบบองค์ความรู้ที่ลึกล้ำ

แบบที่เป็นการรู้เอง เป็นโครงสร้าง เป็นฟังก์ชัน หรือเป็นพฤติกรรมของวัตถุรอบ ๆ ตัวเรา อย่างที่เรียกว่าอุปนัย (inductive)

ระบบกฎฟัซซีเป็นสิ่งที่มีประโยชน์ในการจัดรูปแบบของระบบที่ซับซ้อนที่สามารถสังเกตได้โดยมนุษย์ เพราะระบบเหล่านี้สามารถแสดงด้วยตัวแปรภาษาในข้อนำและข้อตามของกฎได้ ตัวแปรภาษาสามารถนำแสดงเชิงธรรมชาติด้วยฟัซซีเซตและตัวเชื่อมตรรกะของเซตเหล่านั้น

8.7.2 โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซีลอจิก

โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซี ซึ่งประกอบด้วยส่วนที่สำคัญ 4 ส่วนดังนี้ ดังภาพที่ 8-15

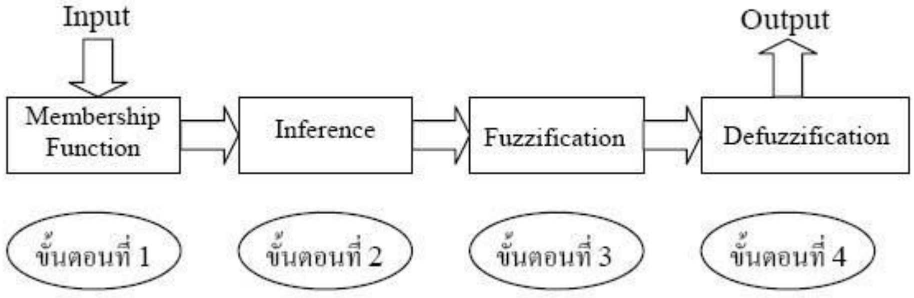


ภาพที่ 8-15 โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซี

- ส่วนที่แปลงข้อมูลอินพุตทั่วไปเป็นอินพุตแบบตัวแปรฟัซซี (Fuzzification) หรือในรูปแบบเซตฟัซซีหรือเรียกว่าเป็นตัวแปรภาษา (Linguistic Variable)
- ฐานความรู้ (Knowledge base) เป็นส่วนที่จัดเก็บรวบรวมข้อมูลในการควบคุมประกอบ 2 ส่วนคือ ฐานกฎ (Rule base) และฐานข้อมูล (Database)
  - ฐานกฎ (Rule base) ส่วนของการกำหนดวิธีการควบคุม ซึ่งได้จากผู้เชี่ยวชาญในรูปแบบของชุดข้อมูลแบบกฎของภาษา (Linguistic rule)
  - ฐานข้อมูล (Database) เป็นการจัดเตรียมส่วนที่จำเป็นเพื่อที่จะใช้ในการกำหนดกฎการควบคุม และการจัดการข้อมูลของตรรกศาสตร์ฟัซซี
- เครื่องอนุมานหรือการตีความ (Inference Engine) เป็นส่วนที่ทำหน้าที่ตรวจสอบข้อเท็จจริงและกฎ เพื่อใช้ในการตีความหาเหตุผล เหมือนกลไกสำหรับควบคุมการใช้ความรู้ในการแก้ไขปัญหา รวมทั้งการกำหนดวิธีการของการตีความเพื่อหาคาตอบ
- ส่วนที่แปลงการเอาต์พุตให้อยู่ในช่วงที่เหมาะสม (Defuzzification) เป็นการแปลงข้อมูลที่อยู่ในรูปแบบฟัซซีให้เป็นค่าที่สรุปผลหรือค่าการควบคุมระบบ

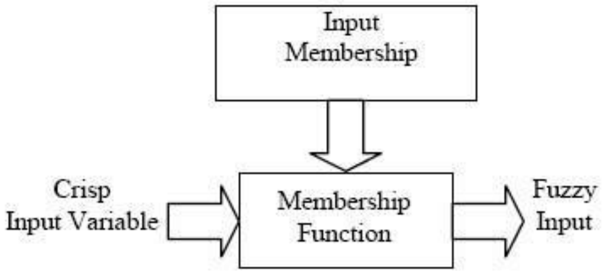
8.7.3 ขั้นตอนการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนการประมวลผลแบบฟัซซีลอจิกมีรูปแบบการทำงานเป็นแบ่งออกเป็น 4 ส่วนดังแสดงในภาพที่ 8-16



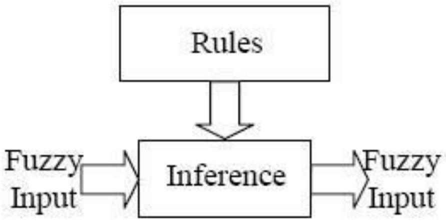
ภาพที่ 8-16 ขั้นตอนการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนที่ 1 เป็นการแปลงการอินพุตแบบทวินัยเปลี่ยนเป็นการอินพุตแบบตัวแปรฟัซซี โดยจะสร้างฟังก์ชันความเป็นสมาชิก โดยไม่จำเป็นต้องมีลักษณะเดียวกัน ขึ้นกับคุณลักษณะของแต่ละการอินพุต และความสำคัญต่อการเอาต์พุตที่น่าสนใจ โดยฟังก์ชันจะมีลักษณะเป็นการกำหนดภาษาสามัญเพื่อให้เป็นอินพุตแบบฟัซซี ดังภาพที่ 8-17



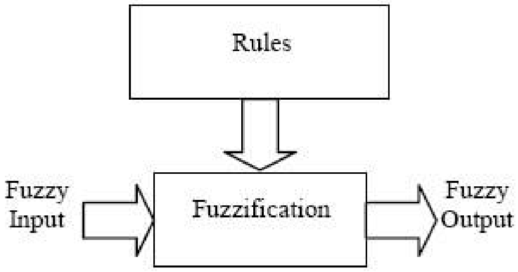
ภาพที่ 8-17 ขั้นตอนที่ 1 ของการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนที่ 2 เป็นการสร้างความสัมพันธ์ระหว่างการอินพุตทั้งหมดที่เกี่ยวข้องกับเอาต์พุตที่อาศัยหลักการของการหาเหตุและผล อาจจะสร้างการเก็บข้อมูล การคาดการณ์จากการตัดสินใจของมนุษย์หรือค่าจากการทดลอง โดยเขียนเป็นกฎการควบคุมระบบ ซึ่งจะมีลักษณะอยู่ในรูปแบบ ถ้า (If) และ (And) หรือ (Or) ซึ่งเป็นภาษาสามัญ นำกฎทั้งหมดมาประมวลผลรวมกัน เพื่อการหาตัดสินใจที่เหมาะสม ดังแสดงในภาพที่ 8-18



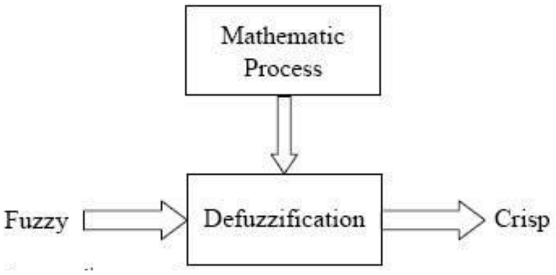
ภาพที่ 8-18 ขั้นตอนที่ 2 ของการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนที่ 3 เป็นการหาฟัซซีเอาต์พุต โดยการนำกฎการควบคุมที่สร้างขึ้น ในขั้นตอนที่ 2 มาประมวลผลกับฟัซซีอินพุต โดยใช้วิธีการทางคณิตศาสตร์ เพื่อนำค่าที่ได้ประมวลผล วิธีการทำเป็นค่าคลุมเครือ (Fuzzification) วิธีการที่นิยมใช้ในการตีความหาเหตุผลเลือกใช้ Max-Min method และ Max-Dot method ดังภาพที่ 8-19



ภาพที่ 8-19 ขั้นตอนที่ 3 ของการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนที่ 4 เป็นขั้นตอนสุดท้ายหรือขั้นตอนการสรุปเหตุผลฟัซซี โดยจะเปลี่ยนฟัซซีเอาต์พุตให้เป็นทวินัยเอาต์พุตตามภาพที่ 8-20 และด้วยวิธีทางคณิตศาสตร์ เช่น วิธีการหาจุดศูนย์ถ่วง (Central of Gravity) เพื่อนำค่าที่ได้มาใช้ในการตัดสินใจเพื่อควบคุมระบบในสถานการณ์นั้นๆ



ภาพที่ 8-20 ขั้นตอนที่ 4 ของการประมวลผลแบบฟัซซีลอจิก

วิธีการทำค่าฟัซซีให้เป็นค่าปกติ (Defuzzification) วิธีการที่เป็นเทคนิคการเลือกค่าสูงสุดหรือสรุปหาเหตุผลจากหลาย ๆ เซตมาเพียงค่าเดียว ซึ่งเป็นการใช้ค่าสูงสุดของค่าระดับการเป็นสมาชิกจากการกระทำหลายๆ แบบ และเลือกกระทำเพียงรูปแบบเดียว

วิธีการหาจุดศูนย์ถ่วง (Central of Gravity: COG) เป็นวิธีการเฉลี่ยผลที่ได้จากการตีความหาเหตุที่นิยมใช้ในปัจจุบัน จะหาได้จากการประมาณค่าจากสมการ

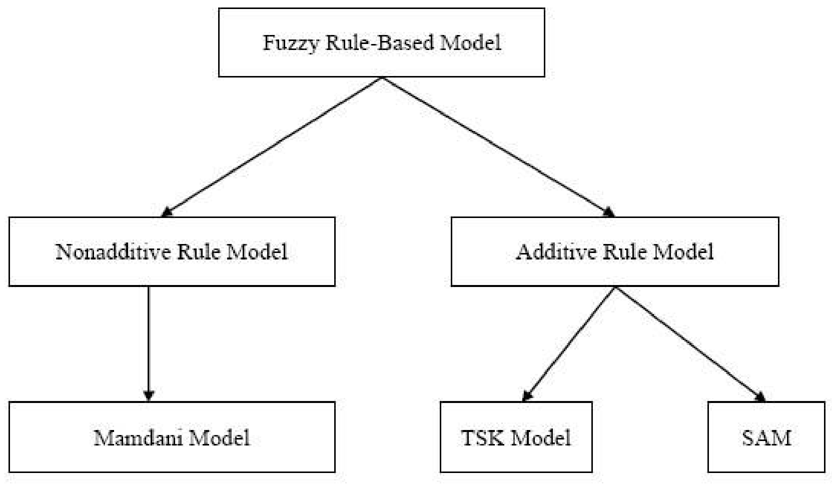
$$COG = \frac{\sum_{i=1}^{n} a_i w_i}{\sum_{i=1}^{n} a_i}$$

โดยสมการ ได้กำหนดค่าของสมการดังนี้

COG	แทน	ค่าของจุดศูนย์ถ่วง
N	แทน	ค่าตั้งแต่ตำแหน่งที่ 1 ถึงตำแหน่งที่ *i*
a_i	แทน	ค่าฟัซซีของเอาต์พุตในเซตฟัซซีของตำแหน่ง *i*
w_i	แทน	พื้นที่ใต้โค้งของเซตฟัซซีตำแหน่งที่ *i*

8.7.4 ชนิดของระบบกฎฟัซซี

ในการประมาณค่าฟังก์ชัน (function approximation) ระบบกฎฟัซซีที่ส่วนใหญ่ใช้กันมีด้วยกัน 3 ชนิด ได้แก่ Madani, Takagi Sugeno Kang (TSK) และ Standard Additive Model (SAM) รูปแบบ Madani รวมผลการอนุมาน (inference) ของกฎ โดยวิธีการซ้อนทับจากกฎหลาย ๆ ข้อ ซึ่งไม่เป็นแบบบวกกัน จึงเรียกระบบแบบนี้ว่าเป็น nonadditive rule model แต่สาหรับ TSK และ SAM มีการอนุนามแบบรวมค่าน้าหนักจากหลาย ๆ กฎ เพื่อรวมเป็นข้อสรุปสุดท้าย จึงเรียกระบบแบบนี้ว่า additive rule model การจัดกลุ่มของระบบกฎแบบฟัซซีแสดงในภาพที่ 8-21



ภาพที่ 8-21 กลุ่มของระบบฟัซซี

8.8 สรุป

ฟัซซีลอจิก คือ ตรรกะกำกวมหรือตรรกะคลุมเครือ นำมาใช้เชื่อมโยงระบบการคำนวณที่แตกต่างกันของสอบระบบ เช่น การทำงานระหว่างสมองกับเครื่องคอมพิวเตอร์ โดยมักถูกนำไปใช้งานกับระบบที่มีความแปรปรวนในกระบวนการดำเนินงานสูง ความคลุมเครือของตัวแปรต่างๆ ที่อาจเกิดจากการตีความที่แตกต่างกัน หรือเป็นข้อมูลที่มาจากการตีความในภาษาธรรมชาติ ด้วยเหตุนี้ฟัซซีลอจิก จึงช่วยให้การตีความดังกล่าวชัดเจนขึ้น โดยกำหนดค่าน้ำหนักเพื่อช่วยให้ระบุความแน่นอนของสิ่งที่ต้องการได้อย่างชัดเจน

แบบฝึกหัด

1. จงอธิบายว่าฟัซซีลอจิกคืออะไร และมีแนวคิดในการพัฒนานำมาใช้ในปัญญาประดิษฐ์อย่างไร
2. จงเขียนตัวอย่างงานที่สามารถนำฟัซซีลอจิกไปประยุกต์ใช้ได้ พร้อมทั้งประโยชน์ที่จะได้รับ

แผนปฏิบัติการสอนสัปดาห์ที่ 9

วิชา	030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์	**ระดับ**	ปริญญาตรี
เรื่อง	ระบบผู้เชี่ยวชาญและการอนุมานแบบฐานกฎ	**เวลา :**	บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. ความเป็นมาของระบบผู้เชี่ยวชาญ	IS 9.1
2. ความแตกต่างระหว่างระบบผู้เชี่ยวชาญกับโปรแกรมทั่วไป	IS 9.2
3. กฎและการเขียนกฎ	IS 9.3
4. กราฟ AND/OR	IS 9.4
5. การอนุมานข้อมูลแบบฐานกฎ	IS 9.5

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2-3-4-5**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 9
ระบบผู้เชี่ยวชาญและการอนุมานแบบฐานกฎ

ระบบผู้เชี่ยวชาญเป็นศาสตร์อีกแขนงหนึ่งของปัญญาประดิษฐ์ที่ได้รับความนิยมอย่างมาก โดยนำมาประยุกต์ใช้ในการแก้ไขปัญหาและตัดสินใจแทนมนุษย์ในด้านต่างๆ เช่น การแพทย์ เป็นต้น โดยอาศัยวิธีการต่างๆ จากหลักการสร้างฐานองค์ความรู้เนื่องจากเป็นแหล่งเก็บรวบรวมองค์ความรู้ที่มีประโยชน์ในการแก้ไขปัญหาที่ต้องเผชิญ ในบทนี้จะกล่าวถึงความเป็นมาของระบบผู้เชี่ยวชาญ และศาสตร์ที่มีความสำคัญต่อระบบผู้เชี่ยว

9.1 ความเป็นมาของระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญ (Expert System) หมายถึง ระบบหรือโปรแกรมคอมพิวเตอร์ที่นำองค์ความรู้ของผู้เชี่ยวชาญที่เป็นมนุษย์ มาจัดเก็บไว้เพื่อใช้ในการประมวลผล วิเคราะห์ และแก้ไขปัญหาต่างๆ ได้ด้วยตนเอง พร้อมทั้งให้คำแนะนำอย่างมีเหตุผล ระบบผู้เชี่ยวชาญถือว่าเป็นวิชาความรู้ส่วนหนึ่งของปัญญาประดิษฐ์ที่นำมาประยุกต์ใช้กับการประมวลผลของเครื่องคอมพิวเตอร์ให้มีประสิทธิภาพมากขึ้น จนมีความใกล้เคียงกับการตัดสินใจของมนุษย์

ระบบผู้เชี่ยวชาญถือกำเนิดจากการค้นคว้าและพัฒนาโครงการ MYCIN ของมหาวิทยาลัยสแตนต์ฟอร์ด ซึ่งเป็นโครงการที่มีจุดประสงค์ในการช่วยเหลือและวินิจฉัยผู้ติดเชื้อแบคทีเรียชนิดหนึ่ง ทำให้สามารถชี้ชัดได้ว่าแบคทีเรียชนิดใดเป็นสาเหตุของโรคอย่างแท้จริง ส่งผลให้การเยียวยารักษาผู้ติดเชื้อาทำได้รวดเร็วและทันท่วงที สำหรับการพัฒนาระบบผู้เชี่ยวชาญเริ่มตั้งแต่ ค.ศ. 1960 จนกระทั่ง ค.ศ. 1970 ซึ่งปัญญาประดิษฐ์เริ่มเข้ามามีบทบาทและถูกนำมาประยุกต์ใช้ในหลายๆ สาขา โดยนำองค์ความรู้มาใช้สนับสนุนการทำงาน จนถูกพัฒนาให้สอดคล้องกับระบบผู้เชี่ยวชาญ บางครั้งเรียกระบบผู้เชี่ยวชาญนี่ว่า “ระบบฐานองค์ความรู้ (Knowledge Based System)” ถือเป็นระบบที่นำเสนอองค์ความรู้เพื่อใช้สนับสนุนการตัดสินใจ

ต่อมาระบบผู้เชี่ยวชาญได้ถูกพัฒนาขึ้นมาจากเทคนิคของปัญญาประดิษฐ์ เพื่อเพิ่มประสิทธิภาพในการวิเคราะห์ให้มีความสามารถใกล้เคียงหรือเทียบเท่าผู้เชี่ยวชาญที่เป็นมนุษย์ และในปัจจุบันระบบผู้เชี่ยวชาญได้ถูกนำไปใช้งานในด้านต่างๆ จนกลายเป็นระบบช่วยตัดสินใจให้กับองค์กรหรือหน่วยงานต่างๆ ได้อย่างมีประสิทธิภาพ ตามความต้องการของแต่ละองค์กร

9.1.1 ระบบ MYCIN

MYCIN เป็นระบบผู้เชี่ยวชาญที่ถูกพัฒนาขึ้นในยุคแรก โดยสร้างขึ้นจากภาษา LISP (List Processing Language) และใช้วิธีการอนุมานแบบ Backward Chaining สำหรับค้นหาคำตอบจากองค์ความรู้ที่จัดเก็บไว้จำนวนมาก MYCIN จะอาศัยเพียงกฎที่มีความเชื่อมโยงสัมพันธ์กันประมาณ 500 กฎ โดยกฎเหล่านี้จะช่วยในการวินิจฉัยปัญหาต่างๆ ของผู้ป่วย เพื่อใช้จำแนกอาการและบ่งบอกถึงโรคของผู้ป่วยได้ โดยแต่ละกฎจะมีความสำคัญและหน้าที่แตกต่างกัน ซึ่งช่วยแก้ไขปัญหาการวินิจฉัยโดยแพทย์ที่อาจมีประสบการณ์ หรือความรู้แตกต่างกัน จนส่งผลให้ผลการตรวจไม่ตรงกับอาการป่วยที่แท้จริงได้

วิธีแก้ไขปัญหาของระบบ MYCIN มี 2 ขั้นตอน คือ

1. **ขั้นตอนการวินิจฉัย** ซึ่งจะเกี่ยวข้องกับการแยกแยะจำแนกอาการ สังเคราะห์ข้อมูล รวมถึงการวินิจฉัยความไม่แน่นอนของข้อมูลที่ได้รับจากผู้ป่วย เพื่อค้นหาคำตอบที่ถูกต้องต่อไป
2. **ขั้นตอนการแก้ปัญหา** จะเกี่ยวข้องกับการวานแผน เพื่อกำหนดแนวทางหรือขอบเขตการแก้ไขปัญหาที่พบ โดยสร้างทางเลือกการแก้ไขปัญหา เพื่อดำเนินการทดสอบทางเลือกดังกล่าว จนได้คำตอบที่น่าพอใจที่สุด

วิธีการดังกล่าวจะช่วยสนับสนุนการวินิจฉัยของแพทย์ได้ เนื่องจากข้อมูลการวินิจฉัยของแพทย์แบบดั้งเดิมจะได้มาจากการสอบถามอาการของผู้ป่วยโดยตรง ซึ่งอาการบางชนิดประกอบด้วยปัจจัยต่างๆ มากมาย ทำให้การวินิจฉัยของแพทย์แต่ละคนไม่สอดคล้องกัน และได้ผลสรุปที่ไม่ตรงกับอาการป่วยที่แท้จริง อีกทั้งโรคบางชนิดจำเป็นต้องอาศัยคำปรึกษาหรือคำแนะนำจากผู้เชี่ยวชาญ ทำให้ใช้เวลานานในการวินิจฉัยและระบุอาการของผู้ป่วย ส่งผลต่อวิธีรักษาที่ต้องล่าช้าออกไป ระบบ MYCIN จะดำเนินการวิเคราะห์ข้อมูลจากองค์ความรู้ที่จัดเก็บไว้ โดยให้ผู้ใช้สอบถามผ่านทางหน้าจอ และแสดงผลเป็นคำตอบหรือคำเสนอแนะที่เหมาะสมที่สุด ผลลัพธ์หรือวิธีการแก้ไขปัญหาที่ดีที่สุดจะแสดงแก่ผู้ใช้ตาม ข้อมูลที่ป้อนลงในระบบ ซึ่งเป็นปัจจัยสำคัญในการวิเคราะห์ข้อมูล

การโต้ตอบระหว่างผู้ใช้กับระบบ MYCIN จะมีลักษณธเสมือนกับผู้ใช้ขอคำปรึกษาจากผู้เชี่ยวชาญที่เป็นมนุษย์ แตกต่างกันตรงที่ผู้ใช้จะต้องใช้คำถามหรือคำตอบที่ค่อนข้างชัดเจน และตรงประเด็น เพื่อให้ระบบสามารถประมวลผลได้อย่างมีประสิทธิภาพในบางกรณีระบบอาจรับรู้ได้เองว่าข้อมูลที่ได้รับจากผู้ใช้นั้นผิดพลาด ระบบจะสอบถามข้อเท็จจริงจากผู้ใช้เพื่อยืนยันข้อมูลดังกล่าว

การแสดงคำตอบของระบ MYCIN นี้จะมีการระบุถึงความเชื่อมั่นในการตัดสินใจโดยใช้ค่าความแน่นอนของข้อมูลเพื่อแสดงว่าข้อมูลดังกล่าวควรเชื่อถือหรือไม่ ด้วยเหตุนี้ระบจึงนำเสนอคำตอบและแนวทางการแก้ปัญหาที่ดีที่สุดแก่ผู้ใช้ได้ รวมทั้งกรณีที่ระบบพบข้อมูลที่ไม่เคยปรากฎอยู่ในองค์ความรู้มาก่อน การแสดงผลลัพธ์จึงต้องอาศัยค่าความแน่นอนที่ผู้เชี่ยวชาญกำหนดไว้ เพื่อแสดงถึงความเชื่อมั่นของ

ข้อมูลซึ่งเป็นคำตอบที่ผู้ใช้เชื่อมั่นได้ว่าถูกต้องที่สุด ทำให้ผู้ใช้ได้รับผลลัพธ์ที่เหมาะสมกับการนำไปแก้ไขปัญหามากยิ่งขึ้น

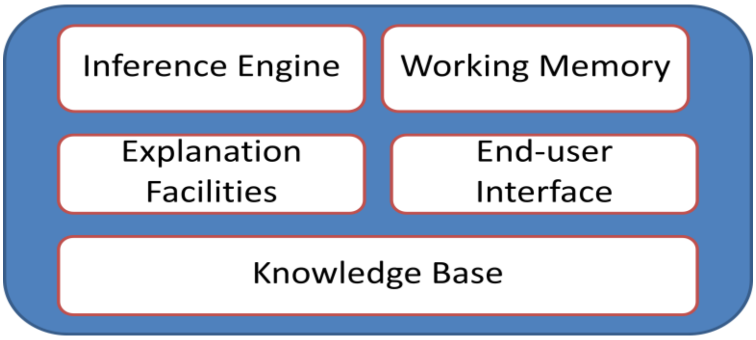
สำหรับแนวทางการประเมินผลของระบบ MYCIN แบ่งออกเป็น 2 ส่วนคือ

- **ส่วนการประเมินผล** เป็นการวัดผลของระบบโดยผู้เชี่ยวชาญเฉพาะด้านที่มีความรู้และประสบการณ์เกี่ยวข้องโดยตรงกับระบบ เพื่อเปรียบเทียบผลลัพธ์ของระบบกับผลลัพธ์ที่ได้จากผู้เชี่ยวชาญ ทำให้สามารถวัดประสิทธิภาพหรือความถูกต้องของระบบได้
- **ส่วนการทบทวนผลการประเมิน** เป็นขั้นตอนเปรียบเทียบผลของการประเมินระบบที่ได้จากขั้นตอนที่หนึ่ง เพื่อพิจารณาว่าผลดังกล่าวเป็นที่น่าพอใจหรือไม่ หากยังไม่เป็นที่น่าพอใจหรือยังไม่เหมาะสมตามความต้องการจึงเริ่มดำเนินการทบทวนผลของการประเมินดังกล่าวใหม่อีกครั้ง เพื่อให้ได้ระบบที่ประเมินผลแล้วน่าพอใจมากที่สุด

9.1.2 องค์ประกอบของระบบ MYCIN

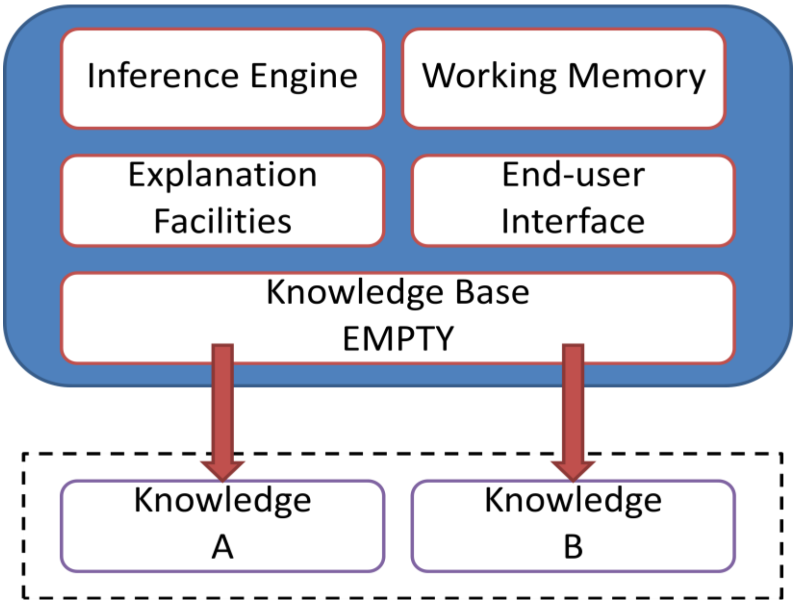
MYCIN มีองค์ประกอบสำคัญด้วยกัน 5 ส่วนดังแสดงในภาพที่ 9-1 ซึ่งประกอบไปด้วย

- **ฐานองค์ความรู้** (Knowledge Base) เป็นส่วนที่จัดเก็บองค์ความรู้ในเรื่องที่ต้องการได้
- **กลไกการอนุมาน** (Inference Engine) เป็นส่วนควบคุมการอนุมาน หรือสรุปความของระบบ
- **พื้นที่หน่วยความจำ** (Working Memory) เป็นส่วนการบันทึกข้อมูลไว้ในหน่วยความจำในขณะดำเนินการ
- **เครื่องมืออธิบายความ** (Explanation Facilities) เป็นส่วนตีความที่ช่วยผู้ใช้สามารถเข้าใจได้
- **ส่วนติดต่อกับผู้ใช้** (End-user Interface) เป็นส่วนแสดงผลบนหน้าจอแก่ผู้ใช้



ภาพที่ 9-1 องค์ประกอบของระบบ MYCIN

จะเห็นได้ว่าระบบ MYCIN จะมีฐานองค์ความรู้ที่ประกอบด้วยองค์ความรู้เพียงเรื่องใดเรื่องหนึ่งเท่านั้น ข้อมูลที่นอกเหนือจากนี้จะไม่ถูกจัดเก็บ ทำให้ระบบ MYCIN สามารถนำไปใช้งานได้เฉพาะด้านเท่านั้น ด้วยเหตุผลนี้จึงมีการพัฒนาให้ระบบ MYCIN มีความยืดหยุ่นมากขึ้น โดยสามารถนำองค์ความรู้ด้านอื่นๆ มาจัดเก็บเพิ่มได้อีก ซึ่งการทำงานของแต่ละองค์ความรู้จะเป็นอิสระต่อกัน การพัฒนารูปแบบดังกล่าวช่วยสนับสนุนให้ระบบผู้เชี่ยวชาญมีประสิทธิภาพและมีขอบเขตที่กว้างไกลมากขึ้น ระบบใหม่ที่ถูกพัฒนาขึ้นนี้เรียกว่า “EMYCIN” โดยมีโครงสร้างดังแสดงในภาพที่ 9-2 ซึ่งจะเห็นได้ว่าระบบสามารถรองรับองค์ความรู้ได้มากกว่า 1 องค์ความรู้ และสามารถเพิ่มเติมลงในฐานองค์ความรู้ว่าง (Empty) ได้ ข้อมูลของแต่ละองค์ความรู้อาจมีความเกี่ยวข้องและสัมพันธ์กันเพื่อช่วยเพิ่มประสิทธิภาพในการค้นหาคำตอบ การเสนอแนะแนวทางการแก้ไขปัญหา และการหาผลลัพธ์ให้ดีขึ้น เสมือนเป็นข้อมูลที่ได้จากผู้เชี่ยวชาญโดยตรง ระบบ MYCIN และ EMYCIN นี้สามารถสร้างประโยชน์ได้อย่างมากมายโดยเฉพาะวงการแพทย์ ซึ่งได้นำมาใช้และพัฒนาตั้งแต่ยุคแรกเริ่มจนถึงปัจจุบัน ระบบดังกล่าวถือได้ว่าเป็นต้นแบบของระบบผู้เชี่ยวชาญที่นำมาใช้กันอย่างแพร่หลายในวงการต่างๆ



ภาพที่ 9-2 องค์ประกอบของระบบ EMYCIN

9.2 ความแตกต่างระหว่างระบบผู้เชี่ยวชาญกับโปรแกรมทั่วไป

ระบบผู้เชี่ยวชาญเป็นการนำเสนอข้อมูลในรูปแบบที่แตกต่างจากโปรแกรมหรือซอฟต์แวร์โดยทั่วไป เนื่องจากใช้ฐานองค์ความรู้เป็นตัวจัดเก็บข้อมูลหรือองค์ความรู้ต่างๆ ไว้ ในขณะที่โปรแกรมทั่วไปจะจัดเก็บเพียงข้อมูลธรรมดาเท่านั้น นอกจากนี้การประมวลผลและการดำเนินงานของระบบผู้เชี่ยวชาญส่วนใหญ่จะขึ้นอยู่กับระบบหลัก แต่โปรแกรมทั่วไปจะต้องอาศัยมนุษย์ในการดำเนินกระบวนการต่างๆ เพื่อให้ได้ผลลัพธ์ที่ต้องการ ดังนั้นระบบผู้เชี่ยวชาญและโปรแกรมทั่วไปจึงมีความแตกต่างกันในหลายๆ ด้าน

ข้อแตกต่างระหว่างระบบผู้เชี่ยวชาญกับโปรแกรมคอมพิวเตอร์ทั่วไป

- ระบบผู้เชี่ยวชาญสามารถจำลองกระบวนการทางความคิดและการแก้ไขปัญหาเชิงเหตุผลแบบมนุษย์ได้ ในขณะที่โปรแกรมไม่สามารถทำได้
- ระบบผู้เชี่ยวชาญสามารถแก้ไขปัญหาด้วยข้อมูลและองค์ความรู้ที่มีความไม่แน่นอนหรือไม่สามารถรับประกันความสำเร็จได้ แต่โปรแกรมทั่วไปจะต้องใช้ข้อมูลที่มีความแน่นอนและถูกต้องเท่านั้น
- ระบบผู้เชี่ยวชาญสามารถตอบสนองกับรูปแบบของสัญลักษณ์ หรือภาษาธรรมชาติได้ดีกว่า แต่โปรแกรมทั่วไปจะตอบสนองได้ดีกับกระบวนการคำนวณด้วยตัวเลข
- การนำเสนอข้อมูลของระบบผู้เชี่ยวชาญจะทำได้ดีกว่าโปรแกรมทั่วไป เนื่องจากมีการแปลองค์ความรู้ของมนุษย์และคำนวณได้ในเชิงเหตุผล แต่โปรแกรมทั่วไปทำได้เพียงคำนวณและนำเสนอข้อมูลเท่านั้น

นอกจากระบบการทำงานและการประมวลผลข้อมูลแล้ว ระบบผู้เชี่ยวชาญยังมีความแตกต่างจากโปรแกรมทั่วไปในเรื่องการจัดเก็บข้อมูลอีกด้วย เนื่องจากระบบผู้เชี่ยวชาญจะอาศัยองค์ความรู้ ซึ่งถูกจัดเก็บไว้ในฐานองค์ความรู้ (Knowledge Base) ในขณะที่โปรแกรมทั่วไปการจัดเก็บข้อมูลจะอยู่ในฐานข้อมูล (Database) รูปแบบของข้อมูลภายในจึงแตกต่างกัน

ข้อแตกต่างระหว่างฐานองค์ความรู้กับระบบฐานข้อมูล

- องค์ความรู้เป็นข้อมูลระดับสูง คือข้อมูลในลักษณะนามธรรมไม่สามารถตีความได้โดยตรง แต่ข้อมูลที่เก็บในฐานข้อมูลจะสามารถนำเสนอข้อเท็จจริงได้ด้วยตัวเอง
- ข้อมูลที่ใช้กับฐานข้อมูลต้องมีความชัดเจน และแน่นอน ในขณะที่ฐานองค์ความรู้สามารถรองรับข้อมูลที่ไม่แน่นอนได้ ซึ่งมีการอนุมานที่ไม่สามารถทำได้ในระบบฐานข้อมูล
- ในฐานข้อมูลจะมีความสัมพันธ์ของข้อมูล แต่ในฐานองค์ความรู้จะอาศัยกฎ ตรรกะ เพื่อสร้างความสัมพันธ์
- จุดประสงค์ของฐานข้อมูลคือนำข้อมูลไปใช้งาน แต่ฐานองค์ความรู้มีจุดประสงค์เพื่อการวิเคราะห์ วางแผน และแก้ไขปัญหาต่างๆ

9.3 กฎและการเขียนกฎ

กฎในฐานกฎจะถูกเขียนเพื่อให้สามารถอนุมานโดยการให้เหตุผลด้วยฐานกฎ (Rule-Based Reasoning) ที่เป็นการอนุมานที่ใช้ หลักการของ Modus Ponens โดยกฎอยู่ในรูปแบบของ IF-THEN เรียกว่า production rules หรือ rules เช่น

IF ไฟข้ามถนน เป็น สีเขียว THEN ให้ข้ามถนนได้

9.3.1 องค์ประกอบของกฎ

กฎจะประกอบ 2 ส่วน คือ

- ส่วน IF : เรียกว่า antecedent (premise หรือ condition)
- ส่วน THEN : เรียกว่า consequent (conclusion หรือ action)

โดยมีรูปแบบไวยกรณ์อย่างง่ายคือ

IF <antecedent>
THEN <consequent>

ในทั่วไปแล้ว กฎสามารถมีหลาย antecedent ที่เชื่อมต่อกันด้วย AND, OR หรือ ผสมกัน แต่อย่างไรก็ตามควรจะหลีกเลี่ยงการเขียนตัวเชื่อมผสมกันในกฎ ตัวอย่างการเชื่อมต่อ condition ในกฎสามารถเขียนได้ดังนี้

IF condition$_1$ OR condition$_2$ .. OR condition$_n$ THEN consequent
IF condition$_1$ AND condition$_2$ .. AND condition$_n$ THEN consequent

ภายใน Antecedent หรือ Condition จะประกอบด้วยข้อมูล 2 ส่วนคือ

- Object เปรียบเสมือนตัวแปร
- Value คือค่าของตัวแปร Object

เช่น

IF ไฟข้ามถนน เป็น สีเขียว THEN ให้ข้ามถนนได้

- Object คือ ไฟข้ามถนน
- Value คือ สีเขียว

Object และ Value จะเชื่อมต่อกันด้วย Operator เช่น เป็น ไม่เป็น หรือ Operator สามารถเป็นเครื่องหมายของคณิตศาสตร์ได้ด้วย เช่น

IF 'age of the customer' < 18
AND 'cash withdrawal' > 1000
THEN 'signature of the parent' is required

9.3.2 การตีความและเขียนกฎ

กำหนดให้ผู้เชี่ยวชาญในการลงทุน เขียนข้อมูลเงื่อนไขเกี่ยวกับการลงทุนมา 5 ข้อดังนี้

- ถ้าคนมีเงิน 10000 บาทที่จะลงทุนและจบปริญญาตรี แล้วเขาควรลงทุนในความปลอดภัย
- ถ้าคนมีรายได้อย่างน้อย 40000 บาทต่อปีและจบปริญญาตรี แล้วเขาควรลงทุนในหุ้น
- ถ้าคนอายุน้อยกว่า 30 ปี และ ลงทุนในความปลอดภัย แล้วเขาควรลงทุนในหุ้น
- ถ้าคนอายุน้อยกว่า 30 ปี แล้วจะจบปริญญาตรี
- ถ้าคนที่ต้องการลงทุนในหุ้น แล้วควรจะลงหุ้น IBM

จากนั้นให้กำหนดตัวแปรสำหรับข้อความเพื่อให้สะดวกในการเขียนกฎ ซึ่งสามารถไล่ตามตัวอักษรภาษาอังกฤษได้ ข้อควรระวังถ้าใช้ตัวแปรมากกว่า 1 ตัวในการแทนประโยคเดียวกัน โดยสามารถกำหนดได้ดังนี้

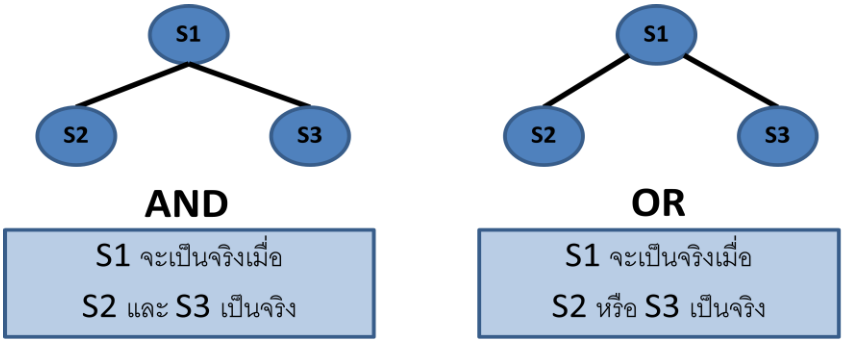
- A แทนประโยค คนมีเงิน 10000 บาทที่จะลงทุน
- B แทนประโยค จบปริญญาตรี
- C แทนประโยค ลงทุนในความปลอดภัย
- D แทนประโยค มีรายได้อย่างน้อย 40000 บาทต่อปี
- E แทนประโยค ลงทุนในหุ้น
- F แทนประโยค อายุน้อยกว่า 30 ปี
- G แทนประโยค ลงหุ้น IBM

ดังนั้นจากประโยคที่ผู้เชี่ยวชาญข้างต้นสามารถนำมาเขียนในอยู่ในรูปแบบฐานกฎได้ดังนี้

- IF A AND B THEN C
- IF D AND B THEN E
- IF F AND C THEN E
- IF F THEN B
- IF E THEN G

9.4 กราฟ AND/OR

ในบางครั้งการแทนความรู้ให้อยู่ในรูปแบบกฎเลยนั้นจะซับซ้อน จึงมีการแทนความรู้ด้วยกราฟให้เข้าใจถึงโครงสร้างรวมของความรู้ เรียกกราฟประเภทนี้ว่า “กราฟ AND/OR” ดังนั้นถ้ารู้เป้าหมายของระบบ ก็จะสามารถแตกเป้าหมาย (goal) ซึ่งเป็นสิ่งที่ต้องการนั้น ออกเป็นเป้าหมายย่อยๆ (sub-goals) ซึ่งเป็นคือสิ่งที่ต้องได้เพื่อทำให้ได้เป้าหมายที่ต้องการ กราฟ AND/OR จะประกอบด้วย 2 คุณลักษณะคือ กราฟที่เชื่อมด้วยเงื่อนไข AND และกราฟที่เชื่อมด้วยเงื่อนไข OR ดังแสดงในภาพที่ 9-3

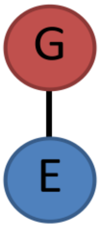


ภาพที่ 9-3 กราฟเงื่อนไข AND และกราฟเงื่อนไข OR

ตัวอย่างการสร้างกราฟ AND/OR จากฐานกฎการลงทุนกับ IBM โดยมีกฎต่างๆ ที่เขียนมาดังนี้

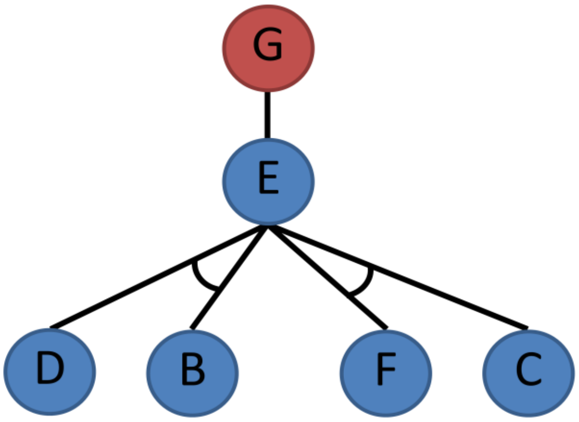
- กฎข้อที่ 1 : IF A AND B THEN C
- กฎข้อที่ 2 : IF D AND B THEN E
- กฎข้อที่ 3 : IF F AND C THEN E
- กฎข้อที่ 4 : IF F THEN B
- กฎข้อที่ 5 : IF E THEN G

ซึ่งเป้าหมายที่ต้องการคือ G ที่แทนประโยค ลงหุ้น IBM ดังนั้นเริ่มต้นจากการหากฎที่ให้ผลลัพธ์ (ค่าหลัง THEN) เป็น G โดยจะพบว่าคือกฎข้อที่ 5 ซึ่งหมายความว่า E จะต้องเป็นจริงก่อนซึ่งจะเป็นเงื่อนไขที่นำมาสู่ G เป็นจริง ดังนั้นจะสามารถเขียนกราฟได้ดังแสดงในภาพที่ 9-4



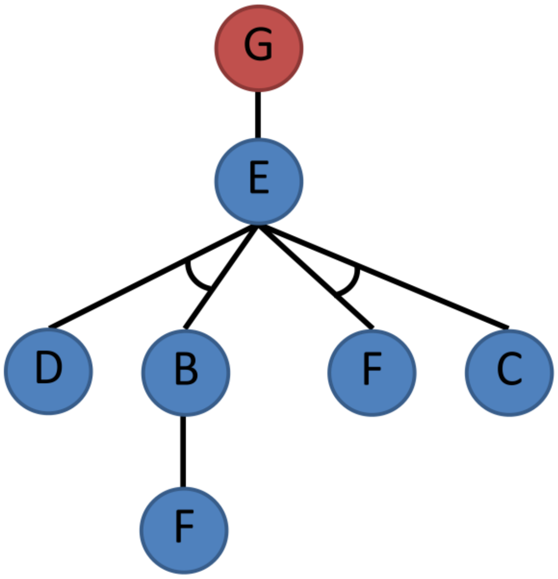
ภาพที่ 9-4 กราฟ AND/OR การลงทุนกับ IBM ขั้นตอนที่ 1

จากนั้นให้หากฎที่จะทำให้ E เป็นจริง (ดูกฎที่มีค่าหลัง THEN เป็น E) จะพบว่ามีกฎข้อที่ 2 และกฎข้อที่ 3 ดังนั้นจึงสามารถเขียนกราฟต่อเติมจากเดิมได้ดังแสดงในภาพที่ 9-5



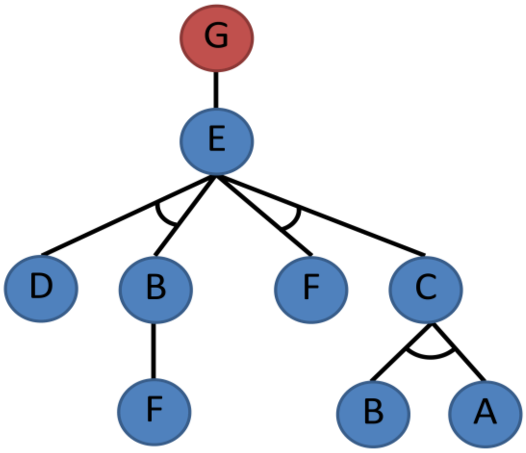
ภาพที่ 9-5 กราฟ AND/OR การลงทุนกับ IBM ขั้นตอนที่ 2

จากนั้นเพื่อให้การสร้างกราฟ AND/OR มีสะดวกมากขึ้นให้พิจารณาการแตกกิ่งก้านจากซ้ายไปขวา ซึ่งในที่นี้ให้พิจารณาที่ D จะเห็นว่าไม่มีกฎข้อไหนที่ให้ผลลัพธ์เป็น D ดังนั้นข้อมูลนี้จะต้องสอบถามจากผู้ใช้งาน ต่อมาพิจารณาที่ B พบว่ากฎข้อที่ 4 จะเป็นเหตุการณ์ที่ทำให้เกิด B ดังนั้นสามารถเขียนกราฟต่อเติมจากเดิมได้ดังแสดงในภาพที่ 9-6



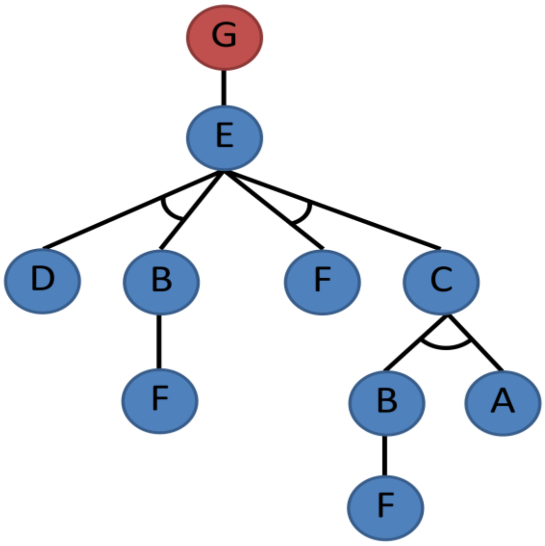
ภาพที่ 9-6 กราฟ AND/OR การลงทุนกับ IBM ขั้นตอนที่ 3

จากนั้นให้พิจารณาจากโหนดที่เกิดขึ้นใหม่คือ F ซึ่งพบว่าไม่มีกฎข้อไหนที่ให้ผลลัพธ์เป็น F ดังนั้นโหนด F ถือเป็นโหนดใบ จากนั้นพิจารณาโหนดถัดไปทางด้านขวาซึ่งก็คือ F ที่เป็นโหนดใบอีกที่ไม่สามารถแตกกิ่งก้านต่อไปได้ ดังนั้นจึงเหลือโหนดสุดท้ายที่อยู่ทางขวาสุดก็คือโหนด C ซึ่งมีกฎข้อที่ 1 รองรับจึงสามารถเขียนกราฟต่อเติมจากเดิมได้ดังแสดงในภาพที่ 9-7



ภาพที่ 9-7 กราฟ AND/OR การลงทุนกับ IBM ขั้นตอนที่ 4

ต่อมาพิจารณาจากโหนดที่เกิดใหม่คือโหนด B ซึ่งพบว่าตรงกับกฎข้อที่ 4 จึงสามารถแตกกิ่งก้านเพิ่มเติมจากเดิมได้ดังแสดงในภาพที่ 9-8



ภาพที่ 9-8 กราฟ AND/OR การลงทุนกับ IBM ขั้นตอนที่ 5

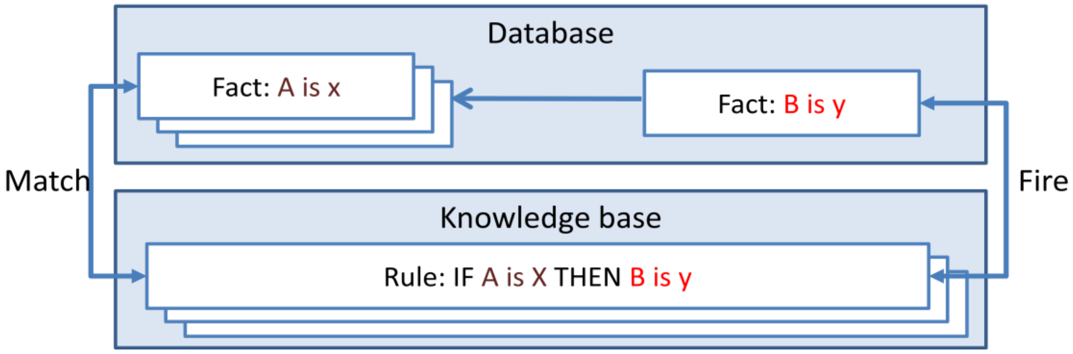
ซึ่งเมื่อทำต่อจะพบว่าทั้ง F และ A เป็นโหนดใบที่ไม่สามารถแตกกิ่งก้านต่อไปได้อีก ดังนั้น การสร้างกราฟ AND/OR จึงเสร็จสิ้นลงแต่เพียงเท่านี้

9.5 การอนุมานข้อมูลแบบฐานกฎ

การอนุมานแบบฐานกฎจะเปรียบเทียบแต่ละกฎที่เก็บอยู่ในฐานองค์ความรู้กับข้อเท็จจริง เมื่อกฎที่อยู่ในส่วนของ IF ตรงกับข้อเท็จจริง (match) กฎนั้นจะถูกเรียกใช้งาน (fired) และ ส่วนของ THEN จะถูกทำงาน ซึ่งมีรากฐานมาจากกฎ Resolution คือ

$$(\neg A \Rightarrow B) \land (B \Rightarrow C) \text{ อนุมานได้เป็น } (\neg A \Rightarrow C)$$

โดยจะมีลักษณะการทำงานดังแสดงในภาพที่ 9-9 กล่าวคือในฐานข้อมูล (Database) จะบรรจุข้อเท็จจริงของประโยคไว้ เรียกว่า Fact จากนั้นจะเปรียบเทียบข้อมูลใน Fact ว่าตรงกับกฎข้อไหนในฐานองค์ความรู้หรือไม่ จากภาพทราบข้อเท็จจริงว่า “A is x” และในฐานองค์ความรู้ก็มีกฎว่า “IF A is x THEN B is y” อยู่ ทำให้กฎเกิดการ “match” กับข้อเท็จจริง และใช้กฎ Resolution เพื่อหาข้อเท็จจริงจะได้ข้อมูลหลัง THEN ซึ่งก็คือ “B is y” เป็นจริงด้วย กรณีนี้เรียกว่าการ “Fire” กฎ จากนั้นนำข้อเท็จจริงใหม่ที่ได้บันทึกลงในฐานข้อมูลเพื่อใช้งานต่อไป



ภาพที่ 9-9 โครงสร้างการทำงานของกลไลการอนุมานข้อมูลแบบฐานกฎ

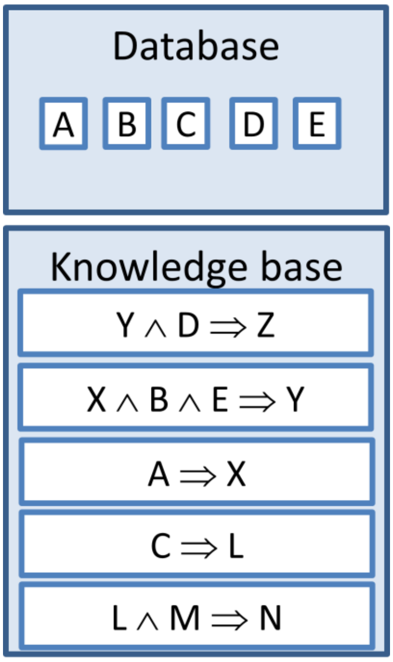
ในการอนุมานข้อมูลที่เก็บอยู่ในรูปแบบฐานกฎนั้นมี 2 วิธีหลักๆคือ

- **Forward chaining** เรียกอีกอย่างหนึ่งว่า “Data Driven Search” เนื่องจากมุ่งเน้นความสำคัญที่ข้อมูลเป็นหลัก (Data Driven) โดยเริ่มจากการค้นหาข้อมูลที่มีอยู่หรือแนวคิดพื้นฐาน เพื่อให้ได้ผลลัพธ์ของการอนุมาน จะทำการทดสอบกฎต่างๆก่อนที่จะประมวลผลจนกว่าจะได้ข้อสรุป
- **Backward chaining** เรียกอีกอย่างหนึ่งว่า “Goal Driven Search” เนื่องจากเป็นวิธีการที่เน้นความสำคัญตรงเป้าหมายเป็นหลัก โดยเริ่มค้นหาจากสิ่งที่คาดหวังไปยังหลักฐานที่จะบ่งบอกได้ว่าเป็นไปอย่างที่คาดหวัง**หรือไม่**

กำหนดให้ฐานกฎมีกฎจำนวน 5 กฎดังนี้

- Rule 1: Y $\wedge$ D $\Rightarrow$ Z
- Rule 2 : X $\wedge$ B $\wedge$ E $\Rightarrow$ Y
- Rule 3: A $\Rightarrow$ X
- Rule 4: C $\Rightarrow$ L
- Rule 5: L $\wedge$ M $\Rightarrow$ N

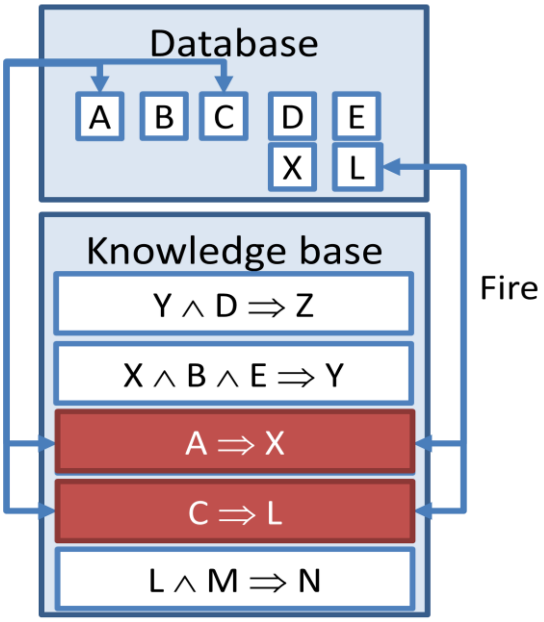
พร้อมทั้งกำหนดข้อเท็จจริงในฐานข้อมูล คือ A, B, C, D และ E (เป็นจริง) ต้องการพิสูจน์ว่า Z เป็นจริงหรือไม่ โดยสามารถนำข้อมูลข้างต้นที่กล่าวมาเขียนให้เป็นรูปภาพได้ดังแสดงในภาพที่ 9-10



ภาพที่ 9-10 แผนผังแสดงข้อมูลในฐานข้อมูลและฐานองค์ความรู้ของปัญหาตัวอย่าง

9.5.1 วิธีการอนุมานแบบ Forward Chaining

ในการอนุมานหาผลลัพธ์ของวิธี Forward Chaining จะใช้ผลลัพธ์เป็นเพียงเงื่อนไขเพื่อหยุดการทำงานของการอนุมาน ซึ่งในขั้นตอนของการ Match หรือ Fire กฎนั้นเป็นเพียงการเปรียบเทียบกฎกับข้อเท็จจริง โดยไม่ขึ้นอยู่กับผลลัพธ์ใดๆ ทั้งสิ้น การทำงานรอบแรกหรืออาจจะเรียกว่า Cycle ที่ 1 จะทำการไล่กฎในฐานองค์ความรู้ทีละกฎ และตรวจสอบดูว่ามีกฎใดที่ match ข้อเท็จจริงหรือไม่ ถ้ามีกฎที่ match แล้วให้ fire ข้อเท็จจริงใหม่ออกมา สิ่งที่ควรระวัง คือถ้ากฎไหนที่ถูก fire ออกมาแล้วจะไม่ถูกนำไปใช้ในการ match ซ้ำ



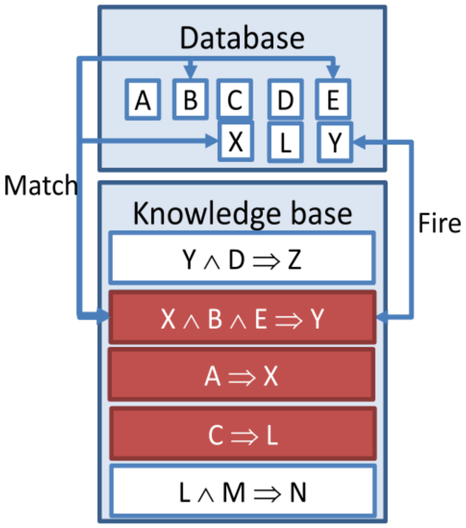
ภาพที่ 9-11 ข้อมูลในฐานข้อมูลและฐานองค์ความรู้ของปัญหาตัวอย่างใน Cycle ที่ 1

การทำงาน Cycle 1 แสดงดังภาพที่ 9-11 :

- กฎข้อที่ 1 ไม่สามารถ match ได้ ถึงแม้จะรู้ว่า D เป็นจริง เนื่องจากระบบยังขาดความรู้เกี่ยวกับ Y
- กฎข้อที่ 2 ไม่สามารถ match ได้เนื่องจากระบบยังขาดความรู้เกี่ยวกับ X
- กฎข้อที่ 3 จะสามารถ match ได้เพราะทราบข้อเท็จจริงของ A และสามารถ fire ข้อเท็จจริงใหม่ซึ่งก็คือ X เข้าไปในฐานข้อมูล
- กฎข้อที่ 4 จะสามารถ match ได้เพราะทราบข้อเท็จจริงของ C และสามารถ fire ข้อเท็จจริงใหม่ซึ่งก็คือ L เข้าไปในฐานข้อมูล
- กฎข้อที่ 5 ไม่สามารถ match ได้ ถึงแม้จะเพิ่งรู้ว่า L เป็นจริง เนื่องจากระบบยังขาดความรู้เกี่ยวกับ M

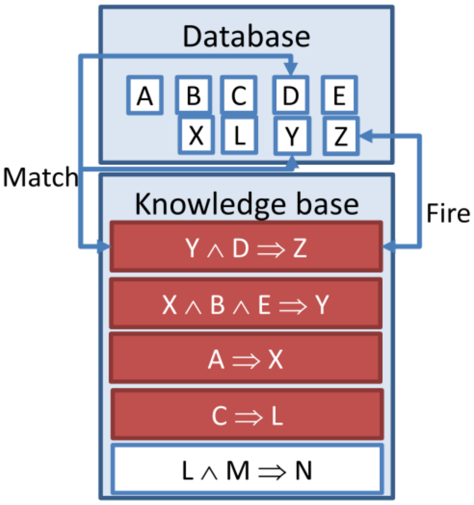
หลังจากสิ้นสุด Cycle ที่ 1 แล้วจะพบว่ามีกฎจำนวน 2 กฎจากฐานองค์ความรู้ที่ถูก match และได้ fire ข้อเท็จจริงใหม่คือ X และ L ไปยังฐานข้อมูล เนื่องจากยังไม่มีข้อเท็จจริงของเป้าหมายที่ต้องการ (Z) จึงทำการเปรียบเทียบกฎกับข้อเท็จจริงซ้ำอีก กลายเป็นการทำงานใน Cycle ที่ 2 โดยแสดงดังภาพที่ 9-12

การทำงาน Cycle ที่ 2 จะพบกฎข้อที่ 3 ที่สามารถ match เพิ่มขึ้นมา เนื่องจากใน Cycle แรกทำให้ทราบข้อเท็จจริงเกี่ยวกับ X ดังนั้นเมื่อกฎข้อที่ 3 นี้ ถูก fired ออกมาทำให้ได้ข้อเท็จจริงใหม่ ซึ่งก็คือ Y



ภาพที่ 9-12 ข้อมูลในฐานข้อมูลและฐานองค์ความรู้ของปัญหาตัวอย่างใน Cycle ที่ 2

ในเมื่อยังไม่ได้ข้อเท็จจริง Z ที่ต้องการจึงต้องทำการเปรียบเทียบกฎซ้ำอีกครั้ง ซึ่งจะนับเป็น Cycle ที่ 3 แสดงดังภาพที่ 9-13 โดยจะเห็นได้ว่า กฎข้อที่ 1 จะถูก match เนื่องจากในขณะนี้ เราทราบถึงข้อเท็จจริง Y (จาก Cycle ที่ 1) และข้อเท็จจริง D (จากความรู้เริ่มต้น) ทำให้กฎข้อนี้ถูก fired และให้ข้อเท็จจริง Z ออกมา ซึ่งเป็นข้อเท็จจริงที่เราต้องการพิสูจน์ ดังนั้นการทำงานของ Forward Chaining จึงหยุดเพียงเท่านี้ พร้อมทั้งสามารถสรุปการพิสูจน์ได้ว่า Z เป็นจริง

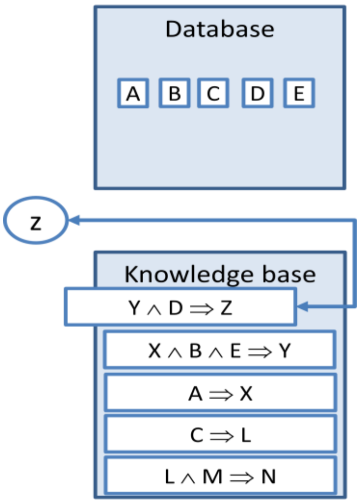


ภาพที่ 9-13 ข้อมูลในฐานข้อมูลและฐานองค์ความรู้ของปัญหาตัวอย่างใน Cycle ที่ 3

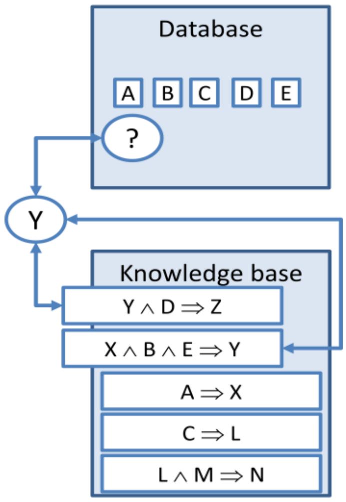
9.5.2 วิธีการอนุมานแบบ Backward Chaining

ในการอนุมานแบบ Backword Chaining จะเริ่มต้นดูจากผลลัพธ์ (goal) และพยายามหาข้อเท็จจริงต้นเหตุที่ทำให้เกิดผลลัพธ์ (sub-goal) ถ้าหากไม่มีข้อเท็จจริงเพียงพอที่จะนำการอนุมานไปสู่เป้าหมายได้ อาจจะมีการสอบถามข้อเท็จจริงเพิ่มเติมได้

การทำงานในขั้นตอนที่ 1 แสดงดังภาพที่ 9-14 โดยการนำเอาเป้าหมายคือ Z และพยายามหากฎที่จะนำไปสู่เป้าหมายคือ Z ในที่นี้จะเห็นได้ว่ากฎข้อที่ 1 จะส่งให้ผลลัพธ์เป็น Z ดังนั้นเพื่อให้ได้เป้าหมายนี้จำเป็นต้องรู้ข้อเท็จจริงของ Y และ D แต่เนื่องด้วยยังไม่ทราบถึงข้อเท็จจริงของ Y จึงให้ตั้ง Y เป็น sub-goal ดังแสดงในภาพที่ 9-15

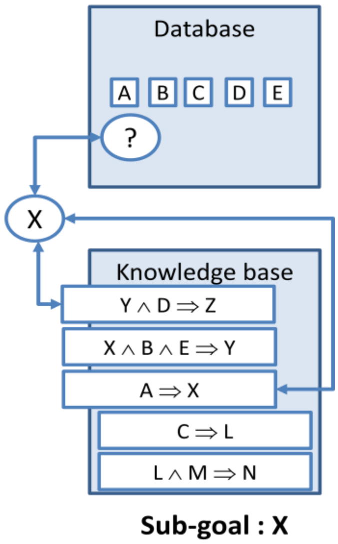


ภาพที่ 9-14 การอนุมานแบบฐานกฎด้วยวิธี Backword Chaining ขั้นตอนที่ 1



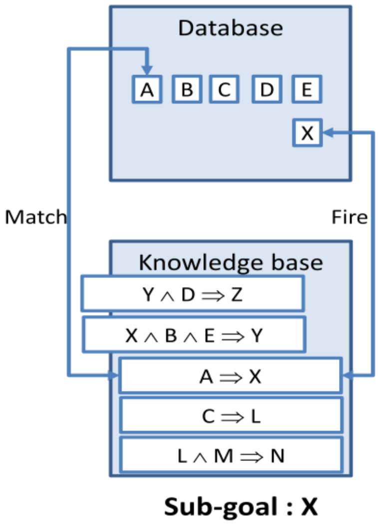
ภาพที่ 9-15 การอนุมานแบบฐานกฎด้วยวิธี Backword Chaining ขั้นตอนที่ 2

สำหรับขั้นตอนที่ 2 ก็ต้องหากฎที่จะมีผลนำไปสู่เป้าหมายย่อยในขณะปัจจุบันซึ่งก็คือ Y โดยจะพบว่ากฎข้อที่ 2 จะนำพาเป็นสู่ข้อเท็จจริงของ Y ได้ ดังนั้นให้พิจารณาจากเหตุการณ์ที่จะส่งผลให้ Y เป็นจริง ซึ่งจะพบว่าจำเป็นจะต้องทราบค่าความเป็นจริง X, B และ E ก่อน แต่อย่างไรก็ตามในฐานข้อมูลยังไม่มีข้อเท็จจริงเกี่ยวกับ X ดังนั้นจึงต้องเปลี่ยนเป้าหมายย่อยใหม่ให้เป็น X และหากฎที่ทำให้ได้ X ซึ่งก็คือกฎข้อที่ 3 แสดงดังภาพที่ 9-16 เป็นขั้นตอนที่ 3



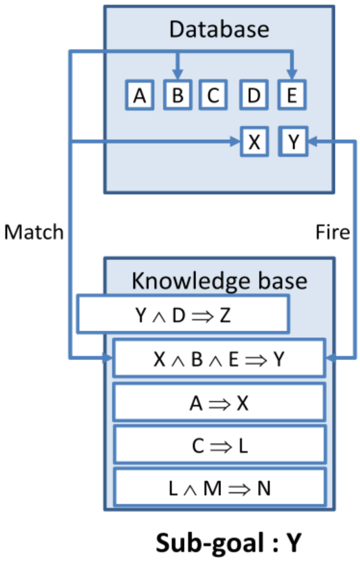
ภาพที่ 9-15 การอนุมานแบบฐานกฎด้วยวิธี Backword Chaining ขั้นตอนที่ 3

หลังจากขั้นตอนที่ 3 แล้วก็พบว่าเพื่อที่จะให้ได้เป้าหมายย่อย X จำเป็นจะต้องรู้ข้อเท็จจริงของ A ก่อน ซึ่งในฐานข้อมูลได้มีข้อเท็จจริงของ A อยู่แล้วทำให้ กฎข้อที่ 3 เกิดการ match กัน และ fire ข้อเท็จจริงใหม่คือ X กลับไปยังฐานข้อมูล ดังแสดงในภาพที่ 9-16 เป็นขั้นตอนที่ 4



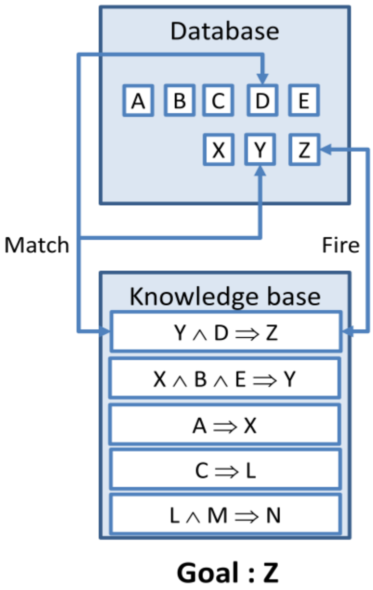
ภาพที่ 9-16 การอนุมานแบบฐานกฎด้วยวิธี Backword Chaining ขั้นตอนที่ 4

ขั้นตอนที่ 5 เมื่อฐานข้อมูลมีข้อเท็จจริงใหม่คือ X แล้วนั้นจะส่งผลให้กฎข้อที่ 2 match เพื่อ fire ข้อเท็จจริง Y ออกมา เพื่อให้สามารถนำไปสู่เป้าหมายย่อย Y ได้ ดังแสดงในภาพที่ 9-17



ภาพที่ 9-17 การอนุมานแบบฐานกฎด้วยวิธี Backword Chaining ขั้นตอนที่ 5

ขั้นตอนสุดท้ายเมื่อฐานข้อมูลมีข้อมูลเกี่ยวกับข้อเท็จจริงของ Y ดังนั้นจึงทำให้กฎข้อที่ 1 match กับข้อมูลและ fire ข้อเท็จจริง Z ออกมา ทำให้สามารถพิสูจน์ได้ว่าจากฐานกฎที่มี และข้อเท็จจริงเริ่มต้นที่กำหนดสามารถสรุปได้ว่า Z เป็นจริง ดังแสดงในภาพที่ 9-18



ภาพที่ 9-18 การอนุมานแบบฐานกฎด้วยวิธี Backword Chaining ขั้นตอนที่ 5

9.5.3 การเลือกใช้วิธีการอนุมานแบบฐานกฎ

การเลือกใช้วิธีการอนุมานแบบฐานกฎขึ้นอยู่กับวิธีที่ผู้เชี่ยวชาญแก้ปัญหาในโดเมนนั้นๆ ถ้าผู้เชี่ยวชาญต้องการรวบรวมข้อมูลก่อนแล้วค่อยหาทางแก้ไขปัญหา จะใช้วิธี Forward Chaining แต่ถ้าผู้เชี่ยวชาญตั้งสมมุติฐานของคำตอบมาก่อนแล้วค่อยหาข้อเท็จจริงเพื่อสรุปว่าสมมุติฐานที่ตั้งขึ้นมาเป็นจริง จะใช้วิธี Backward Chaining

โดยที่ Forward Chaining ส่วนใหญ่ใช้กับระบบที่ต้องการวิเคราะห์ข้อมูลเพื่อสรุป (ต้องมีข้อเท็จจริงพร้อมอยู่แล้ว) และ Backward Chaining ส่วนใหญ่ใช้กับระบบที่ต้องการตรวจสอบสมมุติฐาน (ระบบจะถามข้อเท็จจริงเพิ่มถ้าไม่สามารถตัดสินใจได้)

9.6 สรุป

ระบบผู้เชี่ยวชาญ หมายถึงระบบหรือโปรแกรมคอมพิวเตอร์ที่นำเอาองค์ความรู้ของผู้เชี่ยวชาญที่เป็นมนุษย์ มาจัดเก็บไว้เพื่อใช้ในการประมวลผล วิเคราะห์ และแก้ไขปัญหา พร้อมทั้งให้คำแนะนำอย่างมีเหตุผล ด้วยเหตุนี้ทำให้มีการพัฒนาระบบผู้เชี่ยวชาญขึ้น เพื่อนำมาประยุกต์ใช้งาน ระบบ MYCIN นับว่าเป็นระบบผู้เชี่ยวชาญในยุคแรกที่ถูกสร้างขึ้นมาเพื่อใช้ในการแพทย์ โดยใช้กฎที่มีความเชื่อมโยงกันประมาณ 500 กฎ เพื่อจำแนกอาการของผู้ป่วย ต่อมาได้มีการพัฒนาเป็นระบบ EMYCIN ซึ่งมีความยืดหยุ่นในการเพิ่มฐานองค์ความรู้ใหม่ๆ เข้าไปได้ ช่วยทำให้การทำงานของระบบผู้เชี่ยวชาญมีประสิทธิภาพสูงขึ้น

โครงสร้างของรบบผู้เชี่ยวชาญจะมีส่วนประกอบสำคัญอยู่ 2 อย่าง ได้แก่ 1) ฐานองค์ความรู้ ซึ่งเป็นส่วนที่เก็บองค์ความรู้ที่ประกอบด้วยความจริงและกฎต่างๆ ไว้ เพื่อใช้ในการประมวลผลและ ค้นหาวิธีการแก้ไขปัญหาที่ดีที่สุด ฐานองค์ความรู้ช่วยให้ระบบผู้เชี่ยวชาญมีความยืดหยุ่นในการตัดสินใจและหาคำตอบมากขึ้น โดยเฉพาะฐานองค์ความรู้ที่สามารถเพิ่มเติมและปรับเปลี่ยนได้ และ 2) กลไกการอนุมานเป็นส่วนที่ทำหน้าที่ในการอนุมานองค์ความรู้ ซึ่งจะเป็นตัวแปลกฎตต่างๆ ที่ใช้ในกระบวนการตรวจสอบความจริงและตัดสินใจระหว่างปัญหาที่เผชิญกับองค์ความรู้ที่มีอยู่ กลไกอนุมานมีหน้าที่หลัก คือดำเนินการตรวจสอบความจริงและกฎ และเป็นตัวตัดสินใจลำดับของการอนุมาน

กลไกลการอนุมานแบบฐานกฎนั้นสามารถแบ่งออกเป็น 2 วิธีหลักๆ คือ การอนุมานแบบไปข้างหน้า (Forward Chaining) ซึ่งเป็นวิธีการค้นหาเริ่มต้นจากข้อมูลที่มีนำพไปสู่เป้าหมายที่ต้องการ และการอนุมานแบบย้อนกลับ (Backward Chaining) เป็นวิธีการค้นหาที่เริ่มต้นจากเป้าหมายและนำพาไปถึงสาเหตุที่จะทำให้เป้าหมายนั้นเป็นจริง

แบบฝึกหัด

1. จงอธิบายว่าระบบผู้เชี่ยวชาญ หมายถึงอะไร มีความสัมพันธ์กับปัญหาประดิษฐ์อย่างไร
2. จงอธิบายความแตกต่างของระบบผู้เชี่ยวชาญ MYCIN และ EMYCIN พอสังเขป
3. จากข้อมูลของผู้เชี่ยวชาญในการแนะนำภาษาโปรแกรมที่บุคคลควรจะเรียนออกมาเป็นกฎดังนี้
    - ถ้า เป็นผู้เรียนคอมพิวเตอร์ขั้นต้น และ ต้องการความสนุก แล้วควรเรียน ภาษาเบสิค
    - ถ้า เป็นผู้เรียนคอมพิวเตอร์ขั้นต้น และ ต้องการความยาก แล้วควรเรียน ภาษาซี
    - ถ้า เป็นผู้เรียนคอมพิวเตอร์ขั้นสูง และ ต้องการความยาก แล้วควรเรียน ภาษาจาวา
    - ถ้า เป็นผู้เรียนคอมพิวเตอร์ขั้นสูง และ ต้องการความสนุก แล้วควรเรียน ภาษาไพทอน
    - ถ้า ผู้เรียนไม่คุ้นเคยกับคอมพิวเตอร์ แล้ว เป็นผู้เรียนคอมพิวเตอร์ขั้นต้น
    - ถ้า ผู้เรียนคุ้นเคยกับคอมพิวเตอร์ และ เขียนโปรแกรมไม่เป็น แล้ว เป็นผู้เรียนคอมพิวเตอร์ขั้นต้น
    - ถ้า ผู้เรียนคุ้นเคยกับคอมพิวเตอร์ และ เขียนโปรแกรมเป็น แล้ว เป็นผู้เรียนคอมพิวเตอร์ขั้นสูง
    - ถ้า เข้าใจโฟว์ชาร์ท แล้ว เขียนโปรแกรมเป็น
    - ถ้า ไม่เข้าใจโฟว์ชาร์ท แล้ว เขียนโปรแกรมไม่เป็น

    3.1 จงกำหนด Symbol ให้กับประโยคต่างๆ

    3.2 จงเขียนกฎทั้งหมดในรูปแบบของ Symbolic Logic

    3.3 เขียนกราฟ AND/OR graph ของกฎ
4. จากกฎในข้อที่ 3 ถ้าใช้วิธี Forward Chaining ในการอนุมาน แล้วผู้ใช้ให้ข้อเท็จจริงดังต่อไปนี้ จงแนะนำผู้ใช้ถึงภาษาโปรแกรมที่ควรเรียน

    4.1 ผู้ใช้คุ้นเคยกับคอมพิวเตอร์
    ผู้ใช้เข้าใจโฟลว์ชาร์ท
    ผู้ใช้ต้องการความสนุก

    4.2 ผู้ใช้ไม่เข้าใจโฟลว์ชาร์ท
    ผู้ใช้ไม่คุ้นเคยกับคอมพิวเตอร์
    ผู้ใช้ต้องการความยาก

    4.3 ผู้ใช้คุ้นเคยกับคอมพิวเตอร์
    ผู้ใช้เข้าใจโฟลว์ชาร์ท

แผนปฏิบัติการสอนสัปดาห์ที่ 10

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง พื้นฐานทฤษฎีความน่าจะเป็น **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. สเปซตัวอย่าง	IS 10.1
2. เหตุการณ์	IS 10.2
3. การนับจุดตัวอย่าง	IS 10.3
4. ความน่าจะเป็นของการเกิดเหตุการณ์	IS 10.4

ข. การจัดการเรียนการสอน

เวลา – นาที		0 60 120 180
วัตถุประสงค์		**1-2-3-4**
การนำเข้าสู่บทเรียน		
ให้เนื้อหา		
สรุปเนื้อหา		
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**
วิธีการสอน:	บรรยาย	
	อธิบาย	
	ถาม – ตอบ	
	ปฏิบัติด้วยตนเอง	
สื่อการสอน:	กระดานดำ	
	คอมพิวเตอร์	
	โปรเจคเตอร์	

บทที่ 10
พื้นฐานทฤษฎีความน่าจะเป็น

ในทางสถิติ คำว่า “การทดลอง (Experiment)” หมายถึงกระบวนการในการที่จะก่อให้เกิดชุดของข้อมูล ชุดของข้อมูลในที่นี้หมายถึงผลทั้งหมดที่เป็นไปได้ที่เกิดขึ้นจากการทดลอง ตัวอย่างเช่นในการทดลองโยนเหรียญ 1 เหรียญ ผลที่เกิดขึ้นเป็นไปได้ 2 แบบด้วยกันคือ หัวหรือก้อย ในการทดลองโยนลูกเต๋า 1 ลูก ผลที่เกิดขึ้นก็จะเป็นไปตามแต้มของลูกเต๋า ผลของการทดลองที่ออกมาแตกต่างกันนั้น สะท้อนให้เห็นถึงความหมายของคำว่า “ความไม่แน่นอน (Uncertainty)” ความน่าสนใจจะอยู่ที่การศึกษาโอกาส (Chance) หรือความน่าจะเป็นของการที่จะเกิดผลเป็นแบบใดแบบหนึ่งว่าเป็นเท่าใด

10.1 สเปซตัวอย่าง (Sample Space)

เซตของผลที่เป็นได้ทั้งหมดจากการทดลองจะเรียกว่า *สเปซตัวอย่าง* และเขียนแทนด้วยเซต S ผลของการทดลองแต่ละแบบเรียกว่าเป็นสมาชิก (Element) หรือจุดตัวอย่าง (Sample Point)

ตัวอย่างเช่น ในการโยนเหรียญ 1 เหรียญ

S = {H, T} โดยที่ H หมายถึง หัว และ T หมายถึง ก้อย

ในการโยนลูกเต๋า 1 ลูก ถ้าสนใจแต้มบนลูกเต๋า จะได้

S = {1, 2, 3, 4, 5, 6}

แต่ถ้า สนใจว่าเป็นแต้มคู่หรือคี่ จะได้

S = {คู่, คี่}

ในการทดลองทำข้อสอบปรนัย 3 ข้อถ้าสนใจว่าท าถูกหรือผิด จะได้

S = {TTT, TTF, TFT, FTT, FFT, FTF, TFF, FFF}

โดยที่ T หมายถึงทำข้อสอบถูก และ F หมายถึงทำข้อสอบผิด

แต่ถ้าสนใจจำนวนข้อที่ทำถูก จะได้

S = {0, 1, 2, 3}

ข้อสังเกต จะเห็นได้ว่าในการทดลองหนึ่งๆ สเปซตัวอย่างอาจมีได้หลายแบบ ทั้งนี้ขึ้นอยู่กับความสนใจผลของการทดลองว่าเป็นแบบใด

10.2 เหตุการณ์ (Events)

ในการทดลองนั้นบางครั้งสิ่งที่สนใจอาจเป็นแค่ผลของการทดลองบางส่วน จากผลที่เป็นไปได้ทั้งหมด ตัวอย่างเช่นในการโยนเหรียญ 3 เหรียญ ถ้าพิจารณาถึงจำนวนเหรียญที่ขึ้นหัว จะได้ว่าสเปซตัวอย่าง S = {0, 1, 2, 3} แต่สิ่งที่สนใจ อาจเป็นแค่จำนวนเหรียญที่ขึ้นหัวไม่เกิน 1 เหรียญ เป็นต้น โดยเหตุการณ์ หมายถึง เซตย่อย (Subset) ของสเปซตัวอย่าง

ตัวอย่างเช่น ในการโยนลูกเต๋า 1 ลูก

สเปซตัวอย่าง S = {1, 2, 3, 4, 5, 6}

ถ้า E1 เป็นเหตุการณ์ที่แต้มบนลูกเต๋ามากกว่า 2 จะได้ว่า

E1 = {3, 4, 5, 6}

ถ้า E2 เป็นเหตุการณ์ที่แต้มบนลูกเต๋าเป็นเลขคี่จะได้ว่า

E2 = {1, 3, 5}

มีการกำหนดนิยมเกี่ยวกับเหตุการณ์ดังต่อไปนี้

- กำหนดสเปซตัวอย่าง S และกำหนดเหตุการณ์ A คอมพลีเมนต์ (Complement) ของเหตุการณ์ A เขียนแทนด้วย A’ หมายถึง เซตของสมาชิกของ S ที่ไม่ได้เป็นสมาชิกของเหตุการณ์A
- กำหนดสเปซตัวอย่าง S และกำหนดเหตุการณ์ A และ B ยูเนียน (Union) ของเหตุการณ์ A และ B เขียนแทนด้วย A ∪ B หมายถึงเซตของสมาชิก S ที่อยู่ใน A หรือ B
- กำหนดสเปซตัวอย่าง S และกำหนดเหตุการณ์ A และ B อินเตอร์เซคชัน (Intersection) ของเหตุการณ์ A และ B เขียนแทนด้วย A ∩ B หมายถึงเซตของสมาชิก S ที่อยู่ใน A และ B

ตัวอย่างเช่น กล่องใบหนึ่งมีสลากที่มีหมายเลขตั้งแต่ 1 – 20 อย่างละ 1 ใบ ถ้าหยิบสลาก 1 ใบจากกล่องใบนี้ จะได้ว่า S = {1, 2, 3,..., 20}

ถ้า A เป็นเหตุการณ์ที่สลากนั้นมีหมายเลขที่หารด้วย 2 ลงตัว

B เป็นเหตุการณ์ที่สลากนั้นมีหมายเลขที่หารด้วย 3 ลงตัว

จะได้ว่า A = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}

B = {3, 6, 9, 12, 15, 18}

A’ = เป็นเหตุการณ์ที่สลากนั้นมีหมายเลขที่หารด้วย 2 ไม่ลงตัว

= {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}115

A ∪ B = เป็นเหตุการณ์ที่สลากนั้นมีหมายเลขที่หารด้วย 2 หรือ 3 ลงตัว

= {2, 3, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20}

A ∩ B = เป็นเหตุการณ์ที่สลากนั้นมีหมายเลขที่หารด้วย 2 และ 3 ลงตัว

= {6, 12, 18}

- เหตุการณ์ A และ B จะถูกเรียกว่าเป็นเหตุการณ์ที่ไม่เกิดร่วมกัน (Mutually Exclusive Events) ถ้า A ∩ B = Φ

ตัวอย่างเช่น จากตัวอย่างที่หยิบสลาก ที่ได้กล่าวข้างต้น

ถ้าให้ C เป็นเหตุการณ์ที่สลากมีหมายเลขหารด้วย 5 ลงตัว

D เป็นเหตุการณ์ที่สลากมีหมายเลขหารด้วย 7ลงตัว

C = {5, 10, 15, 20}

D = {7, 14}

ในที่นี้ C ∩ D = Φ ดังนั้นเหตุการณ์ C และ D เรียกว่าเป็นเหตุการณ์ที่ไม่เกิดร่วมกัน

10.3 การนับจุดตัวอย่าง

สิ่งที่สำคัญในการทดลองสถิติก็คือ จุดตัวอย่างที่เป็นไปได้ทั้งหมดเป็นอะไรได้บ้าง และมีจำนวนเท่าใด โดยเฉพาะอย่างยิ่งจำนวนของจุดตัวอย่าง จะเป็นสิ่งสำคัญในการคำนวณหาโอกาสที่จะเกิดขึ้นของเหตุการณ์ต่างๆ หลักการที่สำคัญอันหนึ่งในการนับจำนวนจุดตัวอย่างที่เป็นไปได้ก็คือ กฎการคูณ (Multiplication Rule) โดยมีทฤษฎีบทดังนี้

ทฤษฎีบท 1 ถ้างานหนึ่งสามารถเลือกทำได้ใน n_1 วิธี และในแต่ละวิธีของ n_1 สามารถเลือกทำงานอย่างที่สองได้ใน n_2 วิธี ดังนั้นจำนวนวิธีที่จะทำงานทั้งสองงานเท่ากับ $n_1 \times n_2$ วิธี

ตัวอย่าง 1: *ชายคนหนึ่งมีเสื้อเชิ้ตอยู่ 5 ตัว และกางเกงอยู่ 4 ตัว อยากทราบว่าชายคนนี้จะแต่งตัวได้ทั้งหมดกี่วิธี*

วิธีทำ ชายคนนี้เลือกใส่เสื้อได้ 5 วิธี

และหลังจากใส่เสื้อแล้ว ชายคนนั้นเลือกใส่กางเกงได้อีก 4 วิธี

ดังนั้นจำนวนวิธีที่เขาจะแต่งตัวได้ทั้งหมดเท่ากับ 5 x 4 = 20 วิธี

ตัวอย่าง 2: *ในการโยนเหรียญ 2 เหรียญ จำนวนจุดตัวอย่างที่เป็นไปได้ทั้งหมดมีกี่แบบ*

วิธีทำ การโยนเหรียญที่ 1 เหตุการณ์ที่เป็นไปได้คือเหรียญขึ้นหัวหรือก้อยซึ่งมี 2 แบบ

การโยนเหรียญที่2 ก็เช่นกัน เป็นไปได้อีก 2 แบบ

ดังนั้นจำนวนจุดตัวอย่างที่เป็นไปได้ในการโยนเหรียญ 2 เหรียญคือ 2 x 2= 4 แบบ

ทฤษฎีบท 2 ถ้ามีงานที่ต้องทำ k อย่าง ทั้งนี้งานที่ 1 มีทางเลือก n_1 วิธี งานที่ 2 มีทางเลือก n_2 วิธี ,..., และงานที่ k มีทางเลือก n_k วิธี ดังนั้นในการทำงาน k อย่างนั้น สามารถทำได้ทั้งหมด

$$n_1 \times n_2 \times \ldots \times n_k \text{ วิธี}$$

ตัวอย่าง 1: *ในการสร้างเลข 4 หลักจากเลข 0, 1, 2, 3, 4, 5 ถ้าแต่ละหลักสามารถใช้เลขซ้ำกันได้ อยากทราบว่าเลข 4 หลักที่เป็นไปได้นั้นมีทั้งหมดกี่จำนวนที่เป็นเลขคี่*

วิธีทำ ในหลักพันนั้นสามารถเลือกใช้ตัวเลขได้ 5 วิธี (ใช้เลข 0 ไม่ได้)

ในหลักร้อยนั้นสามารถเลือกใช้ตัวเลขได้ 6 วิธี

ในหลักสิบนั้นสามารถเลือกใช้ตัวเลขได้ 6 วิธี

ในหลักหน่วยนั้นสามารถเลือกใช้ตัวเลขได้ 3 วิธี (เลขคี่ 1, 3, 5)

ดังนั้นจ านวนเลข 4 หลักที่เป็นเลขคี่มีทั้งหมดเท่ากับ 5 x 6 x 6 x 3= 540 จำนวน

ตัวอย่าง 2: *ในการโยนลูกเต๋า 3 ลูก จำนวนจุดตัวอย่างที่เป็นไปได้มีกี่แบบ*

วิธีทำ ในการโยนลูกเต๋า ลูกที่ 1 จำนวนแบบที่เป็นไปได้มี 6 แบบ

ในการโยนลูกเต๋า ลูกที่ 2 จำนวนแบบที่เป็นไปได้มี 6 แบบ

ในการโยนลูกเต๋า ลูกที่ 3 จำนวนแบบที่เป็นไปได้มี 6 แบบ

ดังนั้นจำนวนจุดตัวอย่างของการโยนลูกเต๋า 3 ลูกเท่ากับ 6 x 6 x 6 = 216 แบบ

10.3.1 การเรียงสิ่งของที่แตกต่างกัน n สิ่ง ในแนวเส้นตรง

การจัดลำดับ (Permutation) คือการจัดเรียงลำดับของสิ่งของจำนวนหนึ่ง ซึ่งอาจจะเป็นสิ่งของทั้งหมดหรือเป็นเพียงบางส่วน

ตัวอย่าง 1: *ถ้านำอักษร 4 ตัว a, b, c และ d มาเรียงลำดับกัน จะได้ทั้งหมดกี่วิธี*

วิธีทำ ถ้าพิจารณาอักษร 4 ตัวมาเรียงกันในลักษณะ _ _ _ _

1 2 3 4

จะเห็นได้ว่าในตำแหน่งที่ 1 สามารถเลือกตัวอักษรดังกล่าวได้ 4 วิธี

หลังจากเลือกตัวอักษรที่ตำแหน่งที่ 1 แล้วเหลือตัวอักษรที่จะอยู่ตำแหน่งที่ 2 ได้ 3 วิธี

และหลังจากเลือกตัวอักษรตำแหน่งที่ 2 จะเหลือตัวอักษรที่จะอยู่ในตำแหน่งที่ 3 ได้ 2 วิธี

และจะเหลือตัวอักษรในตำแหน่งที่ 4 ได้อยู่เพียง 1 วิธี

ดังนั้นการนำอักษร 4 ตัวมาเรียงลำดับกันจะได้ = 4 x 3 x 2 x 1 = 24 วิธี (หรือ 4! วิธี)

จากตัวอย่างด้านต้นจึงสามารถสรุปได้ว่า การจัดลำดับสิ่งของ n สิ่งที่แตกต่างกันจะมีวิธีการจัดได้ทั้งหมด n! วิธี

ตัวอย่าง 2: *ถ้านำคน 5 คนมายืนเรียงกันจะได้ทั้งหมดกี่วิธี*

วิธีทำ การนำคน 5 คนมาเรียงกันจะได้ = 5! วิธี

= 120 วิธี

10.3.2 การเรียงสิ่งของที่แตกต่างกัน r สิ่งจากของทั้งหมด n สิ่ง

ในการจัดลำดับสิ่งของนั้น บางครั้งอาจไม่ได้นำมาเรียงกันทั้งหมด กล่าวคือถ้ามีสิ่งของทั้งหมด n สิ่ง อาจจะนำมาเรียงลำดับกันเพียง r สิ่ง โดยที่ r < n ถ้าสิ่งของ n สิ่งที่แตกต่างกัน นำมาเรียงกันทีละ r สิ่งจะเป็นในลักษณะดังนี้

$$\underset{1}{_}\;\underset{2}{_}\;\underset{3}{_}\;\cdots\;\underset{r}{_}$$

จะเห็นได้ว่าในตำแหน่งที่ 1 สามารถเลือกสิ่งของได้ n วิธี

ในตำแหน่งที่ 2 สามารถเลือกสิ่งของได้ n – 1 วิธี

ในตำแหน่งที่ 3 สามารถเลือกสิ่งของได้ n – 2 วิธี

...

ในตำแหน่งที่ n สามารถเลือกสิ่งของได้ n – (r – 1) วิธี

ดังนั้นจำนวนวิธีที่เรียงสิ่งของ r สิ่งจาก n สิ่ง เท่ากับ

$$n \times (n-1) \times (n-2) \times \ldots \times (n-(r-1)) = \frac{n!}{(n-r)!} \text{ วิธี}$$

จึงสามารถนำมาสรุปเป็นทฤษฎีบทได้ว่า การจัดลำดับของ n สิ่งที่แตกต่างกัน แต่นำมาเรียงกัน r สิ่งจะได้จำนวนวิธีทั้งหมดเท่ากับ

$$n\,P\,r = \frac{n!}{(n-r)!} \text{ วิธี}$$

ตัวอย่าง 1: *จากอักษร a, b, c, d และ e นำมาเรียงกันเพียง 3 ตัวจะได้กี่วิธี*

วิธีทำ จำนวนวิธีที่จะเรียงอักษร 3 ตัวจาก 5 ตัวนั้นจะได้ = $\frac{5!}{(5-3)!}$

= 60 วิธี

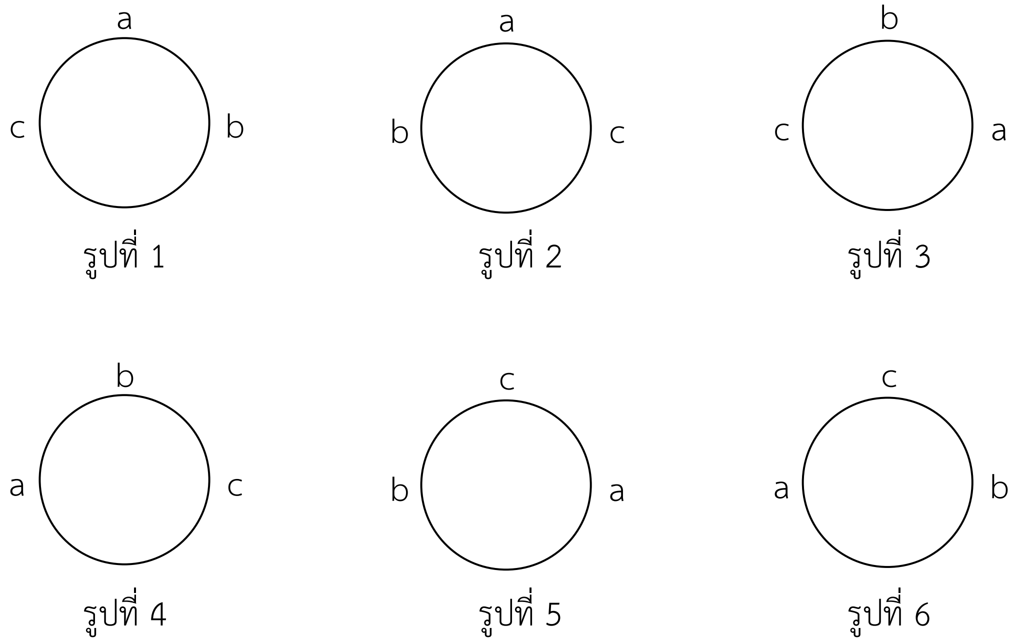
ตัวอย่าง 2: *หนังสือที่แตกต่างกัน 6 เล่ม ถ้านำหนังสือเพียงแค่ 2 เล่มมาเรียงกัน จะได้ทั้งหมดกี่วิธี*

วิธีทำ หนังสือ 6 เล่มเอามาเรียงกัน 2 เล่มจะได้ทั้งหมด = $\frac{6\,!}{(6-2)!}$

= 30 วิธี

10.3.3 การเรียงสิ่งของที่แตกต่างกัน n สิ่งในแนววงกลม

ในการนำของ n สิ่งที่แตกต่างกันมาเรียงกันในแนวเส้นตรงนั้น จำนวนวิธีที่ได้เท่ากับ n! วิธี อย่างเช่นการเอาตัวอักษร a, b, c มาเรียงกันจะได้ทั้งหมด 3! วิธี ซึ่งได้แก่ abc, acb, bac, bca, cab และ cba ถ้าทั้ง 6 แบบดังกล่าวมาเรียงในแนววงกลมจะเห็นดังภาพที่ 10-1



ภาพที่ 10-1 การเรียงตัวอักษร a, b, c ในแนววงกลม

จากภาพจะเห็นได้ว่า abc ในรูปที่ 1 เหมือนกับ bca ในรูปที่ 4 และ cab ในรูปที่ 5 และ acb ในรูปที่ 2 เหมือนกับ bac ในรูปที่3 และ cba ในรูปที่ 6 นั่นคือจาก 6 แบบของการเรียงดังกล่าว จะเหลือเพียง 2 วิธีที่เป็นเช่นนี้เพราะการเรียงในแนวเส้นตรงตัวอักษรที่อยู่ในตำแหน่งแรกหรือตำแหน่งสุดท้ายจะแตกต่างกัน แต่ในแนววงกลมจะไม่มีตำแหน่งแรกและตำแหน่งสุดท้าย ดังนั้นการนับจำนวนวิธีที่เรียงกัน

ในแนววงกลมนั้น ต้องคิดว่าตัวอักษรใดตัวหนึ่งคงที่ ซึ่งจะเป็น a, b หรือ c ก็ได้ ดังนั้นจำนวนวิธีที่จะเรียงสลับกันไปมาก็จะเหลือแค่ตัวอักษร 2 ตัว ซึ่งจะสลับกันได้ 2! วิธี

ทำให้สามารถเขียนเป็นทฤษฎีบทได้ว่า ถ้ามีของ n สิ่งที่แตกต่างกัน นำมาเรียงกันเป็นวงกลม จะได้ทั้งหมด (n – 1)! วิธี

ตัวอย่าง 1: *ถ้าคน 5 คนมายืนเรียงกันเป็นวงกลมจะได้กี่วิธี*

วิธีทำ จำนวนวิธีที่คน 5 คนมายืนเรียงกันเป็นวงกลม = (5 – 1)! วิธี

= 24 วิธี

10.3.4 การเรียงสิ่งของ n สิ่งแต่มีบางสิ่งที่ซ้ำกัน

ในการเรียงตัวอักษร 6 ตัวคือ a1, a2, a3, b1, b2, c สลับกันไปมานั้นจะได้ทั้งหมด 6! แบบด้วยกัน ทั้งนี้ใน 6! แบบนั้นมีที่ตัวอักษร a1, a2 และ a3 สลับกันไปมา 3! แบบ ซึ่งถ้าคิดว่าตัวอักษร a1, a2 และ a3 นั้นเหมือนกันคือเป็น a ตัวเดียวกัน จำนวนแบบที่จะได้ก็จะลดลงเหลือ $\frac{6\,!}{3\,!}$ แบบ และในทำนองเดียวกัน สำหรับตัวอักษร b1 และ b2 ซึ่งถ้าคิดว่าเป็น b เหมือนกัน จำนวนแบบทั้งหมดที่เป็นไปได้ก็จะเท่ากับ $\frac{6\,!}{3!\,2!}$ แบบ

ดังนั้นจึงสามารถเขียนเป็นทฤษฎีบทได้ว่า ในการเรียงลำดับสิ่งของ n สิ่ง ซึ่งใน n สิ่งนี้มีสิ่งของ n_1 สิ่งที่ซ้ำกัน มีสิ่งของ n_2 สิ่งที่ซ้ำกัน ... และสิ่งของ n_k สิ่งที่ซ้ำกัน จะได้ว่าการเรียงลำดับสิ่งของ n สิ่งนั้นเท่ากับ $\frac{n\,!}{n1!\;n2!\;....nk!}$ วิธี

ตัวอย่าง 1: *จากตัวอักษรในคำว่า “STATISTICS” ถ้านำมาเรียงกันไปมาโดยไม่สนใจความหมายจะได้ทั้งหมดกี่วิธี*

วิธีทำ จากตัวอักษรในคำดังกล่าว จะมี “S” 3 ตัว “T” 3 ตัว “I” 2 ตัว และ “A” และ “C” อย่างละ 1 ตัว

ดังนั้นจำนวนวิธีที่ตัวอักษรในค าดังกล่าว เรียงสลับกันจะได้ = $\frac{10\,!}{3!\;3!\,2!\,1!\,1!}$ วิธี

= 50, 400 วิธี

10.3.5 การจัดหมู่ (Combination)

จากนิยาม ในการนำสิ่งของ n สิ่งที่แตกต่างกัน และนำมาเรียงกันทีละ r สิ่ง จำนวนวิธีที่ได้เท่ากับ $\frac{n\,!}{(n-r)!}$ วิธี แต่ถ้าของ r สิ่งนั้นไม่ได้นำมาเรียงกัน กล่าวคือไม่ได้สนใจลำดับของสิ่งของ r สิ่งดังกล่าว

จะเรียกว่าเป็นการจัดหมู่ ซึ่งสามารถเขียนเป็นทฤษฎีบทใหม่ได้ว่า ถ้ามีของ n สิ่งที่แตกต่างกัน แล้วเอามา r สิ่ง จำนวนวิธีที่ได้จะเท่ากับ $\frac{n\,!}{r!\,(n-r)!}$ วิธี ซึ่งจะใช้สัญลักษณ์ n C r หรือ $\binom{n}{r}$

ตัวอย่าง 1: *จะเลือกคน 6 คนมาเป็นกรรมการชุดหนึ่งซึ่งมี 4 คนได้กี่วิธี*

วิธีทำ จำนวนชุดของกรรมการ = $\binom{6}{4}$ = $\frac{6\,!}{4!\,(6-4)!}$ = $\frac{6\,!}{4!\;2!}$

= 15 วิธี

10.4 ความน่าจะเป็นของการเกิดเหตุการณ์

ในการทดลองที่มีสเปซตัวอย่างคือเซต S ถ้า E เป็นเหตุการณ์หนึ่งที่เกิดขึ้น สิ่งที่น่าสนใจเกี่ยวกับเหตุการณ์ E ก็คือโอกาสที่จะเกิดเหตุการณ์ E ได้นั้นเป็นเท่าใด ซึ่งมีการให้นิยามดังนี้

ถ้า E เป็นเหตุการณ์ที่มีสเปซตัวอย่างคือ S ความน่าจะเป็นของการเกิดเหตุการณ์ E เขียนแทนด้วย P(E) จะหมายถึง ผลรวมของน้ำหนักของจำนวนจุดตัวอย่างใน E ทั้งนี้

- $0 \leq P(E) \leq 1$
- $P(\Phi) = 0$
- $P(S) = 1$

ตัวอย่าง 1: *ในการโยนเหรียญที่เที่ยงตรง 2 เหรียญ ความน่าจะเป็นที่เหรียญขึ้นหัวและก้อยอย่างละ 1 เหรียญ เป็นเท่าใด*

วิธีทำ ในที่นี้ สเปซตัวอย่าง S = {HH, HT, TH, TT}

ในเซต S มี 4 จุดตัวอย่าง ซึ่งแต่ละจุดตัวอย่างมีน้ำหนักหรือโอกาสเกิดขึ้นเท่าๆ กัน เพราะเหรียญมีความเที่ยงตรง ซึ่งถ้าให้น้ำหนักแต่ละจุดตัวอย่างเท่ากับ x

ดังนั้นจะได้ว่า 4 x = 1 หรือ x = (1/4) = 0.25

ให้ E เป็นเหตุการณ์ที่เหรียญขึ้น H และ T อย่างละ 1 เหรียญ

ดังนั้น E = {HT, TH}

และความน่าจะเป็นของเหตุการณ์ E = P(E)

= 0.25 + 0.25 = 0.5

ตัวอย่าง 2: *ในการโยนลูกเต๋าที่เที่ยงตรง 1 ลูก ความน่าจะเป็นที่จะได้แต้มบนลูกเต๋ามากกว่าหรือเท่ากับ 3 เป็นเท่าใด*

วิธีทำ ในที่นี้ S = {1, 2, 3, 4, 5, 6}

ทั้งนี้จะได้ว่าน้ำหนักของแต่ละจุดตัวอย่างเท่ากับ $\frac{1}{6}$

ให้ E = เหตุการณ์ที่แต้มบนลูกเต๋ามากกว่าหรือเท่ากับ 3

= {3, 4, 5, 6}

$$P(E) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6}$$

$$= \frac{2}{3}$$

ทฤษฎีบทอีกทฤษฎีกล่าวว่า ในการทดลองหนึ่งที่มีจำนวนจุดตัวอย่างที่เป็นไปได้ N แบบที่แตกต่างกัน และมีโอกาสเกิดขึ้นเท่าๆ กัน ถ้า E เป็นเหตุการณ์หนึ่งโดยที่ $E \subset S$ โดยที่จำนวนจุดตัวอย่างใน E = n แล้วจะได้ว่า

$$P(E) = \frac{n}{N}$$

ตัวอย่าง 3: *ไพ่สำรับหนึ่งมี 52 ใบ ถ้าสุ่มหยิบไพ่ 1 ใบจากสำรับนี้ แล้วจงหาความน่าจะเป็นที่จะได้*

1. ไพ่ที่มีแต้ม 7

2. ไพ่ที่เป็นโพดำ

วิธีทำ ให้ E_1 เป็นเหตุการณ์ที่ไพ่ที่หยิบได้มีแต้ม 7

E_2 เป็นเหตุการณ์ที่ไพ่ที่หยิบได้เป็นโพดำ

เนื่องจากไพ่ที่มีแต้ม 7 มี 4 ใบด้วยกัน

ดังนั้น $P(E_1) = \frac{4}{52} = \frac{1}{13}$

และไพ่ที่เป็นโพดำ มี 13 ใบ

จะได้ $P(E_2) = \frac{13}{52} = \frac{1}{4}$

ตัวอย่าง 4: *ในการโยนลูกเต๋าที่เที่ยงตรง 2 ลูก จงหาความน่าจะเป็นที่*

1. ผลรวมของแต้มบนลูกเต๋าเท่ากับ 6

2. ได้แต้มเหมือนกันทั้ง 2 ลูก

วิธีทำ ในที่นี้จำนวนจุดตัวอย่างในสเปซตัวอย่าง = 6 x 6 = 36 จุดตัวอย่าง

ให้ E_1 เป็นเหตุการณ์ที่ผลรวมของแต้มบนลูกเต๋าเท่ากับ 6

ดังนั้น E_1 = {(1, 5), (5, 1), (2, 4), (4, 2), (3, 3)}

จะได้ $P(E_1) = \frac{5}{36}$

ให้ E2 เป็นเหตุการณ์ที่ลูกเต๋า 2 ลูกได้แต้มเหมือนกัน

ดังนั้น E_2 = {(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)}

จะได้ $P(E_2) = \frac{6}{36} = \frac{1}{6}$

10.4.1 กฎที่สำคัญต่างๆ

ในการคำนวณหาความน่าจะเป็นของเหตุการณ์หนึ่งๆ นั้น อาจหาได้ง่ายขึ้นถ้าทราบกฎบางข้อที่เกี่ยวข้องกับการหาความน่าจะเป็นดังกล่าวนั้น โดยทฤษฎีบทที่สำคัญมีดังนี้

- กำหนดสเปซตัวอย่าง S และกำหนดเหตุการณ์ A และ B จะได้ว่า

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

- กำหนดสเปซตัวอย่าง S และกำหนดเหตุการณ์ A และ B ว่าเป็นเหตุการณ์ที่ไม่เกิดร่วมกัน จะได้ว่า

$$P(A \cup B) = P(A) + P(B)$$

- กำหนดสเปซตัวอย่าง S และกำหนดเหตุการณ์ A จะได้ว่า

$$P(A) + P(A') = 1$$

ตัวอย่าง 1: *กล่องใบหนึ่งมีสลากที่มีหมายเลข 1 - 30 อย่างละใบ หยิบสลาก 1 ใบจากกล่องใบนี้จงหาความน่าจะเป็นที่*

1. สลากที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 3 ลงตัว

2. สลากที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 3 ไม่ลงตัว

3. สลากที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 5 ลงตัว

4. สลากที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 3 และ 5 ลงตัว

5. สลากที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 3 หรือ 5 ลงตัว

วิธีทำ 1. ให้ A เป็นเหตุการณ์ที่สลากที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 3 ลงตัว

ในที่นี้ A = {3, 6, 9, 12, 15, 18, 21, 24, 27, 30}

จะได้ว่า $P(A) = \frac{10}{30} = \frac{1}{3}$

2. ให้ A’ เป็นเหตุการณ์ที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 3 ไม่ลงตัว

จาก $P(A') = 1 - P(A)$

$= 1 - \frac{1}{3} = \frac{2}{3}$

3. ให้ B เป็นเหตุการณ์ที่หยิบได้เป็นสลากที่มีหมายเลขที่หารด้วย 5 ลงตัว

ในที่นี้ B = {5, 10, 15, 20, 25, 30}

จะได้ว่า P(B) = $\frac{6}{30}$ = $\frac{1}{5}$

4. A ∩ B เป็นเหตุการณ์ที่หยิบได้สลากที่มีหมายเลขที่หารด้วย 3 และ 5 ลงตัว

ในที่นี้ A ∩ B = {15, 30}

จะได้ว่า P(A ∩ B) = $\frac{2}{30}$ = $\frac{1}{15}$

5. A U B เป็นเหตุการณ์ที่หยิบได้สลากที่มีหมายเลขที่หารด้วย 3 หรือ 5 ลงตัว

จาก P(A U B) = P(A) + P(B) – P(A ∩ B)

จะได้ว่า P(A U B) = $\frac{10}{30} + \frac{6}{30} - \frac{2}{30}$

$= \frac{14}{30}$

$= \frac{7}{15}$

ตัวอย่าง 2: *หยิบไพ่ 1 ใบจากสำรับหนึ่งที่มี 52 ใบ จงหาความน่าจะเป็นที่จะได้ไพ่ที่มีแต้ม 2 หรือเป็นโพแดง*

วิธีทำ ให้ A เป็นเหตุการณ์ที่ได้ไพ่ที่มีแต้ม 2

B เป็นเหตุการณ์ที่ได้ไพ่เป็นโพแดง

จะได้ P(A) = $\frac{4}{52}$

P(B) = $\frac{13}{52}$

A ∩ B หมายถึงเหตุการณ์ที่ได้ 2 และเป็นโพแดง

จะได้ P(A ∩ B) = $\frac{1}{52}$

เนื่องด้วย P(เหตุการณ์ที่ได้ไพ่แต้ม 2 หรือเป็นโพแดง) = P(A U B)

ดังนั้น P(A U B) = P(A) + P(B) – P(A ∩ B)

$= \frac{4}{52} + \frac{13}{52} - \frac{1}{52}$

$= \frac{16}{52}$

$= \frac{4}{13}$

ตัวอย่าง 3: *ในการโยนเหรียญที่เที่ยงตรง 4 เหรียญ 1 ครั้ง จงหาความน่าจะเป็นที่เหรียญจะขึ้นหัวอย่างน้อย 1 เหรียญ*

วิธีทำ ถ้าให้ A เป็นเหตุการณ์ที่เหรียญขึ้นหัวอย่างน้อย 1 เหรียญ

จะได้ A’ เป็นเหตุการณ์ที่เหรียญไม่ขึ้นหัวเลย (ขึ้นก้อยทั้ง 4 เหรียญ)

$P(A') = \frac{1}{16}$

ดังนั้น P(A) = 1 – P(A’)

$= 1 - \frac{1}{16}$

$= \frac{15}{16}$

10.4.2 ความน่าจะเป็นแบบมีเงื่อนไข (Conditional Probability)

กำหนดเหตุการณ์ A และ B จะใช้สัญลักษณ์ P(A | B) หมายถึงความน่าจะเป็นของการเกิดเหตุการณ์ A เมื่อกำหนดว่าเหตุการณ์ B ได้เกิดขึ้นแล้ว

ตัวอย่าง 1: *ในการโยนลูกเต๋าที่เที่ยงตรง 2 ลูก*

ถ้าให้ A เป็นเหตุการณ์ที่ผลรวมของแต้มบนลูกเต๋าทั้ง 2 ลูกเป็น 7

B เป็นเหตุการณ์ที่แต้มบนลูกที่ 1 มากกว่าแต้มบนลูกที่ 2

จงหา P(A | B) และ P(B | A)

วิธีทำ ในที่นี้ A = {(1, 6), (6, 1), (2, 5), (5, 2), (3, 4), (4, 3)}

B = {(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3), (5, 1), (5, 2), (5, 3), (5, 4), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5)}

P(A | B) = ความน่าจะเป็นที่ผลรวมของแต้มบนลูกเต๋าทั้ง 2 ลูกเป็น 7 เมื่อกำหนดว่าแต้มบนลูกเต๋าที่ 1 มากกว่าแต้มบนลูกเต๋าที่ 2

จะเห็นได้ว่า เมื่อกำหนดว่าเหตุการณ์ B ได้เกิดขึ้นนั้น จากจำนวนจุดตัวอย่างที่เป็นไปได้ 36 แบบนั้น ก็จะลดลงเหลือเท่ากับจำนวนจุดตัวอย่างของเหตุการณ์ B คือ 15 แบบ และใน 15 แบบนี้ จำนวนจุดตัวอย่างที่ผลรวมเป็น 7 ก็จะมีเพียง 3 แบบเท่านั้น คือ (6,1), (5,2) และ (4, 3)

ดังนั้น $P(A \mid B) = \frac{3}{15}$

$= \frac{1}{5}$

P(B | A) = ความน่าจะเป็นที่แต้มบนลูกที่ 1 มากกว่าแต้มบนลูกที่ 2 เมื่อกำหนดว่าผลรวมของแต้มบนลูกเต๋าทั้ง 2 ลูกเป็น 7

เช่นเดียวกับข้างต้น คือเมื่อกำหนดว่าเหตุการณ์ A ได้เกิดขึ้นนั้น จากจำนวนจุดตัวอย่างที่เป็นไปได้ 36 แบบนั้น ก็จะลดลงเหลือเท่ากับจำนวนจุดตัวอย่างของเหตุการณ์ A คือ 6 แบบ และใน 6 แบบดังกล่าว จำนวนจุดตัวอย่างที่แต้มบนลูกที่ 1 มากกว่าแต้มบนลูกที่ 2 ก็มีเพียง 3 แบบเท่านั้น คือ (6, 1), (5, 2) และ (4, 3)

นั่นคือ P(B | A) $= \frac{3}{6}$

$= \frac{1}{2}$

จาก P(A | B) และ P(B | A) ดังกล่าว จะเห็นได้ว่าการหาความน่าจะเป็นแบบมีเงื่อนไข หรือการหาความน่าจะเป็นของเหตุการณ์ได้กำหนดว่ามีอีกเหตุการณ์หนึ่งได้เกิดขึ้นนั้น เป็นการหาจำนวนจุดของเหตุการณ์นั้น จากจำนวนจุดของตัวอย่างที่สเปซตัวอย่างลดลง (Reduced Sample Space) หรือเท่ากับจำนวนจุดตัวอย่างของเหตุการณ์ที่กำหนดว่าได้เกิดขึ้นแล้ว จึงสามารถเขียนเป็นทฤษฎีบทได้ว่า ถ้ากำหนด A และ B เป็นเหตุการณ์ โดยที่ P(B) > 0 จะได้ว่า

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

ตัวอย่าง 2: จากตัวอย่าง 1

P(A) $= \frac{6}{36}$, P(B) $= \frac{15}{36}$

P(A ∩ B) $= \frac{3}{36}$

ดังนั้น P(A | B) $= \frac{3}{36} / \frac{15}{36}$

$= \frac{3}{15}$

$= \frac{1}{5}$

และ P(B | A) $= \frac{3}{36} / \frac{6}{36}$

$= \frac{3}{6}$

$= \frac{1}{2}$

ตัวอย่าง 3: *จากข้อมูลของพนักงานบริษัทแห่งหนึ่ง ปรากฏผลดังตาราง*

	สำเร็จ การศึกษาต่ำกว่าปริญญาตรี	สำเร็จ การศึกษาปริญญาตรีหรือสูงกว่า	รวม
ผู้ชาย	100	300	400
ผู้หญิง	150	450	600
รวม	250	750	1000

ถ้าสุ่มเลือกพนักงานของบริษัทแห่งนี้มา 1 คน พบว่าเป็นผู้หญิง แล้วความน่าจะเป็นที่พนักงานนั้นจะสำเร็จการศึกษาที่ต่ำกว่าปริญญาตรีเป็นเท่าใด

วิธีทำ ให้ A เป็นเหตุการณ์ที่พนักงานสำเร็จการศึกษาที่ต่ำกว่าปริญญาตรี

B เป็นเหตุการณ์ที่พนักงานเป็นผู้หญิง

P(A) = $\frac{250}{1000}$, P(B) = $\frac{600}{1000}$

P(A ∩ B) = $\frac{150}{1000}$

จาก P(A | B) = $\frac{P(A \cap B)}{P(B)}$

$= \frac{150}{1000} / \frac{600}{1000}$

$= \frac{150}{600} = \frac{1}{4}$

10.4.3 เหตุการณ์ที่เป็นอิสระต่อกัน (Independent Events)

ในการหาความน่าจะเป็นแบบมีเงื่อนไข P(A | B) ตามที่ได้กล่าวมาแล้วนั้นจะเห็นได้ว่า เมื่อได้กำหนดว่าเหตุการณ์ B ได้เกิดขึ้น ก็จะมีผลกระทบต่อการหาความน่าจะเป็นของเหตุการณ์ A อย่างไรก็ตาม ในบางสถานการณ์ ารกำหนดว่าเหตุการณ์ B ได้เกิดขึ้น อาจไม่ส่งผลกระทบต่อการเกิดเหตุการณ์ A ก็เป็นไปได้ และนั่นหมายถึงว่า P(A | B) = P(A)

ดังนั้นถึงสามารถนิยามได้ว่า เหตุการณ์ A และ B จะเรียกว่าเป็นเหตุการณ์ที่อิสระต่อกันก็ต่อเมื่อ P(A | B) = P(A) หรือ P(B | A) = P(B) และจะกล่าวได้ว่าเหตุการณ์ A และ B จะเป็นอิสระต่อกันก็ต่อเมื่อ P(A ∩ B) = P(A) x P(B)

ตัวอย่าง 1: *เหรียญอันหนึ่งถูกถ่วงน้ำหนัก ให้โอกาสที่จะขึ้นหัว เป็นสองเท่าของโอกาสที่จะขึ้นก้อย ถ้าโยนเหรียญนี้ 2 ครั้ง จงหาความน่าจะเป็นที่*

1. เหรียญขึ้นก้อยทั้ง 2 ครั้ง

2. เหรียญขึ้นหัว 1 ครั้ง

วิธีทำ ให้โอกาสที่เหรียญจะขึ้นก้อยเท่ากับ y

ดังนั้นโอกาสที่เหรียญขึ้นหัวเท่ากับ 2y

จะได้ว่า y + 2y = 1 หรือ $y = \frac{1}{3}$

ถ้า T, H หมายถึงเหรียญขึ้นหน้าก้อย และหัวตามลำดับ

ความน่าจะเป็นที่เหรียญขึ้นก้อยทั้ง 2 ครั้ง = P (TT)

$= P(T) \times P(T)$

$= \frac{1}{3} \times \frac{1}{3}$

$= \frac{1}{9}$

ที่ P (TT) = P(T) x P(T) เป็นเพราะการที่ครั้งที่ 1 เหรียญขึ้นหน้าก้อย ไม่ได้ส่งผลอะไรกับการโยนครั้งที่สอง กล่าวคือการโยนเหรียญแต่ละครั้ง เป็นอิสระต่อกัน

ความน่าจะเป็นที่เหรียญขึ้นหัว 1 ครั้ง = P (TH) + P (HT)

$= [P(T) \times P(H)] + [P(H) \times P(T)]$

$= [\frac{1}{3} \times \frac{2}{3}] + [\frac{2}{3} \times \frac{1}{3}]$

$= \frac{2}{9} + \frac{2}{9}$

$= \frac{4}{9}$

สามารถสรุปเป็นทฤษฎีบทได้ว่า เมื่อกำหนด {E1, E2, … En} เป็นเซตของเหตุการณ์จากสเปซตัวอย่าง S โดยที่

1. $P(E_i) \neq 0$ ทุกค่า i = 1, 2, …, n

2. $E_i \cap E_j = 0$ ทุกค่า $i \neq j$

3. $E_1 \cup E_2 \cup E_3 \ldots \cup E_n$

ถ้า A เป็นเหตุการณ์หนึ่งของ S แล้ว $P(A) = \sum_{i=1}^{n} P(E_i)P(A \mid E_i)$

ตัวอย่าง 2: *โรงงานผลิตหลอดไฟฟ้าแห่งหนึ่งมีเครื่องจักรที่ใช้ในการผลิต 3 เครื่องคือ เครื่องจักร A, B และ C ซึ่งมีอัตราการผลิต 40%, 30% และ 30% ตามลำดับ หลอดไฟฟ้าที่ผลิตจากเครื่องจักร 3 เครื่องดังกล่าวนี้ พบว่ามีหลอดไฟฟ้าเสีย 3%, 5% และ 4% สำหรับเครื่องจักร A, B และ C ตามลำดับ ถ้าสุ่มหยิบหลอดไฟฟ้าจากโรงงานนี้มา 1 หลอด ความน่าจะเป็นที่หลอดไฟฟ้านี้จะเป็นหลอดเสียมีค่าเท่าใด*

วิธีทำ ให้ D เป็นเหตุการณ์ที่หลอดไฟฟ้าที่สุ่มหยิบมาเป็นหลอดเสีย

จะได้ว่า $P(D) = P(A)P(D \mid A) + P(B)P(D \mid B) + P(C)P(D \mid C)$

$= (0.4)(0.03) + (0.3)(0.05) + (0.3)(0.04)$

$= 0.012 + 0.015 + 0.012$

$= 0.039$

10.5 สรุป

ความน่าจะเป็นอธิบายถึงความไม่แน่นอนต่างๆ ที่เกิดขึ้นกับการทดลอง (experiment) ซึ่งจะประกอบด้วยสเปซตัวอย่าง (ผลลัพธ์ทุกกรณีที่สามารถเป็นไปได้) และการวัดหาความน่าจะเป็น (โอกาสที่กรณีที่กำหนดจะออกมาเป็นผลลัพธ์) และได้นิยามคำว่า เหตุการณ์ อีกเซตของผลลัพธ์ที่เกิดจากการทดลอง

กฎที่สำคัญเกี่ยวกับความน่าจะเป็นได้แก่ การที่เหตุการณ์ 2 เหตุการณ์มีความสัมพันธ์ต่อกัน [$P(A \cup B) = P(A) + P(B) - P(A \cap B)$], เหตุการณ์ทั้ง 2 เหตุการณ์อิสระต่อกัน [$P(A \cup B) = P(A) + P(B)$] และ ความน่าจะปีนของเหตุการณ์หนึ่งที่จะเกิดกับเหตุการณ์เดียวกันที่จะไม่เกิดผลรวมของความน่าจะเป็นจะเท่ากับหนึ่ง [$P(A) + P(A') = 1$]

นอกจากนี้ความน่าจะเป็นของ 2 เหตุการณ์ยังอาจเกิดความน่าจะเป็นแบบมีเงื่อนไข กล่าวคือเมื่อมีเหตุการณ์หนึ่งเกิดขึ้นก่อนจะส่งผลให้ความน่าจะเป็นของเหตุการณ์ที่ตามเป็นอย่างไร โดยมีการกำหนดสัญลักษณ์ว่า $P(A \mid B)$ คือความน่าจะเป็นที่เกิดเหตุการณ์ B ก่อนแล้วส่งผลให้เกิดเหตุการณ์ A

ทฤษฎีความน่าจะเป็นในบทนี้เป็นเพียงหลักการและพื้นฐานเพื่อที่จะนำไปใช้กับความน่าจะเป็นที่ซับซ้อนขึ้น เช่น ความน่าจะเป็นของเบย์ซึ่งจะกล่าวในบทถัดไป โดยจะเกี่ยวข้องโดยตรงกับการตัดสินใจในปัญหาที่เกี่ยวข้องกับปัญญาประดิษฐ์ต่อไป

แบบฝึกหัด

1. การทดลองต่อไปนี้มีจุดตัวอย่างเป็นจำนวนเท่าใด

    1.1 โยนลูกเต๋า 1ลูก 3 ครั้ง

    1.2 โยนเหรียญ 1 เหรียญ 4 ครั้ง

    1.3 ทำข้อสอบที่มี 3 ตัวเลือก 5 ข้อ

2. กำหนดอักษร “a, b, c, d, e, f” จงหาจำนวนวิธีที่

    2.1 นำอักษรทั้งหมด 6 ตัวมาเรียงกัน

    2.2 นำอักษรเพียง 4 ตัวจาก 6 ตัวมาเรียงกัน

    2.3 นำอักษรมา 4 ตัวแต่ไม่ได้สนในลำดับ

    2.4 นำอักษร 4 ตัว มาเรียงในแนววงกลม

3. จากอักษรคำว่า “Mississippi”

    3.1 ถ้านำมาเรียงสลับกันทั้งหมดจะได้กี่วิธี

    3.2 ถ้านำมาเรียงกันทีละ 4 ตัวได้กี่วิธี

4.จากตัวเลข 0, 1, 2, 3, 4, 5 ถ้านำมาสร้างตัวเลข 3 หลัก จะได้กี่วิธี

    4.1 ถ้าตัวเลขใช้ซ้ำกันได้

    4.2 ถ้าตัวเลขใช้ซ้ำกันไม่ได้

    4.3 จากข้อ 4.2 มีกี่วิธีที่

        4.3.1 เป็นเลขคี่

        4.3.2 เป็นเลขคู่

        4.3.3 เป็นเลขที่มากกว่า 300

        4.3.4 เป็นเลขที่น้อยกว่า 250

        4.3.5 เป็นเลขที่มากกว่า 150 แต่น้อยกว่า 450

5. ในการโยนลูกเต๋า 1 ลูก 5 ครั้ง จงหาความน่าจะเป็นที่

    5.1 ลูกเต๋าขึ้นหน้า 2 หรือ 3 จำนวน 3 ครั้ง

    5.2 ลูกเต๋าขึ้นหน้า 2 หรือ 3 อย่างน้อย 2 ครั้ง

6. กล่องใบหนึ่งมีลูกบอลสีแดง 5 ลูก สีขาว 3 ลูก และสีดำ 2 ลูก สุ่มหยิบลูกบอล 4 ลูก จากกล่องใบนี้ จงหาความน่าจะเป็นที่

    6.1 ได้ลูกบอลสีแดง 2 ลูกและสีขาวและสีดำอย่างละ 1 ลูก

    6.2 ได้ลูกบอลสีแดง 3 ลูก

    6.3 ได้ลูกบอลสีแดงอย่างน้อย 1 ลูก

7. จากกล่องที่มีลูกบอล 10 ลูกจากข้อ 6 สุ่มหยิบลูกบอลมา 3 ลูก ความน่าจะเป็นที่จะได้ลูกบอลสีละลูกเป็นเท่าใด ถ้าการหยิบลูกบอล 3 ลูกนั้นเป็นแบบ

7.1 หยิบทีละลูกแล้วใส่คืน

7.2 หยิบทีละลูกแล้วไม่ใส่คืน

7.3 หยิบทีเดียว 3 ลูก

8. กล่องใบหนึ่งมีสลากที่มีหมายเลขตั้งแต่ 1-100 อย่างละ 1 ใบ สุ่มหยิบสลากมา 1 ใบ ถ้าให้ A เป็นหมายเลขสลากที่หารด้วย 3 ลงตัว และ B เป็นหมายเลขสลากที่หารด้วย 5 ลงตัว จงหา

8.1 $P(A)$ และ $P(B)$

8.2 $P(A \cap B)$ และ $P(A \cap B)$

8.3 $P(A \mid B)$ และ $P(B \mid A)$

แผนปฏิบัติการสอนสัปดาห์ที่ 11

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง ทฤษฎีของเบย์และเครือข่ายความเชื่อแบบเบย์ **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. ความหมายของตัวเลขค่าความน่าจะเป็น	IS 11.1
2. ประวัติของความน่าจะเป็นแบบเบย์	IS 11.2
3. ทฤษฎี Bayes	IS 11.3
4. เครือข่ายความเชื่อแบบเบย์	IS 11.4

ข. การจัดการเรียนการสอน

เวลา – นาที		0			60		120	180
วัตถุประสงค์		**1-2-3-4**						
การนำเข้าสู่บทเรียน		■						
ให้เนื้อหา			■	■	■	■		
สรุปเนื้อหา							■	
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**						
วิธีการสอน:	บรรยาย		■					
	อธิบาย			■	■	■		
	ถาม – ตอบ					■		
	ปฏิบัติด้วยตนเอง							■
สื่อการสอน:	กระดานดำ	■	■	■	■	■		
	คอมพิวเตอร์	■	■	■	■	■	■	■
	โปรเจคเตอร์	■	■	■	■	■	■	■

บทที่ 11
ทฤษฎีของเบย์และเครือข่ายความเชื่อแบบเบย์

ในทฤษฎีความน่าจะเป็น สถิติ การอนุมาน และ ปัญญาประดิษฐ์ บางครั้งจะพบคำว่า แบบเบย์ (Bayesian) มาขยายชื่อทฤษฎีหรือโมเดลต่างๆ โดยทุกครั้งที่พบคำขยายนี้ หมายความว่าได้มีการนำปรัชญาหรือหลักการของ ทฤษฎีความน่าจะเป็นแบบเบย์ (บางครั้งเรียก การอนุมานแบบเบย์ หรือ สถิติแบบเบย์) มาใช้กับสาขาความรู้นั้นๆ ถ้าจะกล่าวอย่างไม่เป็นทางการ ทฤษฎีความน่าจะเป็นแบบเบย์แปลความหมายของคำว่า ความน่าจะเป็น เป็นความเชื่อมั่นส่วนบุคคลในเหตุการณ์หนึ่ง ๆ ซึ่งต่างจากทฤษฎีความน่าจะเป็นของคอลโมโกรอฟ (ที่มักถูกเรียกว่าทฤษฎีความน่าจะเป็นเชิงความถี่) ที่มักแปลความหมายของความน่าจะเป็น โดยต้องแปลควบคู่ไปกับการทดลองเสมอ ดังนี้ ความน่าจะเป็นของเหตุการณ์ A คือ อัตราส่วนของจำนวนครั้งของเหตุการณ์ A ที่ทดลองสำเร็จเทียบกับจำนวนครั้งที่ทดลองทั้งหมด

ความหมายของความน่าจะเป็น

- พวกเบย์มอง ความน่าจะเป็น เป็นความเชื่อส่วนบุคคล
- พวกเชิงความถี่มอง ความน่าจะเป็น เป็นคุณสมบัติหนึ่งที่ถูกฝังอยู่ในวัตถุ (ไม่ขึ้นกับตัวบุคคล)

ในการนำทฤษฎีความน่าจะเป็นเชิงความถี่ไปใช้งานจะต้องมีการทดลองเชิงแนวคิด (conceptual experiment) ควบคู่ไปด้วยเสมอ เหตุการณ์ใดๆ ก็ตามที่ไม่มีการทดลองเชิงแนวคิดที่สมเหตุสมผลพอ จะไม่สามารถนำทฤษฎีความน่าจะเป็นเชิงความถี่ไปใช้งานได้ เช่น เราไม่สามารถจินตนาการการทดลองเพื่อทดสอบว่า มีมนุษย์ต่างดาวอยู่หรือไม่ได้ ฉะนั้นประโยคความน่าจะเป็นที่จะมีมนุษย์ต่างดาวไม่มีความหมายในทฤษฎีความน่าจะเป็นเชิงความถี่ แต่เราสามารถนำทฤษฎีความน่าจะเป็นแบบเบย์มาอ้างความน่าจะเป็นประเภทนี้ได้

ในมุมมองนี้ของปัญหาประเภทนี้อาจกล่าวได้ว่า ทฤษฎีความน่าจะเป็นแบบเบย์สามารถนำไปประยุกต์ใช้งานได้กว้างขวางมากกว่า กล่าวโดยสรุปทฤษฎีความน่าจะเป็นแบบเบย์ มีปรัชญาที่ต่างจากทฤษฎีความน่าจะเป็นเชิงความถี่เกือบสิ้นเชิง ถึงแม้จะมีสัจพจน์พื้นฐานแบบเดียวกัน โดยในทฤษฎีความน่าจะเป็นแบบเบย์นั้นมอง ความน่าจะเป็น สถิติ หรือการอนุมานเป็นเรื่องเดียวกัน

11.1 ความหมายของตัวเลขค่าความน่าจะเป็น

ความหมายของตัวเลขของความน่าจะเป็นแบบเบย์ (เช่นตัวเลข 0.72 ใน "มีความน่าจะเป็น 0.72 ที่ ...") มีการแปลในหลายความหมาย (เนื่องจากในหมู่ผู้สนับสนุนความน่าจะเป็นแบบเบย์ก็มีความเห็นไม่ค่อยตรงกันในรายละเอียดหลายๆ ประเด็น) การแปลความหมายหนึ่งที่นิยมใช้กันมากมีดังนี้

- มองในแง่ของการพนัน ดังเช่นที่ลาปลาซใช้ เช่น ความน่าจะเป็น 1/2 มีความหมายเหมือนพนัน 1 ต่อ 2 (แทง 1 ได้ 2 ไม่รวมทุน) มุมมองนี้ถูกเสนออีกครั้งโดยบรูโน เด ฟิเนตติ หนึ่งในผู้บุกเบิกทฤษฎีความน่าจะเป็นแบบเบย์ในศตวรรษที่ 20
- มองในแง่ที่ว่าทฤษฎีความน่าจะเป็น เป็นส่วนขยายของตรรกศาสตร์ นั่นคือในตรรกศาสตร์ดั้งเดิม ประพจน์จะมีค่าความจริง ได้แค่ จริง หรือ เท็จ นั่นคือ 1 หรือ 0 เท่านั้น การเพิ่มตัวเลขในช่วง 0 ถึง 1 จึงเป็นการเพิ่มความไม่แน่นอน เข้าไปในระบบตรรกศาสตร์ดั้งเดิม

สังเกตว่าตัวเลขความน่าจะเป็นต้องอยู่ในช่วง 0 ถึง 1 ตามสัจพจน์ของความน่าจะเป็น มุมมองนี้ถูกเสนอโดย แฮโรลด์ เจฟฟรีย์, ริชาร์ด คอกซ์ และ เอ็ดวิน ทอมป์สัน เจนส์ อีกทั้งยังมีผู้บุกเบิกทฤษฎีความน่าจะเป็นแบบเบย์ที่มีชื่อเสียงคนอื่นๆ คือ จอห์น เมย์นาร์ด เคนส์, เลโอนาร์ด ซาเวจ, แฟรงค์ แรมซีย์, รูดอร์ฟ คาร์นาพ และเดนนิส ลินด์ลีย์

เจนส์ให้ข้อสังเกตไว้ว่าผู้สนับสนุนทฤษฎีแบบเบย์ที่มีชื่อเสียง มักเป็นบุคคลจากสาขาอื่นที่ไม่ใช่คณิตศาสตร์ ไม่ว่าจะเป็นตัวเขาเอง และคอกซ์ที่เป็นนักฟิสิกส์, เซอร์ แฮโรลด์ เจฟฟรีย์ที่เป็นนักธรณีวิทยา, เคนส์ที่เป็นนักเศรษฐศาสตร์ หรือ คาร์นาพที่เป็นนักปรัชญาวิทยาศาสตร์ ทั้งนี้อาจเป็นเพราะบุคคลเหล่านี้ต้องการนำทฤษฎีความน่าจะเป็น ไปใช้งานจริง และต่างก็พบว่า ทฤษฎีความน่าจะเป็นเชิงความถี่ไม่กว้างขวางพอที่จะเอาไปใช้จริงได้ อีกทั้งสถิติเชิงความถี่ก็ไม่น่าเชื่อถือ และสมเหตุสมผลพอ บุคคลเหล่านี้จึงต้องพัฒนาทฤษฎีความน่าจะเป็น ที่สามารถนำไปใช้ได้จริงขึ้นมา และต่างก็ค้นพบแนวทางเดียวกัน ซึ่งก็คือสิ่งที่ลาปลาซได้แสดงไว้แล้วเมื่อราวต้นคริสต์ศตวรรษที่ 19

11.1.1 สูตรพื้นฐานของความน่าจะเป็น

สูตรพื้นฐานของความน่าจะเป็น ที่ถูกใช้บ่อยกับการเรียนรู้แบบ Bayes

- **กฎผลคูณ** : ความน่าจะเป็น $P(A \cap B)$ ที่ 2 เหตุการณ์ A และ B จะเกิดขึ้นพร้อมๆ กัน (เขียนย่อได้เป็น P(A, B))

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

- **กฎผลรวม** : ความน่าจะเป็น $P(A \cup B)$ ที่ 2 เหตุการณ์ A หรือ B เหตุการณ์ใดเหตุการณ์หนึ่งจะเกิดขึ้นหรือเกิดพร้อมกัน

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

11.2 ประวัติของความน่าจะเป็นแบบเบย์

ชื่อเรียก "แบบเบย์" เพิ่งจะมาใช้ในราวปี ค.ศ. 1950 โดยมีต้นกำเนิดมาจากชื่อของ โทมัส เบย์ ผู้ซึ่งเสนอทฤษฎีบทของเบย์เป็นคนแรก (เท่าที่ทราบในประวัติศาสตร์) ต่อมาปิแยร์ ซิมง ลาปลาซได้เสนอทฤษฎีบทของเบย์เช่นกัน โดยในขณะนั้นลาปลาซไม่ทราบว่ามีงานของเบย์อยู่ ทฤษฎีบทของเบย์เวอร์ชันลาปลาซถูกนำไปใช้งานอย่างกว้างขวางชนิดที่ตัวของเบย์เองก็อาจคาดไม่ถึง (ทั้งนี้เนื่องจากการแปลความหมายของความน่าจะเป็น ของลาปลาซนั้นกว้างมากอย่างที่ได้กล่าวไว้)

โดยลาปลาซได้นำไปในประยุกต์ใช้ในปัญหาของกลศาสตร์ ดาราศาสตร์ สถิติการแพทย์ (medical statistics) หรือแม้แต่ นิติศาสตร์ ลาปลาซได้ใช้ทฤษฎีความน่าจะเป็น (แบบเบย์) ในการทำนายมวลของดาวเสาร์โดยใช้ข้อมูลของวงโคจรดาวเสาร์ที่มีอยู่ในขณะนั้น โดยลาปลาซมั่นใจผลการทำนายมากถึงขนาดกล่าวว่า "ผมพนัน 1 ต่อ 11,000 ว่ามวลของดาวเสาร์จะคลาดเคลื่อนไม่เกิน 1/100 ของมวลที่ผมคำนวณได้" ถ้าเขายังมีชีวิตอยู่ไปอีก 150 ปี ลาปลาซคงจะได้ทราบว่าตัวเองชนะพนัน เนื่องจากในเวลานั้น พบว่ามวลของดาวเสาร์มีความคลาดเคลื่อนจากผลการคำนวณของลาปลาซเพียง 0.63% สังเกตว่าไม่มีทางที่เราจะใช้ทฤษฎีความน่าจะเป็นเชิงความถี่ ในปัญหานี้ได้เลย (ไม่สามารถสร้างการทดลองเชิงแนวคิดที่ว่า "ทดลองสร้างดาวเสาร์มา N ครั้ง มี M ครั้งที่ ..." ได้อย่างสมเหตุสมผล)

11.3 ทฤษฎี Bayes

ทฤษฎี Bayes เป็นทฤษฎีทางด้านสถิติ โดยนำความน่าจะเป็นมาใช้ประเมินความไม่แน่นอนเป็นตัวเลขได้ โดยกล่าวถึงความน่าจะเป็นของเหตุการณ์ที่จะเกิด (A) ถ้ามีเหตุการณ์อีกเหตุการณ์หนึ่งเกิดมาแล้ว (B)

ทฤษฎี Bayes สามารถเขียนอยู่ในรูปอย่างง่ายคือ

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A | B) = ความน่าจะเป็นที่เหตุการณ์ A จะเกิดถ้าเหตุการณ์ B เกิดขึ้นแล้ว

P(B | A) = ความน่าจะเป็นที่เหตุการณ์ B จะเกิดถ้าเหตุการณ์ A เกิดขึ้นแล้ว

P(A) = ความน่าจะเป็นที่จะเกิดเหตุการณ์ A

P(B) = ความน่าจะเป็นที่จะเกิดเหตุการณ์ B

ทฤษฎี Bayes สามารถนำมาใช้สำหรับเหตุการณ์ที่มีสมมุติฐาน (H) หลายๆ สมมุติฐานได้จากสูตร

$$P(H_i \mid E) = \frac{P(E|H_i)(P\,H_i)}{\sum_{n=1}^{k} P(E|\,H_n)P\,(H_n)}$$

$P(H_i \mid E)$ = ความน่าจะเป็นที่สมมุติฐาน H_i จะเป็นจริงภายใต้เหตุการณ์ E

$P(E \mid H_i)$ = ความน่าจะเป็นที่เหตุการณ์ E จะเป็นจริงภายใต้สมมุติฐาน H_i

$P(H_i)$ = ความน่าจะเป็นที่สมมุติฐาน H_i จะเป็นจริง

สูตรของ Bayes เมื่อปัญหาอยู่ในลักษณะของ Binary

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|\neg A)P(\neg A)}$$

พิสูจน์ที่มาของ Bayes' Theorem

จากความน่าจะเป็นแบบมีเงื่อนไขว่า

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad \text{...(1)}$$

และ

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \qquad \text{...(2)}$$

ดังนั้นจาก (2) *จะได้ว่า* $P(A \cap B) = P(B \mid A) \times P(A)$ ....(3)

ในขณะเดียวกัน

$$P(B \mid \neg A) = \frac{P(\neg A \cap B)}{P(\neg A)} \qquad \text{...(4)}$$

ดังนั้นจาก (4) *จะได้ว่า* $P(\neg A \cap B) = P(B \mid \neg A) \times P(\neg A)$ ....(5)

จากนิยามของความน่าจะเป็น

$P(B) = P(A \cap B) + P(\neg A \cap B)$ ....(6)

แทนค่า (3) *และ* (5) *ลงใน* (6) *จะได้ว่า*

$P(B) = P(B \mid A) \times P(A) + P(B \mid \neg A) \times P(\neg A)$ ....(7)

แทนค่า (3) และ (7) ลงใน (1) จะได้ว่า

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)}$$

การประยุกต์ใช้ความน่าจะเป็นแบบเบย์ในการอธิบายปัญหามอนตี้ฮอลล์ โดยในตัวปัญหาจะมีประตูให้เลือก 3 บาน คือ ประตู A, ประตู B, และประตู C ด้านหลังประตู 1 ใน 3 บานจะมีรถยนต์ ส่วนด้านหลังประตูอีก 2 บานที่เหลือจะมีแกะ สมมุติว่าผู้เล่นเลือกประตู A (ไม่ว่าเลือกอะไรก็มีโอกาสถูกเท่ากัน 1/3) มอนตี้จะเปิดประตูที่มีแกะให้ดู 1 บาน สมมติว่าเป็นประตู B มอนตี้ถามผู้เล่นว่าจะเปลี่ยนใจเลือกประตูใหม่หรือไม่ ถ้าคิดในมุมมองปกติความน่าจะเป็นที่หลังประตูจะมีรถก็คือ 1/3 ทำให้ไม่น่ามีผลกระทบต่อการเปลี่ยนใจ แต่ในความเป็นจริงไม่เป็นเช่นนั้น โดยสามารถนำมาคำนวณโดยใช้ทฤษฎีเบย์ได้ดังนี้

ความน่าจะเป็นที่จะมีรถหลังประตู A เมื่อเหตุการณ์มอนตี้ได้เปิดประตู B ออกมาให้ดูแล้ว

P(มีรถหลังประตู A | มอนตี้เปิดประตู B) = $\dfrac{\text{P(มีรถหลังประตู A) x P(มอนตี้เปิดประตู B | มีรถหลังประตู A)}}{\text{P(มอนตี้เปิดประตู B)}}$

P(มีรถหลังประตู A) = $\frac{1}{3}$ (เรียกว่า *Prior probability* ความน่าจะเป็นก่อนมีข้อมูลเพิ่มเติม)

P(มอนตี้เปิดประตู B | มีรถหลังประตู A) = $\frac{1}{2}$ (ถ้ามีรถหลัง A มอนตี้จะเลือกเปิด B หรือ C ก็ได้)

P(มอนตี้เปิดประตู B) การที่มอนตี้เปิด B แสดงว่ามี 2 กรณีคือรถอาจอยู่หลังประตู A หรือหลังประตู C ก็ได้ ดังนั้น

P(มอนตี้เปิดประตู B) = P(มอนตี้เปิดประตู B และ รถอยู่หลังประตู A) +
P(มอนตี้เปิดประตู B และ รถอยู่หลังประตู C)

P(มอนตี้เปิดประตู B) = P(มอนตี้เปิดประตู B | รถอยู่หลังประตู A) x P(รถอยู่หลังประตู A) +
P(มอนตี้เปิดประตู B | รถอยู่หลังประตู C) x P(รถอยู่หลังประตู C)

$$= \left(\frac{1}{2} \text{ x } \frac{1}{3}\right) + \left(1 \text{ x } \frac{1}{3}\right)$$

$$= \frac{1}{2}$$

จุดที่ให้สังเกตคือ P(มอนตี้เปิดประตู B | รถอยู่หลังประตู C) = 1 ในกรณีที่มีรถอยู่หลังประตู C ผู้เล่นเลือกประตู A มอนตี้หมดสิทธิ์เลือกประตูที่จะเปิด เขาถูกมัดมือชกให้เปิดประตู B เท่านั้น ส่วน P(มอนตี้เปิดประตู B | รถอยู่หลังประตู A) = 1/2 กรณีที่มีรถอยู่หลังประตู A แล้วผู้เล่นเลือกประตู A มอนตี้จะเลือกเปิดอะไรก็ได้ระหว่าง B กับ C ด้วยโอกาสเท่า ๆ กัน เมื่อแทนค่าสมการตั้งต้นจะได้ว่า

P(มีรถหลังประตู A | มอนตี้เปิดประตู B) = $\left(\frac{1}{3} \text{ x } \frac{1}{2}\right) / \frac{1}{2}$

$= \frac{1}{3}$

จะเห็นได้ว่าถ้ายังยืนยันการเลือกประตู A เหมือนเดิมโอกาสได้รถก็เท่าเดิม แต่ถ้าเลือกเปลี่ยนเป็นเลือกประตู C โอกาสได้มีรถเพิ่มขึ้นเป็น $\frac{2}{3}$

11.4 เครือข่ายความเชื่อแบบเบย์ (Bayesian belief network)

เครื่อข่ายงานความเชื่อแบบเบย์ (Bayesian belief network) หรือเรียกโดยย่อว่าข่ายงานเบย์(Bayes net) เป็นวิธีการเรียนรู้ที่ลดข้อจำกัดของการเรียนรู้เรื่องเบย์อย่างง่ายในสมมุติฐานของความไม่ขึ้นต่อกันระหว่างคุณสมบัติ ในวิธีการเรียนรู้เบย์อย่างง่ายจะตั้งสมมติฐานว่าคุณสมบัติใดๆ ไม่ขึ้นต่อกัน แต่ในความเป็นจริงพบว่าคุณสมบัติบางตัวจะขึ้นต่อกันบ้าง และควรที่นำความขึ้นต่อกันนี้เข้ามาใส่ไว้ในโมเดลด้วย ดังนั้นจึงสามารถใช้ข่ายงานความเชื่อเบย์ในการอธิบายความไม่ขึ้นต่อกันอย่างมีเงื่อนไข (condition independent) ระหว่างตัวแปร และในโมเดลนี้จะสามารถใช้ (1) ความรู้ก่อน (prior knowledge) เกี่ยวกับความ(ไม่)ขึ้นต่อกันระหว่างตัวแปร ร่วมกับ (2) ตัวอย่างสอน เพื่อทำให้กระบวนการเรียนรู้มีประสิทธิภาพโดยสามารถใส่ความรู้ก่อนในข่ายงานความเชื่อเบย์ให้อยู่ในรูปของโครงสร้างข่ายงานและตารางความน่าจะเป็นมีเงื่อนไข

เริ่มต้นจากนิยามความไม่ขึ้นต่อกันอย่างมีเงื่อนไข (condition independent) Z กำหนดความน่าจะเป็นของ X ไม่ขึ้นกับค่าของ Y เมื่อรู้ค่าของ Z นั่นคือ

$$(\forall\, x_{\mathrm{i}}, \mathrm{y_j}, \mathrm{z_k}) P(X = x_i | Y = \mathrm{y_j}, \mathrm{Z} = \mathrm{z_k}) = \mathrm{P}(X = x_i |, \mathrm{Z} = \mathrm{z_k})$$

หรือในรูปง่าย

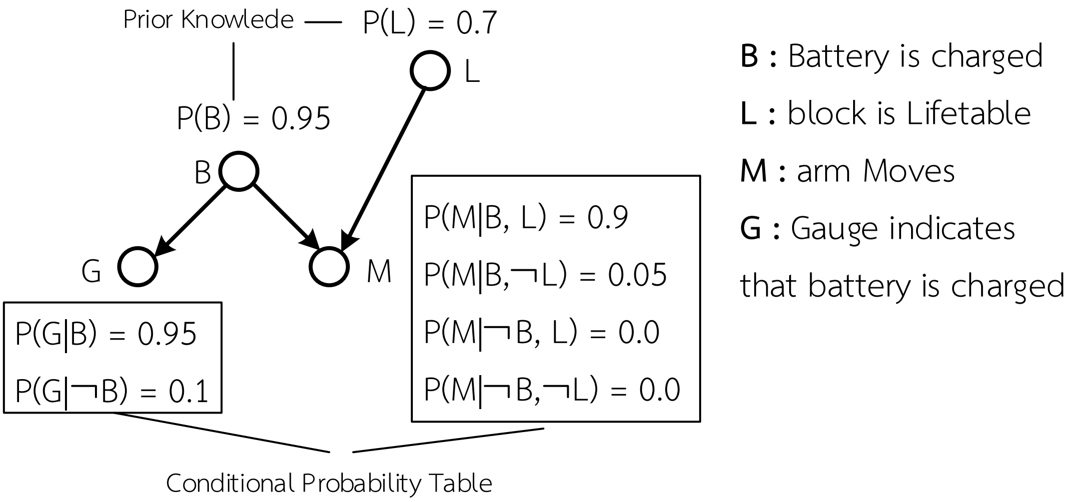
$$P(X|Y,Z) = P(X|Z)$$

นิยามด้านบนนี้หมายความว่าสำหรับ x_{i}, $\mathrm{y_j}$, $\mathrm{z_k}$ ใดๆ ความน่าจะเป็นที่ X จะมีค่าเป็น x_{i} เมื่อรู้ว่า Y มีค่าเป็น $\mathrm{y_j}$ และ Z มีค่าเป็น $\mathrm{z_k}$ ในกรณีที่ความน่าจะเป็นทั้งสองเท่ากันเช่นนี้ เราเรียกว่าค่าของ X ไม่ขึ้นกับค่าของ Y อย่างมีเงื่อนไขเมื่อรู้ค่าของ Z เราจังสามารถตัด Y ทิ้งไปได้

ตัวอย่างเช่น ความน่าจะเป็นที่ฟ้าร้องจะขึ้นอยู่กับการที่ฝนตกและฟ้าแลบ

P(Thunder | Rain, Lighting) = P (Thunder | Lighting)

ดังนั้นถ้ามีฟ้าแลบสามารถบอกได้เลยว่าจะต้องได้ยินเสียงฟ้าร้องด้วยความน่าจะเป็นเท่าไรโดยไม่ต้องสนใจว่าเกิดฝนตก



ภาพที่ 11-1 ตัวอย่างข่ายงานเบย์

จากภาพที่ 11-1 จะเห็นได้ว่าข่ายงานประกอบด้วยโหนดหลายโหนด โดยแต่ละโหนดหมายถึงคุณสมบัติของข้อมูลหรือตัวแปร และในแต่ละโหนดจะมีตารางความน่าจะเป็นมีเงื่อนไข - ซีพีที (conditional probability table - CPT) ติดอยู่ด้วย ข่ายงานเบย์นี้แสดงในรูปของกราฟมีทิศทางซึ่งสามารถบอกได้ว่ามีตัวแปรใดบ้างที่ขึ้นกับตัวแปรอื่น และตัวแปรตัวใดบ้างที่ไม่ขึ้นกับตัวอื่น ตัวอย่างเช่น โหนด M ขึ้นกับโหนด B และโหนด L หรือถ้ามองเป็นนลักษณะความสัมพันธ์ของโหนดพ่อแม่กับโหนดลูก จะเห็นว่าโหนดพ่อแม่ของ M คือ B และ L ส่วนโหนดพ่อแม่ของ G คือ B และสามารถบอกต่อได้ว่าโหนด G จะไม่ขึ้นกับโหนด L ถ้ารู้ B และได้ว่า G ไม่ขึ้นกับ M เมื่อรู้ B (สมมุติว่าตัวแปรทั้งสี่คือ G, M, B และ L เป็นตัวแปรแบบบูลีน และเขียนแทนค่าของตัวแปรอย่างง่ายโดยใช้ตัวแปรนั้นแทนค่าจริง และใส่เครื่องหมาย ¬ แทนค่าเท็จ เช่น G แทนค่าตัวแปร G เป็นจริง และ ¬G แทนค่าตัวแปร G เป็นเท็จ

โหนดใดๆ จะไม่ขึ้นกับโหนดอื่นถ้ารู้โหนดพ่อแม่โดยตรงของมัน จึงได้ว่า G จะไม่ขึ้นกับโหนดอื่นถ้ารู้โหนด B ส่วน L ไม่มีโหนดพ่อแม่ แสดงว่า L ไม่ขึ้นกับโหนดอื่นๆ เช่นเดียวกับโหนด B ก็ไม่ขึ้นกับโหนดอื่น ส่วน M ขึ้นกับ B และ L

จากข่ายงานเบย์ข้างต้น สมมุติว่าเป็นข่ายงานที่อธิบายหุ่นยนต์ตัวหนึ่งที่กำลังจะย้ายของในโดเมนโลกของบล็อก หุ่นยนต์ตัวนี้จะชาร์จแบตเตอรี และมีเกจ(G) คอยวัดว่าขณะนี้แบตเตอรีเหลืออยู่หรือไม่ หุ่นยนต์ทำงานด้วยการเคลื่อนแขนไปยกบล็อก เมื่อจำลองเหตุการณ์นี้ในข่ายงานเบย์จะได้ว่าแบตเตอรี่ (B) จะส่งผลต่อเกจ G นอกจากนั้นยังส่งผลต่อ M (การเคลื่อนแขนของหุ่นยนต์) จากนั้นจึงใส่ความรู้ก่อนหน้าเข้าไปในรูปของตารางความน่าจะเป็นมีเงื่อนไขว่า 70% ของบล็อกทั้งหมดสามารถยกได้ (P(L) = 0.7) และในเวลา 100 ชั่วโมงมี 95 ชั่วโมงที่แบตเตอรี่มีไฟ (P(B) = 0.95)

เมื่อดูที่ซีพีทีของ G พบว่าแบตเตอรี่เมื่อมีไฟ เกจซึ่งมีความบกพร่องอยู่บ้างนี้จะแสดงผลว่ามีไฟด้วยความน่าจะเป็นเท่ากับ 0.95 (P(G | B) = 0.95) และถ้าไฟหมดแต่เกจยังแสดงว่ามีไฟด้วยความน่าจะเป็นเท่ากับ 0.1 (P(G | ¬B) = 0.1)

ในตารางซีพีทีของโหนด M นั้น P(M | B, L) = 0.9 หมายความว่าหุ่นยนต์จะเคลื่อนแขนถ้าแบตเตอรีมีไฟและบล็อกสามารถยกได้ด้วยความน่าจะเป็น 0.9 แต่ถ้าแบตเตอรีมีไฟแต่บล็อกไม่สามารถยกได้ความน่าจะเป็นที่แขนกลจะเคลื่อนที่คือ 0.05 (P(M | B, ¬L) = 0.05) สำหรับกรณีที่แบตเตอรีไม่มีไฟแต่บล็อกสามารถยกได้ หุ่นยนต์ก็จะไม่เคลื่อนแขน (P(M | ¬B, L) = 0.0) และกรณีสุดท้ายคือถ้าแบตเตอรีไม่มีไฟและบล็อกยกไม่ได้แขนกลก็จะไม่เคลื่อนเช่นกัน (P(M | ¬B, ¬L) = 0.0)

ทั้งหมดนี้คือความน่าจะเป็นทั้งหมดที่ต้องป้อนให้ระบบในรูปของซีพีที โดยผู้ที่ป้อนข้อมูลคือผู้เชี่ยวชาญที่ทำงานเกี่ยวกับหุ่นยนต์ เมื่อทราบค่าต่างๆ ทั้งหมดนี้แล้วก็สามารถที่จะคำนวณความน่าจะเป็นต่างๆ ที่เกิดขึ้นภายในระบบนี้ได้ เช่น ถ้าต้องการคำนวณหาว่าความน่าจะเป็นที่ แบตเตอรี่มีไฟ บล็อกสามารถยกได้ เกจแสดงไฟ และหุ่นยนต์เคลื่อนแขน ทั้งสี่เหตุการณ์เกิดขึ้นพร้อมกันว่ามีค่าเท่าไหร่ก็สามารถคำนวณได้จากข่ายงานเบย์นี้

เพื่อให้สามารถหาความน่าจะเป็นต่างๆ จากเครือข่ายงานเบย์ได้จำเป็นต้องรู้พื้นฐานความน่าจะเป็นหนึ่งตัวก่อน เรียกว่า ความน่าจะเป็นร่วม *(Joint probability)* ระหว่างตัวแปรคือความน่าจะเป็นที่ตัวแปรหลายตัวจะมีค่าตามที่กำหนด เช่น P(Battery, Liftable, Gauge, Move) เป็นต้น โดยสามารถเขียนความน่าจะเป็นร่วมให้อยู่ในรูปทั่วไปได้เป็น

$$P(y_1,\ldots y_n) = \prod_{i=1}^{n} P(y_i \mid \text{Parents}(y_i))$$

โดยที่ Parents (y_1) หมายถึงโหนดพ่อแม่โดยตรงของโหนด y_1

ตัวอย่าง ความน่าจะเป็นที่แบตเตอรี่มีไฟ, บล็อกยกได้, เกจแสดงไฟ และหุ่นยนต์เคลื่อนแขน

P(B, L, G, M) = P(B) x P(L) x P(G | B) x P(M | B, L)

= 0.95 x 0.7 x 0.95 x 0.9

= 0.57

กรณีตัวอย่างที่ยกมาเป็นกรณีง่ายๆ เพราะรู้ค่าความน่าจะเป็นครบทั้งสี่ตัวแล้ว แต่ในกรณีเช่นทราบค่าของตัวแปรเพียงแค่ 2 หรือ 3 ตัว โดยไม่ทราบทั้งหมดก็สามารถใช้เทคนิคในการอนุมานของข่ายงานเบย์เพื่อหาความน่าจะเป็นร่วมได้เช่นกัน โดยใช้เทคนิคการอนุมานทั่วไปสำหรับข่ายงานเบย์ ซึ่งจะมีใช้งานกันอยู่ 3 เทคนิค เพื่อหาความน่าจะเป็นของตัวแปรที่เราสนใจ

11.4.1 การอนุมานจากเหตุ (Causal reasoning)

เมื่อทราบเหตุ ก็สามารถหาได้ว่าผลจะเกิดขึ้นด้วยความน่าจะเป็นเท่าไหร่ เช่น ความน่าจะเป็นที่แขนจะเคลื่อนเมื่อรู้ว่าบล็อกสามารถยกได้ หรือ P(M | L) (บล็อกยกได้เป็นสาเหตุหนึ่งของการที่หุ่นยนต์จะเคลื่อนแขน) แต่เมื่อไม่ทราบค่า B (ไม่ทราบว่าขณะนั้นแบตเตอรี่มีไฟหรือไม่) ถ้าหากย้อนกลับไปดูในข่ายงานเบย์ในภาพที่ 11-1 จะเห็นได้ว่าไม่สามารถคำนวณ P(M|L) ได้โดยตรง เพราะไม่มีค่าบอกไว้ในตาราง ในตารางมีค่าที่ใกล้เคียงที่สุดคือ P(M | B, L) ดังนั้นจึงต้องพยายามกระจาย P(M | L) ให้อยู่ในรูปที่เกี่ยวข้อง ในที่นี้จะใช้ทฤษฎีความน่าจะเป็นทั้งหมดที่กล่าวว่า ถ้าเหตุการณ์ $A_1,\ldots, A_n$ ไม่เกิดร่วมกันและ $\sum P(A_i) = 1$ แล้ว $P(B) = \sum P(B \mid A_i)P(A_i) = \sum P(B, A)$ จะสามารถนำ $A_1,\ldots, A_n$ กระจายเข้ามาได้ ดังนั้นจึงสามารถกระจาย P(M | L) ให้อยู่ในรูปของผลรวมของความน่าจะเป็นร่วมระหว่าง M กับโหนดพ่อแม่อื่นนอกจาก L (ซึ่งก็คือ B ที่เป็นโหนดพ่อแม่ของ M ด้วย) ได้เป็น

$$\begin{aligned}
P(M \mid L) &= P(M \mid B, L) \times P(B \mid L) + P(M \mid \neg B, L) \times P(\neg B \mid L) \\
&= P(M \mid B, L) \times P(B) + P(M \mid \neg B, L) \times P(\neg B) \\
&= (0.9 \times 0.95) + (0.0 \times 0.05) \\
&= 0.855
\end{aligned}$$

11.4.2 การอนุมานจากผล (diagnosis reasoning)

วิธีนี้จะตรงข้ามกับวิธีแรกกล่าวคือ เราทราบผลแล้วแต่อยากทราบว่าสาเหตุจะเกิดขึ้นด้วยความน่าจะเป็นเท่าไหร่ เช่นต้องการคำนวณหาความน่าจะเป็นที่บล็อกจะไม่สามารถยกได้เนื่องมาจากแขนหุ่นยนต์ไม่เคลื่อนไหว P(¬L | ¬M)

จากกฎของเบย์จะได้ว่า

$$P(\neg L \mid \neg M) = \frac{P(\neg M \mid \neg L)P(\neg L)}{P(\neg M)}$$

ในส่วนของ P(¬M | ¬L) สามารถคำนวณได้โดยใช้การอนุมานจากเหตุดังนี้

$$\begin{aligned}
P(\neg M \mid \neg L) &= P(\neg M \mid B, \neg L) \times P(B \mid \neg L) + P(\neg M \mid \neg B, \neg L) \times P(\neg B \mid \neg L) \\
&= P(\neg M \mid B, \neg L) \times P(B) + P(\neg M \mid \neg B, \neg L) \times P(\neg B) \\
&= (0.95 \times 095) + (1.0 \times 0.05) \\
&= 0.9525
\end{aligned}$$

$P(\neg L) \quad = 1 - P(L) = 1 - 0.7 = 0.3$

แต่พบว่ายังมี $P(\neg M)$ ที่ไม่ทราบค่าอีก ซึ่งการหาค่าโดยตรงค่อนข้างยุ่งยาก ดังนั้นจึงควรไป $P(L \mid \neg M)$ เนื่องจาก

$$P(\neg L \mid \neg M) + P(L \mid \neg M) = 1$$

จากกฎของเบย์

$$P(L \mid \neg M) = \frac{P(\neg M \mid L)P(L)}{P(\neg M)}$$

ในส่วนของ $P(\neg M \mid L)$ สามารถคำนวณได้โดยใช้การอนุมานจากเหตุดังนี้

$$\begin{aligned} P(\neg M \mid L) &= P(\neg M \mid B, L) \times P(B \mid L) + P(\neg M \mid \neg B, L) \times P(\neg B \mid L) \\ &= P(\neg M \mid B, L) \times P(B) + P(\neg M \mid \neg B, L) \times P(\neg B) \\ &= (0.1 \times 0.95) + (1.0 \times 0.05) \\ &= 0.145 \\ P(L) &= 0.7 \end{aligned}$$

จากสมการ

$$P(\neg L \mid \neg M) + P(L \mid \neg M) = 1$$

$$\frac{P(\neg M \mid \neg L)P(\neg L)}{P(\neg M)} + \frac{P(\neg M \mid L)P(L)}{P(\neg M)} = 1$$

ย้ายข้างสมการ

$$\begin{aligned} P(\neg M) &= P(\neg M \mid \neg L)P(\neg L) + P(\neg M \mid L)P(L) \\ &= (0.9525 \times 0.3) + (0.145 \times 0.7) \\ &= 0.38725 \end{aligned}$$

นำค่าของ *P(¬M)* ที่หาได้ไปแทนในสมการ

$$\begin{aligned} P(\neg L \mid \neg M) &= \frac{P(\neg M \mid \neg L)P(\neg L)}{P(\neg M)} \\ &= (0.9525 \times 0.3) / 0.38725 \\ &= 0.738 \end{aligned}$$

11.4.3 การอธิบายลดความเป็นไปได้ (explaining away)

เป็นการทำการอนุมานจากเหตุภายในการอนุมานจากผล เป็นการผสมระหว่างวิธีการทั้งสองแบบข้างต้น เช่นถ้าทราบ ¬M (แขนไม่เคลื่อน) ก็จะสามารถคำนวณ ¬L หรือความน่าจะเป็นที่บล็อกไม่สามารถยกได้ แต่ถ้ารู้ ¬B แล้ว ¬L ควรจะมีค่าความน่าจะเป็นน้อยลง ในกรณีเรียกว่า ¬B อธิบาย ¬M ทำให้ ¬L มีความน่าจะเป็นไปได้น้อยลง

$$P(\neg L \mid \neg B, \neg M) = \frac{P(\neg B, \neg M \mid \neg L)P(\neg L)}{P(\neg B, \neg M)}$$

$$= \frac{P(\neg M \mid \neg B, \neg L)P(\neg B \mid \neg L)P(\neg L)}{P(\neg M \mid \neg B)P(\neg B)}$$

$$= \frac{P(\neg M \mid \neg B, \neg L)P(\neg B)P(\neg L)}{P(\neg M \mid \neg B)P(\neg B)}$$

จากค่าความน่าจะเป็นในตารางของแต่ละโหนดพบว่า

$P(\neg M \mid \neg B, \neg L) = 1.0 - 0 = 1.0$

$P(\neg B) = 1 - 0.95 = 0.05$

$P(\neg L) = 1 - 0.7 = 0.3$

ยังขาดค่าของ $P(\neg M \mid \neg B)$ ซึ่งสามารถคำนวณได้ด้วยการอนุมานจากเหตุดังนี้

$$P(\neg M \mid \neg B) = [P(\neg M \mid \neg B, L) \times P(L \mid \neg B)] + [P(\neg M \mid \neg B, \neg L) \times P(\neg L \mid \neg B)]$$

$$= [P(\neg M \mid \neg B, L) \times P(L)] + [P(\neg M \mid \neg B, \neg L) \times P(\neg L)]$$

$$= [1.0 \times 0.7] + [1 \times 0.3] = 1.0$$

แทนค่าความน่าจะเป็นทั้งหมดลงในสมการ $\frac{P(\neg M \mid \neg B, \neg L)P(\neg B)P(\neg L)}{P(\neg M \mid \neg B)P(\neg B)}$

จะได้ว่า

$$P(\neg L \mid \neg B, \neg M) = (1.0 \times 0.05 \times 0.3) / (1.0 \times 0.05)$$

$$= 0.015 / 0.05$$

$$= 0.3$$

11.5 สรุป

ทฤษฎีเบย์เป็นหลักการที่ใช้ในการแก้ปัญหาเรื่องของความไม่แน่นอน เกี่ยวข้องโดยตรงกับความน่าจะเป็นและทฤษฎีทางสถิติ เพื่อใช้ประเมินค่าความไม่แน่นอนออกมาเป็นตัวเลข ทฤษฎีของเบย์ถูกนำมาประยุกต์ใช้กับปัญญาประดิษฐ์ เพื่อคำนวณหาตัวเลขของการวิเคราะห์ และช่วยให้ตัดสินใจความน่าเชื่อถือของเหตุการณ์ได้

สำหรับเครือข่ายความเชื่อเบย์ เป็นแบบจำลองกราฟของความน่าจะเป็น ซึ่งเป็นวิธีการนำเสนอกลุ่มของตัวแปรและค่าความน่าจะเป็นที่อิสระต่อกัน โดยนำองค์ความรู้ทั่วไปที่มีความสัมพันธ์หรือเกี่ยวข้องกันมานำเสนอเป็นโครงข่าย โหนดในแบบจำลองอาจสัมพันธ์หรือม่สัมพันธ์กันทั้งหมดก็ได้ การเชื่อมโยงแต่ละโหนดจะมีเงื่อนไขที่แสดงถึงเหตุการณ์หรือความน่าจะเป็นต่างๆ ซึ่งระบุความสัมพันธ์ตามเหตุการณ์ที่สนใจ ซึ่งแบบจำลองดังกล่าวจะนำไปสู่การคำนวณหาความน่าจะเป็นโดยอาศัยตาราง CPT เพื่อให้ทราบค่าความน่าจะเป็นของเหตุการณ์ที่มีความสัมพันธ์กันต่อไป

แบบฝึกหัด

1. บ๊อบสามารถเดินทางจากบ้านไปทำงานได้ 3 วิธี รถยนต์, รถเมล์ หรือ รถไฟ
   - เนื่องจากรถจะติดมาก ถ้าบ๊อบขับรถมาเองมีโอกาสเข้างานสาย 50%
   - ถ้าบ๊อบขึ้นรถเมล์ BRT ก็ยังมีบางครั้งที่รถติด บ๊อบจะมีโอกาสเข้างานสาย 20%
   - แต่ถ้าบ๊อบขึ้นรถไฟมาทำงาน โอกาสที่บ๊อบจะเข้างานสายมีแค่ 1% แต่มีราคาที่ค่อนข้างแพง

   วันหนึ่งบ๊อบมาทำงานสาย หัวหน้างานอยากทราบว่ามีความน่าจะเป็นเท่าไรที่บ๊อบขับรถมาทำงาน โดยหัวหน้าให้ความสำคัญต่อวิธีการเดินทางที่บ๊อบจะเลือกเท่าๆกัน

2. ในคลีนิกแห่งหนึ่งมี 15% คนไข้มีเชื้อ HIV จากผลการตรวจเลือดของคนไข้พบว่า ถ้าคนไข้มีเชื้อ HIV แล้วจะตรวจพบ 95% ถ้าคนไข้ไม่มีเชื้อ HIV ผลตรวจผิดที่บอกว่าคนไข้นั้นมีเชื้อ HIV อยู่ที่ 2%
   ถ้าผลตรวจพบว่าคนไข้มีเชื้อ HIV จงหาความน่าจะเป็นที่คนไข้
   - มีเชื้อ HIV
   - ไม่มีเชื้อ HIV

   ถ้าผลตรวจพบว่าคนไข้นั้นไม่มีเชื้อ HIV จงหาความน่าจะเป็นที่คนไข้
   - มีเชื้อ HIV
   - ไม่มีเชื้อ HIV

3. ในการตรวจสอบผู้ติดยาเสพติดในบริษัทแห่งหนึ่ง ซึ่งเครื่องมือวัดผู้ติดยาเสพติดมีความแม่นยำถึง 99% ในการตรวจสอบผู้ที่ติดยาเสพติด และมีความแม่นยำ 99% ในการบ่งบอกผู้ที่ไม่ติดยาเสพติด ในบริษัทนี้มีผู้ติดยาเสพติดจริงประมาณ 0.5% จงหาความน่าจะเป็นที
    - การตรวจสอบจะให้ผลติดยาสำหรับผู้ที่ติดยา
    - การตรวจสอบจะให้ผลติดยาสำหรับผู้ที่ไม่ติดยา
4. แมรี่จะแต่งงานพรุ่งนี้ซึ่งจะจัดงานกลางแจ้งในเขตประเทศทะเลทราย ซึ่งจากสถิติที่ผ่านมาจะมีฝนตก 5 วันต่อปีเท่านั้น โชคร้ายที่นักพยากรณ์อากาศได้คาดการณ์ว่าฝนจะตกในวันพรุ่งนี้ เมื่อนักพยากรณ์อากาศคาดการณ์เรื่องฝนตก ฝนจะตกจริง 90% ถ้าฝนไม่ตกนักพยากรณ์อากาศคาดการณ์ผิดว่าฝนตก 10% จงหาความน่าจะเป็นที่ฝนจะตกในวันแต่งงานของแมรี่ (กำหนดให้ 1 ปี มี 365 วัน)
5. จากข้อมูลดังต่อไปนี้
    - สูบบุหรี่ หรือ การกินอาหารไม่ถูกสุขลักษณะ หรือทั้งคู่จะทำให้เกิดโรคหัวใจ
    - โรคหัวใจ จะทำให้เกิด ความดันสูง หรือ การเต้นของหัวใจผิดจังหวะ
    - โดยกำหนดให้
        - S คือ สูบบุหรี่ (smoking)
        - D คือ การกินอาหารไม่ถูกสุขลักษณะ (bad diet)
        - H คือ โรคหัวใจ (Heart disease)
        - B คือ ความดันสูง (High blood pressure)
        - E คือ การเต้นของหัวใจผิดจังหวะ (abnormal electrocardiogram)
    - จากการสำรวจพบว่า
        - คนที่สูบบุหรี่คิดเป็น 30% ของคนทั้งหมด
        - คนที่กินอาหารไม่ถูกสุขลักษณะคิดเป็น 40% ของคนทั้งหมด
        - คนที่เป็นโรคหัวใจ เนื่องด้วยสูบบุหรี่และกินอาหารไม่ถูกสุขลักษณะมี 80%
        - คนที่เป็นโรคหัวใจ เนื่องด้วยการกินอาหารไม่ถูกสุขลักษณะแต่ไม่สูบบุหรี่มี 50%
        - คนที่เป็นโรคหัวใจ เนื่องด้วยสูบบุหรี่แต่ไม่กินอาหารที่ไม่ถูกสุขลักษณะมี 40%
        - คนที่เป็นโรคหัวใจ โดยที่ไม่สูบบุหรี่และไม่กินอาหารที่ไม่ถูกสุขลักษณะมี 10%
        - คนที่เป็นความดันสูงเนื่องจากโรคหัวใจมี 70%
        - คนที่เป็นความดันสูงที่ไม่ได้มีสาเหตุมาจากโรคหัวใจมี 10%
        - คนที่เป็นโรคหัวใจเต้นไม่เป็นจังหวะเนื่องจากโรคหัวใจมี 80%
        - คนที่เป็นโรคหัวใจเต้นไม่เป็นจังหวะที่ไม่ได้มีสาเหตุมาจากโรคหัวใจมี 10%

    จงเขียน Bayesian network ให้สมบูรณ์

แผนปฏิบัติการสอนสัปดาห์ที่ 12

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง ความไม่แน่นอนและการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. ความไม่แน่นอน	IS 12.1
2. การจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด	IS 12.2

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 12
ความไม่แน่นอนและการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด

ในบทนี้จะแบ่งออกเป็น 2 หัวข้อหลักๆ ที่ไม่เกี่ยวข้องกัน หัวข้อแรกจะเกี่ยวกับความไม่แน่นอนที่ถูกนำมาประยุกต์ใช้กับระบบผู้เชี่ยวชาญแบบฐานกฎ เนื่องจากองค์ความรู้ที่อยู่ในระบบผู้เชี่ยวชาญส่วนใหญ่มักเป็นองค์ความรู้ที่มีใจความสมบูรณ์ และเนื้อหาชัดเจน สามารถหาข้อพิสูจน์ได้จากการค้นหาค่าความจริง แต่แท้จริงแล้วประโยคหรือเนื้อหาบางส่วนในอีกแง่มุมอาจถูกมองว่าขาดความชัดเจนและมีความคลุมเครือ ทำให้เกิดความสงสัยและไม่มั่นใจ สิ่งที่เกิดขึ้นดังนี้เรียกว่า “ความไม่แน่นอน” และหัวข้อที่ 2 จะเกี่ยวกับการเรียนรู้ของเครื่องจักรเป็นเทคนิคการหาหมวดหมู่ของสิ่งที่ไม่เคยพบเห็นให้อยู่ในหมวดหมู่ที่มีอยู่แล้วเป็นข้อมูลเดิม โดยเทคนิคการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุดนี้ เป็นเทคนิคการจัดหมวดหมู่ที่สามารถเข้าใจและพัฒนาอัลกอริทึมได้ง่าย

12.1 ความไม่แน่นอน

ความไม่แน่นอน (uncertainty) หมายถึง สถานะภาพความลังเล สงสัย ไม่แน่ใจ หรือไม่มั่นใจ ส่งผลให้สถานการณ์เกิดความไม่มั่นคง และผลลัพธ์ที่ได้มีความไม่แน่นอน เนื่องจากขัดปัจจัยที่จะรับประกันความถูกต้องและสมบูรณ์ ความไม่แน่นอนเหล่านี้อาจเกิดได้จากหลายสาเหตุขึ้นอยู่กับสถานการณ์ต่างๆ เช่น การสำรวจด้วยแบบสอบถาม ซึ่งผู้ตอบแบบสอบถามแต่ละคนอาจมีความเห็นไม่ตรงกัน ในบางส่วนอาจะมีความเห็นแตกต่างจากคนส่วนใหญ่โดยสิ้นเชิง ทำให้มีค่าความไม่แน่นอนเกิดขึ้น ทั้งนี้อาจแตกต่างกันตามกลุ่มเป้าหมายที่ตอบแบบสอบถามด้วย ความไม่แน่นอนอาจทำให้เกิดการเบี่ยงเบนของข้อมูลซึ่งเกิดจากปัจจัยหลายอย่าง เช่น

- ข้อมูลไม่ครบถ้วน มีเพียงบางส่วนเท่านั้น
- หลักฐานยืนยันความน่าเชื่อถือมีไม่เพียงพอที่จะสร้างความมั่นใจในตัวข้อมูลได้
- การใช้ภาษามีรูปแบบที่เข้าใจยาก หรือตีความได้ยาก
- ข้อมูลจากแหล่งอ้างอิงแตกต่างกัน จนเกิดความขัดแย้ง
- ไม่สามารถชี้แจ้งข้อมูลด้วยเหตุและผลได้อย่างชัดเจน
- ข้อมูลอาจได้มาจากการประมาณแบบคร่าวๆ ทำให้เกิดความไม่มั่นใจ
- แหล่งที่มาของข้อมูลขาดความน่าเชื่อถือ

ในโลกของความเป็นจริงมีความซับซ้อนและมีปัจจัยแวดล้อมอยู่มากมาย สิ่งเหล่านี้อาจทำให้เกิดข้อสงสัยและคำถามอยู่เสมอถ้ามีการเปลี่ยนแปลงใดๆ เกิดขึ้น อีกทั้งยังเป็นตัวการทำให้เกิดความไม่มั่นคงและความไม่แน่นอนด้วย การแก้ไขปัญหาหรือการแทนองค์ความรู้จึงจำเป็นต้องพิจารณาอย่างรอบคอบ เนื่องจากการตีความของประโยคหรือปัญหาอาจทำได้หลายรูปแบบ เช่น น้ำหนักของผู้หญิง ต่างคนต่างมีมุมมองที่แตกต่างกันว่าผู้หญิงที่มีน้ำหนัก 50 กิโลกรัมเป็นคนอ้วนหรือผอม อาจได้ผลลัพธ์ที่ไม่ถูกต้องเสมอไป หากต้องการผลลัพธ์ที่ถูกต้องควรอาศัยหลักเกณฑ์หรือมาตรฐานน้ำหนักของผู้หญิง และปัจจัยแวดล้อมอื่นๆ มาช่วยในการตัดสิน เช่น อายุ และส่วนสูง เป็นต้น

ดังนั้นในการแก้ไขปัญหาเรื่องของความไม่แน่นอน จึงต้องอาศัยหลักการทางคณิตศาสตร์และทฤษฎีทางสถิติเข้ามาช่วย เนื่องจากหลักการดังกล่าวสามารถวิเคราะห์ค่าความไม่แน่นอนได้ในระดับหนึ่ง อีกทั้งมีทฤษฎีต่างๆ ที่เกี่ยวข้องกับความไม่แน่นอนของข้อมูลโดยตรง เช่น ทฤษฎีความน่าจะเป็น เป็นต้น นอกจากนี้ยังมีทฤษฎีต่างๆ ทางด้านปัญญาประดิษฐ์ที่สามารถนำมาใช้แก้ปัญหาความไม่แน่นอนได้ เช่น ตรรกศาสตร์กำกวมหรือตรรกะคลุมเครือ (Fuzzy Logic) และเทคนิคการอนุมาน เป็นต้น

12.1.1 การแทนค่าความไม่แน่นอน

การแทนค่าความไม่แน่นอนสามารถทำได้หลายวิธี ซึ่งแต่ละวิธีจะมีการนำเสนอค่าความไม่แน่นอนด้วยข้อมูลหลายลักษณะ เพื่อทำให้เข้าใจได้ง่ายขึ้นเช่น แบบตัวเลข แผนภูมิ และสัญลักษณ์ เป็นต้น วิธีการพื้นฐานในการแทนค่าความไม่แน่นอนมีดังนี้

12.1.1.1 การแทนค่าด้วยตัวเลข

การแทนค่าด้วยตัวเลขเป็นวิธีพื้นฐานที่สะดวกและง่ายที่สุด โดยกำหนดช่วงของตัวเลข เช่น เลข 0 แทนค่าความไม่แน่นอน และเมื่อตัวเลขเพิ่มขึ้นแสดงว่าค่าความแน่นอนเพิ่มขึ้นด้วย เป็นต้น การแทนค่าด้วยตัวเลขจะช่วยให้เข้าใจง่ายขึ้น เนื่องจากมีค่าของตัวเลขเป็นตัวบ่งบอกระดับความน่าเชื่อถือของข้อมูลได้ แต่ทั้งนี้ขึ้นอยู่กับทัศนคติของผู้เชี่ยวชาญด้วย โดยผู้เชี่ยวชาญแต่ละคนอาจแทนระดับค่าด้วยตัวเลขที่แตกต่างกัน ดังนั้นในบางกรณีอาจเกิดความขัดแย้งขึ้นได้ เนื่องจากทัศนคติของผู้เชี่ยวชาญกับบุคคลทั่วไปอาจแตกต่างกันมากเกินไปทำให้การยอมรับค่าตัวเลขเป็นไปได้ยาก ส่งผลให้ค่าตัวเลขดังกล่าวไม่มีมาตรฐาน

12.1.1.2 การแทนค่าด้วยกราฟและแผนภูมิ

การแทนค่าความไม่แน่นอนด้วยกราฟและแผนภูมิเป็นวิธีการที่ช่วยลดความไม่มั่นใจและความขัดแย้งที่เกิดจากการกำหนดระดับความแน่นอนด้วยตัวเลขได้ เนื่องจากการตีความของค่าความไม่แน่นอนมีการกำหนดช่วงความถี่ของกราฟ เพื่อแสดงความแตกต่างของระดับค่าความแน่นอน ถึงแม้ว่าจะไม่สามารถบอกค่าความแม่นยำได้มากเท่าที่ควร แต่ถ้าผู้เชี่ยวชาญมีความชำนาญและมีทัศนคติที่ดี การกำหนดช่วงความถี่ของกราฟจะช่วยให้ผู้อื่นตีความได้ง่าย และลดความขัดแย้งของผู้เชี่ยวชาญที่ไม่เห็นด้วยกับการแทนค่าความไม่แน่นอนด้วยตัวเลขได้ในระดับหนึ่ง

12.1.1.3 การแทนค่าด้วยสัญลักษณ์

เป็นการแทนค่าความไม่แน่นอนโดยการกำหนดมาตรวัดหรือสเกล ซึ่งจะคล้ายกับวิธีการแทนค่าด้วยกราฟและแผนภูมิ แต่จะมีการกำหนดสเกลเป็นส่วนๆ หรือเป็นระดับอย่างชัดเจน โดยส่วนใหญ่นิยมใช้สเกลวัดความเป็นไปได้ของ Likert ซึ่งใช้แสดงความคิดเห็นโดยแบ่งความคิดเห็นออกเป็น 5 ระดับได้แก่

- ระดับที่ 1 ไม่น่าเป็นไปได้มากที่สุด (Very Unlikely)
- ระดับที่ 2 ไม่น่าเป็นไปได้ (Unlikely)
- ระดับที่ 3 เป็นกลาง (Nature)
- ระดับที่ 4 น่าจะเป็นไปได้ (Likely)
- ระดับที่ 5 น่าจะเป็นไปได้มากที่สุด (Very Likely)

การแทนค่าในลักษณะนี้ผู้เชี่ยวชาญนิยมใช้ในการอ้างอิงข้อมูลเชิงปริมาณ ซึ่งเป็นข้อมูลที่สามารถบอกได้ว่ามีค่ามากหรือน้อย เพื่อบ่งบอกปริมาณได้อย่างถูกต้องด้วยตัวเลขระดับของสเกลที่กำหนดขึ้น สเกลสามารถเรียงระดับความแน่นอนของข้อมูลได้หลายลักษณะจากมากไปน้อยหรือจากน้อยไปมาก ทั้งนี้ขึ้นอยู่กับข้อมูลที่สนใจ

12.1.2 ค่าความไม่แน่นอน (Certainty Factor)

ค่าความไม่แน่นอน (Certainty Factor: CF) เป็นค่าของตัวเลขที่ถูกกำหนดขึ้นบนพื้นฐานของเหตุการณ์ขณะนั้น เพื่อช่วยให้ยอมรับข้อสรุปของเหตุการณ์นั้นได้ ค่าของ CF มีตั้งแต่ -1 ถึง 1 เมื่อข้อสมมุติฐานที่ตั้งขึ้นมาสรุปได้ค่าของ CF เป็น 1 นั่นแสดงว่าข้อสมมุติฐานดังกล่าวมีความถูกต้อง แน่นอนและเป็นจริงตามที่กล่าวอ้าง และเมื่อ CF มีค่าเป็น -1 แสดงว่าข้อสมมุติฐานไม่เป็นจริง ในขณะเดียวกันถ้าค่า CF เป็น 0 จะมีความหมายว่า ข้อสมมุติฐานนั้นไม่สามารถสรุปได้ว่าเป็นจริงหรือเท็จ ช่วงของค่า CF และการตีความหมายแสดงดังภาพที่ 12-1

CF ถูกพัฒนาขึ้นเพื่อใช้งานกับระบบ MYCIN ซึ่งเป็นระบบผู้เชี่ยวชาญประเภทหนึ่ง สำหรับ CF อาจไม่ใช่วิธีการที่เป็นทางการในการกำหนดระดับค่าความแน่นอน ต่อการสรุปผลภายใต้สถานการณ์ที่กำหนด ค่า CF ของกฎแต่ละข้อในระบบจะถูกกำหนดขึ้นโดยผู้เชี่ยวชาญที่เกี่ยวข้อง กับสถานการณ์นั้น ดังนั้นค่า CF ที่ได้จะขึ้นอยู่กับประสบการณ์ของผู้เชี่ยวชาญ ค่า CF จะบ่งบอก ถึงความน่าเชื่อถือของข้อสรุปภายใต้สถานการณ์ เช่น รถยนต์คันหนึ่งมีเครื่องยนต์และแบตเตอรีในสภาพปกติ มีน้ำมันอยู่ปริมาณหนึ่ง ดังนั้นรถยนต์คันนี้สามารถใช้งานได้ โดยผู้เชี่ยวชาญอาจกำหนดให้มีค่า CF เป็น 0.8 ในขณะเดียวกันหากมีผู้เชี่ยวชาญที่มีประสบการณ์มากกว่าอาจจะกำหนดค่า CF เป็น 0.6 ตามปัจจัยแวดล้อมที่ใช้สรุปผลเป็นต้น

Term	Certainty Factor
ไม่ใช่อย่างแน่นอน	-1.0
เกือบจะไม่ใช่อย่างแน่นอน	-0.8
ไม่น่าจะเป็นไปได้	-0.6
อาจจะไม่ใช่	-0.4
ไม่ทราบแน่ชัด	-0.2 ถึง +0.2
อาจจะใช่	+0.4
มีความเป็นไปได้ว่าใช่	+0.6
เกือบจะใช่อย่างแน่นอน	+0.8
ใช่อย่างแน่นอน	+1.0

ภาพที่ 12-1 ตารางค่า CF ในช่วงต่างๆ และการตีความหมาย

ค่า CF ในระบบผู้เชี่ยวชาญจะนำมาใช้ควบคู่กับฐานองค์ความรู้ ซึ่งมีความเกี่ยวข้องกับกฎต่างๆ ที่ใช้ในการสร้างฐานองค์ความรู้ โดยมีโครงสร้างดังนี้

IF <Evidence (E)>

THEN <Hypothesis (H)> **{CF}**

CF ใช้เพื่อบ่งบอกความน่าเชื่อถือได้ว่า เหตุการณ์ (E) จะส่งผลให้เกิด สมมุติฐาน(H) ด้วยค่าความไม่แน่นอนดังสมการต่อไปนี้

$$CF(H, E) = CF(E) \times CF$$

โดยกำหนด

- CF(H, E) คือ ค่าความไม่แน่นอนของข้อสรุป
- CF(E) คือ ค่าความไม่แน่นอนของเหตุการณ์
- CF คือ ค่าความไม่แน่นอนของกฎ

12.1.3 การใช้งานค่าความไม่แน่นอน

ตัวอย่างการใช้งานค่าความไม่แน่นอนอย่างง่ายที่สุดคือการหาค่าความไม่แน่นอนของข้อสรุปที่มาจากกฎเพียงข้อเดียว เช่น

ตัวอย่างที่ 1 : **IF** ท้องฟ้าปลอดโปร่ง
THEN จะพยากรณ์ได้ว่าวันนี้อากาศแจ่มใส **{ CF 0.8 }**

จากตัวอย่างจะเห็นได้ว่าผู้เชี่ยวชาญได้กำหนดค่าน่าเชื่อถือของกฎไว้ด้วยค่า CF เท่ากับ 0.8 ถ้าหากในขณะนี้ท้องฟ้าปลอดโปร่งแต่ไม่มากนักโดยกำหนดให้มีค่า CF เท่ากับ 0.5 ก็จะสามารถ หาค่าความไม่แน่นอนของข้อสรุปได้ดังนี้

$$CF(H, E) = CF(E) \times CF$$
$$= 0.5 \times 0.8$$
$$= 0.4$$

ดังนั้นจึงสามารถสรุปได้ว่า วันนี้อากาศจะแจ่มใสด้วยค่า CF เท่ากับ 0.4 หรือถ้าตีความเป็นประโยคคำพูดจะได้ว่า “วันนี้อากาศ**อาจจะ**แจ่มใส”

อย่างไรก็ตามในความเป็นจริงกฎที่ใช้ในระบบผู้เชี่ยวชาญจะมีความซับซ้อน และมีการเชื่อมโยงกฎของเหตุการณ์มากกว่า 1 เหตุการณ์ เช่น ถ้า A และ B แล้ว C เป็นต้น ในกรณีที่จะต้องเชื่อมโยงกฎของเหตุการณ์มากกว่า 1 เหตุการณ์ การหาค่า CF จะแตกต่างจากเดิมตามสมการโดยมีการประยุกต์ใช้กฎการ Conjunction และกฎการ Disjunction

- กฎการ Conjunction

  $$CF(H, E_1 \wedge E_2 \wedge \ldots \wedge E_n) = \min[CF(E_1), CF(E_2),\ldots,CF(E_n)] \times CF$$

- กฎการ Disjunction

  $$CF(H, E_1 \vee E_2 \vee \ldots \vee E_n) = \max[CF(E_1), CF(E_2),\ldots,CF(E_n)] \times CF$$

ตัวอย่างที่ 2 : **IF** ท้องฟ้าปลอดโปร่ง
AND การพยากรณ์ว่าอากาศแจ่มใส
THEN ควรจะใส่แว่นตากันแดด **{CF 0.8}**

ถ้ากำหนดให้ CF ของท้องฟ้าปลอดโปร่ง (E_1) คือ 0.9
และกำหนดให้ CF ของการพยากรณ์ว่าอากาศแจ่มใส่ (E_2) คือ 0.7
ต้องการหาค่า CF ของข้อสรุปในการใส่แว่นตากันแดด

$$
\begin{aligned}
CF(H, E_1 \wedge E_2) &= \min[(CF(E_1), CF(E_2)] \times CF \\
&= \min[0.9, 0.7] \times 0.8 \\
&= 0.7 \times 0.8 = 0.56
\end{aligned}
$$

ดังนั้นสามารถสรุปได้ว่า วันนี้มี**อาจจะ**สวมแว่นตากันแดด (ด้วยค่า CF 0.56)

ในกรณีสุดท้ายคือเหตุการณ์หรือข้อมูลที่มีสามารถอนุมานจากกฎได้หลายข้อ และให้ข้อสรุปเดียวกัน จำเป็นต้องนำข้อสรุปที่ได้จากกฎแต่ละข้อนั้นมาสรุปเป็นค่าความไม่แน่นอนรวมของข้อสรุปจากสูตรต่อไปนี้

$$CF(CF_1, CF_2) = CF_1(H, E_1) + (CF_2(H, E_2) \times [1 - CF_1(H, E_1)])$$

ตัวอย่างที่ 3 : กำหนดกฎ 2 ข้อ

Rule 1 :

IF วันนี้ฝนตก

THEN พรุ่งนี้ฝนจะตก **{CF 0.5}**

Rule 2 :

IF วันนี้ร้อน

THEN พรุ่งนี้ฝนจะตก **{CF 0.8}**

ถ้าข้อเท็จจริงของวันนี้คือ ฝนกำลังตกอยู่ (CF = 1.0) และอากาศร้อน (CF = 0.7)

สำหรับกฎข้อที่ 1 : $CF_1(H, E_1) = CF(E_1) \times CF_1$

$= 1.0 \times 0.5$

$= 0.5$

สำหรับกฎข้อที่ 2 : $CF_2(H, E_2) = CF(E_2) \times CF_2$

$= 0.7 \times 0.8$

$= 0.56$

ดังนั้น CF รวมคือ $CF(CF_1, CF_2) = 0.5 + (0.56 \times [1 - 0.5])$

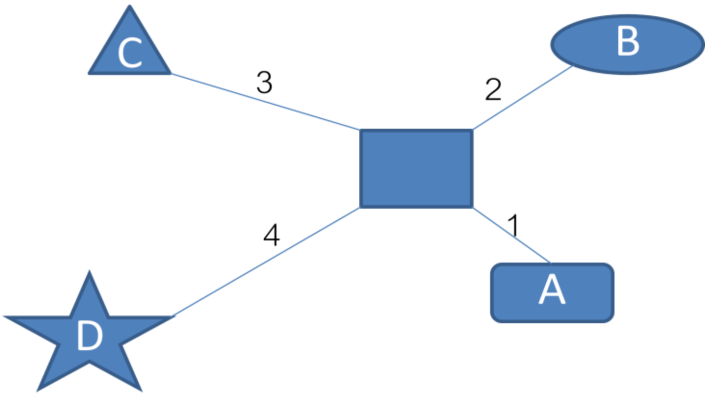
$= 0.5 + 0.28$

$= 0.78$

จึงสามารถสรุปได้ว่า พรุ่งนี้**มีความเป็นไปได้**ว่าฝนจะตกอย่างแน่นอน (ด้วยค่า CF 0.78)

12.2 การจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด

การจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด (Nearest neighbor classification) เป็นการเรียนรู้ประเภท Unsupervised Learning ซึ่งเป็นการจำแนกหรือจัดกลุ่มที่มีวิธีการไม่ซับซ้อนมาก โดยพิจารณาจากชุดข้อมูลใกล้เคียง กล่าวคือข้อมูลมีค่าใกล้เคียงกับค่าของข้อมูลที่พิจารณามากที่สุด ในที่นี้ค่าความใกล้เคียงจะหมายถึง ระยะทาง (distance) ที่มีค่าน้อยที่สุดระหว่างชุดข้อมูลกับข้อมูลที่พิจารณา ข้อมูลดังกล่าวนี้เรียกว่า “nearest neighbor” ดังแสดงในภาพที่ 12-2



ภาพที่ 12-2 ตัวอย่างระยะทางระหว่างชุดข้อมูลกับข้อมูลที่พิจารณา

จากภาพแสดงระยะทางระหว่างชุดข้อมูล A, B, C, D และข้อมูลที่พิจารณา (รูปสี่เหลี่ยมตรงกลาง) โดยตัวเลขบนเส้นเชื่อมระหว่างชุดข้อมูลและข้อมูลที่พิจารณาคือระยะทาง จะเห็นได้ว่าข้อมูลที่พิจารณา มีระยะทางใกล้กับข้อมูล A ในชุดข้อมูล (มีระยะทางเท่ากับ 1) ในทฤษฎีของ Nearest neighbor การจำแนกหมวดหมู่โดยพิจารณาจากข้อมูลในชุดข้อมูลที่ใกล้ที่สุดเพียงตัวเดียวนั้น จะเรียกว่า “1-NN” หรือ One Nearest Neighbor แต่ผลที่ได้จากการจำแนกหมวดหมู่นั้นอาจจะไม่แม่นยำเพียงพอจึงมีการประยุกต์ไปสู่ “k-NN” หรือ K - Nearest Neighbor โดย k แทนจำนวนของข้อมูลในชุดข้อมูลที่ใกล้กับ ข้อมูลที่พิจารณา

การหาระยะทางระหว่างชุดข้อมูลและข้อมูลที่พิจารณานั้น วิธีที่ง่ายที่สุดคือการใช้ทฤษฎีการวัดค่าของ Euclidean หรือเรียกว่า ระยะทาง Euclidean ซึ่งมีสูตรดังนี้

$$d = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

โดย

- d คือ ค่าระยะทาง Euclidean
- p_i คือ ค่าในชุดข้อมูลที่ต้องการจำแนก
- q_i คือ ค่าในชุดข้อมูลที่มีอยู่แล้ว

12.2.1 ตัวอย่างการจำแนกหมวดหมู่ด้วยวิธี 1-NN

กำหนดให้มีสัตว์ 5 ชนิด จำแนกจากคุณสมบัติที่สามารถบ่งบอกถึงสายพันธุ์ได้ ดังแสดงในภาพที่ 12-3 ต้องการหาหมวดหมู่ของสัตว์ปริศนาที่รู้คุณสมบัติบางประการด้วยวิธี 1-NN

	ออกลูกเป็นตัว	ออกลูกเป็นไข่	เลี้ยงลูกด้วยนม	มีขนตามร่างกาย	ผิวหยาบหรือมีเกล็ด	สัตว์เลือดอุ่น	สัตว์เลือดเย็น	อยู่ทั้งบกและน้ำ	หายใจด้วยเหงือก	
กบ		1					1	1		สัตว์ครึ่งบกครึ่งน้ำ
เป็ด		1		1		1		1		สัตว์จำพวกนก
ค้างคาว	1		1	1						สัตว์เลี้ยงลูกด้วยนม
งู		1			1		1			สัตว์เลื้อยคลาน
ปลาแซลมอล		1			1		1		1	สัตว์จำพวกปลา
สัตว์ปริศนา		1						1		???

ภาพที่ 12-3 ตัวอย่างข้อมูลสำหรับการจำแนกหมวดหมู่ด้วยวิธี 1-NN

จากชุดข้อมูลของสัตว์ปริศนา จะนำมาใช้หาค่าระยะทางระหว่างสัตว์ทั้ง 5 ชนิดในชุดข้อมูล ด้วยวิธีการคำนวณระยะทาง Euclidean เมื่อแทนค่าในแต่ละชุดข้อมูลจะได้ดังนี้

- $d_{\text{กบ}} = \sqrt{0^2 + (1-1)^2 + 0^2 + 0^2 + 0^2 + 0^2 + (0-1)^2 + (1-1)^2 + 0^2}$
  $= \sqrt{1}$
- $d_{\text{เป็ด}} = \sqrt{0^2 + (1-1)^2 + 0^2 + (0-1)^2 + 0^2 + (0-1)^2 + 0^2 + (1-1)^2 + 0^2}$
  $= \sqrt{2}$
- $d_{\text{ค้างคาว}} = \sqrt{(0-1)^2 + (1-0)^2 + (0-1)^2 + (0-1)^2 + 0^2 + 0^2 + 0^2 + (1-0)^2 + 0^2}$
  $= \sqrt{5}$
- $d_{\text{งู}} = \sqrt{0^2 + (1-1)^2 + 0^2 + 0^2 + (0-1)^2 + 0^2 + (0-1)^2 + (1-0)^2 + 0^2}$
  $= \sqrt{3}$
- $d_{\text{ปลาแซลมอล}} = \sqrt{0^2 + (1-1)^2 + 0^2 + 0^2 + (0-1)^2 + 0^2 + (0-1)^2 + (1-0)^2 + (0-1)^2}$
  $= \sqrt{4}$

จากค่าระยะทางระหว่างสัตว์ปริศนากับสัตว์ทั้ง 5 ประเภทในชุดข้อมูล พบว่าระยะทางของสัตว์ปริศนาใกล้กับ กบ มากกว่าที่สุด ดังนั้นจึงสรุปได้ว่า สัตว์ปริศนานี้เป็น **สัตว์ครึ่งบกครึ่งน้ำ**

12.2.2 ตัวอย่างการจำแนกหมวดหมู่ด้วยวิธี 3-NN

กำหนดให้มีสัตว์ 8 ชนิด จำแนกจากคุณสมบัติที่สามารถบ่งบอกถึงสายพันธุ์ได้ ดังแสดงในภาพที่ 12-4 ต้องการหาหมวดหมู่ของสัตว์ปริศนาที่รู้คุณสมบัติบางประการด้วยวิธี 3-NN

	ออกลูกเป็นตัว	ออกลูกเป็นไข่	เลี้ยงลูกด้วยนม	มีขนตามร่างกาย	ผิวหยาบหรือมีเกล็ด	สัตว์เลือดอุ่น	สัตว์เลือดเย็น	อยู่ทั้งบกและน้ำ	หายใจด้วยเหงือก	
กบ		1					1	1		สัตว์ครึ่งบกครึ่งน้ำ
เป็ด		1		1		1		1		สัตว์จำพวกนก
ค้างคาว	1		1	1						สัตว์เลี้ยงลูกด้วยนม
งู		1			1		1			สัตว์เลื้อยคลาน
ปลาแซลมอล		1			1		1		1	สัตว์จำพวกปลา
จระเข้		1			1		1			สัตว์เลื้อยคลาน
แมว	1		1	1		1				สัตว์เลี้ยงลูกด้วยนม
นกกระจอกเทศ		1		1		1				สัตว์จำพวกนก
สัตว์ปริศนา			1	1		1				???

ภาพที่ 12-4 ตัวอย่างข้อมูลสำหรับการจำแนกหมวดหมู่ด้วยวิธี 3-NN

จากชุดข้อมูลของสัตว์ปริศนา จะนำมาใช้หาค่าระยะทางระหว่างสัตว์ทั้ง 8 ชนิดในชุดข้อมูล ด้วยวิธีการคำนวณระยะทาง Euclidean เมื่อแทนค่าในแต่ละชุดข้อมูลจะได้ดังนี้

- $d_{\text{กบ}} = \sqrt{6}$
- $d_{\text{เป็ด}} = \sqrt{3}$
- $d_{\text{ค้างคาว}} = \sqrt{2}$
- $d_{\text{งู}} = \sqrt{6}$
- $d_{\text{ปลาแซลมอล}} = \sqrt{7}$
- $d_{\text{จระเข้}} = \sqrt{6}$
- $d_{\text{แมว}} = \sqrt{1}$
- $d_{\text{นกกระจอกเทศ}} = \sqrt{2}$

วิธี 3-NN ก็คือการหาข้อมูลในชุดข้อมูลที่มีอยู่ที่มีระยะทางใกล้กับข้อมูลที่พิจารณามากที่สุด 3 อันดับแรกขึ้นมาพิจารณา เมื่อคำนวณระยะทางได้แล้วจะเห็นได้ว่าสัตว์ที่มีระยะทางใกล้สุดกับสัตว์ปริศนาคือ แมว ค้างคาว และ นกกระจอกเทศ โดยมีระยะทางจากสัตว์ปริศนาคือ $\sqrt{1}$, $\sqrt{2}$ และ $\sqrt{2}$ ตามลำดับ จากนั้นให้พิจารณาจากสายพันธุ์ของสัตว์ทั้ง 3 ชนิด โดยจากข้อมูลเบื้องต้นพบว่า

- แมว มีสายพันธุ์เป็น สัตว์เลี้ยงลูกด้วยนม
- ค้างคาว มีสายพันธุ์เป็น สัตว์เลี้ยงลูกด้วยนม
- นกกระจอกเทศ มีสายพันธุ์เป็น สัตว์จำพวกนก

โดยจะเห็นได้ว่าจากสัตว์ 3 ชนิด เป็นสายพันธุ์สัตว์เลี้ยงลูกด้วยนมอยู่ 2 ชนิด และเป็นสายพันธุ์สัตว์จำพวกนก อยู่ 1 ชนิด ดังนั้นจึงสรุปตามเสียงข้างมาก ก็คือ สัตว์ปริศนามีสายพันธุ์เป็น**สัตว์เลี้ยงลูกด้วยนม**

12.3 สรุป

ในบทนี้ได้อธิบายถึงหัวข้อสำคัญที่อิสระต่อกัน 2 หัวข้อคือ ความไม่แน่นอน และการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด ซึ่งความไม่แน่นอน หมายถึง สถานะภาพความลังเล สงสัย ไม่แน่ใจ หรือไม่มั่นใจ ส่งผลให้สถานการณ์เกิดความไม่มั่นคง ความไม่แน่นอนอาจเป็นสาเหตุที่ทำให้เกิดการเบี่ยงเบนของข้อมูล การแทนค่าความไม่แน่นอนสามารถทำได้หลายวิธี แต่ละวิธีจะมีการนำเสนอค่าตวามไม่แน่นอนด้วยข้อมูลหลายลักษณะ วิธีการแทนค่าความไม่แน่นอนขั้นพื้นฐาน ได้แก่ การแทนด้วยตัวเลข การแทนค่าด้วยกราฟและแผนภูมิ และการแทนค่าด้วยสัญลักษณ์ นอกจากนี้ยังต้องอาศัยการหาเหตุผลในรูปแบบต่างๆ เช่น ค่าความไม่แน่นอน ที่เป็นตัวเลขที่ถูกกำหนดขึ้นเพื่อช่วยให้ยอมรับหรือเชื่อข้อสรุปของเหตุการณ์นั้นได้เป็นต้น

สำหรับการจัดหมวดหมู่ด้วยวิธีเพื่อนบ้านที่ใกล้ที่สุด หรือ Nearest Neighbor Classification เป็นวิธีการจำแนกชุดข้อมูล โดยพิจารณาถึงความสัมพันธ์ของชุดข้อมูลและค่าในชุดข้อมูลที่ต้องการจำแนก วิธีการนี้จะทำการคำนวณหาระยะทางระหว่างข้อมูลข้างเคียง เพื่อให้ทราบว่าชุดข้อมูลใดมีความใกล้เคียงมากที่สุด ซึ่งความใกล้เคียงดังกล่าวจะสามารถนำมาสรุปได้ว่าข้อมูลที่ทำการจำแนกจัดอยู่ในกลุ่มใด การคำนวณชุดข้อมูลใกล้เคียงอาจคัดเลือกเฉพาะชุดข้อมูลที่มีค่าใกล้เคียงมากที่สุดเพียงหนึ่งเดียว หรือชุดข้อมูลที่ใกล้เคียงหลายๆ ชุด มาประกอบการพิจารณาก็ได้ ทั้งนี้เพื่อให้ได้การจำแนกที่ถูกต้องที่สุด

แบบฝึกหัด

1. จากกฎ 3 ข้อ ต่อไปนี้

กฎข้อที่ 1 : IF วันนี้ฝนตก THEN พรุ่งนี้ฝนจะตก {CF 0.5}

กฎข้อที่ 2 : IF วันนี้อากาศแห้ง THEN พรุ่งนี้ฝนจะไม่ตก {CF 0.5}

กฎข้อที่ 3 : IF วันนี้ฝนตก AND มีความชื้นในอากาศต่ำ THEN พรุ่งนี้ฝนจะไม่ตก {CF 0.6}

จงหาค่าข้อสรุปพร้อมทั้งค่า CF ถ้าทราบข้อมูลของเหตุการณ์ดังต่อไปนี้

- หากวันนี้ฝนตกอย่างแน่นอนอากาศวันพรุ่งนี้จะเป็นเช่นไร
- หากวันนี้ไม่น่าจะเป็นไปได้ที่อากาศแห้งแล้วอากาศวันพรุ่งนี้จะเป็นเช่นไร
- ถ้าวันนี้มีความเป็นไปได้ว่าฝนจะตกและไม่มีความชื้นในอากาศต่ำอย่างแน่นอน แล้วพรุ่งนี้อากาศจะเป็นเช่นไร

2. จากข้อมูลของสัตว์ 7 ชนิด ดังตารางด้านล่าง จงหาสายพันธุ์ของสัตว์ปริศนาทั้ง 2 ชนิด โดยกำหนดให้ใช้วิธี 1-NN สำหรับการหาสายพันธุ์ของสัตว์ปริศนาตัวแรก และให้ใช้วิธี 3-NN สำหรับการหาสายพันธุ์ของสัตว์ปริศนาตัวที่สอง

	ออกลูกเป็นตัว	ออกลูกเป็นไข่	เลี้ยงลูกด้วยนม	มีขนตามร่างกาย	ผิวหยาบหรือมีเกล็ด	สัตว์เลือดอุ่น	สัตว์เลือดเย็น	อยู่ทั้งบกและน้ำ	หายใจด้วยเหงือก	
กบ		1					1	1		สัตว์ครึ่งบกครึ่งน้ำ
ปลาหมอ		1			1		1		1	สัตว์จำพวกปลา
จระเข้		1			1		1			สัตว์เลื้อยคลาน
ปลาทอง		1			1		1		1	สัตว์จำพวกปลา
เต่า		1					1	**1**		สัตว์เลื้อยคลาน
สุนัข	1		1	1		1				สัตว์เลี้ยงลูกด้วยนม
นกกระจอกเทศ		1		1		1				สัตว์จำพวกนก
สัตว์ปริศนา (1NN)	1			1						???
สัตว์ปริศนา (3NN)		1			1		1			???

แผนปฏิบัติการสอนสัปดาห์ที่ 13

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง ต้นไม้การตัดสินใจ **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. การวิเคราะห์และสร้างต้นไม้การตัดสินใจ	IS 13.1
2. โครงสร้างต้นไม้การตัดสินใจ	IS 13.2
3. การสร้างต้นไม้การตัดสินใจ	IS 13.3
4. ตัวอย่างการสร้างต้นไม้การตัดสินใจ	IS 13.4
5. ข้อด้อยของต้นไม้การตัดสินใจ	IS 13.5

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2-3-4-5**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 13
ต้นไม้การตัดสินใจ

ต้นไม้การตัดสินใจเป็นวิธีการใช้ผังต้นไม้ในการเขียนอธิบายการประมวลผลโดยจำลองกิ่งก้านสาขาของต้นไม้เป็นหลักโดยแตกจากรากทางซ้ายมือ แตกเป็นกิ่งอยู่ทางขวามือ หรืออาจจะแตกจากรากที่อยู่ด้านบน ลงมาเป็นกิ่งที่อยู่ด้านล่าง และแตกกิ่งไปเรื่อย ๆ จนครบเงื่อนไขทั้งหมดและกิ่งสุดท้ายของทุกกิ่งคือกิจกรรมที่ต้องกระทำ เมื่อเป็นตามเงื่อนไขนั้น นิยมใช้เมื่อมีเหตุการณ์ตั้งแต่ 2 เหตุการณ์ขึ้นไปที่เกิดต่อเนื่องกันโดยผู้ตัดสินใจมีทางเลือกหลายทาง แต่ไม่ทราบผลลัพธ์ของแต่ละทางเลือก จะบอกทางเลือก (วิธีการตัดสินใจ ความน่าจะเป็นของแต่ละทางเลือก รวมทั้งบอกค่าใช้จ่ายหรือสิ่งที่ต้องเสียไป เพื่อให้ได้ซึ่งผลลัพธ์ของแต่ละทางเลือก) เมื่อปัญหาคลี่คลายหรือเวลาผ่านไป อาจจะพบทางเลือกใหม่ๆ หรือได้ผลลัพธ์/ผลตอบแทนในขั้นสุดท้าย และทำการตัดสินใจคัดสรรทางเลือกที่คาดว่าจะได้ผลตอบแทนที่ดีที่สุด

13.1 การวิเคราะห์และสร้างต้นไม้การตัดสินใจ

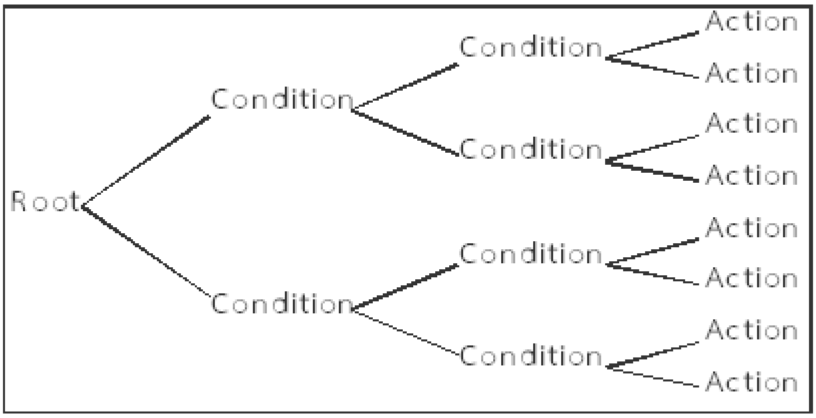
13.1.1 ขั้นตอนการวิเคราะห์แบบกิ่งก้านสาขา

1. ระบุปัญหา
2. ร่างโครงสร้างการวิเคราะห์แบบกิ่งก้านสาขา
3. ระบุความน่าจะเป็นของแต่ละทางเลือก
4. ประเมินผลตอบแทนที่คาดหวังของแต่ละทางเลือก
5. วิเคราะห์ผลตอบแทนที่คาดว่าจะได้รับ ด้วยการคำนวณย้อนหลังไปจากทางขวาสุดของกิ่งก้านมาทางซ้ายสุด

13.1.2 ขั้นตอนการสร้างผังแบบกิ่งก้านสาขา

1. เขียนเริ่มต้นจากรากแตกกิ่งไปตามจำนวนเงื่อนไขที่เป็นไปตามกฎโดยพิจารณาเงื่อนไขที่เกิดขึ้นในการตัดสินใจเริ่มแรก
2. แตกกิ่งของเงื่อนไขแรกนั้นเป็นเงื่อนไขถัดไป
3. แตกกิ่งของเงื่อนไขต่อไป จนกระทั่งหมดเงื่อนไขที่เกิดขึ้น
4. เขียนกิ่งของกิจกรรมที่จะต้องกระทำเมื่อเป็นไปตามเงื่อนไขจากรากไปปลายกิ่งเงื่อนไขแต่ละกิ่ง

13.2 โครงสร้างต้นไม้การตัดสินใจ



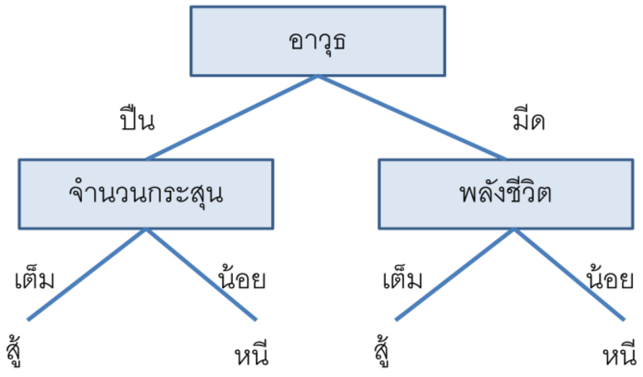
ภาพที่ 13-1 ลักษณะของต้นไม้การตัดสินใจ

ลักษณะของต้นไม้การตัดสินใจแสดงดังภาพที่ 13-1 ซึ่งมีลักษณะโครงสร้างเหมือนต้นไม้ทั่วไปทางด้านวิทยาการคอมพิวเตอร์คือ ประกอบไปด้วยส่วนสำคัญ 3 ส่วน

- ราก (Root) เป็นจุดเริ่มต้นของเหตุการณ์
- กิ่ง คือ เงื่อนไข (Condition) เป็นทางเลือกหรือเหตุการณ์ที่เกิดขึ้น
- ใบ คือ การกระทำ (Action) เป็นจุดสิ้นสุดหรือเป็นการกระทำที่เกิดขึ้นตามเงื่อนไขและค่าที่เป็นไปได้

ข้อสังเกตบางประการในการเขียนแขนงการตัดสินใจของต้นไม้การตัดสินใจคือ สร้างจากด้านซ้ายไปขวาหรือจากด้านบนลงล่าง, ทางเลือกของแต่ละโหนดต้องมากกว่า 1 ทาง และเหตุการณ์ต้องเกิดอย่างน้อย 1 เหตุการณ์ สำหรับการตัดสินใจเลือกวิธีการตัดสินใจบนต้นไม้การตัดสินใจ **คือ**การเกิดเงื่อนไขต่าง ๆ และการกระทำกิจกรรมเป็นไปตามลำดับก่อนหลัง และกรณีที่มีเงื่อนไขหลากหลายแบบ ในการแตกกิ่งที่แตกต่างกันไป โดยเงื่อนไขไม่จำกัด

ตัวอย่างต้นไม้การตัดสินใจแสดงดังภาพที่ 13-2 เป็นต้นไม้การตัดสินใจในการพิจารณาว่าผู้เล่นควรจะสู้ หรือ หนี จากการต่อสู้ โดยโหนดรากเป็นการพิจารณาจาก อาวุธ ซึ่งมี 2 ประเภทคือ ปืน หรือมีด ถ้าผู้เล่นมีอาวุธเป็นปืน (ตามกิ่งด้านซ้าย) ก็ให้พิจารณาจากจำนวนกระสุนที่มีถ้ามีกระสุนเต็มให้สู้ แต่ถ้าหากมีกระสุนน้อยให้หนี ในทางกลับกันถ้าผู้เล่นมีอาวุธเป็นมีด ให้พิจารณาจากพลังชีวิตที่เหลืออยู่ ถ้าพลังชีวิตเต็มก็ให้สู้ แต่ถ้าพลังชีวิตน้อยก็ให้หนี



ภาพที่ 13-2 ตัวอย่างต้นไม้การตัดสินใจในการพิจารณาการสู้หรือหนีในเกม

13.3 การสร้างต้นไม้การตัดสินใจ

ขั้นตอนที่สำคัญในเทคนิคการเรียนรู้แบบต้นไม้การตัดสินใจก็ คือการสร้างแบบจำลองของต้นไม้การตัดสินใจ โดยจะต้องเริ่มพิจารณาที่โหนดราก เป็นอันดับแรก ก่อนจะดำเนินการพิจารณาใบ และกิ่งก้าน การหาโหนดรากจะใช้วิธีการคำนวณหาค่า Information gain ที่มากที่สุด

13.3.1 ทฤษฎี Information Gain

จากผลการศึกษาการสร้างต้นไม้การตัดสินใจเพื่อใช้ในการจัดหมวดหมู่ (Classification) มีทฤษฎีที่นิยมใช้ในการสร้างต้นไม้การตัดสินใจคือ ทฤษฎี Information gain เป็นทฤษฎีที่นำมาใช้ในการเลือก attribute ในแต่ละโหนดของต้นไม้ ซึ่ง attribute ตัวใดที่มีค่า Information gain สูงสุด หรือว่ามีค่า Entropy น้อย ก็จะถูกเลือกเป็น attribute ของโหนดนั้น

ทฤษฎีนี้มีสมการการคำนวณ Information gain และค่า Entropy ดังนี้

สมการ Entropy

$$Entropy(s) \equiv \sum_{i=1}^{n} -P(V_i)\log_2 V_i$$

โดยที่

- Entropy(s) หรือ E(S) คือ ค่า Entropy ของเซตข้อมูล (S)
- S คือ เซตข้อมูลความน่าจะเป็น $P(V_1), P(V_2), \ldots, P(V_n)$
- $P(V_i)$ คือ ความน่าจะเป็นของข้อมูล I ที่สนใจ

ตัวอย่างการคำนวณหาค่า Entropy โดยกำหนดฟังก์ชันเป้าหมาย: จะเล่นกอล์ฟเมื่อ? จากตารางทางสถิติที่ผ่านมาแสดงดังภาพที่ 13-3 สรุปเป็น attribute ที่เกี่ยวข้องกับการเล่นกอล์ฟว่า การเล่นกอล์ฟ (PlayGolf) ขึ้นอยู่กับสภาพอากาศ (Outlook) และ กำลังลม (Wind) โดยสภาพอากาศมีความเป็นไปได้อยู่ 3 ค่า คือ ฝนตก (rain), แดดออก (sunny) และ มีเมฆมาก (overcast) สำหรับกำลังลม มีค่าอยู่ด้วยกัน 2 ค่าคือ ลมแรง (strong) และ ลมเบา (weak) ตัวอย่างในบรรทัดแรกสามารถตีความได้ว่า ถ้าฝนตก และลมแรง ก็ไม่เล่นกอล์ฟ

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no

ภาพที่ 13-3 ตารางทางสถิติในการเล่นกอล์ฟ

ในการคำนวณค่า Entropy ของข้อมูลทั้งหมดนั้น ให้พิจารณาจากฟังก์ชันเป้าหมายหรือผลลัพธ์ที่ต้องการ ซึ่งฟังก์ชันเป้าหมายของโจทย์นี้คือต้องการทราบถึงจะมีการเล่นกอล์ฟหรือไม่ ดังนั้นให้นับความน่าจะเป็นที่จะเกิดขึ้นจากข้อมูลทั้งหมดว่าโอกาสที่จะเล่นกอล์ฟมีค่าเท่าไร และความน่าจะเป็นที่จะไม่เล่นกอล์ฟมีค่าเท่าไร

จากตารางทางสถิติจะเห็นได้ว่า มีข้อความตัวอย่างทั้งหมด 7 ข้อมูล โดยผลลัพธ์เป็นการเล่นกอล์ฟ (PlayGolf = yes) มีอยู่ 4 ข้อมูล และผลลัพธ์เป็นไม่เล่นกอล์ฟ (PlayGolf = no) มีอยู่ 3 ข้อมูล ดังนั้นสามารถนำมาคำนวณความน่าจะเป็นได้ดังนี้

- P(เล่นกอล์ฟ) $= \frac{4}{7}$
- P(ไม่เล่นกอล์ฟ) $= \frac{3}{7}$

และคำนวณ Entropy ของโจทย์ได้เป็น

E(S) = - P(เล่นกอล์ฟ)$\log_2$P(เล่นกอล์ฟ) – P(ไม่เล่นกอล์ฟ)$\log_2$P(ไม่เล่นกอล์ฟ)

$$= -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.461 + 0.524 = 0.985$$

13.3.2 การค่า Information Gain

เมื่อมีการคำนวณหาค่า Entropy ของระบบแล้ว ต่อไปคือการพิจาณา attribute ว่า attribute ไหนที่มีผลกระทบต่อผลลัพธ์มากที่สุด วิธีที่นิยมใช้พิจารณาหา attribute ที่มีผลกระทบต่อผลลัพธ์มากที่สุดคือ การคำนวณหาค่า Information Gain ซึ่งสามารถคำนวณได้จากสมการต่อไปนี้

$$Gain(S,A) \equiv Entropy(s) - \sum_{v \varepsilon Values(A)} \left(\frac{s_v}{s} Entropy(s_v) \right)$$

โดยที่

- Gain(S,A) คือ ค่า Gain ของ Attribute ที่สนใจ
- $E(S_v)$ คือ ค่า Entropy ของเซ็ตข้อมูลที่สนใจ
- S_v คือ เซ็ตข้อมูลที่สนใจ

ในการคำนวณค่า Entropy นั้นให้พิจารณาข้อมูลในแต่ละ attribute (Outlook และ Wind) โดยการจำแนกผลลัพธ์กับค่าของ attribute

13.3.2.1 สภาพอากาศ (Outlook)

เมื่อพิจารณาจากข้อมูลในตารางพบว่า ค่าของสภาพอากาศมีด้วยกัน 3 ค่าคือ

- Rain(R) มีข้อมูลอยู่ 3 ข้อมูล (S_R = 3) จากข้อมูล 7 ข้อมูล (S = 7)
- Sunny(S) มีข้อมูลอยู่ 2 ข้อมูล (S_s = 2) จากข้อมูล 7 ข้อมูล (S = 7)
- Overcast(O) มีข้อมูลอยู่ 2 ข้อมูล (S_o = 2) จาก 7 ข้อมูล (S = 7)

เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_R), Entropy(S_S) และ Entropy(S_O) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_R) ให้พิจารณาจากข้อมูลที่มี attribute outlook เป็น rain พบว่ามีข้อมูลทั้งหมด 3 ข้อมูลโดยเป็นเล่นกอล์ฟ 1 ข้อมูล [P(เล่นกอล์ฟ) = 1/3] และไม่เล่นกอล์ฟ 2 ข้อมูล [P(ไม่เล่นกอล์ฟ) = 2/3]

ดังนั้น Entropy(S_R) $= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}$

= 0.528 + 0.39

= 0.918

สำหรับ Entropy(S_s) ให้พิจารณาจากข้อมูลที่มี attribute outlook เป็น sunny พบว่ามีข้อมูลทั้งหมด 2 ข้อมูลโดยเป็นเล่นกอล์ฟ 2 ข้อมูล [P(เล่นกอล์ฟ) = 2/2 = 1] และไม่เล่นกอล์ฟ 0 ข้อมูล [P(ไม่เล่นกอล์ฟ) = 0/2 = 0]

ดังนั้น Entropy(S_s) $= -\log_2 1 \; - \; 0 \log_2 0$

$= 0 + 0$

$= 0$

และสำหรับ Entropy(S_o) ให้พิจารณาจากข้อมูลที่มี attribute outlook เป็น overcast พบว่ามีข้อมูลทั้งหมด 2 ข้อมูลโดยเป็นเล่นกอล์ฟ 1 ข้อมูล [P(เล่นกอล์ฟ) = 1/2] และไม่เล่นกอล์ฟ 1 ข้อมูล [P(ไม่เล่นกอล์ฟ) = 1/2]

ดังนั้น Entropy(S_s) $= -\frac{1}{2}\log_2\frac{1}{2} \; - \frac{1}{2}\log_2\frac{1}{2}$

$= 0.5 + 0.5$

$= 1$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute outlook แล้ว สามารถนำมาคำนวณหาค่า Information Gain ของ Outlook ได้ดังนี้

$$\text{Gain(S, Outlook)} = E(S) - \left[\frac{S_R}{S} E(S_R) + \frac{S_S}{S} E(S_S) + \frac{S_O}{S} E(S_O) \right]$$

$$= 0.985 - \left[\frac{3}{7}(0.918) + \frac{2}{7}(0) + \frac{2}{7}(1) \right]$$

$$= 0.985 - [\, 0.393 + 0 + 0.286 \,]$$

$$= 0.985 - 0.679$$

$$= 0.306$$

13.3.2.2 กำลังลม (Wind)

เมื่อพิจารณาจากข้อมูลในตารางพบว่า ค่าของกำลังลมมีด้วยกัน 2 ค่าคือ

- Strong (S) มีข้อมูลอยู่ 4 ข้อมูล (S_S = 4) จากข้อมูล 7 ข้อมูล (S = 7)
- Weak (W) มีข้อมูลอยู่ 3 ข้อมูล (S_W = 3) จากข้อมูล 7 ข้อมูล (S = 7)

เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_S) และ Entropy(S_W) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_S) ให้พิจารณาจากข้อมูลที่มี attribute wind เป็น Strong พบว่ามีข้อมูลทั้งหมด 4 ข้อมูลโดยเป็นเล่นกอล์ฟ 1 ข้อมูล [P(เล่นกอล์ฟ) = 1/4] และไม่เล่นกอล์ฟ 3 ข้อมูล [P(ไม่เล่นกอล์ฟ) = 3/4]

ดังนั้น Entropy(S_S) $= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}$

$= 0.5 + 0.311$

$= 0.811$

และสำหรับ Entropy(S_W) ให้พิจารณาจากข้อมูลที่มี attribute wind เป็น Weak พบว่ามีข้อมูลทั้งหมด 3 ข้อมูลโดยเป็นเล่นกอล์ฟ 3 ข้อมูล [P(เล่นกอล์ฟ) = 3/3 = 1] และไม่เล่นกอล์ฟ 0 ข้อมูล [P(ไม่เล่นกอล์ฟ) = 0/3 = 0]

ดังนั้น Entropy(S_W) $= -\log_2 1 - 0\log_2 0$

$= 0 + 0$

$= 0$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute Wind แล้ว สามารถนำมาคำนวณหาค่า Information Gain ของ Outlook ได้ดังนี้

$$\text{Gain}(S, \text{Wind}) = E(S) - [\,\frac{S_S}{S}E(S_S) + \frac{S_W}{S}E(S_W)\,]$$

$$= 0.985 - [\,\frac{4}{7}(0.811) + \frac{3}{7}(0)\,]$$

$$= 0.985 - [\,0.463 + 0\,]$$

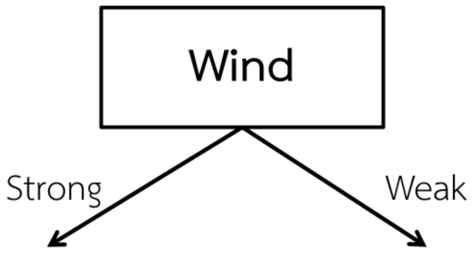
$$= 0.522$$

13.3.2.3 เปรียบเทียบค่า Gain และการเลือก Attribute

จากการคำนวณหาค่า Information Gain ของ attribute Outlook และ Wind ในหัวข้อที่ 13.3.2.1 และ 13.3.2.2 พบว่าค่า Gain ของแต่ละ attribute มีค่าดังนี้

- Gain(S, Outlook) = 0.306
- Gain(S, Wind) = 0.522

ในการเลือก attribute ที่มีผลกระทบกับฟังก์ชันเป้าหมายหรือผลลัพธ์จะให้พิจารณาจาก attribute ที่มีค่า Information Gain สูงสุด ซึ่งในที่นี้พบว่า attribute Wind มีค่า Information Gain สูงกว่า attribute Outlook ดังนั้นจึงเลือก attribute Wind ขึ้นมาเป็นโหนดราก และนำมาเขียนต้นไม้การตัดสินใจได้ดังภาพที่ 13-4



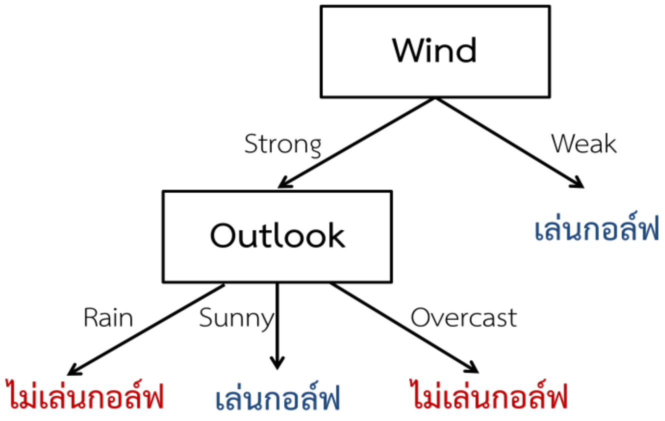
ภาพที่ 13-4 การกำหนดโหนดรากให้กับปัญหาตัวอย่าง

เมื่อสามารถหาโหนดรากให้กับต้นไม้การตัดสินใจได้แล้ว ให้พิจารณาต่อว่าการเลือกทางเดินนั้น พาไปสู่คำตอบที่ต้องการหรือยัง ถ้ายังจะต้องนำ attribute ที่เหลือมาช่วยประกอบ ในการตัดสินต่อไป ในที่นี้จะเห็นได้ว่าถ้ากำลังลมเบา (Wind เป็น Weak) จากข้อมูลที่มีสามารถตัดสินใจได้ทันทีว่าจะเล่นกอล์ฟ เนื่องจากข้อมูลที่เกี่ยวกับกำลังลมเบาทั้ง 3 ข้อมูล บ่งชี้ถึงการตัดสินใจจะเล่นกอล์ฟ แต่ในทางกลับกันถ้ากำลังลมแรง (Wind เป็น Strong) จะยังไม่สามารถทำการตัดสินใจว่าจะเล่นกอล์ฟหรือไม่ได้ เนื่องด้วยผลลัพธ์ยังแบ่งออกเป็น 2 คำตอบ โดยมีโอกาสที่จะเล่นกอล์ฟ 2 ข้อมูลจาก 4 ข้อมูล ดังนั้นจึงต้องนำ attribute ที่เหลือก็คือ Outlook มาช่วยในการตัดสินใจสำหรับกิ่งนี้ถัดไป

เมื่อพิจารณาในต้นไม้ระดับถัดไปให้ทำการตัดข้อมูลที่สามารถพิจารณาได้แล้วออกไป ทำให้ข้อมูลทั้งหมดที่จะพิจารณาต่อเหลือดังภาพที่ 13-5 จากโจทย์ปัญหานี้พบว่าโชคดีที่ข้อมูลที่ให้มาสามารถแยกนำไปเป็นคำตอบได้เลย โดยจะสังเกตเห็นว่า ถ้าสภาพอากาศมีฝนตกผลลัพธ์ที่ได้ไม่ขัดแย้งกันคือ จะไม่เล่นกอล์ฟ, ถ้าสภาพอากาศมีแสงแดดผลลัพธ์ก็คือ จะเล่นกอล์ฟ และสุดท้ายถ้าสภาพอากาศ มีเมฆมาก ก็จะไม่เล่นกอล์ฟ แต่สำหรับปัญหาบางประเภทที่เมื่อให้ attribute จนครบทุกตัวแล้ว ยังไม่สามารถนำพาไปสู่คำตอบที่ไม่ขัดแย้งกันได้ จำเป็นจะต้องเพิ่ม attribute หรือข้อมูลทางสถิติ ให้มากขึ้น หรือแม้กระทั่งบางครั้งอาจจำเป็นต้องลบข้อมูลส่วนน้อยที่ทำให้เกิดการขัดแย้งของผลลัพธ์ออกไป อย่างไรก็ตามจากโจทย์ที่กำหนดสามารถนำมาเขียนต้นไม้การตัดสินใจสุดท้ายได้ดังภาพที่ 13-6

Outlook	Wind	PlayGolf
rain	strong	no
sunny	strong	yes
rain	strong	no
overcast	strong	no

ภาพที่ 13-5 ข้อมูลในการตัดสินใจเมื่อตัดข้อมูลที่พิจารณาได้แล้วออกไป



ภาพที่ 13-6 ต้นไม้การตัดสินใจเกี่ยวกับการเล่นกอล์ฟ

13.4 ตัวอย่างการสร้างต้นไม้การตัดสินใจ

เพื่อให้เกิดความชำนาญในการคำนวณหาค่า Entropy และ Information Gain เพื่อใช้สำหรับการสร้างต้นไม้การตัดสินใจ จึงขอยกตัวอย่างในกรณีที่ปัญหามีความซับซ้อนมากขึ้น โดยกำหนดปัญหาเกี่ยวกับการนั่งรออาหารที่โต๊ะของลูกค้า ว่าอะไรบ้างเป็นสาเหตุให้ลูกค้าตัดสินใจที่จะรออาหาร หรือตัดสินใจที่จะไม่รออาหาร ซึ่งมีการสำรวจจากข้อมูลดิบแสดงในภาพที่ 13-7 โดยมีข้อมูลที่นำมาพิจารณาคือ สถานะของโต๊ะ, ราคา, ประเภทของร้านอาหาร, และระยะเวลาที่ต้องรอ

ลูกค้า	สถานะของโต๊ะ	ราคา	ร้านอาหาร	ระยะเวลารอ (นาที)	ลูกค้ารอ
A	เต็ม	แพง	ญี่ปุ่น	10-30	รอ
B	เต็ม	ถูก	ไทย	10-30	รอ
C	เต็ม	แพง	ไทย	10-30	ไม่รอ
D	เต็ม	แพง	ญี่ปุ่น	มากกว่า 30	ไม่รอ
E	เต็ม	แพง	อิตาลี	0-10	รอ
F	ว่าง	ถูก	ไทย	10-30	รอ
G	เต็ม	แพง	อิตาลี	10-30	ไม่รอ
H	ว่าง	แพง	ไทย	0-10	รอ

ภาพที่ 13-7 ข้อมูลดิบที่ใช้ในการตัดสินใจเกี่ยวกับการรออาหารของลูกค้าในร้านอาหาร

ขั้นตอนแรกคือการหาโหนดรากจาก attribute ทั้งหมด โดยเริ่มต้นจากการคำนวณค่า Entropy รวมของระบบก่อน ซึ่งพิจารณาจากผลลัพธ์เป็นหลัก (รอ หรือไม่รอ) พบว่าข้อมูลทั้งหมดมีด้วยกัน 8 ข้อมูล แบ่งออกเป็นลูกค้ารอ 5 ข้อมูล และลูกค้าไม่รอ 3 ข้อมูล ดังแสดงต่อไปนี้

Entropy(S) $= [-P(\text{รอ}) \times \log_2 P(\text{รอ})] + [-P(\text{ไม่รอ}) \times \log_2 P(\text{ไม่รอ})]$

$= -\frac{5}{8}\ \log 2\ \frac{5}{8}\ \ -\frac{3}{8}\ \log 2\ \frac{3}{8}$

$= 0.424 + 0.53 = 0.954$

เมื่อได้ค่า Entropy รวมของระบบแล้วให้บันทึกเก็บไว้ก่อน จากนั้นมาคำนวณหาค่า Information Gain ของแต่ละ attribute ที่มีคือ สถานะของโต๊ะ, ราคา, ประเภทของอาหาร และระยะเวลาที่ต้องรอ

- สถานะของโต๊ะ

ค่าที่เป็นไปได้ของสถานะของโต๊ะจากข้อมูลดิบคือ เต็ม (F) และว่าง(E) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับสถานะของโต๊ะเต็ม (S_F) และว่าง (S_E) คือ 6 และ 2 ตามลำดับจากข้อมูลทั้งหมด 8 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_F) และ Entropy(S_E) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_F) ให้พิจารณาจากข้อมูลที่มี attribute สถานะของโต๊ะเป็น “เต็ม” พบว่ามีข้อมูลทั้งหมด 6 ข้อมูล โดยผลลัพธ์เป็นรอ 3 ข้อมูล [P(รอ) = 3/6] และผลลัพธ์เป็นไม่รอจำนวน 3 ข้อมูล [P(ไม่รอ) = 3/6]

ดังนั้น Entropy(S_F) $= -\frac{3}{6}\log_2\frac{3}{6}\ \ -\frac{3}{6}\log_2\frac{3}{6}$

$= 0.5 + 0.5 = 1$

และสำหรับ Entropy(S_E) ให้พิจารณาจากข้อมูลที่มี attribute สถานะของโต๊ะเป็น “ว่าง” พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/2 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/2 = 0]

ดังนั้น Entropy(S_E) $= -\log_2 1\ \ -\ 0 \log_2 0$

$= 0 + 0 = 0$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute สถานะของโต๊ะแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, สถานะของโต๊ะ) $= E(S) - [\frac{S_F}{S} E(S_F) + \frac{S_E}{S} E(S_E)]$

$= 0.954 - [\frac{6}{8}(1) + \frac{2}{8}(0)] = 0.204$

- ราคา

ค่าที่เป็นไปได้ของราคาจากข้อมูลดิบคือ ถูก (C) และแพง(E) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับราคาถูก (S_C) และแพง (S_E) คือ 2 และ 6 ตามลำดับจากข้อมูลทั้งหมด 8 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_C) และ Entropy(S_E) ซึ่งสามารถคำนวณได้ด้วยการพิจารณาผลลัพธ์

สำหรับ Entropy(S_C) ให้พิจารณาจากข้อมูลที่มี attribute ราคาเป็น “ถูก” พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/2 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/2 = 0]

ดังนั้น Entropy(S_C) $= -\log_2 1 \;-\; 0\log_2 0$

$= 0 + 0$

$= 0$

และสำหรับ Entropy(S_E) ให้พิจารณาจากข้อมูลที่มี attribute ราคาเป็น “แพง” พบว่ามีข้อมูลทั้งหมด 6 ข้อมูล โดยผลลัพธ์เป็นรอ 3 ข้อมูล [P(รอ) = 3/6] และผลลัพธ์เป็นไม่รอจำนวน 3 ข้อมูล [P(ไม่รอ) = 3/6]

ดังนั้น Entropy(S_E) $= -\frac{3}{6}\log_2\frac{3}{6} \;-\; \frac{3}{6}\log_2\frac{3}{6}$

$= 0.5 + 0.5$

$= 1$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute ราคาแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, ราคา) $= E(S) - [\,\frac{S_C}{S}E(S_C) + \frac{S_E}{S}E(S_E)\,]$

$= 0.954 - [\,\frac{2}{8}(0) + \frac{6}{8}(1)\,]$

$= 0.954 - 0.75$

$= 0.204$

- ประเภทร้านอาหาร

ค่าที่เป็นไปได้ของประเภทร้านอาหารจากข้อมูลดิบคือ ญี่ปุ่น (J), ไทย(T) และอิตาลี (I) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับประเภทร้านอาหารญี่ปุ่น (S_J), ประเภทร้านอาหารไทย (S_T) และประเภทร้านอาหารอิตาลี (S_I) คือ 2, 4 และ 2 ตามลำดับจากข้อมูลทั้งหมด 8 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_J), Entropy(S_T) และ Entropy(S_I) ซึ่งคำนวณได้ด้วยการพิจารณาผลลัพธ์

สำหรับ Entropy(S_J) ให้พิจารณาจากข้อมูลที่มี attribute ประเภทร้านอาหารเป็น "ญี่ปุ่น" พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/2] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/2]

ดังนั้น Entropy(S_J) $= -\frac{1}{2}\log_2\frac{1}{2} \quad -\frac{1}{2}\log_2\frac{1}{2}$

$= 0.5 + 0.5 = 1$

สำหรับ Entropy(S_T) ให้พิจารณาจากข้อมูลที่มี attribute ประเภทร้านอาหารเป็น "ไทย" พบว่ามีข้อมูลทั้งหมด 4 ข้อมูล โดยผลลัพธ์เป็นรอ 3 ข้อมูล [P(รอ) = 3/4] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/4]

ดังนั้น Entropy(S_T) $= -\frac{3}{4}\log_2\frac{3}{4} \quad -\frac{1}{4}\log_2\frac{1}{4}$

$= 0.311 + 0.5 = 0.811$

และสำหรับ Entropy(S_I) ให้พิจารณาจากข้อมูลที่มี attribute ประเภทร้านอาหารเป็น "อิตาลี" พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/2] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/2]

ดังนั้น Entropy(S_I) $= -\frac{1}{2}\log_2\frac{1}{2} \quad -\frac{1}{2}\log_2\frac{1}{2}$

$= 0.5 + 0.5 = 1$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute ประเภทของร้านอาหารแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, ประเภทร้านอาหาร) $= E(S) - [\frac{S_J}{S}E(S_J) + \frac{S_T}{S}E(S_T) + \frac{S_I}{S}E(S_I)]$

$= 0.954 - [\frac{2}{8}(1) + \frac{4}{8}(0.811) + \frac{2}{8}(1)]$

$= 0.954 - [0.25 + 0.4055 + 0.25]$

$= 0.0485$

- ระยะเวลาในการรอ

ค่าที่เป็นไปได้ของระยะเวลาในการรอจากข้อมูลดิบคือ 0-10 นาที (1), 10-30 นาที (2) และมากกว่า 30 นาที (3) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับระยะเวลาในการรอในช่วง 0-10 นาที (S_1), 10-30 นาที (S_2) และมากกว่า 30 นาที (S_3) คือ 2, 5 และ 1 ตามลำดับจากข้อมูลทั้งหมด 8 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_1), Entropy(S_2) และ Entropy(S_3) ซึ่งคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_1) ให้พิจารณาจากข้อมูลที่มี attribute ระยะเวลาในการรอมีค่าเป็น "0-10" พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/2 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/2 = 0]

ดังนั้น Entropy(S_1) $= -\log_2 1 \quad - \quad 0 \log_2 0$

$= 0 + 0 = 0$

สำหรับ Entropy(S_2) ให้พิจารณาจากข้อมูลที่มี attribute ระยะเวลาในการรอมีค่าเป็น "10-30" พบว่ามีข้อมูลทั้งหมด 5 ข้อมูล โดยผลลัพธ์เป็นรอ 3 ข้อมูล [P(รอ) = 3/5] และผลลัพธ์เป็นไม่รอจำนวน 2 ข้อมูล [P(ไม่รอ) = 2/5]

ดังนั้น Entropy(S_2) $= -\frac{3}{5}\log_2\frac{3}{5} \quad -\frac{2}{5}\log_2\frac{2}{5}$

$= 0.442 + 0.529 = 0.971$

และสำหรับ Entropy(S_3) ให้พิจารณาจากข้อมูลที่มี attribute ระยะเวลาในการรอมีค่าเป็น "มากกว่า 30" พบว่ามีข้อมูลทั้งหมด 1 ข้อมูล โดยผลลัพธ์เป็นรอ 0 ข้อมูล [P(รอ) = 0/1 = 0] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/1 = 1]

ดังนั้น Entropy(S_3) $= -\log_2 1 \quad - \quad 0 \log_2 0$

$= 0 + 0 = 0$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute ระยะเวลาในการรอแล้ว สามารถ นำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, ระยะเวลาในการรอ) $= E(S) - [\frac{S_1}{S}E(S_1) + \frac{S_2}{S}E(S_2) + \frac{S_3}{S}E(S_3)]$

$= 0.954 - [\frac{2}{8}(0) + \frac{5}{8}(0.971) + \frac{1}{8}(0)]$

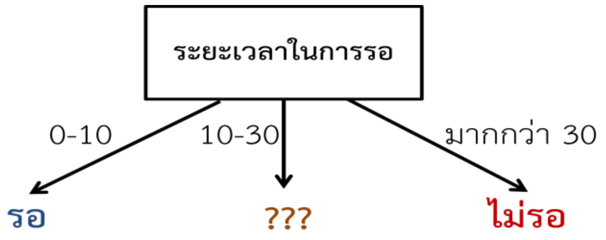
$= 0.954 - 0.607$

$= 0.347$

เมื่อพิจารณาจาก Information Gain ของแต่ละ attribute ที่คำนวณได้ สรุปได้ดังนี้

- Gain(S, สถานะของโต๊ะ) = 0.204
- Gain(S, ราคา) = 0.204
- Gain(S, ประเภทของร้านอาหาร) = 0.0485
- Gain(S, ระยะเวลาที่รอ) = 0.347

ดังนั้นจะเห็นได้ว่า attribute ระยะเวลาที่รอ มีผลกระทบกับการตัดสินใจของลูกค้าว่าจะรอหรือไม่รออาหารมากที่สุด ดังนั้นจึงเลือก attribute ระยะเวลาที่รอ มาเป็นรากของต้นไม้การตัดสินใจ และเมื่อพิจารณาถึงค่าของ attribute ทั้ง 3 ค่าพบว่า ถ้าระยะเวลาในการรออยู่ในช่วง 0-10 นาที ลูกค้าจะตัดสินใจรอ และถ้าระยะเวลาในการรออยู่ในช่วงมากกว่า 30 นาที ลูกค้าจะตัดสินใจไม่รอ แต่สำหรับระยะเวลาในการรออยู่ในช่วง 10-30 นาที ลูกค้ายังไม่สามารถตัดสินใจได้ จึงต้องพิจารณา attribute อื่นเพื่อประกอบการตัดสินใจของลูกค้า โดยต้นไม้การตัดสินใจเมื่อพิจารณาโหนดรากได้แล้ว แสดงดังภาพที่ 13-8



ภาพที่ 13-8 ต้นไม้การตัดสินใจเกี่ยวกับการรออาหารของลูกค้าเมื่อพิจารณาโหนดรากได้แล้ว

ขั้นตอนต่อไปคือการพิจารณาโหนดในการตัดสินสำหรับกรณีที่ลูกค้าต้องรออาหารในช่วงเวลา 10-30 นาที ดังนั้นจึงต้องตัดข้อมูลอื่นที่สามารถพิจาณาได้แล้วออกไปทำให้เหลือข้อมูลดังแสดงในภาพที่ 13-9

ลูกค้า	สถานะของโต๊ะ	ราคา	ร้านอาหาร	ลูกค้ารอ
A	เต็ม	แพง	ญี่ปุ่น	รอ
B	เต็ม	ถูก	ไทย	รอ
C	เต็ม	แพง	ไทย	ไม่รอ
F	ว่าง	ถูก	ไทย	รอ
G	เต็ม	แพง	อิตาลี	ไม่รอ

ภาพที่ 13-9 ข้อมูลดิบที่เหลือหลังจากพิจารณาโหนดรากได้แล้ว

จากข้อมูลดิบที่เหลือให้นำมาคำนวณ Entropy ของระบบใหม่อีกครั้ง ซึ่งพิจารณาจากผลลัพธ์เป็นหลัก (รอ หรือไม่รอ) พบว่าข้อมูลทั้งหมดมีด้วยกัน 5 ข้อมูลแบ่งออกเป็นลูกค้ารอ 3 ข้อมูล และลูกค้าไม่รอ 2 ข้อมูล ดังแสดงต่อไปนี้

$$\text{Entropy(S)} = [-P(\text{รอ}) \times \log_2 P(\text{รอ})] + [-P(\text{ไม่รอ}) \times \log_2 P(\text{ไม่รอ})]$$

$$= -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$= 0.442 + 0.529 = 0.971$$

เมื่อได้ค่า Entropy รวมของระบบใหม่แล้วให้บันทึกเก็บไว้ก่อน จากนั้นมาคำนวณหาค่า Information Gain ของแต่ละ attribute ที่เหลือ ซึ่งก็คือ สถานะของโต๊ะ, ราคา และประเภทของอาหาร

- สถานะของโต๊ะ

ค่าที่เป็นไปได้ของสถานะของโต๊ะจากข้อมูลดิบคือ เต็ม (F) และว่าง(E) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับสถานะของโต๊ะเต็ม (S_F) และว่าง (S_E) คือ 4 และ 1 ตามลำดับจากข้อมูลทั้งหมด 5 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_F) และ Entropy(S_E) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_F) ให้พิจารณาจากข้อมูลที่มี attribute สถานะของโต๊ะเป็น “เต็ม” พบว่ามีข้อมูลทั้งหมด 4 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/4] และผลลัพธ์เป็นไม่รอจำนวน 2 ข้อมูล [P(ไม่รอ) = 2/4]

$$\text{ดังนั้น Entropy}(S_F) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$$

$$= 0.5 + 0.5 = 1$$

และสำหรับ Entropy(S_E) ให้พิจารณาจากข้อมูลที่มี attribute สถานะของโต๊ะเป็น “ว่าง” พบว่ามีข้อมูลทั้งหมด 1 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/1 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/1 = 0]

$$\text{ดังนั้น Entropy}(S_E) = -\log_2 1 - 0\log_2 0$$

$$= 0 + 0 = 0$$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute สถานะของโต๊ะแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

$$\text{Gain(S, สถานะของโต๊ะ)} = E(S) - [\frac{S_F}{S}E(S_F) + \frac{S_E}{S}E(S_E)]$$

$$= 0.971 - [\frac{4}{5}(1) + \frac{1}{5}(0)] = 0.171$$

- ราคา

ค่าที่เป็นไปได้ของราคาจากข้อมูลดิบคือ ถูก (C) และแพง(E) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับราคาถูก (S_C) และแพง (S_E) คือ 2 และ 3 ตามลำดับจากข้อมูลทั้งหมด 5 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_C) และ Entropy(S_E) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_C) ให้พิจารณาจากข้อมูลที่มี attribute ราคาเป็น “ถูก” พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/2 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/2 = 0]

ดังนั้น Entropy(S_C) $= -\log_2 1 \quad - \quad 0 \log_2 0$

$= 0 + 0$

$= 0$

และสำหรับ Entropy(S_E) ให้พิจารณาจากข้อมูลที่มี attribute ราคาเป็น “แพง” พบว่ามีข้อมูลทั้งหมด 3 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/3] และผลลัพธ์เป็นไม่รอจำนวน 2 ข้อมูล [P(ไม่รอ) = 2/3]

ดังนั้น Entropy(S_E) $= -\frac{1}{3}\log_2\frac{1}{3} \quad -\frac{2}{3}\log_2\frac{2}{3}$

$= 0.528 + 0.39$

$= 0.918$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute ราคาแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, ราคา) $= E(S) - [\frac{S_C}{S} E(S_C) + \frac{S_E}{S} E(S_E)]$

$= 0.971 - [\frac{2}{5}(0) + \frac{3}{5}(0.918)]$

$= 0.971 - 0.551$

$= 0.42$

- ประเภทร้านอาหาร

ค่าที่เป็นไปได้ของประเภทร้านอาหารจากข้อมูลดิบคือ ญี่ปุ่น (J), ไทย(T) และอิตาลี (I) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับประเภทร้านอาหารญี่ปุ่น (S_J), ประเภทร้านอาหารไทย (S_T) และประเภทร้านอาหารอิตาลี (S_I) คือ 1, 3 และ 1 ตามลำดับจากข้อมูลทั้งหมด 5 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_J), Entropy(S_T) และ Entropy(S_I) ซึ่งคำนวณได้ด้วยการพิจารณาผลลัพธ์

สำหรับ Entropy(S_J) ให้พิจารณาจากข้อมูลที่มี attribute ประเภทร้านอาหารเป็น “ญี่ปุ่น” พบว่ามีข้อมูลทั้งหมด 1 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/1 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/1 = 0]

ดังนั้น Entropy(S_J) $= -\log_2 1 \quad - \quad 0 \log_2 0$

$= 0 + 0 = 0$

สำหรับ Entropy(S_T) ให้พิจารณาจากข้อมูลที่มี attribute ประเภทร้านอาหารเป็น “ไทย” พบว่ามีข้อมูลทั้งหมด 3 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/3] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/3]

ดังนั้น Entropy(S_T) $= -\frac{2}{3}\log_2\frac{2}{3} \quad -\frac{1}{3}\log_2\frac{1}{3}$

$= 0.39 + 0.528 = 0.918$

และสำหรับ Entropy(S_I) ให้พิจารณาจากข้อมูลที่มี attribute ประเภทร้านอาหารเป็น “อิตาลี” พบว่ามีข้อมูลทั้งหมด 1 ข้อมูล โดยผลลัพธ์เป็นรอ 0 ข้อมูล [P(รอ) = 0/1 = 0] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/1 = 1]

ดังนั้น Entropy(S_I) $= - \quad 0 \log_2 0 \quad -\log_2 1$

$= 0 + 0 = 0$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute ประเภทของร้านอาหารแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, ประเภทร้านอาหาร) $= E(S) - [\frac{S_J}{S}E(S_J) + \frac{S_T}{S}E(S_T) + \frac{S_I}{S}E(S_I)]$

$= 0.971 - [\frac{1}{5}(0) + \frac{3}{5}(0.918) + \frac{1}{5}(0)]$

$= 0.971 - 0.551$

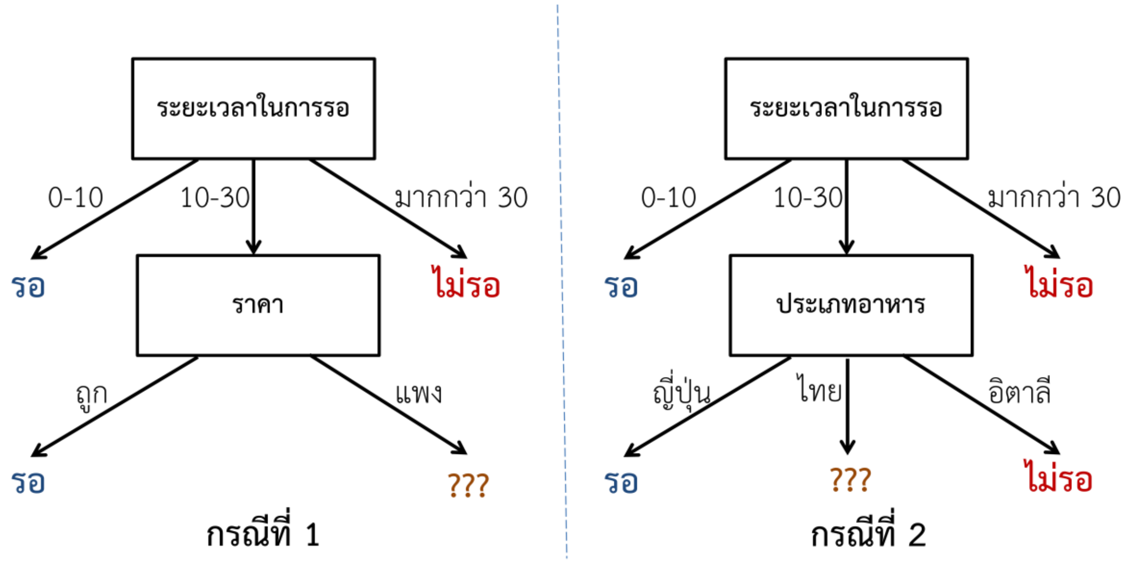
$= 0.42$

เมื่อพิจารณาจาก Information Gain ของแต่ละ attribute ที่คำนวณได้ สรุปได้ดังนี้

- Gain(S, สถานะของโต๊ะ) = 0.171
- Gain(S, ราคา) = 0.42
- Gain(S, ประเภทของร้านอาหาร) = 0.42

จากค่า Gain ที่คำนวณได้จะเห็นว่า attribute ราคา และ ประเภทของร้านอาหาร มีผลกระทบกับการตัดสินใจของลูกค้าว่าจะรอหรือไม่รออาหารมากที่สุดเท่านั้นทั้งสอง attribute ดังนั้นการเลือกโหนดเพื่อพิจารณาจึงสามารถแบ่งออกได้เป็น 2 กรณี ดังแสดงในภาพที่ 13-10

- **กรณีที่** 1 : ถ้าเลือก attribute ราคา สำหรับการตัดสินใจจะพบว่า ถ้าราคาของอาหารถูกลูกค้าจะตัดสินใจรอ แต่ถ้าราคาอาหารแพงจะยังไม่สามารถตัดสินใจได้ว่าลูกค้าจะรอหรือไม่
- **กรณีที่ 2 :** ถ้าเลือก attribute ประเภทของร้านอาหาร สำหรับการตัดสินใจจะพบว่า ถ้าประเภทของอาหารเป็นอาหารญี่ปุ่นลูกค้าจะตัดสินใจรอ และถ้าอาหารเป็นอาหารอิตาลีลูกค้าจะตัดสินใจไม่รอ แต่สำหรับอาหารไทยจะยังไม่สามารถตัดสินใจได้ว่าลูกค้าจะรอหรือไม่



ภาพที่ 13-10 ต้นไม้การตัดสินใจหลังจากการเลือกโหนดในการตัดสินใจระดับที่ 2

13.4.1 การสร้างต้นไม้การตัดสินใจต่อในกรณีที่ 1

ในกรณีนี้คือหลังจากที่เลือก attribute ราคามาเพื่อประกอบการตัดสินใจ ขั้นตอนต่อไปคือการพิจารณาโหนดในการตัดสินสำหรับกรณีที่อาหารมีราคาแพง ดังนั้นจึงต้องตัดข้อมูลอื่นที่สามารถพิจาณาได้แล้วออกไป ทำให้เหลือข้อมูลดังแสดงในภาพที่ 13-11

ลูกค้า	สถานะของโต๊ะ	ร้านอาหาร	ลูกค้ารอ
A	เต็ม	ญี่ปุ่น	รอ
C	เต็ม	ไทย	ไม่รอ
G	เต็ม	อิตาลี	ไม่รอ

ภาพที่ 13-11 ข้อมูลดิบกรณีที่ 1 ที่เหลือหลังจากพิจารณาโหนดระดับที่ 2 ได้แล้ว

จากข้อมูลดิบที่เหลือให้นำมาคำนวณ Entropy ของระบบใหม่อีกครั้ง ซึ่งพิจารณาจากผลลัพธ์เป็นหลัก (รอ หรือไม่รอ) พบว่าข้อมูลทั้งหมดมีด้วยกัน 3 ข้อมูลแบ่งออกเป็นลูกค้ารอ 1 ข้อมูล และลูกค้าไม่รอ 2 ข้อมูล ดังแสดงต่อไปนี้

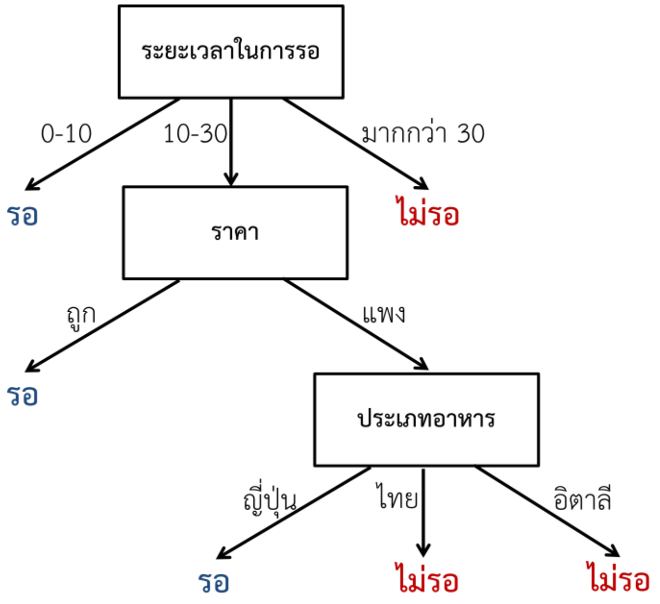
$$\text{Entropy}(S) = [-P(\text{รอ}) \times \log_2 P(\text{รอ})] + [-P(\text{ไม่รอ}) \times \log_2 P(\text{ไม่รอ})]$$

$$= -\frac{1}{3}\log_2\frac{1}{3} \quad -\frac{2}{3}\log_2\frac{2}{3}$$

$$= 0.528 + 0.39$$

$$= 0.918$$

เมื่อได้ค่า Entropy รวมของระบบใหม่แล้วให้บันทึกเก็บไว้ก่อน จากนั้นมาคำนวณหาค่า Information Gain ของแต่ละ attribute ที่เหลือ ซึ่งก็คือ สถานะของโต๊ะ และประเภทของอาหาร แต่อย่างไรก็ตาม attribute สถานะของโต๊ะ ไม่สามารถนำมาคำนวณหาค่า Gain ได้เนื่องจากมีทางเลือกเดียวคือ “เต็ม” ดังนั้นจึงต้องพิจาณาจาก attribute ประเภทของอาหารพบว่ายังมี 3 ทางเลือกและเป็น attribute ตัวสุดท้ายที่เหลืออยู่ ดังนั้นจึงเลือก attribute ประเภทของอาหารมาใช้เป็นตัวตัดสินใจ และเพิ่มเข้าไปยังต้นไม้การตัดสินใจ โดยมีผลลัพธ์ในการตัดสินใจคือ ถ้าประเภทของอาหารเป็นอาหารญี่ปุ่น ลูกค้าจะตัดสินใจที่จะรอ แต่ถ้าประเภทของอาหารเป็นอาหารไทยหรืออาหารอิตาลี ลูกค้าจะตัดสินใจที่จะไม่รอ โดยต้นไม้การตัดสินใจที่สมบูรณ์ถ้าเลือกทำแบบกรณีที่ 1 แสดงดังภาพที่ 13-12



ภาพที่ 13-12 ต้นไม้การตัดสินใจที่สมบูรณ์ของกรณีที่ 1

13.4.2 การสร้างต้นไม้การตัดสินใจต่อในกรณีที่ 2

ในกรณีนี้คือหลังจากที่เลือก attribute ประเภทอาหารมาเพื่อประกอบการตัดสินใจขั้นตอนต่อไปคือการพิจารณาโหนดในการตัดสินสำหรับกรณีที่อาหารมีประเภทเป็นอาหารไทย ดังนั้นจึงต้องตัดข้อมูลอื่นที่สามารถพิจาณาได้แล้วออกไป ทำให้เหลือข้อมูลดังแสดงในภาพที่ 13-13

ลูกค้า	สถานะของโต๊ะ	ราคา	ลูกค้ารอ
B	เต็ม	ถูก	รอ
C	เต็ม	แพง	ไม่รอ
F	ว่าง	ถูก	รอ

ภาพที่ 13-13 ข้อมูลดิบกรณีที่ 2 ที่เหลือหลังจากพิจารณาโหนดระดับที่ 2 ได้แล้ว

จากข้อมูลดิบที่เหลือให้นำมาคำนวณ Entropy ของระบบใหม่อีกครั้ง ซึ่งพิจารณาจากผลลัพธ์เป็นหลัก (รอ หรือไม่รอ) พบว่าข้อมูลทั้งหมดมีด้วยกัน 3 ข้อมูลแบ่งออกเป็นลูกค้ารอ 2 ข้อมูล และลูกค้าไม่รอ 1 ข้อมูล ดังแสดงต่อไปนี้

$$\text{Entropy(S)} = [-P(\text{รอ}) \times \log_2 P(\text{รอ})] + [-P(\text{ไม่รอ}) \times \log_2 P(\text{ไม่รอ})]$$

$$= -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$$

$$= 0.528 + 0.39 = 0.918$$

เมื่อได้ค่า Entropy รวมของระบบใหม่แล้วให้บันทึกเก็บไว้ก่อน จากนั้นมาคำนวณหาค่า Information Gain ของแต่ละ attribute ที่เหลือ ซึ่งก็คือ สถานะของโต๊ะ และราคา

- สถานะของโต๊ะ

ค่าที่เป็นไปได้ของสถานะของโต๊ะจากข้อมูลดิบคือ เต็ม (F) และว่าง(E) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับสถานะของโต๊ะเต็ม (S_F) และว่าง (S_E) คือ 2 และ 1 ตามลำดับจากข้อมูลทั้งหมด 3 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_F) และ Entropy(S_E) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_F) ให้พิจารณาจากข้อมูลที่มี attribute สถานะของโต๊ะเป็น “เต็ม” พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/2] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/2]

$$\text{ดังนั้น Entropy}(S_F) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$$

$$= 0.5 + 0.5$$

$$= 1$$

และสำหรับ Entropy(S_E) ให้พิจารณาจากข้อมูลที่มี attribute สถานะของโต๊ะเป็น “ว่าง” พบว่ามีข้อมูลทั้งหมด 1 ข้อมูล โดยผลลัพธ์เป็นรอ 1 ข้อมูล [P(รอ) = 1/1 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/1 = 0]

$$\text{ดังนั้น Entropy}(S_E) = -\log_2 1 - 0\log_2 0$$

$$= 0 + 0$$

$$= 0$$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute สถานะของโต๊ะแล้ว สามารถนำมา คำนวณหาค่า Information Gain ได้ดังนี้

$$\text{Gain(S, สถานะของโต๊ะ)} = E(S) - [\frac{S_F}{S}E(S_F) + \frac{S_E}{S}E(S_E)]$$

$$= 0.918 - [\frac{2}{3}(1) + \frac{1}{3}(0)]$$

$$= 0.918 - 0.667$$

$$= 0.251$$

- ราคา

ค่าที่เป็นไปได้ของราคาจากข้อมูลดิบคือ ถูก (C) และแพง (E) โดยพบว่าจากข้อมูลดิบมีข้อมูลเกี่ยวกับราคาถูก (S_C) และแพง (S_E) คือ 2 และ 1 ตามลำดับจากข้อมูลทั้งหมด 3 ข้อมูล เพื่อคำนวณค่า Gain จำเป็นต้องทราบค่า Entropy(S_C) และ Entropy(S_E) ซึ่งสามารถคำนวณได้ด้วยการพิจาณาผลลัพธ์

สำหรับ Entropy(S_C) ให้พิจารณาจากข้อมูลที่มี attribute ราคาเป็น “ถูก” พบว่ามีข้อมูลทั้งหมด 2 ข้อมูล โดยผลลัพธ์เป็นรอ 2 ข้อมูล [P(รอ) = 2/2 = 1] และผลลัพธ์เป็นไม่รอจำนวน 0 ข้อมูล [P(ไม่รอ) = 0/2 = 0]

ดังนั้น Entropy(S_C) $= -\log_2 1 - 0\log_2 0$

$= 0 + 0$

$= 0$

และสำหรับ Entropy(S_E) ให้พิจารณาจากข้อมูลที่มี attribute ราคาเป็น “แพง” พบว่ามีข้อมูลทั้งหมด 1 ข้อมูล โดยผลลัพธ์เป็นรอ 0 ข้อมูล [P(รอ) = 0/1 = 0] และผลลัพธ์เป็นไม่รอจำนวน 1 ข้อมูล [P(ไม่รอ) = 1/1 = 1]

ดังนั้น Entropy(S_E) $= -0\log_2 0 - \log_2 1$

$= 0 + 0$

$= 0$

เมื่อได้ค่า Entropy ของแต่ละค่าของ attribute ราคาแล้ว สามารถนำมาคำนวณหาค่า Information Gain ได้ดังนี้

Gain(S, ราคา) $= E(S) - [\frac{S_C}{S} E(S_C) + \frac{S_E}{S} E(S_E)]$

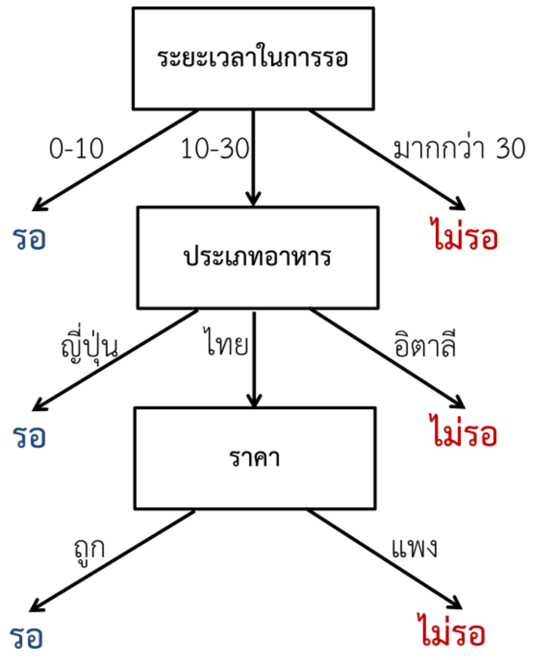
$= 0.918 - [\frac{2}{3}(0) + \frac{1}{3}(0)]$

$= 0.971$

เมื่อพิจารณาจาก Information Gain ของแต่ละ attribute ที่คำนวณได้ สรุปได้ดังนี้

- Gain(S, สถานะของโต๊ะ) = 0.251
- Gain(S, ราคา) = 0.971

จากค่า Gain ที่คำนวณได้จะเห็นว่า attribute ราคา มีผลกระทบกับการตัดสินใจของลูกค้าว่าจะรอหรือไม่รออาหารมากที่สุด เมื่อนำมาเขียนต้นไม้การตัดสินใจจะแสดงได้ดังภาพที่ 13-14



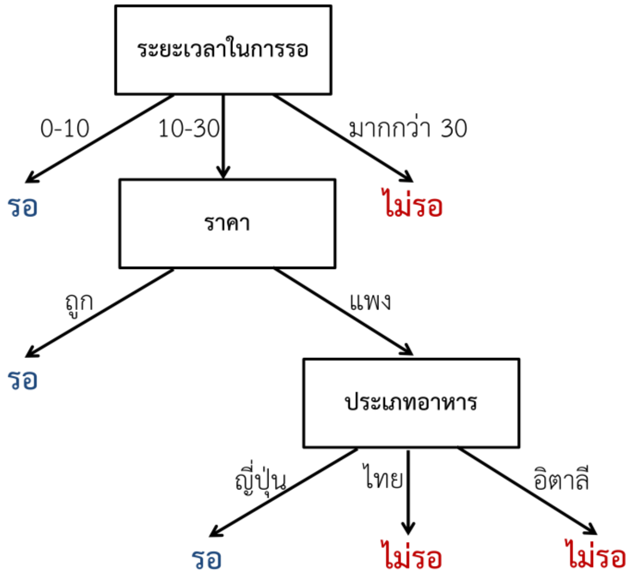
ภาพที่ 13-14 ต้นไม้การตัดสินใจที่สมบูรณ์ของกรณีที่ 2

ภาพที่ 13-15 แสดงต้นไม้การตัดสินใจของทั้ง 2 กรณีโดยสามารถสรุปได้ดังนี้ ทั้ง 2 กรณีมีเงื่อนไขในการดูการตัดสินใจของลูกค้าที่จะรออาหารเริ่มต้นเหมือนกัน คือ ดูจากระยะเวลาในการรอ ถ้าลูกค้าต้องรออาหารในช่วงเวลา 0-10 นาที ลูกค้าจะเลือกที่จะรอ แต่ถ้าลูกค้าต้องรอมากกว่า 30 นาที ไม่ว่าจะราคาอาหารถูกหรือแพง ประเภทอาหารเป็นไทย, ญี่ปุ่น หรือ อิตาลี และไม่ว่าสถานะของโต๊ะจะเต็มหรือว่างไม่มีผลต่อการตัดสินใจในเรื่องนี้ แต่อย่างไรก็ตามถ้าระยะเวลาในการรออยู่ในช่วง 10-30 นาที จะต้องคำนึงถึงเงื่อนไขถัดไป

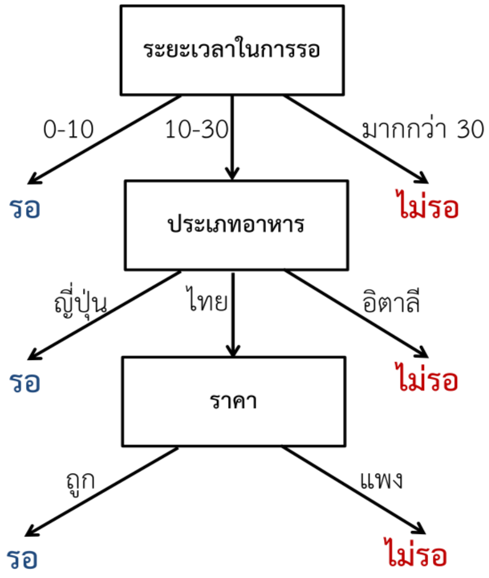
สำหรับกรณีที่ 1 ถ้าลูกค้าต้องรออาหารอยู่ในช่วง 10-30 นาที เงื่อนไขที่จะพิจารณาในการรอต่อไปของลูกค้าก็คือราคาของอาหาร ถ้าอาหารราคาถูกลูกค้าจะรอ แต่ถ้าราคามีราคาแพง ลูกค้าจะพิจารณาจากเงื่อนไขต่อไปคือประเภทของอาหาร ซึ่งถ้าเป็นอาหารญี่ปุ่นลูกค้าจะรอ แต่ถ้าเป็นอาหารประเภทอื่นลูกค้าจะตัดสินใจไม่รออาหาร

สำหรับกรณีที่ 2 ถ้าลูกค้าต้องรออาหารอยู่ในช่วง 10-30 นาที เงื่อนไขที่จะพิจารณาในการรอต่อไปของลูกค้าก็คือประเภทของอาหาร ถ้าเป็นอาหารญี่ปุ่นลูกค้าจะรอ แต่ถ้าเป็นอาหารอิตาลีลูกค้าจะไม่รอ สำหรับอาหารไทย ลูกค้าจะพิจารณาจากเงื่อนไขเพิ่มเติม คือ ราคา ถ้าอาหารมีราคาถูกลูกค้าจะเลือกที่จะรอ แต่ถ้าอาหารมีราคาแพงลูกค้าจะเลือกที่จะไม่รอ

ถ้าสังเกตให้ดีทั้ง 2 กรณีจะไม่มีการใช้ attribute สถานะของโต๊ะ ซึ่งสามารถสรุปได้ว่า สถานะของโต๊ะไม่มีผลกระทบมายังผลลัพธ์ที่ลูกค้าจะรอหรือไม่รออาหารแต่อย่างใด



กรณีที่ 1



กรณีที่ 2

ภาพที่ 13-15 ต้นไม้การตัดสินใจของการรออาหารของลูกค้าทั้ง 2 กรณี

13.5 ข้อด้อยของต้นไม้การตัดสินใจ

อย่างไรก็ตามการใช้งานของต้นไม้การตัดสินใจก็มีข้อจำกัดบางประการดังต่อไปนี้

- การแบ่งกลุ่มกิ่งก้านสาขา กรณีเป็นข้อมูลที่มีค่าต่อเนื่อง เช่น ข้อมูลรายได้ ข้อมูลราคา ต้องทำการแปลงให้อยู่ในช่วงหรือตัดเป็นกลุ่มก่อน
- เมื่อ ขั้นตอนวิธี (Algorithm) เลือกว่าจะใช้ค่าไหนเป็นตัวแบ่งกลุ่มแล้วก็จะไม่สนใจค่าอื่นที่อาจมีความสำคัญเช่นเดียวกัน
- การจัดการกับข้อมูลที่ไม่ทราบค่า อาจมีผลกระทบกับผลลัพธ์ของการตัดสินใจ
- การตัดสินใจแบบต้นไม่การตัดสินใจ ที่มีระดับชั้นมากเกินไป จะทำให้ข้อมูลที่ผ่านเงื่อนไข แตกออกเป็นชิ้นเล็กชิ้นน้อย ซึ่งข้อมูลเหล่านั้นจะไม่มีประโยชน์ในการนำมาใช้ทำการวิเคราะห์
- ปัญหาเรื่องที่เกิดจากการที่แบบจำลองได้เรียนรู้เข้าไปถึงรายละเอียดของข้อมูลมากเกินไปจะทำให้เกิดเงื่อนไขที่เป็นส่วนเฉพาะเจาะจงกับกลุ่มข้อมูลที่ใช้ในการเรียนรู้ ซึ่งจะต้องหาวิธี การในการตัดกิ่งนี้ออกไป

13.6 สรุป

ต้นไม้การตัดสินใจเป็นเทคนิคหนึ่งของการเรียนรู้ของเครื่องจักร ที่ใช้ในการพัฒนาการเรียนรู้ของเครื่องจักรหรือเครื่องคอมพิวเตอร์ ซึ่งเป็นแบบจำลองที่ใช้สำหรับคาดคะเนหรือทำนายเหตุการณ์ที่จะเกิดขึ้น บางครั้งอาจเรียกว่า ต้นไม้สำหรับจำแนกหมวดหมู่ (Classification Tree) ซึ่งประกอบด้วย ใบที่เป็นส่วนของข้อมูลที่สนใจ แต่ละใบถูกเชื่อมด้วยกิ่งก้านซึ่งเป็นข้อมูลที่ใช้เชื่อมระหว่างใบด้วยกัน กำเนิดจากส่วนบนสุดคือ ราก สำหรับแบบจำลองต้นไม้การตัดสินใจ จะเกี่ยวข้องกับกลุ่มของข้อมูล โดยมีกระบวนการสร้างที่สามารถคำนวณแบบย้อนกลับหรือทำซ้ำขั้นตอนเดิมได้ การสร้างจะพิจารณาที่ โหนดเริ่มต้นหรือราก เป็นอันดับแรก และกิ่งก้านที่แตกแขนงต่อไป ซึ่งอาศัยการคำนวณหาค่า Information Gain ที่ประกอบด้วยค่าของข้อมูลอีกชนิดหนึ่งคือ Entropy เพื่อใช้เปรียบเทียบ และคัดเลือกชุดข้อมูลที่เหมาะสมที่สุด

แบบฝึกหัด

1. จงอธิบายขั้นตอนการสร้างต้นไม้การตัดสินใจพอสังเขป

2. พิจารณาจากข้อมูลดิบของซูปเปอร์มาร์เก็ตแห่งหนึ่งต้องการสำรวจความต้องการซื้อสินค้าชนิดหนึ่ง ซึ่งมีทั้งหมด 3 ยี่ห้อคือ A, B และ C โดยพิจารณาข้อมูลที่สำคัญ 3 อย่างคือ ยี่ห้อ, ของแถม, และราคา จงเขียน Decision tree โดยให้ความสนใจกับการตัดสินใจที่จะซื้อสินค้า

ลูกค้า	ยี่ห้อ	ของแถม	ราคา	ความต้องการที่จะซื้อสินค้า
สมชาย	A	มี	ถูก	ซื้อ
สมศรี	B	มี	แพง	ไม่ซื้อ
สมปอง	A	ไม่มี	ถูก	ซื้อ
สมศักดิ์	C	มี	แพง	ซื้อ
สมหมาย	C	ไม่มี	ถูก	ไม่ซื้อ
สมหญิง	B	มี	ถูก	ซื้อ
สมใจ	C	ไม่มี	แพง	ไม่ซื้อ
สมจิตร	A	ไม่มี	แพง	ซื้อ

แผนปฏิบัติการสอนสัปดาห์ที่ 14

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี

เรื่อง โครงข่ายประสาทเทียม **เวลา :** บรรยาย 180 นาที

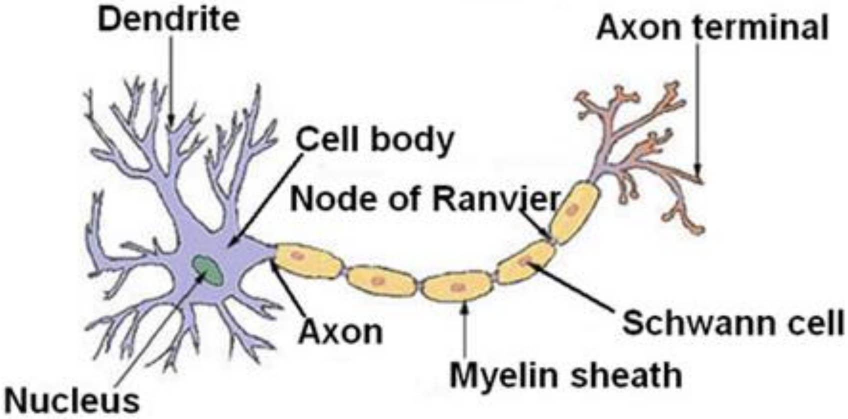
ก. วัตถุประสงค์การสอน	**รายละเอียดตามที่ระบุไว้ใน**
1. ประวัติความเป็นมาของโครงข่ายประสาทเทียม	IS 14.1
2. หลักการทำงานของระบบโครงข่ายประสาทเทียมในความพิวเตอร์	IS 14.2
3. การเรียนรู้ของโครงข่ายประสาทเทียม	IS 14.3
4. สถาปัตยกรรมโครงข่ายประสาทเทียม	IS 14.4
5. การใช้อัลกอริทึมแบบ Back-Propagation	IS 14.5
6. ประโยชน์ของโครงข่ายประสาทเทียม	IS 14.6
7. การประยุกต์ใช้งานโครงข่ายประสาทเทียม	IS 14.7

ข. การจัดการเรียนการสอน

เวลา – นาที		0	60	120	180
วัตถุประสงค์		**1-2-3-4-5-6-7**			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 14
โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial Neural Network: ANN) ที่มักจะเรียกสั้น ๆ ว่า ระบบโครงข่ายประสาท (Neural Network หรือ Neural Net) หมายถึง ระบบคอมพิวเตอร์ที่สร้างขึ้นเลียนแบบเส้นใยประสาทของมนุษย์ ที่เรียกว่า นิวรอน (Neurons) โดยโครงข่ายประสาท คือโมเดลทางคณิตศาสตร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์(Connectionist) เพื่อจำลองการทำงานของเครือขายประสาทในสมองมนุษย ดวยวัตถุประสงคที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรูการจดจำแบบรูป (Pattern Recognition) และการอุปมานความรู (Knowledge Deduction) เชนเดียวกับความสามารถที่มีในสมองมนุษย์ แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาข่ายงานไฟฟ้าชีวภาพ (Bioelectric Network) ในสมองดังภาพที่ 14-1 ซึ่งประกอบด้วย เซลล์ประสาท หรือ นิวรอน (neurons) และ จุดประสานประสาท (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า เดนไดรท์ (Dendrite) ซึ่งเป็น Input และปลายในการส่งกระแสประสาทเรียกว่า แอคซอน (Axon) ซึ่งเป็นเหมือน Output ของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียส (Nucleus) ซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่นๆ ต่อไปผ่านทางแอคซอน



ภาพที่ 14-1 โครงสร้างของเซลล์ประสาทในสมองมนุษย์
(ที่มาจาก: http://wiki.bethanycrane.com/artificial-neural-networks)

14.1 ประวัติความเป็นมาของโครงข่ายประสาทเทียม

ในปี พ.ศ. ๒๔๘๖ เป็นปีแห่งการกำเนิดของสาขาโครงข่ายประสาทเทียมในวงการวิทยาศาสตร์ โดย แม็กคัลลอช (Mcculloch) และพิตส์ (Pitts) ได้เสนอแบบจำลองของเซลล์ประสาทและได้แสดงให้เห็นว่า ในทางทฤษฎีแล้วโครงข่ายของแบบจำลองเซลล์ประสาทดังกล่าวสามารถทำงานเป็นโปรแกรมคอมพิวเตอร์ใดๆ ก็ได้

ปี พ.ศ. ๒๔๙๒ โดนัลด์ เฮบบ์ (Donald Hebb) ได้เสนอผลงานวิจัยว่า การเรียนรู้ของสมองสามารถอธิบายได้ด้วยรูปแบบของการประกอบเซลล์ประสาทเข้าด้วยกันเป็นโครงข่าย และได้เสนอกฎการเรียนรู้ของเฮบบ์ (Hebb's rule) ที่ทำให้โครงข่ายของเซลล์ประสาทเทียมที่แม็กคัลลอชและพิตส์ได้เสนอไว้สามารถเรียนรู้ปัญหาง่ายๆได้สำเร็จ การเรียนรู้ในแบบของเฮบบ์บนเซลล์ประสาทเทียมของแม็กคัลลอชและพิตส์นั้นเป็นการเรียนรู้แบบ "ไม่มีผู้สอน" ซึ่งในทางปฏิบัติแล้ว โครงข่ายประสาทเทียมที่ทำการเรียนรู้จะพยายามทำการจัดกลุ่มข้อมูลที่โครงข่ายมองว่าคล้ายคลึงกัน นำไปไว้ในกลุ่มเดียวกัน ซึ่งไม่เหมาะสมกับปัญหาประเภทที่ต้องมีการควบคุมกระบวนการเรียนรู้

ในช่วงพุทธทศวรรษ ๒๔๙๐ คอมพิวเตอร์ที่ทำงานเลียนแบบสมองเครื่องแรกของโลกถูกสร้างและทดสอบโดยมินสกี (Minsky) ซึ่งได้เสนอผลงานดังกล่าวในปี พ.ศ.๒๕๑๑ เมื่อคอมพิวเตอร์ดังกล่าวได้รับการป้อนตัวอย่างสำหรับการเรียนรู้เข้าไป ก็จะสามารถปรับอัตราการขยายสัญญาณในการเชื่อมโยงหรือ "ความแข็งแรงของการเชื่อมโยง" ระหว่างเซลล์ประสาทเทียมได้เองโดยอัตโนมัติ ซึ่งเป็นการแสดงการเรียนรู้ตัวอย่างที่ถูกป้อนเข้าไป

ในปี พ.ศ. ๒๕๐๑ แฟรงค์ โรเซนแบลตต์ (Frank Rosenblatt) ได้พัฒนาสาถาปัตยกรรมโครงข่ายประสาทเทียมขึ้น โดยใช้แบบจำลองของแม็กคัลลอชและพิตส์เป็นแนวทาง รวมทั้งเสนอวิธีการเรียนรู้แบบใหม่สำหรับสถาปัตยกรรมโครงข่ายประสาทเทียมดังกล่าวด้วย โครงข่ายประสาทเทียมดังกล่าวเรียกว่าเพอร์เซปตรอน (Perceptron) ซึ่งมีการเรียนรู้แบบ "มีผู้สอน" (supervised learning) โดยการปรับความแข็งแรงของการเชื่อมโยง ซึ่งจะพิจารณาได้จากการเปรียบเทียบความรู้ของโครงข่ายประสาทเทียมกับความรู้ของ "ผู้สอน" (teacher) เพอร์เซปตรอนมีความเหมาะสมกับงานประเภท "การระบุชนิด" ซึ่งในระหว่างการเรียนรู้นั้น เพอร์เซปตรอนจะถูกสอนว่าข้อมูลตัวอย่างที่สอนเข้าไปแต่ละแบบนั้นจัดเป็นชนิดใดบ้าง หากปัญหาและข้อมูลตัวอย่างมีความเหมาะสม เพอร์เซปตรอนจะสามารถระบุชนิดของข้อมูลที่ไม่เคยเห็นมาก่อนได้ถูกต้อง

ในช่วงต้นพุทธทศวรรษ ๒๕๐๐ เบอร์-นาร์ด วิโดรว์ (Bernard Widrow) และมาร์เชียนฮอฟฟ์ (Marcian Hoff) ได้พัฒนาอุปกรณ์ที่เรียกว่าอดาไลน์ (ADALINE;Adaptive Linear combiner) และกฎการเรียนรู้แบบใหม่ที่มีประสิทธิภาพสูงเรียกว่า กฎการเรียนรู้ของวิโดรว-ฮอฟฟ์ (Windrow-Hoff learning rule) ที่เป็นการเรียนรู้แบบ "มีผู้สอน" ซึ่งในเวลาต่อมาอุปกรณ์ดังกล่าวได้รับการขยายแนวคิดไปเป็นมาดาไลน์ (MADALINE;Many ADALINEs) และได้ถูกนำไปประยุกต์ใช้ในการรู้จำรูปแบบ (Pattern

Recognition) การพยากรณ์อากาศ และระบบควบคุมที่จำเป็นต้องมีการปรับเปลี่ยนระบบไปตามสภาพแวดล้อมต่างๆ

ในขณะที่การค้นคว้าเพื่อพัฒนาคอมพิวเตอร์ที่มี "ปัญญา" ได้ดำเนินไปในแนวทางของการศึกษาและได้พยายามอธิบายการทำงานของระบบประสาทของสิ่งมีชีวิต อีกแนวทางหนึ่งที่สามารถดำเนินควบคู่กันไปได้ คือแนวทางการค้นคว้าที่พยายามจะอธิบายพฤติกรรมที่เกี่ยวข้องกับปัญญาของมนุษย์ในเชิงโครงสร้างของเหตุและผล โดยใช้สัญลักษณ์ในการแทนแนวคิดของมนุษย์ และดำเนินการกับสัญลักษณ์เหล่านั้นด้วยกระบวนการในลักษณะของคณิตศาสตร์ผลลัพธ์ที่ได้จากกระบวนการดังกล่าวอาจถือได้ว่าเป็นการตอบสนองที่มนุษย์น่าจะทำภายใต้เงื่อนไขเดียวกัน แนวทางดังกล่าวนำไปสู่พัฒนาการของสาขาปัญญาประดิษฐ์ (Artificial Intelligence) และจิตวิทยาการเรียนรู้ (Cognitive Psychology) ในช่วงพุทธทศวรรษ ๒๕๑๐ และพุทธทศวรรษ ๒๕๒๐

ในปี พ.ศ. ๒๕๑๒ มินสกี (Minsky) และพาเพิร์ต (Papert) ได้เสนอผลงานตีพิมพ์ในรูปของหนังสือซึ่งในหนังสือดังกล่าวมีการระบุข้อจำกัดทางทฤษฎีของเพอร์เซปตรอน ผลงานตีพิมพ์ดังกล่าวก่อให้เกิดแนวคิดด้านลบต่อการพัฒนาระบบประมวลผลแบบโครงข่ายของเซลล์ประสาทและเป็นจุดเริ่มต้นจุดหนึ่งของกระแสแนวความคิดที่ว่า การคิดของมนุษย์เป็นการประมวลผลแบบลำดับขั้น (Serial Processing) เนื่องจากในช่วงเวลาดังกล่าวไม่มีผู้ที่สามารถแก้ไขข้อจำกัดทางทฤษฎีของโครงข่ายประสาทเทียมได้ วิทยาการด้านโครงข่ายประสาทเทียมจึงไม่ได้รับความสนใจจากวงการคอมพิวเตอร์ในช่วงพุทธทศวรรษ ๒๕๒๐ ผลงานวิจัยด้านโครงข่ายประสาทเทียมในช่วงเวลาดังกล่าวจึงเป็นที่รู้จักในวงแคบๆ ของผู้ที่สนใจเท่านั้น

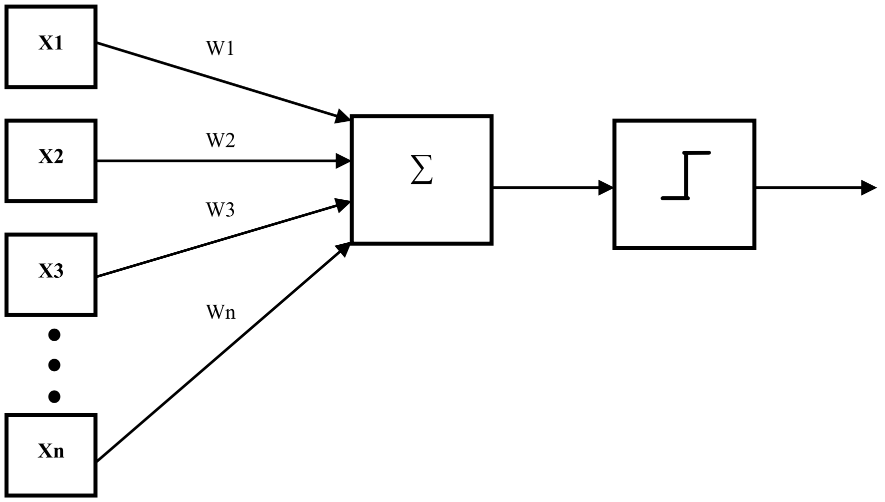
อย่างไรก็ตาม ในช่วงพุทธทศวรรษ ๒๕๑๐ มีผลงานวิจัยด้านโครงข่ายประสาทเทียมจากประเทศญี่ปุ่น โดย ชุนอิจิ อามาริ (Shun-Ichi Amari) ในปี พ.ศ. ๒๕๑๕ และ พ.ศ. ๒๕๒๐ ต่อมาในปีพ.ศ. ๒๕๒๓ คูนิฮิโก ฟูกูชิมา (Kunihiko Fuku-shima) ได้พัฒนาสถาปัตยกรรมโครงข่ายประสาทเทียมที่มีชื่อว่านีโอค็อกนิตรอน (neocognitrons) สำหรับการรู้จำรูปแบบภาพลักษณ์ โดยเลียนแบบแนวทางการมองเห็นของสิ่งมีชีวิต

ในปี พ.ศ. ๒๕๒๕ จอห์น ฮอปฟิลด์ (John Hopfield) ได้เสนอสถาปัตยกรรมโครงข่ายประสาทเทียมสำหรับเป็นหน่วยความจำแบบแอสโซซิเอทิฟ (Associatiov memory) ที่ดึงความจำออกมา โดยใช้ข้อมูลที่มีความเกี่ยวข้องกับความจำนั้นเป็นตัวชี้นำ ซึ่งเป็นจุดเริ่มที่ทำให้นักวิทยาศาสตร์หันกลับมาสนใจโครงข่ายประสาทเทียมอีกครั้ง ในช่วงเวลาใกล้เคียงกัน สตีเฟน กรอสเบอร์ก (StephenGrossberg) และเกล คาร์เพนเตอร์ (Gail Carpen-ter) ก็ได้เสนอทฤษฎีของการกำทอนแบบอะแดปทิฟ (Adaptive resonance) และได้พัฒนาโครงข่ายประสาทเทียมอาร์ต (ART Network) ซึ่งในปัจจุบันถือว่าเป็นโครงข่ายประสาทเทียมที่มีสมรรถนะสูงที่สุดแบบหนึ่ง

จากนั้นโครงข่ายประสาทเทียมกลับมาเฟื่องฟูเต็มที่เนื่องจากผลงานตีพิมพ์ของเจมส์ แม็กคลีแลนด์ (James McClelland) และเดวิด รูเมลฮาร์ต (David Rumelhart) ในปีพ.ศ. ๒๕๒๙ ซึ่งเสนอกฎการเรียนรู้แบบใหม่สำหรับสถาปัตยกรรมโครงข่ายประสาทเทียมที่มีความซับซ้อนขึ้น โดยอิงกับเพอร์เซปตรอน โครงข่ายประสาทเทียมดังกล่าวสามารถก้าวข้ามขีดจำกัดที่เคยถูกเสนอไว้โดยมินสกีและพาเพิร์ต ตั้งแต่ปี พ.ศ. ๒๕๑๒ ได้สำเร็จ อย่างไรก็ตาม มีการค้นพบว่า แนวทางคล้ายๆ กันได้เคยถูกเสนอแล้วโดย พอล เวอร์โบส (Paul Werbos) ตั้งแต่ปี พ.ศ. ๒๕๑๗ แต่ไม่มีผู้ใดสนใจในช่วงเวลาดังกล่าว ตั้งแต่นั้นมางานวิจัยค้นคว้าด้านโครงข่ายประสาทเทียมก็เกิดขึ้นอีกอย่างมากมายมาจนถึงปัจจุบัน

14.2 หลักการทำงานของระบบโครงข่ายประสาทเทียมในคอมพิวเตอร์

นักวิจัยส่วนใหญ่ในปัจจุบันเห็นตรงกันว่าโครงข่ายประสาทเทียมมีโครงสร้างแตกต่างจากโครงข่ายงานในสมองแต่ก็ยังเหมือนสมอง ในแง่ที่ว่าโครงข่ายประสาทเทียมคือการรวมกลุ่มแบบขนานของหน่วยประมวลผลย่อย ๆ และการเชื่อมต่อนี้เป็นส่วนสำคัญที่ทำให้เกิดสติปัญญาของข่ายงาน เมื่อพิจารณาขนาดแล้วสมองมีขนาดใหญ่กว่าโครงข่ายประสาทเทียมอย่างมาก รวมทั้งเซลล์ประสาทยังมีความซับซ้อนกว่าหน่วยย่อยของโครงข่าย อย่างไรก็ดีหน้าที่สำคัญของสมอง เช่น การเรียนรู้ยังคงสามารถถูกจำลองขึ้นอย่างง่ายด้วยโครงข่ายประสาทนี้



ภาพที่ 14-2 โครงสร้างของเซลล์ประสาท (Neurons) ในคอมพิวเตอร์

สำหรับในคอมพิวเตอร์ Neurons ประกอบด้วย Input และ Output เหมือนกัน โดยจำลองให้ Input แต่ละอันมี Weight เป็นตัวกำหนดน้ำหนักของ Input โดย Neuron แต่ละหน่วยจะมีค่า Threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ Input ต้องมากขนาดไหนจึงจะสามารถส่ง Output ไปยัง

Neurons ตัวอื่นได้ เมื่อนำ Neuron แต่ละหน่วยมาต่อกันให้ทำงานร่วมกันการทำงานนี้ในทางตรรกะแล้วก็จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง

การทำงานโครงข่ายประสาท (Neural Network) คือเมื่อมี Input เข้ามายังเครือข่าย (Network) ก็จะนำเอา Input มาคูณกับ Weight ของแต่ละขา ผลที่ได้จาก Input ทุกๆ ขาของ Neuron จะเอามารวมกันแล้วก็เอามาเทียบกับ Threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า Threshold แล้ว Neuron ก็จะส่ง Output ออกไป Output นี้ก็จะถูกส่งไปยัง Input ของ Neuron อื่น ๆ ที่เชื่อมกันใน Network ถ้าค่าน้อยกว่า Threshold ก็จะไม่เกิด Output ซึ่งแสดงได้ดังตัวโปรแกรมเทียมนี้

*if (sum(input * weight) > threshold) then output*

สิ่งสำคัญคือต้องทราบค่า Weight และ Threshold สำหรับสิ่งที่ผู้ใช้ต้องการเพื่อให้คอมพิวเตอร์รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้โดยการสอนให้คอมพิวเตอร์รู้จัก Pattern ของสิ่งที่ต้องการให้คอมพิวเตอร์รู้จำ เรียกว่า "Back Propagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำ ในการฝึก Feed-Forward Neural Networks จะมีการใช้อัลกอริทึมแบบ Back-Propagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของเครือข่าย (Network Weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว ค่าที่ได้รับ (Output) จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนักคะแนนต่อไป

14.3 การเรียนรู้ของโครงข่ายประสาทเทียม

ผลการวิจัยทางพฤติกรรมศาสตร์พบว่าการเรียนรู้ของสิ่งมีชีวิตชนิดต่างๆ นั้นมีกระบวนการแตกต่างกันไปหลายๆ แบบ แต่ละแบบก็อาจเหมาะสมกับแต่ละเผ่าพันธุ์ของสิ่งมีชีวิตนั้นๆในสาขาโครงข่ายประสาทเทียมนั้น แนวคิดของกระบวนการเรียนรู้จะประยุกต์มาจากผลการศึกษาทางพฤติกรรมศาสตร์ อาจกล่าวโดยทั่วไปได้ว่าการเรียนรู้คือ กระบวนการซึ่งระบบประสาทปรับตัวเองไปตามสิ่งเร้า จนกระทั่งสามารถให้ผลตอบได้ตามต้องการ โดยใช้การปรับตัวแปรที่ควบคุมสภาพของตัวระบบเอง

การเรียนรู้ยังสามารถถูกมองได้ว่า เป็นกระบวนการจัดชนิดของสิ่งเร้าทั้งหลายที่เข้ามาอย่างต่อเนื่องด้วย นั่นคือ เมื่อได้รับสิ่งเร้า หากระบบประสาทรู้จักสิ่งเร้านั้น ก็จะให้ผลตอบได้ตามที่เคยเข้าใจไว้ แต่หากไม่รู้จัก ก็พยายามปรับความเข้าใจในการจัดชนิดขึ้นใหม่ ในทางปฏิบัตินั้น ระบบประสาทของสิ่งมีชีวิตจะปรับความเหนียวแน่นของการเชื่อมต่อที่ซิแนปส์ จนสร้างผลตอบต่อสิ่งเร้าได้ตามที่ต้องการ สถานะที่กระบวนการของการเรียนรู้ก็จะสิ้นสุดลง เป็นสถานะที่ถือว่าระบบประสาทได้รับความรู้ไปแล้วคำจำกัด

ความของกระบวนการเรียนรู้หมายถึงขั้นตอนต่อไปนี้

ขั้นที่ 1 โครงข่ายประสาทถูกกระตุ้นด้วยสิ่งแวดล้อม

ขั้นที่ 2 โครงข่ายประสาทเกิดการเปลี่ยนแปลง อันเป็นผลมาจากการกระตุ้นดังกล่าว

ขั้นที่ 3 โครงข่ายประสาทตอบสนองต่อสิ่งแวดล้อมในแนวทางใหม่ อันเป็นผลมาจากการเปลี่ยนแปลงที่เกิดขึ้นในโครงสร้างภายในโครงข่าย

เมื่อพิจารณาไปที่เฉพาะบริเวณหนึ่งๆ ของโครงข่ายประสาทจะพบว่า การเชื่อมต่อของเซลล์ประสาทที่บริเวณต่างๆ นั้นสามารถมีรูปแบบที่แตกต่างกันได้หลายๆ แบบ และกระบวนการเรียนรู้ของแต่ละบริเวณก็ไม่เหมือนกันด้วย ในทำนองเดียวกัน เทคนิคการเรียนรู้ของโครงข่ายประสาทเทียมก็แตกต่างกันไปสำหรับแต่ละชนิดของโครงข่าย

การสอนโครงข่ายให้ทำการคำนวณข้อมูล Output พร้อมปรับปรุงค่าน้ำหนักโดยใช้ข้อมูล Input ที่ป้อนให้กับโครงข่ายโดยอาศัยกระบวนการทำซ้ำสามารถแบ่งเป็น 2 ประเภท คือ

- การเรียนรู้แบบมีผู้สอน(Supervised Learning)
- การเรียนรู้แบบไม่มีผู้สอน(Unsupervised Learning)

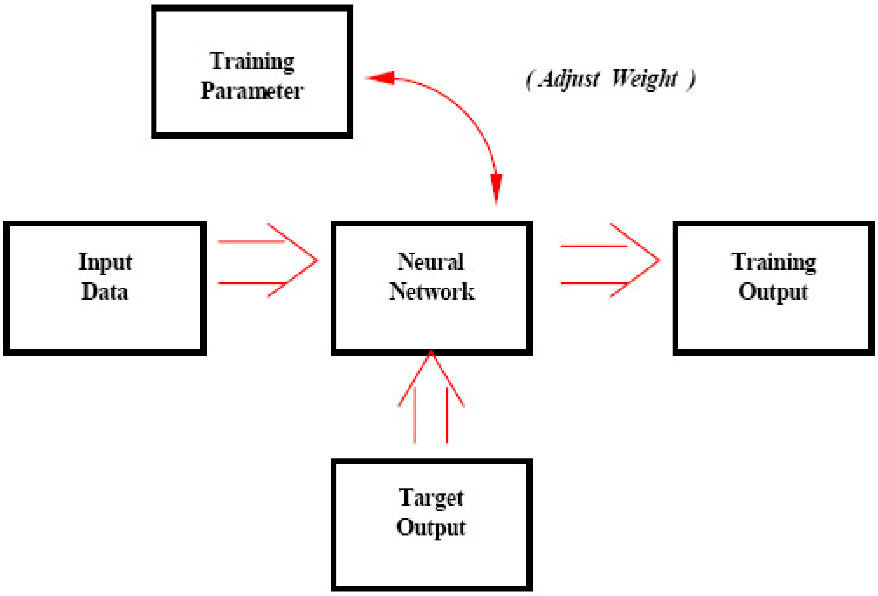
14.3.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

รูปแบบการเรียนรู้แบบมีผู้สอนเริ่มด้วยการส่งสิ่งเร้าที่ใช้ในการสอนเข้าไปเป็นอินพุต (Input) ในโครงข่ายประสาทเทียม เพื่อให้โครงข่ายประสาทเทียมสร้างผลตอบออกมาเป็นเอาต์พุต (Output) ซึ่งผลตอบจะเป็นอย่างไร ก็ขึ้นอยู่กับสภาวะในตอนที่เริ่มเรียนรู้ของโครงข่ายประสาทเทียม ผลตอบดังกล่าวจะถูกนำมาเปรียบเทียบกับผลตอบเป้าหมาย (Target Response) ซึ่งผู้สอน (Teacher) จะเป็นผู้สร้างขึ้น หากผลตอบทั้งสองมีความแตกต่างกัน นั่นคือมีความคลาดเคลื่อน (Error) เกิดขึ้นความคลาดเคลื่อนดังกล่าวจะถูกนำไปคำนวณการปรับแต่งค่าน้ำหนักต่างๆ ในโครงข่ายประสาทเทียม เพื่อลดความคลาดเคลื่อนลงให้เหลือน้อยที่สุด

การปรับแต่งค่าน้ำหนักโดยพิจารณาจากความคลาดเคลื่อนนี้ จะขึ้นอยู่กับกฎการเรียนรู้หรือขั้นตอนการคำนวณซึ่งเรียกว่า "อัลกอริทึม"(Algorithm) ที่แตกต่างกันโดยแต่ละอัลกอริทึมจะมีคุณลักษณะและสมรรถนะแตกต่างกัน วิธีการส่วนใหญ่ของการเรียนรู้แบบมีผู้สอนนี้จะดัดแปลงมาจากวิธีการทางคณิตศาสตร์ในเรื่องของเทคนิคการหาค่าเหมาะสม (Optimization Technique)

เนื่องจากเจตนาของมนุษย์ในการพัฒนาเครื่องมือขึ้นมาใช้งานนั้น จะอิงอยู่กับการที่มนุษย์ต้องการสั่งการและควบคุมเครื่องมือนั้นๆ ให้ทำงานได้ตามที่ต้องการ จึงทำให้โครงข่ายประสาทเทียมประเภทที่ใช้การเรียนรู้แบบมีผู้สอนได้รับความนิยมในการนำไปประยุกต์ใช้มากที่สุด เนื่องจากเป็นแบบที่สามารถควบคุมได้ การสั่งการโครงข่ายประสาทเทียมจะเป็นไปโดยทางอ้อม ในลักษณะของการฝึกสอนโครงข่ายประสาทเทียม โดยการสร้างข้อมูลตัวอย่าง(รวมทั้งค่าเป้าหมาย) ที่จะให้โครงข่ายเรียนรู้ เมื่อ

โครงข่ายประสาทเทียมเรียนรู้ข้อมูลตัวอย่างได้ถูกต้องหมดแล้ว ความรู้ที่โครงข่ายประสาทเทียมได้เก็บไว้ในลักษณะของค่าน้ำหนักต่างๆ จะเป็นสิ่งที่ถูกนำไปใช้งานจริงเพื่อสร้างผลตอบต่อข้อมูลใหม่ๆ ที่โครงข่ายไม่เคยเห็นมาก่อน ดังนั้น สำหรับโครงข่ายประสาทเทียมหนึ่งๆ และวิธีการเรียนรู้แบบมีผู้สอนวิธีการหนึ่งๆ นั้น ความรู้ของโครงข่ายประสาทเทียมจะสามารถนำไปใช้งานจริงได้เพียงใด ก็ขึ้นอยู่กับคุณภาพของข้อมูลตัวอย่างที่นำมาใช้สอนนั้น อาจจะกล่าวได้ว่า หากข้อมูลตัวอย่างมีจำนวนมากพอโครงข่ายประสาทเทียมก็จะสามารถสร้างความรู้ได้อย่างถูกต้อง อย่างไรก็ตามด้วยระดับความเจริญก้าวหน้าด้านโครงข่ายประสาทเทียมในปัจจุบันเทคนิคการวิเคราะห์ปัญหาเพื่อสร้างข้อมูลตัวอย่างสำหรับการเรียนรู้ที่มีประสิทธิภาพนั้นยังคงเป็นงานวิจัยที่ต้องมีการค้นคว้ากันต่อไป ลักษณะการเรียนรู้แบบมีผู้สอนแสดงดังภาพที่ 14-3



ภาพที่ 14-3 การเรียนรูแบบมีผู้สอน (Supervised Learning)

14.3.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

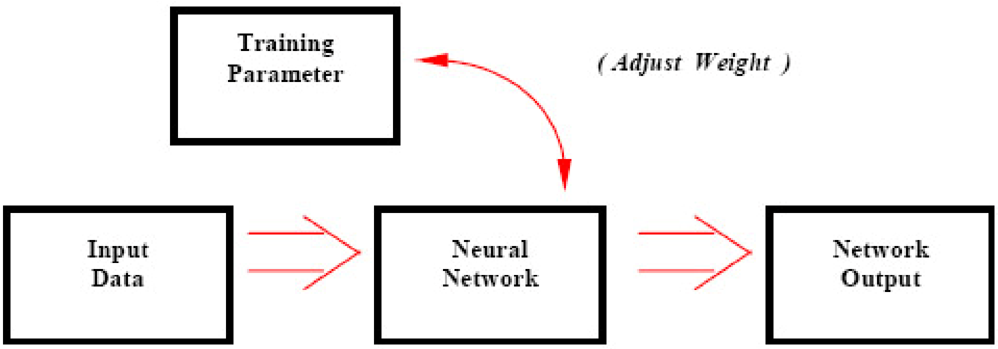
การเรียนรู้แบบไม่มีผู้สอนนั้นไม่จำเป็นต้องมีค่าเป้าหมายของแต่ละข้อมูลตัวอย่างในระหว่างการเรียนรู้ โครงข่ายประสาทเทียมจะได้รับข้อมูลกระตุ้นในรูปแบบต่างๆ และจะทำการจัดกลุ่มรูปแบบต่างๆ เหล่านั้นเองตามต้องการ ผลตอบของโครงข่ายประสาทเทียมที่ใช้การเรียนรู้แบบไม่มีผู้สอนนี้จะเป็นการระบุกลุ่มของข้อมูลที่ใส่เข้าไป โดยจะอิงกับวิธีการจัดกลุ่มซึ่งได้เรียนรู้จากข้อมูลที่โครงข่ายที่เคยพบมา

ตัวอย่างการเรียนรู้แบบนี้ในมนุษย์คือ การให้เด็กเล็กๆ จัดเก็บสิ่งของไว้บนชั้นวางของให้เป็นระเบียบเรียบร้อย สมมุติว่า เด็กคนหนึ่งเลือกเก็บหนังสือต่างๆ ไว้ที่ชั้นบน เก็บตุ๊กตาไว้ที่ชั้นล่างและเก็บของเล่นอื่นๆ ไว้ที่ชั้นกลางๆ หลังจากนั้นหากเด็กคนนั้นซื้อตุ๊กตามาใหม่ ก็นำไปเก็บไว้ที่ชั้นล่าง เป็นต้น

แม้ว่าการเรียนรู้แบบไม่มีผู้สอนนี้จะไม่ต้องการผู้สอน แต่ก็ต้องการแนวทางในการจัดกลุ่ม เช่น การจัดกลุ่มอาจจะจัดตามรูปทรง สีหรือวิธีการใช้งานของวัตถุต่างๆ ที่จะนำมาจัดเป็นต้น ดังนั้น หากไม่มีการให้แนวทางที่ชัดเจนว่า การจัดกลุ่มควรเป็นไปตามคุณลักษณะใด การจัดกลุ่มอาจไม่ประสบความสำเร็จในแง่การนำมาใช้งานจริงก็ได้ ตัวอย่างเช่น การให้เด็กจัดของไว้บนชั้นวางของนั้น เด็กอาจจะจัดตามใจชอบและไม่เป็นหมวดหมู่ ทำให้ไม่สะดวกต่อการนำสิ่งของใหม่ๆ เข้าไปเก็บรวมด้วยก็ได้ การใช้งานโครงข่ายประสาทเทียมที่ใช้การเรียนรู้แบบนี้ จึงมักต้องมีการดำเนินการปรับแต่งข้อมูล เพื่อให้เกิดการเน้นสภาพของคุณลักษณะสำคัญ ที่ต้องการนำมาเป็นแนวทางในการจัดกลุ่มให้เด่นชัดขึ้นหรืออาจเป็นการปรับกฎการเรียนรู้ เพื่อให้เน้นไปที่คุณลักษณะที่ต้องการก็ได้

เนื่องจากขั้นตอนการเรียนรู้แบบไม่มีผู้สอนนี้จะมีการระบุกลุ่มของข้อมูลตัวอย่างก่อน เมื่อตัดสินใจได้แล้วว่า ข้อมูลใหม่มีลักษณะที่ควรจะจัดรวมเข้ากลุ่มใด (หรืออาจถือเป็นกลุ่มใหม่ก็ได้ในกรณีที่เห็นว่าไม่ควรจัดเข้ากลุ่มใดเลย) หลังจากนั้น จึงมีการปรับคุณลักษณะของกลุ่ม โดยการนำลักษณะของข้อมูลใหม่นี้มาช่วยกำหนดแนวทางการจัดด้วย ในการตัดสินว่าข้อมูลใหม่นี้ควรจัดรวมเข้ากลุ่มใด ในโครงข่ายประสาทเทียมบางชนิดอาจจะใช้วิธีการแข่งขันกันของกลุ่มต่างๆ ว่ากลุ่มใดควรได้ข้อมูลดังกล่าวไป การเรียนรู้ในลักษณะนี้จะถูกเรียกว่า การเรียนรู้แบบแข่งขันกัน (Competitive Learning)

ในอีกแง่มุมหนึ่ง นับจากจุดเริ่มต้นของการเรียนรู้ซึ่งไม่มีการจัดกลุ่มข้อมูลในแบบใดๆ เลย จนถึงเวลาที่การจัดเสร็จสิ้นแล้ว จะพบว่า การจัดกลุ่มข้อมูลเกิดขึ้นตามคุณลักษณะบางอย่างของข้อมูลตัวอย่าง ซึ่งการจัดกลุ่มนี้เกิดจากการที่โครง-ข่ายประสาทเทียมประเมินข้อมูลต่างๆ ที่ถูกป้อนเข้าไปในระหว่างการเรียนรู้ จนสร้างเป็นวิธีการจัดกลุ่มขึ้นมาได้ ดังนั้น การเรียนรู้ในลักษณะดังกล่าวจึงถูกเรียกว่า การเรียนรู้แบบจัดตัวเอง (Self-Organizing) ด้วย ลักษณะการเรียนรู้แบบไม่มีผู้สอนแสดงดังภาพที่ 14-4



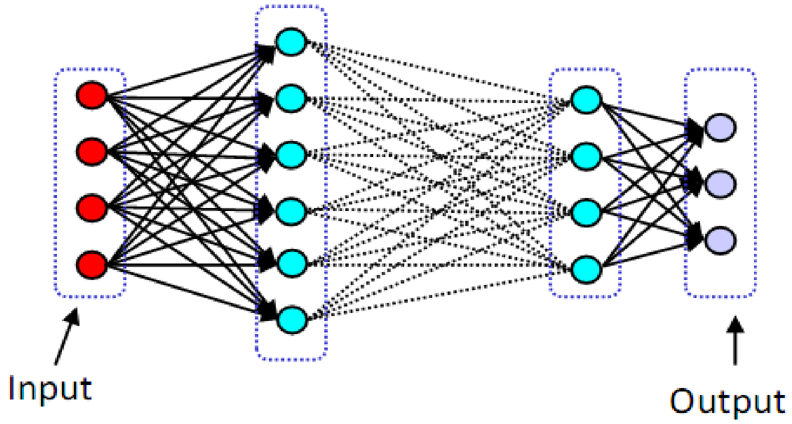
ภาพที่ 14-4 การเรียนรู้แบบไม่มีผู้สอน Unsupervised Learning

14.4. สถาปัตยกรรมโครงข่ายประสาทเทียม

สถาปัตยกรรมโครงข่ายประสาทเทียม สามารถแบ่งเป็น 4 ประเภทใหญ่ๆ คือ

14.4.1 Feed–Forward Network

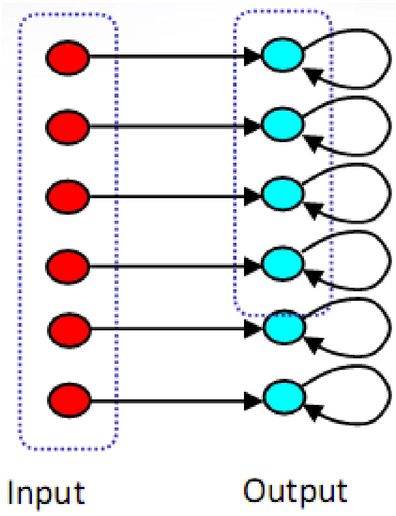
ข้อมูลที่ประมวลผลในวงจรข่ายจะถูกส่งไปในทิศทางเดียวจากโหนด Input ส่งต่อมาเรื่อยๆ จนถึงโหนด Output โดยไม่มีการย้อนกลับของข้อมูล หรือแม้แต่โหนดในชั้น (Layer) เดียวกันก็ไม่มีการเชื่อมต่อกัน ดังแสดงในภาพที่ 14-5



ภาพที่ 14-5 สถาปัตยกรรมของ Feed-Forward Network

14.4.2 Feed-Back Network

ข้อมูลที่ประมวลผลในวงจรข่าย จะมีการป้อนกลับเข้าไปยังวงจรข่ายหลาย ๆ ครั้ง จนกระทั่งได้คำตอบออกมา (บางทีเรียกว่า Recurrent Network) โครงข่ายมีลักษณะดังแสดงในภาพที่ 14-6



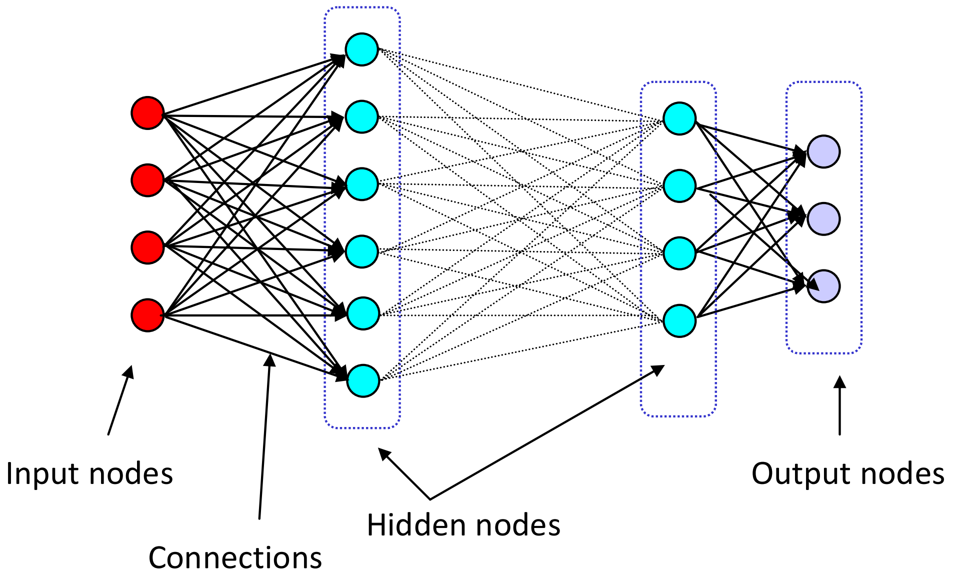
ภาพที่ 14-6 สถาปัตยกรรมของ Feed–Back Network

14.4.3 Network Layer

พื้นฐานสามัญที่สำคัญของโครงข่ายประสาทเทียม ประกอบไปด้วย 3 ส่วน หรือ 3 layer ได้แก่ ชั้นของ Input Units ที่ถูกเชื่อมต่อกับชั้นของ Hidden Units ซึ่งเชื่อมต่อกับชั้นของ Output Units

- Input Units จะทำหน้าที่แทนส่วนของข้อมูลดิบ ที่จะถูกป้อนเข้าสู่เครือข่าย
- Hidden Units จะถูกกำหนด โดยการทำงานของ Input Units และค่าน้ำหนักบนความสัมพันธ์ระหว่าง Input Units และ Hidden Units
- Output Units จะขึ้นอยู่กับการทำงานของ Hidden Units และค่าน้ำหนักระหว่าง Hidden Units และ Output Units

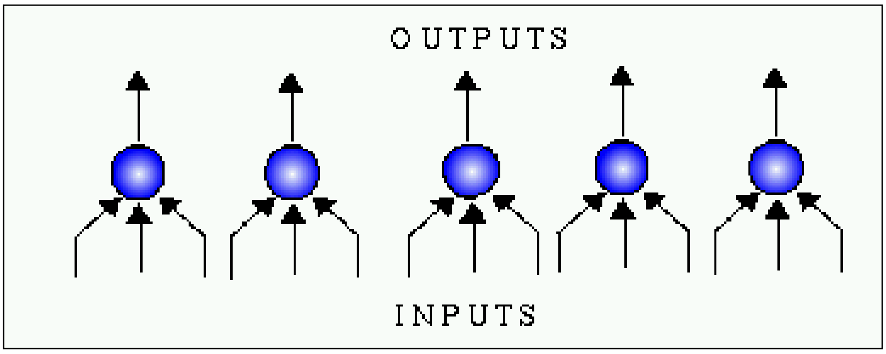
ประเภทของเครือข่ายนี้เป็นที่น่าสนใจ เพราะเราสามารถกำหนดการแทนค่าให้แก่ Input Units ได้อย่างอิสระ ค่าน้ำหนักระหว่าง Input Units และ Hidden Units จะถูกกำหนดเมื่อ Hidden Unit กำลังทำงานเพราะฉะนั้นเวลาที่แก้ไขค่าน้ำหนัก Hidden Units จะสามารถเลือกว่าอะไรคือค่าที่เราแทนเข้ามา ดังแสดงในภาพที่ 14-7



ภาพที่ 14-7 สถาปัตยกรรมของ Network Layer

14.4.4 Architecture of Layer

Architecture of Layer สามารถจำแนกสถาปัตยกรรมของชั้น (Layer) ออกเป็น 2 ประเภทคือ Single-Layer และ Multi-Layer



ภาพที่ 14-8 Single-Layer Perceptron

14.4.4.1 Single-Layer Perceptron

เครือข่ายใยประสาทที่ประกอบด้วยชั้นเพียงชั้นเดียว จำนวนโหนด Input ขึ้นอยู่กับจำนวน Components ของ Input Data และ Activation Function ขึ้นอยู่กับลักษณะข้อมูลของ Output ดังแสดงในภาพที่ 14-8 เช่น ถ้า Output ที่ต้องการเป็น “ใช่” หรือ “ไม่ใช่” เราจะต้องใช้ Threshold Function

$$f(x) = \begin{cases} 1 & \text{if } x \geq T \\ 0 & \text{if } x < T \end{cases}$$ T = Threshold level

หรือถ้า Output เป็นค่าตัวเลขที่ต่อเนื่อง เราต้องใช้ Continuous Function เช่น Sigmoid Function

$$f(x) = \frac{1}{1+e^{-\alpha x}}$$

14.4.4.2 Multi-layer Perceptron

เครือข่ายใยประสาทจะประกอบด้วยหลายชั้นโดยในแต่ละชั้นจะประกอบด้วยโหนด (Nodes) หรือเปรียบได้กับตัวเซลล์ประสาท (Neurons) ค่าน้ำหนักของเส้นที่เชื่อมต่อระหว่างโหนดของแต่ละชั้น (เมทริก W), ค่า Bias Vector (b) และค่า Output Vector (a) โดย m เป็นตัวเลขบอกลำดับชั้นกำกับไว้ด้านบน เมื่อ p เป็น Input Vector การคำนวณค่าเอาต์พุตสำหรับเครือข่ายใยประสาทที่มี M ชั้นจะเป็นดังสมการ

$$a^{m+1} = f^{m+1}(W^{m+1}a^{m} + b^{m+1})$$

เมื่อ $m = 0, 2, \ldots, M-1$

$a^0 = p$

$a = a^m$

และ f เป็น Transfer Function

14.5 การใช้อัลกอริทึมแบบ Back-Propagation

Back-Propagation เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครือข่ายใยประสาทวิธีหนึ่งที่นิยมใช้ใน Multilayer Perceptron เพื่อปรับค่าน้ำหนักในเส้นเชื่อมต่อระหว่างโหนดให้เหมาะสม โดยการปรับค่านี้จะขึ้นกับความแตกต่างของค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่ต้องการ

ขั้นตอนของ Back-propagation Algorithm มีดังนี้

1. กำหนดค่าอัตราเร็วในการเรียนรู้ (Rate parameter : r)
2. สำหรับแต่ละตัวอย่างอินพุตให้ทำตามขั้นตอนต่อไปนี้จนกว่าได้ระดับ performance ที่ต้องการ

- คำนวณหาค่าเอาต์พุตโดยใช้ค่าน้ำหนักเริ่มต้นซึ่งอาจได้จากการสุ่ม
- คำนวณหาค่า β : แทนประโยชน์ที่จะได้รับสำหรับการเปลี่ยนค่าเอาต์พุตของแต่ละโหนด
- ในชั้นเอาต์พุต (Output Layer)

$$\beta_z = d_z - o_z$$

เมื่อ d_z = ค่าเอาต์พุตที่ต้องการ

o_z = ค่าเอาต์พุตที่คำนวณได้

- ในชั้นซ่อน (Hidden Layer)

$$\beta_j = \sum w_{jk} o_k (1 - o_k) \beta_k$$

เมื่อ w_{jk} = น้ำหนักของเส้นเชื่อมระหว่างชั้นที่ j กับ k

- คำนวณค่าน้ำหนักที่เปลี่ยนแปลงไปสำหรับในทุกน้ำหนัก ด้วยสมการต่อไปนี้

$$\Delta w_{ij} = \sum r o_i o_j (1 - o_j) \beta_j$$

- เพิ่มค่าน้ำหนักที่เปลี่ยนแปลง สำหรับตัวอย่างอินพุตทั้งหมด และเปลี่ยนค่าน้ำหนัก

14.6 ประโยชน์ของโครงข่ายประสาทเทียม

14.6.1 เกิดข้อผิดพลาดได้ยาก (Fault Tolerance)

หากระบบโครงข่ายใยประสาทเทียมประกอบไปด้วยโครงข่ายที่ใช้ในการประมวลผลมากมายหลายโครงข่าย ความผิดพลาดที่เกิดขึ้นจากเพียงหนึ่งหรือสองโครงข่ายจะไม่ทำให้ทั้งระบบเกิดข้อผิดพลาดได้

14.6.2 ความสามารถในการหาเหตุผล (Generalization)

เมื่อระบบโครงข่ายใยประสาทเทียมได้รับข้อมูลนำเข้าที่ไม่ครบถ้วนหรือไม่เพียงพอต่อการหาข้อสรุป หรือได้รับข้อเท็จจริงที่ไม่เคยได้รับมาก่อน ระบบจะสามารถลำดับการเชื่อมโยงข้อเท็จจริงจนสามารถให้ข้อสรุปและเหตุผลได้

14.6.3 ความสามารถในการปรับเปลี่ยน (Adaptability)

โครงข่ายใยประสาทเทียมสามารถเรียนรู้สภาพแวดล้อมใหม่ได้ ดังนั้นเมื่อมีเหตุการณ์ใหม่ๆ เข้าสู่ระบบก็จะสามารถปรับเปลี่ยนหรือปรับปรุงองค์ความรู้ให้ทันสมัยตามเหตุการณ์ใหม่นั้น

14.6.4 ความสามารถในการพยากรณ์ (Forecasting Capability)

โครงข่ายใยประสาทเทียมสามารถนำข้อมูลทางสถิติเดิมที่มีอยู่ในระบบ มาใช้คาดการณ์หรือพยากรณ์ข้อมูลในอนาคตได้

14.7 การประยุกต์ใช้งานโครงข่ายประสาทเทียม

แบบข่ายงานระบบประสาท(Neural Network) เนื่องจากความสามารถในการจำลองพฤติกรรมทางกายภาพของระบบที่มีความซับซ้อนจากข้อมูลที่ป้อนให้เรียนรู การประยุกต์ใช้ข่ายงานระบบประสาทจึงเป็นหนทางเลือกใหม่ในการควบคุม ซึ่งมีผู้นำมาประยุกต์ใช้งานหลายประเภท ไดแก

- งานการจดจำรูปแบบที่มีความไม่แน่นอน เช่น ลายมือ ลายเซ็น ตัวอักษร รูปหน้า
- งานการประมาณค่าฟังก์ชันหรือการประมาณความสัมพันธ์ (มี Inputs และ Outputs แต่ไม่ทราบว่า Inputs กับ Outputs มีความสัมพันธ์กันอย่างไร)
- งานที่สิ่งแวดล้อมเปลี่ยนแปลงอยู่เสมอ(วงจรข่ายนิวรอนสามารถปรับตัวเองได้)
- งานจัดหมวดหมู่และแยกแยะสิ่งของ
- งานทำนาย เช่นพยากรณ์อากาศ พยากรณ์หุ้น
- การประยุกต์ใช้ข่ายงานระบบประสาทควบคุมกระบวนการทางเคมีโดยวิธีพยากรณ์แบบจำลอง (Model Predictive Control)
- การประยุกต์ใช้ข่ายงานระบบประสาทแบบแพรกระจายกลับในการทำนายพลังงานความร้อนที่สะสมอยู่ในตัวอาคาร

14.8 สรุป

โครงข่ายประสาทเทียม (Artificial Neural Network) คือการสร้างคอมพิวเตอร์ที่จำลองเอาวิธีการทำงานของสมองมนุษย์หรือทำให้คอมพิวเตอร์รู้จักคิดและจดจำในแนวเดียวกับโครงข่ายประสาทของมนุษย์ เพื่อช่วยให้คอมพิวเตอร์ฟังภาษามนุษย์ได้เข้าใจ อ่านออก และรู้จำได้ซึ่งอาจเรียกได้ว่าเป็น “สมองกล”

โครงสร้างของโครงข่ายประสาทเทียม ประกอบด้วย input units ,output units โดยมีการกำหนดค่าน้ำหนักให้แก่เส้นทางการนำเข้าของ input แต่ละตัว ในการเรียนรู้ของเครือข่ายใยประสาท จะอาศัย Back-propagation Algorithm ในการเขียน การสร้างการเรียนรู้สำหรับ Neural Network เพื่อให้มีความคิดเสมือนมนุษย์ซึ่งมีสองวิธี คือ Supervised Learning การเรียนรู้แบบมีการสอนเปรียบเทียบกับคน เหมือนกับการสอนนักเรียนโดยมีครูผู้สอนคอยแนะนำ และ Unsupervised Learning การเรียนรู้แบบไม่มีการสอน เปรียบเทียบกับคน เช่น การที่เราสามารถแยกแยะพันธุ์พืช พันธุ์สัตว์ตามลักษณะรูปร่างของมันได้เองโดยไม่มีใครสอน อีกทั้งสถาปัตยกรรมโครงข่ายประสาทเทียมยังสามารถแบ่งเป็น 4 แบบ คือ Feed-Forward Network, Feed-Back network, Network Layer และ Perceptrons

เนื่องจากความสามารถในการจำลองพฤติกรรมทางกายภาพของระบบที่มีความซับซ้อน จากข้อมูลที่ป้อนให้เพื่อการเรียนรู้ การประยุกตใช้โครงข่ายประสาทเทียมผู้นำมาประยุกตใช้งานหลายประเภท เช่น งานการจดจำรูปแบบที่มีความไม่แน่นอน งานการประมาณค่าฟังก์ชันหรือการประมาณความสัมพันธ์ งานที่สิ่งแวดล้อมเปลี่ยนแปลงอยู่เสมอ เนื่องจากวงจรข่ายนิวรอลสามารถปรับตัวเองได้และนอกจากนี้ยังสามารถนำไปประยุกต์ในงานต่าง ๆ อีกหลายงาน

แบบฝึกหัด

1. จงอธิบายหลักการทำงานของโครงข่ายประสาทเทียม
2. จงอธิบายส่วนประกอบที่สำคัญของโครงข่ายประสาทเทียมที่สำคัญๆ ว่ามีอะไรบ้าง
3. จงอธิบายประโยชน์ของการใช้งานโครงข่ายประสาทเทียม
4. จงเขียนตัวอย่างของงานที่สามารถนำโครงข่ายประสาทเทียมไปประยุกต์ใช้เพื่อเพิ่มประสิทธิของการทำงาน
5. จงอธิบายการทำงานขออัลกอริทึมแบบ Back-Propagation

แผนปฏิบัติการสอนสัปดาห์ที่ 15

วิชา 030523111 ความรู้เบื้องต้นทางปัญญาประดิษฐ์ **ระดับ** ปริญญาตรี
เรื่อง อัลกอริทึมเชิงพันธุกรรม **เวลา :** บรรยาย 180 นาที

ก. วัตถุประสงค์การสอน	รายละเอียดตามที่ระบุไว้ใน
1. ประวัติความเป็นมาของอัลกอริทึมเชิงพันธุกรรม	IS 15.1
2. องค์ประกอบขั้นตอนวิธีเชิงพันธุกรรม	IS 15.2
3. ตัวอย่างการแก้ปัญหา 8 queen ด้วยอัลกอริทึมเชิงพันธุกรรม	IS 15.3

ข. การจัดการเรียนการสอน

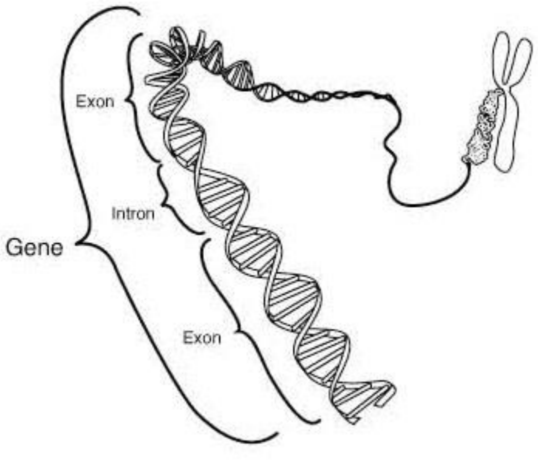
เวลา – นาที		0	60	120	180
วัตถุประสงค์		1-2-3			
การนำเข้าสู่บทเรียน					
ให้เนื้อหา					
สรุปเนื้อหา					
ประเมินผล		**พิจารณาจากคำถาม คำตอบระหว่างการสอน และการทำแบบฝึกหัดในชั่วโมงของนักศึกษา**			
วิธีการสอน:	บรรยาย				
	อธิบาย				
	ถาม – ตอบ				
	ปฏิบัติด้วยตนเอง				
สื่อการสอน:	กระดานดำ				
	คอมพิวเตอร์				
	โปรเจคเตอร์				

บทที่ 15
อัลกอริทึมเชิงพันธุกรรม

อัลกอริทึมเชิงพันธุกรรมหรือจีนเนติกอัลกอริทึม หมายถึงวิธีการแก้ปัญหาที่ใช้แนวทางเดียวกับวิธีการที่สิ่งมีชีวิตปรับตัวเองให้เข้ากับสภาพแวดล้อม หรือวิวัฒนาการ Evolution วิธีการที่ถูกโปรแกรมให้ทำการเปลี่ยนแปลง หรือปรับปรุงส่วนประกอบ ของระบบโดยการสร้างขึ้นมาใหม่ การดัดแปลงและการคัดสรรวิธีธรรมชาติใช้ ให้เกิดสิ่งใหม่ๆมาใช้งาน ประโยชน์ของการแก้ปัญหา ได้แก่ การออกแบบสินค้า การตรวจสอบระบบการทำงานต่างๆ รวมถึงสามารถนำไปประยุกต์ใช้งานทางด้านวิศวกรรม เช่น บทความวิจัยเกี่ยวกับการแก้ไขปัญหาทางเดินที่สั้นที่สุดของพนักงานขาย (Traveling Salesman Problems) ที่แนบไว้ในภาคผนวก เป็นต้น

15.1 ประวัติความเป็นมาของอัลกอริทึมเชิงพันธุกรรม (Genetic Algorithm)

Evolutionary strategies ค้นพบโดย I.Rechenber ในเยอรมันปี 1960 เป็นจุดเริ่มต้นการพัฒนา Genetic Algorithm ในสหรัฐอเมริกาปี 1975 Genetic Algorithm ถูกค้นพบโดย John Holland คือ กระบวนการเรียนแบบวิวัฒนาการของสิ่งมีชีวิตที่เกิดขึ้นในธรรมชาติโดยเกี่ยวข้องกับยีน ต่อมาในปี 1992 John Koza ได้ใช้ Genetic Algorithm พัฒนา Evolve Programs หรือ เรียกว่า Genetic Programming ทั้ง Genetic Algorithm, Genetic Programming อยู่บนพื้นฐานการพัฒนาการของสิ่งมีชีวิตตามหลักของธรรมชาติที่ว่าด้วย ทฤษฎีการคัดเลือกตามธรรมชาติ และทฤษฎีการสืบทอดลักษณะทางพันธุกรรมของยีน ดังแสดงในภาพที่ 15-1



ภาพที่ 15-1 ลักษณะของยีนในพันธุกรรม (ที่มาจาก: http://en.wikipedia.org)

Genetic Algorithm เป็นการรวมกันระหว่างคำว่า Genetic และ Algorithm ซึ่งแต่ละคำมีความหมายดังนี้

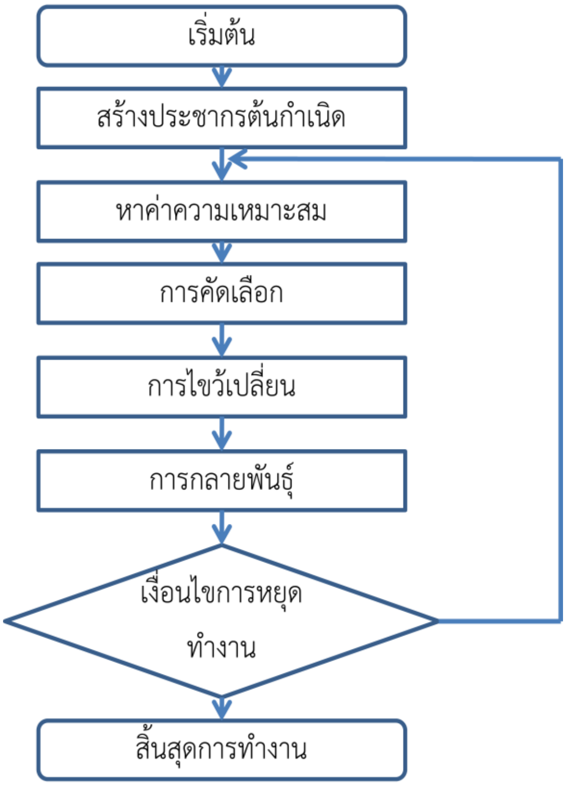
Genetic คือ วิชาพันธุศาสตร์ที่ว่าด้วยการศึกษากระบวนการถ่ายทอดลักษณะทางพันธุกรรมจากบรรพบุรุษสู่รุ่นลูกรุ่นหลานโดย โครโมโซมจะเป็นตัวแทนในการถ่ายทอด โครโมโซมคือสายรหัสของ DNA ประกอบไปด้วยยีนซึ่งแต่ละยีนจะมีการบ่งบอกลักษณะพิเศษเอาไว้ เช่น สีตา สีผม ซึ่งโครโมโซมเหล่านี้มีบทบาทอย่างมากในกระบวนการถ่ายทอดลักษณะทางพันธุกรรมจากรุ่นหนึ่งไปยังรุ่นหนึ่ง ซึ่งจะมีปรากฏการณ์ธรรมชาติเกิดขึ้นในระหว่างการถ่ายทอดลักษณะทางพันธุกรรมทำให้เกิดการเปลี่ยนแปลงในแต่ละรุ่น ซึ่งจะมีกระบวนการที่เกิดขึ้นเหล่านี้ ทำให้เกิดวิวัฒนาการ ซึ่งจะเกิดปรากฎการณ์ Crossover การข้ามสายพันธ์ และ mutation การกลายพันธ์เป็นกระบวนการเปลี่ยนแปลงที่สำคัญที่เรียกว่า วิวัฒนาการ Evolution ที่หมายถึง กระบวนการเปลี่ยนแปลงหรือคลี่คลายไปสู่ฐานะที่ดีขึ้นหรือเจริญขึ้น เป็นการเปลี่ยนแปลงในทางชีววิทยาจากสิ่งที่ง่ายๆ ไปสู่สิ่งที่ซับซ้อนมากขึ้น หรือซับซ้อนน้อยลง เพื่อการดำรงอยู่ตามความเหมาะสมของสิ่งมีชีวิตชนิดนั้น การเปลี่ยนแปลงนี้จะต้องเปลี่ยนในลักษณะค่อยเป็นค่อยไป และต้องใช้เวลานานในการถ่ายทอดลักษณะทางพันธุกรรมจะมีกระบวนการที่ทำให้เกิดการเปลี่ยนแปลงที่เรียกว่า วิวัฒนาการ นั้นคือ กระบวนการ Selection, Crossover และ Mutation โดยคำนวณหาค่าความเหมาะสม Fitness Function ที่สอดคล้องกับวัตถุประสงค์ของปัญหา Objective Function กำหนดให้กับโครโมโซมแต่ละตัว เพื่อนำไปสู่กระบวนการคัดเลือก แต่บางปัญหาไม่สามารถที่จะคำนวณหาค่าเพื่อให้ได้ค่าความเหมาะสม ที่ตรงกับความจริงที่โครโมโซมนั้นควรจะได้รับ และสิ่งนี้จึงทำให้เกิดปัญหาขึ้น ดังนั้นในหารกำหนดค่าความเหมาะสมของโครโมโซมแต่ละตัว จึงเป็นสิ่งที่สำคัญ และจะต้องถูกประมาณค่าที่ดี เพราะถ้าหากค่าความเหมาะสมสูงกว่าความเป็นจริงที่โครโมโซมนั้นควรได้รับ อาจทำให้เกิดความเสียหายในหารสืบค้นข้อมูลในแต่ละครั้ง และครั้งต่อไปได้ซึ่งในที่นี้หมายความว่า สมการหรือฟังก์ชันที่ใช้ในการหาค่าความเหมาะสมให้กับโครโมโซมแต่ละตัวไม่สามารถกำหนดได้แน่นอน และชัดได้ดีเท่ากับค่าความจริงที่คำนวณได้จากมนุษย์หรือผู้ใช้ เช่น ทางด้านศิลปะ เพลงสถาปัตยกรรม และการออกแบบ เป็นต้น ผู้ใช้สามารถให้ค่าความจริงว่าสิ่งที่ผู้ใช้เห็น หรือได้ยินนั้นอยู่ในค่าระดับใด

Algorithm คือ วิธีคิดหรือชุดคำสั่งที่มีการเรียงลำดับขั้นตอนด้วยกระบวนการทางคณิตศาสตร์เพื่อแก้ปัญหา และปัญหามีหลายแบบจึงทำให้มีวิธีการแก้ปัญหาที่เหมาะกับแต่ละแบบ

ดังนั้น Genetic Algorithm สองคำนี้ได้มารวมกันจึงเป็นวิธีการค้นหาและแก้ปัญหาเพื่อได้สิ่งที่เหมาะสมที่สุด Optimum points โดยพัฒนาและจำลองมาจากกระบวนการทางพันธุกรรม คือ ทฤษฏีการอยู่รอดของสิ่งมีชีวิต Genetic Algorithm เป็นทางเลือกที่ใช้แก้ปัญหาที่ดีเพราะวิธีการใช้แก้ปัญหาเป็นวิธีการแบบสุ่มและช่วยแก้ปัญหาที่มีขนาดใหญ่และซับซ้อนได้เนื่องจากมีคุณสมบัติการเรียนแบบการถ่ายทอดทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่เหมาะสมที่สุดจากประชากรรุ่นก่อนมาใช้ในการพิจารณาในการหาคำตอบของประชาชนรุ่นถัดมา ซึ่งจะนำถ่ายทอดลักษณะทางพันธุกรรมจากบรรพบุรุษ

ไปสู่รุ่นลูกหลานโดยใช้ค่า Fitness Function ที่สอดคล้องกับ Objective Function ในการพิจารณาหาคำตอบโดยมีการพิจารณาว่า โครโมโซมใดควรที่จะนำมาสืบสายพันธุ์ หรือไม่ควรนำมาสืบสาย จะทำให้สามารถหาคำตอบที่มีค่าสูงสุดหรือต่ำสุดที่สมบูรณ์ได้

15.2 องค์ประกอบขั้นตอนวิธีเชิงพันธุกรรม



ภาพที่ 15-2 ขั้นตอนวิธีเชิงพันธุกรรม

ภาพที่ 15-2 แสดงขั้นตอนการทำงานของอัลกอริทึมเชิงพันธุกรรมโดยสามารถแบ่งออกเป็นขั้นตอนหลักๆ ได้ 6 ขั้นตอน

15.2.1 การสร้างประชากรต้นกำเนิด

สิ่งที่ควรพิจารณาเป็นอันดับแรกคือการเข้ารหัสโครโมโซม (Chromosome Encoding) ซึ่งเป็นวิธีการที่ใช้ในการนำเสนอทางเลือกที่สามารถจะเป็นไปได้ของแต่ละปัญหา รูปแบบโครโมโซมที่ได้จากปัญหา ในการถอดรหัสนั้นจะขึ้นอยู่กับปัญหา และในปัจจุบันปัญหามีมากมาย จึงทำให้รูปแบบของโครโมโซมมีความแตกต่างกันออกไปตามปัญหานั้นๆ

- Binary Encoding ทุกตำแหน่งของยีนบนโครโมโซมจะมีค่าเป็นบิต 0 หรือ 1 ตัวอย่างเช่น

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

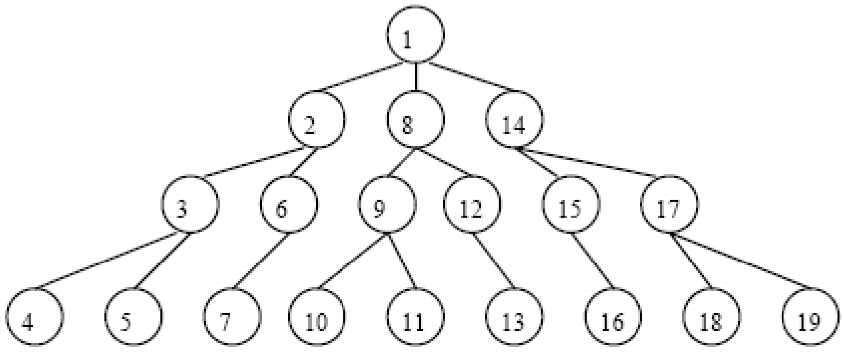
- Value Encoding หรือ Directทุกตำแหน่งของยีนบนโครโมโซมจะมีค่าบางค่า ที่สามารถเชื่อมโยงไปยังปัญหาได้ เช่น ตัวอักษร จำนวนจริง คำสั่ง หรืออื่นๆ รูปแบบโครโมโซมแบบนี้สามารถใช้ได้กับปัญหาที่ค่อนข้างซับซ้อนค่าได้

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGTYTGABA
Chromosome C	(back), (back), (right), (forward), (left)

- Permutation Encoding เป็นการกระทำดับแรกก่อนที่จะเข้ากระบวนการของ Genetic Algorithm ประชากรที่เกิดจากการสุ่ม Random เพื่อนำประชากรเข้าไปในกระบวน ในการสุ่มจะต้องสุ่มให้ได้จำนวนเท่ากับขนาดของรุ่นที่ได้กำหนดไว้โดยที่ยังไม่มีการสนใจค่าความเหมาะสมของแต่ละโครโมโซม B: 9 5 2 1 4 6 7 8 3

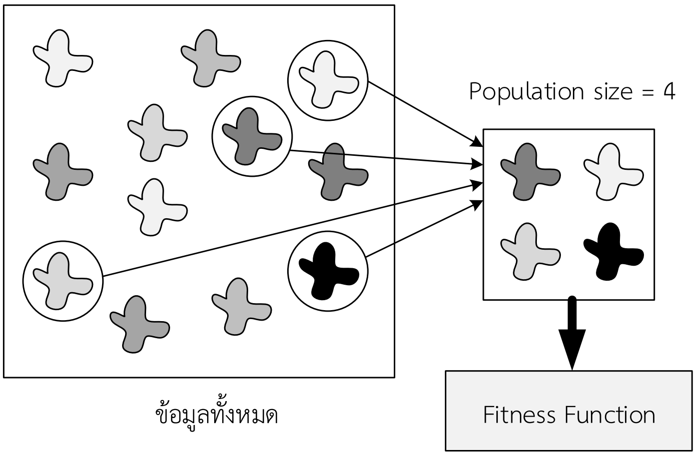
Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

- Tree Encoding ทุกตำแหน่งของยีนจะเป็น node ของต้นไม้ดังภาพที่ 15-3



ภาพที่ 15-3 การเข้ารหัสโครโมโซมแบบต้นไม้

เมื่อหาวิธีเข้ารหัสโครโมโซมได้แล้ว จากนั้นจะทำการสร้างประชากรต้นกำเนิด ซึ่งหมายถึงลักษณะที่เป็นต้นแบบหรือต้นกำเนิดที่จะนำเข้าไปในกระบวนการถ่ายทอดลักษณะทางพันธุกรรม โดยการสุ่มเลือกเรื่องสร้างประชาการต้นแบบขึ้นมาเพื่อใช้เป็นจุดเริ่มต้นของขั้นตอนการวิวัฒนาการขั้นตอนนี้จะเป็นขั้นตอนแรกที่เกิดขึ้นก่อนที่จะเริ่มเข้ากระบวนการของ จีนเนติกอัลกอริทึม โดยประชากรกลุ่มแรก หรือประชากรต้นกำเนิด จะเกิดจากการสุ่มเลือกขึ้นมาจาก กลุ่มของประชากรทั้งหมดที่มีอยู่ โดยในการสุ่มเลือกจะทำการสุ่มตามจำนวนของประชากรที่ได้กำหนดไว้เป็นพารามิเตอร์ของอัลกอริทึม แสดงดังภาพที่ 15-4



ภาพที่ 15-4 การสุ่มหาประชากรเริ่มต้นจำนวน 4 โครโมโซม

15.2.2 การหาค่าความเหมาะสม

Fitness Function คือ เป็นวิธีการสำหรับประเมินค่าความเหมาะสม เพื่อให้คะแนนแต่ละทางเลือกของคำตอบต่างๆอย่างเหมาะสม โครโมโซมทุกตัวจะมีค่าความเหมาะสมของตังเองเพื่อใช้สำหรับพิจารณาว่า โครโมโซมตัวนั้น เหมาะสมหรือไม่ ที่จะนำมาใช้ในการสืบทอดพันธุกรรมสำหรับสร้างโครโมโซมรุ่นใหม่ โดยวิธีการสำหรับคิดค่าความเหมาะสมนั้น จะใช้สมการที่สอดคล้องกับแต่ละปัญหา เช่น กำหนดให้ค่าความเหมาะสม = จำนวนของบิต 1 ทั้งหมดในโครโมโซม A: 100011100 ดังนั้นโครโมโซม A มีค่าความเหมาะสมเท่ากับ 4 หลังจากประเมินค่าความเหมาะสมของแต่ละโครโมโซมเรียบร้อยแล้ว จะเป็นขั้นตอนการปรับเปลี่ยนองค์ประกอบของโครโมโซมตามวิธีเชิงพันธุกรรม (Genetic operations) ได้แก่ การคัดเลือก (Selection), การไขว้เปลี่ยน (Crossover) และ การกลายพันธุ์ (Mutation)

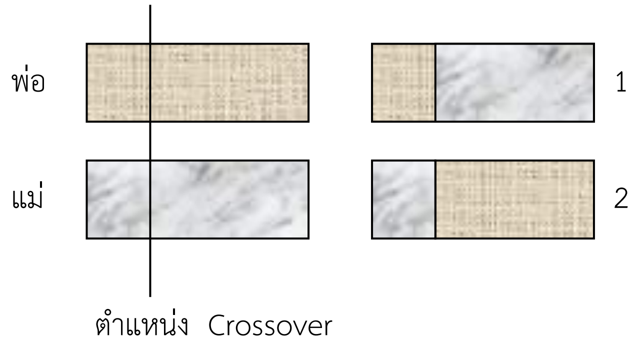
15.2.3 การคัดเลือก

เพื่อให้เกิดการอยู่รอดของสิ่งมีชีวิตนั้น โดยคัดเลือกมาเป็นโครโมโซมพ่อและโครโมโซมแม่หรือที่เรียกว่า Parent ในการสืบสายพันธ์ ทำให้เกิดปัญหาว่าจะทำอย่างไรให้เกิดจากคัดเลือกโครโมโซมที่น่าพอใจเพื่อที่จะเกิดการอยู่รอดของสิ่งมีชีวิตตามทฤษฎีของ Charles Darwin จึงทำให้เกิดรูปแบบมากมายในการเลือกโครโมโซมที่น่าพอใจที่สุดเพื่อนำไปสืบสายพันธุ์ทำให้เกิดรูปแบบการคัดเลือกมากมายเพื่อให้เกิด

ผลลัพธ์น่าพอใจที่สุด เช่น การคัดเลือกแบบ Roulette Wheel, การคัดเลือกแบบ Ranking, การคัดเลือกแบบ Tournament, การคัดเลือกแบบ Elitist, การคัดเลือกแบบ Steady-state และอื่นๆมากมายหลายวิธีเพื่อให้ได้มาซึ่งวิธีการคัดเลือกโครโมโซมที่ดี

15.2.4 การไขว้เปลี่ยน

การไขว้เปลี่ยน (Crossover) เป็นกระบวนการที่สำคัญของ Genetic Algorithm เมื่อเกิดการ Crossover ขึ้นในทางพันธุศาสตร์จะทำให้เกิดการเปลี่ยนแปลงของสิ่งมีชีวิตที่หลากหลาย ซึ่งการ Crossover จะต้องอาศัยวิวัฒนาการเป็นเวลานาน จึงสามารถเลือกเอาคำตอบที่เหมาะสมกับความต้องการได้มากที่สุดและ ขั้นตอนในการ Crossover คือ นำ 2 โครโมโซม Parent มาผสมกันเพื่อให้ได้โครโมโซมใหม่ขึ้นมา จากนั้นใช้วิธีการที่ง่ายที่สุด คือ สุ่มตำแหน่ง Crossover และทำการคัดลอกทุกอย่างที่อยู่หน้าตำแน่ง Crossover ของพ่อและคัดลอกทุกอย่างหลังตำแหน่ง Crossover ของแม่รวมกันจะได้ลูกตัวที่ 1 ออกมา จากนั้นทำการคัดลอกทุกอย่างที่อยู่หน้าตำแหน่ง Crossover ของแม่ และคัดลอกหลังตำแหน่ง Crossover ของพ่อรวมกันจะได้ลูกตัวที่ 2 ออกมาดังแสดงในภาพที่ 15-5



ภาพที่ 15-5 ตัวอย่างการไขว้เปลี่ยนของโครโมโซม

นอกจากการไขว้เปลี่ยนอย่างง่ายที่เรียกว่า การไขว้เปลี่ยนแบบจุดเดียวแล้ว ยังมีการไขว้เปลี่ยนแบบสองจุดอีกด้วย อีกทั้งยังการไขว้เปลี่ยนนั้นผู้พัฒนาโปรแกรมสามารถเลือกตำแหน่งอื่นๆ ในการไขว้เปลี่ยนได้ขึ้นอยู่กับการเข้ารหัสโครโมโซม ลักษณะของการไขว้เปลี่ยนจะสร้างตามลักษณะของปัญหาซึ่งสามารถปรับปรุงได้ตามประสิทธิภาพของจีเนติกอัลกอริทึมได้

15.2.5 การกลายพันธุ์

การกลายพันธุ์ (Mutation) จะเกิดขึ้นหลังจากการ Crossover เสร็จสิ้นจะทำการสุ่มประชากรเปลี่ยนแปลงผลที่ได้จากการ Crossover หมายความว่า รุ่นลูกที่เกิดจากผสมจากรุ่นพ่อแม่แล้วจึงนำรุ่นลูกมาดำเนินการ Mutation ต่อไป ซึ่งการ Mutation ทางพันธุศาสตร์จะทำให้ได้ลักษณะใหม่ๆ

เกิดขึ้น ขั้นตอนในการ Mutation เมื่อได้ตำแหน่ง Mutation แล้วเปลี่ยน แปลงค่า ณ ตำแหน่งที่สุ่มนั้น ตัวอย่างการกลายพันธุ์แสดงดังภาพที่ 15-6 โดยกำหนดโครโมโซมในรูปแบบ Binary จำนวน 6 ตำแหน่ง ซึ่งจะทำการสุ่มเลือกตำแหน่ง bit จำนวน 2 ตำแหน่ง (ตำแหน่งที่ 2 และ 5) และทำการเปลี่ยนแปลงค่าจาก 1 เป็น 0 หรือจาก 0 เป็น 1

ภาพที่ 15-6 การกลายพันธุ์ของโครโมโซม

15.2.6 เงื่อนไขการหยุดทำงาน

เมื่อได้โครโมโซมรุ่นใหม่แล้ว จะมาถึงเงื่อนไขการหยุดทำงานว่าควรจะมีการดำเนินการทางพันธุกรรมต่อหรือไม่ ถ้าต้องการดำเนินการต่อก็จะกลับไปยังขั้นตอนการคัดเลือกโครโมโซมอีกครั้ง แต่ถ้าต้องการหยุดก็จะนำโครโมโซมที่มีค่าความเหมาะสมดีที่สุดมาเป็นผลลัพธ์ โดยเงื่อนไขการหยุดทำงานนั้นสามารถแบ่งออกเป็น 2 แบบใหญ่ๆ คือ แบบแรกนั้นใช้วิธีการกำหนดจำนวนรอบที่ต้องการดำเนินการทางพันธุกรรม เช่น กำหนดว่าต้องการคำตอบที่ดีที่สุดของโครโมโซมรุ่นที่ 10000 ดังนั้นเจนีติกอัลกอริทึมก็จะวนการทำงานตามขั้นตอนไปทั้งหมด 10000 รอบและคืนโครโมโซมที่ดีที่สุดเป็นคำตอบ หรือแบบที่สองคือการกำหนดเงื่อนไขการหยุดทำงานจากค่าความเหมาะสมเช่น ถ้าได้ค่าเหมาะสมในช่วงที่ยอมรับได้ก็ให้หยุดการทำงานทันที

15.3 ตัวอย่างการแก้ปัญหา 8 queen ด้วยอัลกอริทึมเชิงพันธุกรรม

ปัญหา 8-queen คือปัญหาที่ต้องการวางตัวหมาก queen ของหมากรุกลงบนตารางหมากรุกขนาด 8 x 8 โดยที่ queen แต่ละตัวจะต้องไม่ทับตาเดินซึ่งกันและกัน (queen จะสามารถเดินได้ 8 ทิศด้วยระยะทางเท่าไรก็ได้) โดยมีการกำหนดพารามิเตอร์ต่างๆ ของอัลกอริทึมเชิงพันธุกรรมดังนี้

- ใช้เลขจำนวนเต็มในข้อมูลของโครโมโซม
- ใช้วิธีไขว้เปลี่ยนแบบจุดเดียว (single point crossover)
- ขนาดของประชากรเป็น 4
- อัตราการไขว้เปลี่ยนเป็น 1 (หมายถึงไขว้เปลี่ยนทุกๆครั้ง)
- อัตราการกลายพันธุ์เป็น 1/8 = 0.125 (จะมีอย่างมาก 1 ข้อมูลในโครโมโซมที่ถูกกลายพันธุ์)

15.3.1 การสร้างประชากรเริ่มต้น

กำหนดให้โครโมโซมเข้ารหัสเป็นตัวเลขจำนวนเต็มที่มีค่าตั้งแต่ 1 ถึง 8 โดยค่าที่กล่าวถึงคือตำแหน่งของแถวที่วาง queen และแต่ละโครโมโซมมีความยาวของโครโมโซมเท่ากับ 8 (ตามจำนวนหลักของตารางหมากรุก) ภาพที่ 15-7 แสดงตัวอย่างการเข้ารหัสโครโมโซมของปัญหา 8-queen

แถวที่ \หลักที่	1	2	3	4	5	6	7	8
8					Q			
7			Q					
6								
5		Q				Q	Q	
4				Q				
3								Q
2	Q							
1								

โครโมโซม	2	4	7	4	8	5	5	2

ภาพที่ 15-7 ตัวอย่างการเข้ารหัสโครโมโซมของปัญหา 8-queen

จากพารามิเตอร์ที่กำหนดขนาดของประชากรทั้ง 4 ประชากร ซึ่งหมายถึงจะทำการสุ่มโครโมโซมออกมาทั้งหมด 4 โครโมโซม กำหนดให้โครโมโซมที่ได้จากการสุ่มแสดงดังภาพที่ 15-8

โครโมโซม A	2	4	7	4	8	5	5	2
โครโมโซม B	3	2	7	5	2	4	1	1
โครโมโซม C	2	4	4	1	5	1	2	4
โครโมโซม D	3	2	5	4	3	2	1	3

ภาพที่ 15-8 ตัวอย่างโครโมโซมทั้ง 4 โครโมโซมที่ได้จากการสุ่ม

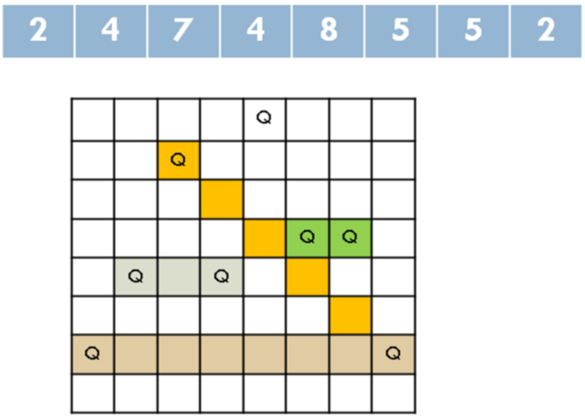
15.3.2 การหาค่าความเหมาะสม

หลังจากทำการสุ่มประชากรต้นกำเนิดจำนวน 4 โครโมโซมแล้ว จะทำการคำนวณค่าความเหมาะสมของแต่ละโครโมโซม ซึ่งสำหรับปัญหา 8-queen นั้น จะใช้คำนวณหาค่าความเหมาะสมจาก Fitness function คือ

$$Q(n) = 28 - n$$

โดยที่ 28 มาจากจำนวนเส้นทางเดินที่ทับกันระหว่าง queen มากที่สุดที่เมื่อวาง queen ครบ 8 ตัว และ n คือจำนวนเส้นทางเดินที่ทับกันระหว่าง queen ของโครโมโซมนั้นๆ ซึ่งหมายความว่าถ้าสามารถวาง queen ครบทั้ง 8 ตัวโดยไม่ทับเส้นทางกันเลยค่า Q(n) จะเท่ากับ 28 – 0 = 28 ซึ่งเป็นค่าความเหมาะสมผลลัพธ์ของปัญหาที่ต้องการ

โครโมโซม A



ภาพที่ 15-9 จำนวนเส้นทางที่ทับกันของ queen ทั้ง 8 ตัว

จากภาพที่ 15-9 แสดงการคำนวณค่าความเหมาะสมของโครโมโซม A โดยจะเห็นได้ว่ามีเส้นทางการเดินทับกันของ queen ทั้งหมดเป็นจำนวน 4 เส้นทาง ดังนั้นค่าความเหมาะสมของโครโมโซม A ก็คือ $Q_A = 28 - 4 = 24$ ในกรณีเดียวกันเมื่อคำนวณค่าความเหมาะสมของโครโมโซมทั้ง 3 ตัวที่เหลือจะได้ค่าความเหมาะสมดังนี้ $Q_B = 23$, $Q_C = 20$ และ $Q_D = 11$

15.3.3 การคัดเลือกโครโมโซม

ขั้นตอนการเลือกโครโมโซมนั้นวิธีที่ง่ายที่สุดคือ Roulette Wheel ซึ่งเป็นการสุ่มเพียงโดยมีการคำนึงถึงค่าความเหมาะสมสูงนั้น โครโมโซมที่มีค่าความเหมาะสมสูงนั้นจะทำให้มีโอกาสถูกคัดเลือกขึ้นมามากกว่าโครโมโซมที่มีค่าความเหมาะสมน้อย ตัวอย่างการคัดเลือกโครโมโซมแสดงดังภาพที่ 15-10

โครโมโซม A	2	4	7	4	8	5	5	2	Q_A = 24 ร้อยละ 31
โครโมโซม B	3	2	7	5	2	4	1	1	Q_B = 23 ร้อยละ 29
โครโมโซม C	2	4	4	1	5	1	2	4	Q_C = 20 ร้อยละ 26
โครโมโซม D	3	2	5	4	3	2	1	3	Q_D = 11 ร้อยละ 14

ภาพที่ 15-10 การคำนวณร้อยละของโครโมโซมเพื่อการคัดเลือก

การคำนวณร้อยละของโครโมโซมที่จะถูกคัดเลือกตามสัดส่วนของค่าความเหมาะสม จะเห็นได้ว่าร้อยละของโครโมโซม A, B, C, และ D คือ 31, 29, 26 และ 14 ตามลำดับ ตัวอย่างการคำนวณร้อยละของโครโมโซม A คือ นำเอาค่าความเหมาะสมของโครโมโซม A (24) หารด้วยผลรวมค่าความเหมาะสมของทุกโครโมโซม (24 + 23 + 20 + 11 = 78) แล้วนำได้คูณกับ 100 ซึ่งก็คือ (24 / 78) * 100 = 30.77 ก็ประมาณร้อยละ 31

เมื่อได้ค่าร้อยละที่มีโอกาสสุ่มเลือกของแต่ละโครโมโซม ขั้นตอนการคัดเลือกก็จะสุ่มเลือกโครโมโซมขึ้นมาตามที่กำหนดไว้ในพารามิเตอร์ ซึ่งตัวอย่างได้กำหนดประชากรเป็น 4 และอัตราการไข้วเปลี่ยนเป็น 1 ดังนั้นจึงต้องสุ่มโครโมโซมขึ้นมาทั้งหมด 2 คู่ (4 โครโมโซม) ข้อเตือนใจคือไม่จำเป็นว่าโครโมโซมที่มีค่าความเหมาะมากที่สุดจะถูกเลือกขึ้นมาเสมอ เพียงแต่ว่าโอกาสที่จะถูกเลือกขึ้นมาสูงกว่าโครโมโซมอื่น ที่เป็นแบบนี้ก็เพราะว่าบางครั้งโครโมโซมที่ให้ค่าความเหมาะสมไม่ดีนักเมื่อไขว้เปลี่ยนกันอาจจะเกิดโครโมโซมที่ดีขึ้นก็เป็นได้ ซึ่งอันที่จริงก็คือเป็นการกันไม่ให้คำตอบลู่เข้าหาจุดดีสุดแบบท้องถิ่น

กำหนดให้โครโมโซม 2 คู่ที่ได้สุ่มคัดเลือกมาคือ คู่โครโมโซม B, A และคู่โครโมโซม B และ C ดังภาพที่ 15-11

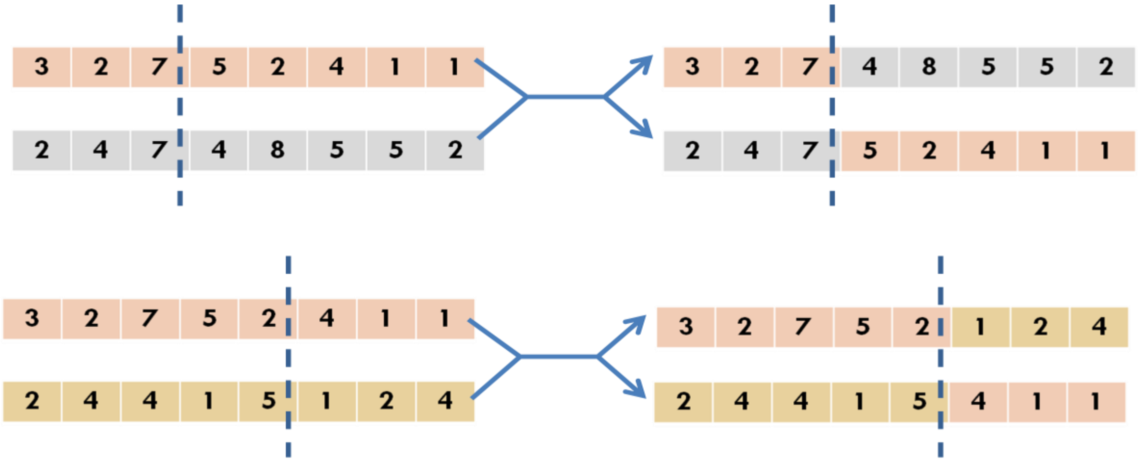
โครโมโซม B	3	2	7	5	2	4	1	1
โครโมโซม A	2	4	7	4	8	5	5	2

โครโมโซม B	3	2	7	5	2	4	1	1
โครโมโซม C	2	4	4	1	5	1	2	4

ภาพที่ 15-11 โครโมโซม 2 คู่ที่ผ่านการคัดเลือก

15.3.4 การไขว้เปลี่ยน

ตัวอย่างได้กำหนดการไขว้เปลี่ยนแบบจุดเดียวก็คือ การสุ่มตำแหน่งในโครโมโซมออกมาเพื่อสร้างประชากรใหม่ โดยโครโมโซมที่เกิดใหม่นั้นจะมี 2 โครโมโซม โดยโครโมโซมที่เกิดใหม่ตัวแรกนั้นจะมีค่าของส่วนก่อนจุดสุ่มเหมือนโครโมโซมพ่อ และส่วนก่อนจุดสุ่มเหมือนโครโมโซมแม่ สำหรับโครโมโซมที่เกิดใหม่ตัวที่สองจะม่ค่าของส่วนก่อนจุดสุ่มเหมือนโครโมโซมแม่ และส่วนก่อนจุดสุ่มเหมือนโครโมโซมพ่อ

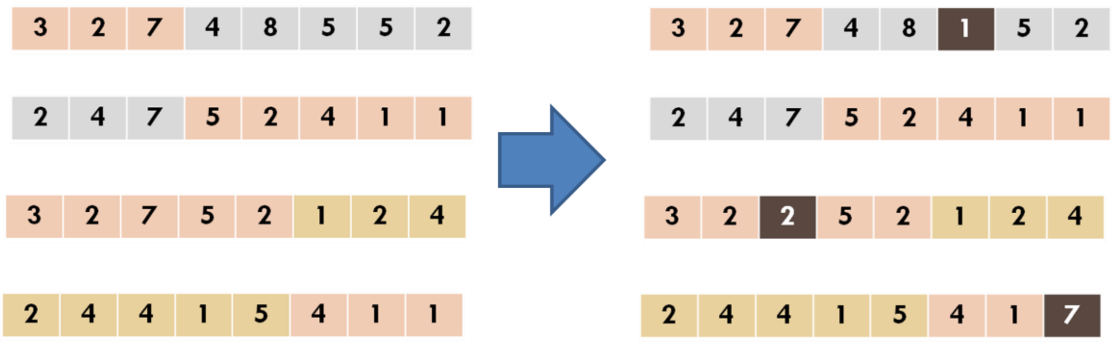


ภาพที่ 15-11 การไขว้เปลี่ยนโครโมโซมทั้ง 2 คู่

จากภาพที่ 15-11 แสดงให้เห็นว่าโครโมโซมคู่แรกมีจุดตัดอยู่ที่ตำแหน่งที่ 3 และโครโมโซมคู่ที่สองมีจุดตัดที่ตำแหน่งที่ 5 ซึ่งการไขว้เปลี่ยนนี้ทำให้เกิดประชากรใหม่ขึ้นมา 4 โครโมโซม

15.3.5 การกลายพันธุ์

อัตราการกลายพันธุ์ได้กำหนดไว้เป็น 1/8 หมายถึงจะมีอย่างมาก 1 ข้อมูลในโครโมโซมที่ถูกกลายพันธุ์ ภาพที่ 15-12 แสดงการกลายพันธุ์ซึ่งเกิดจากการสุ่มตำแหน่งในโครโมโซมที่จะกลายพันธุ์ และค่าใหม่ที่จะแทนที่ก็เกิดจากการสุ่มเช่นเดียวกัน



ภาพที่ 15-12 การกลายพันธุ์ของโครโมโซมใหม่

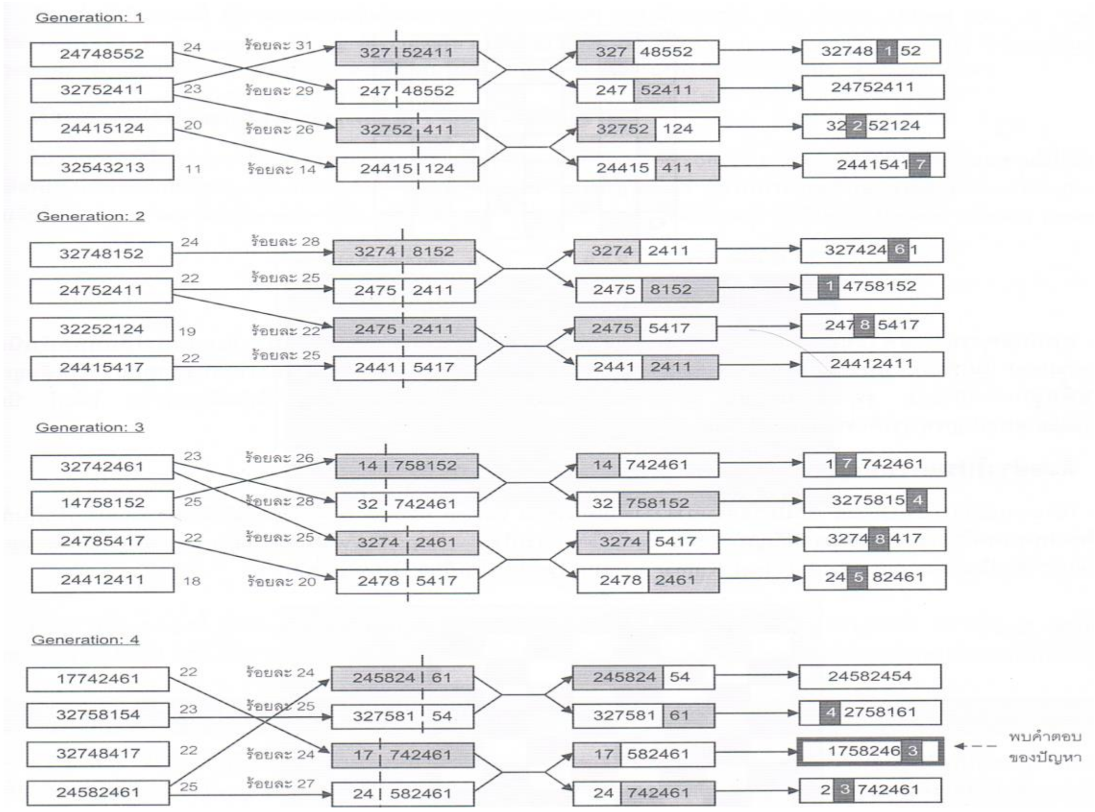
โดยจากภาพจะเห็นได้ว่าโครโมโซมใหม่

- ตัวที่ 1 นั้นได้มีการกลายพันธุ์ในตำแหน่งที่ 6 โดยเปลี่ยนค่าจาก 5 เป็น 1
- ตัวที่ 2 นั้นไม่มีการกลายพันธุ์
- ตัวที่ 3 นั้นได้มีการกลายพันธุ์ในตำแหน่งที่ 3 โดยเปลี่ยนค่าจาก 7 เป็น 2
- ตัวที่ 4 นั้นได้มีการกลายพันธุ์ในตำแหน่งที่ 8 โดยเปลี่ยนค่าจาก 1 เป็น 7

15.3.6 การตรวจสอบเงื่อนไขและทำซ้ำ

จาก 5 ขั้นตอนข้างต้นถือว่าครบการทำงาน 1 รอบของอัลกอริทึมเชิงพันธุกรรม ปกติจะเรียกว่า 1 รุ่น หรือ 1 generation เมื่อครบรอบ 1 generation ก็จะทำการตรวจสอบเงื่อนไข การหยุดทำงานแล้วแต่ผู้พัฒนาโปรแกรมกำหนด หยุดตามรอบหรือหยุดเมื่อได้ค่าความเหมาะสมที่ต้องการ จากตัวอย่างนี้เนื่องจากเป็นปัญหา 8-queen ดังนั้นการหยุดการทำงานของอัลกอริทึมเชิงพันธุกรรม ควรจะหยุดเมื่อได้โครโมโซมที่มีค่าเหมาะสมเป็น 28 (ไม่มีการทับกันของเส้นทางเดินของ queen ทั้ง 8 ตัว)

ภาพที่ 15-13 แสดงการทำงานของอัลกอริทึมเชิงพันธุกรรมในแต่ละ generation โดยการสุ่มและสามารถพบคำตอบได้ใน generation ที่ 4 ซึ่งโครโมโซมที่เป็นคำตอบคือ 1|7|5|8|2|4|6|3 และสามารถนำไปใช้ในการวาง queen ทั้ง 8 ตัวบนกระดานหมากรุกได้ดังภาพที่ 15-14



ภาพที่ 15-13 การทำงานของอัลกอริทึมเชิงพันธุกรรมในแต่ละ generation

(ที่มาจาก : ณัฐพงษ์ วารีประเสริฐ. (2552). ปัญญาประดิษฐ์, กรุงเทพฯ: เคทีพี คอมพ์ แอน คอนซัลท์)

			Q				
	Q						
						Q	
		Q					
					Q		
							Q
				Q			
Q							

1	7	5	8	2	4	6	3

ภาพที่ 15-14 คำตอบของปัญหา 8-queen ด้วยวิธีอัลกอริทึมเชิงพันธุกรรม

15.4 สรุป

ปัจจุบันการวิเคราะห์ข้อมูลมีหลายวิธี ซึ่งแต่ละวิธีก็จะให้ผลลัพธ์การวิเคราะห์ที่แตกต่างกันไป ตามความสัมพันธ์ของสมการ ซึ่งถ้าหากกำหนดรูปแบบสมการไม่ถูกต้อง ผลลัพธ์ที่ได้อาจมีความคลาดเคลื่อนจากความเป็นจริงซึ่งส่งผลให้การวิเคราะห์ข้อมูลเกิดความผิดพลาดขึ้นได้ ดังนั้นจึงมีแนวคิดวิธีการของอัลกอริทึมเชิงพันธุกรรม เพื่อเป็นทางเลือกในการแก้ปัญหาความผิดพลาดในการวิเคราะห์ข้อมูล โดยได้มีการนำกระบวนการอัลกอริทึมเชิงพันธุกรรมซึ่งเป็นความรู้เกี่ยวกับทฤษฎีทางธรรมชาติมาประยุกต์ใช้ เพื่อหาคำตอบที่เหมาะสมและตรงกับความสัมพันธ์ของข้อมูล โดยการเริ่มสร้างประชากรต้นกำเนิดตามรูปแบบโครโมโซมที่ได้กำหนดไว้ เมื่อได้ประชากรต้นกำเนิดแล้วก็ทำการวัดค่าความเหมาะสมของแต่ละโครโมโซม โครโมโซมทุกตัวที่ได้กำหนดไว้จะถูกนำมาทำขั้นตอน Selection เพื่อคัดเลือกเข้าสู่กระบวนการ Genetic Operator โดยทำการคัดเลือกเอาเฉพาะโครโมโซมที่มีค่าความเหมาะสม Fitness Function เพื่อที่จะได้โครโมโซมที่ยอมรับได้ตามที่ได้กำหนดเอาไว้ โครโมโซมที่คัดเลือกไว้นั้นจะถูกนำมาทำการ Crossover การข้ามสายพันธ์ และ Mutation การกลายพันธุ์ ได้เป็นโครโมโซมชุดใหม่ ซึ่งเราจะนำโครโมโซมชุดใหม่นี้มาดำเนินการต่อไปจนสิ้นสุดตามเงื่อนไขที่ได้กำหนดไว้ ก็จะได้โครโมโซมที่มีค่าความเหมาะสมที่ดีที่สุด เป็นคำตอบของปัญหา

แบบฝึกหัด

1. จงอธิบายขั้นตอนการทำงานของอัลกอริทึมเชิงพันธุกรรมแบบคร่าวๆ
2. ถ้ากำหนดให้โครโมโซมมีความยาว 6 ตำแหน่ง แต่ละตำแหน่งเก็บค่าตัวเลขจำนวนเต็ม โดยมี Fitness function คือ Fitness = ผลรวมของค่าในแต่ละตำแหน่ง * 2 จงหาค่าความเหมาะสมของโครโมโซมต่อไปนี้

    2.1 4 | 3 | 2 | 1 | 6 | 2

    2.2 5 | 4 | 1 | 1 | 9 | 3

    2.3 4 | 2 | 2 | 5 | 8 | 7

    2.4 7 | 6 | 3 | 2 | 1 | 5

    2.5 9 | 3 | 2 | 1 | 5 | 7
3. จากโครโมโซมทั้ง 5 โครโมโซมในข้อที่ 2 จงคำนวณหาอัตราส่วนร้อยละของแต่ละโครโมโซมเพื่อใช้สำหรับการคัดเลือก
4. กำหนดคู่ของโครโมโซมดังต่อไปนี้

    โครโมโซม 1 : 4 | 3 | 2 | 1 | 5 | 2

    โครโมโซม 2 : 3 | 1 | 8 | 9 | 3 | 1

    จงเขียนโครโมโซมรุ่นลูกทั้ง 2 ตัวที่เกิดจากการไขว้เปลี่ยนของโครโมโซม 1 และโครโมโซม 2 ที่จุดตำแหน่งกึ่งกลางของโครโมโซม
5. จงคำนวณค่าความเหมาะสมของโครโมโซมรุ่นลูกในข้อ 4 เมื่อกำหนด Fitness function คือ ค่ามากที่สุดในโครโมโซม ลบกับค่าที่น้อยที่สุดในโครโมโซม

ภาคผนวก
การประยุกต์ใช้ปัญญาประดิษฐ์ในทางวิศวกรรม

ความรู้เบื้องต้นทางปัญญาประดิษฐ์สามารถนำไปประยุกต์ใช้ในงานทางด้านวิศวกรรมได้อย่างหลากหลาย โดยในภาคผนวกนี้ขอนำเสนอบทความวิจัยเกี่ยวกับการค้นหาเส้นทางเดินทางที่สั้นที่สุดของพนักงานขาย โดยได้ประยุกต์ใช้อัลกอริทึมเชิงพันธุกรรมหรือจีนเนติกอัลกอริทึม ในการแก้ไขปัญหา ซึ่งในงานวิจัยได้เน้นการใช้ GPU (Graphical Processing Unit) ผ่านกรอบการทำงานของ CUDA ในการแก้ไขปัญหาแทนที่ CPU (Central Processing Unit) เพื่อให้ใช้เวลาในการแก้ไขปัญหาที่สั้นลง

บทความวิจัยนี้ชื่อว่า “Traveling Salesman Problem Solver using Genetics Algorithm on CUDA Framework” และได้เผยแพร่ผ่านงานประชุมวิชาการนานาชาติ The 1st Engineer Science and Innovative Technology (ESIT 2014)

Traveling Salesman Problem Solver using Genetics Algorithm on CUDA Framework

Choopan Rattanapoka[1,*]

Abstract

This paper presents a design and implementation of a traveling salesman problem solver, which is a foundation problem in artificial intelligence field, program by using parallel genetics algorithm on graphical processing unit (GPU), Nvidia GTX560, via CUDA framework and compares the computing time between this implementation and a traditional genetic algorithm implementation computing by the central processing unit (CPU), Core i5-750. From the experiments, we found that the computation time using the solve traveling salesman problems on GPU is faster than on CPU around 4 times.

Keywords : Traveling Salesman Problem, Genetics Algorithm, CUDA, Java

1. Introduction

The Traveling Salesman Problem (TSP) is one of the most widely discussed problems in combinatorial optimization and it is also a foundation problem in artificial intelligence field. The problem involves a traveling salesman that wants to visit each of a set of cities exactly once, starting from and returning to his home city. TSP demands high computation time especially when the number of cities becomes larger.

A Genetic Algorithm (GA) is a simple and elegant heuristic for which many researchers, for example in [1] and [2], used to find a solution of TSP in reasonable amounts of time. However, the basic genetic algorithm was designed to work in sequential fashion. Thus, when GAs is applied to harder and bigger problems, there is an increase in the time required to find adequate solutions. As a consequences, there

have been multiple efforts to make GAs faster, and one of the most promising choices is to use parallel implementation such as in [3] and [4].

Currently, modern Graphic Processing Units (GPU) can be seen as low cost and very fast highly parallel general-purpose computing units due to the ability of these devices that can solve parallelizable problems much faster than traditional sequential processors. Moreover, several general purpose high-level languages for GPUs have become available such as CUDA and OpenCL and thus developers can easily develop applications using GPU. There are many researches and business products that use GPU to accelerate the computation time of the program such as in [5] and [6].

In this paper, we presents a design and implementation of a traveling salesman problem solver using parallel genetic algorithm on GPU via CUDA framework and compare the computation time used to find the solution and the result of solution with a traditional genetic algorithm on CPU.

2. Parallel Genetic Algorithm

A genetic algorithm has several distinctive features borrowed from an abstract formulation of biological systems. First, a representation of problem solutions called *genetic encoding* must be found which can be manipulated through some sort of genetic operations (*crossover*, and *mutation)* to yield other candidate solutions to the problem. Second, acting on an initial population, these transformations create the next *generation* of candidate solutions. Third, calculation of the objective function for each candidate solution supplies a measure of *fitness* which affects its likelihood of leaving surviving offspring in the next generation.

In [7], Erick Cantú-Paz presents a survey of parallel genetic algorithm which can be categorized into three main types: global single-population master-slave, single-population fine-grained, and multiple-population coarse-grained. Single-population fine-grained GAs are suited for massively parallel computers which also suit to our needs for using in GPU. The algorithm that we use to implement our program called *Cellular Genetic Algorithm*.

In a Cellular Genetic Algorithm, the population is usually structured in a regular grid of d dimension (d = 1, 2, 3) and the concept of small *neighborhood* is intensively used; this means that an individual may only interact with its nearby neighbors in the crossover step. The overlapped small neighborhoods helps in exploring the search space because the induced slow diffusion of solutions through the population provides a kind of exploration, while exploitation takes place inside each neighborhood by genetic operations.

3. Design and Implementation

In this paper, we use jCUDA which is a java bindings for CUDA, parallel programming paradigm by NVIDIA, to develop our TSP solver.

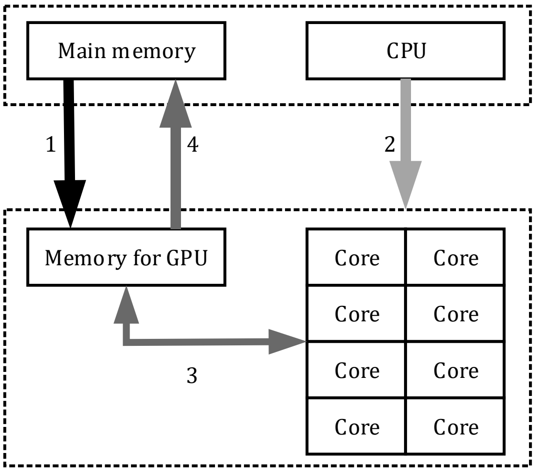


Fig. 1. Processing flow on CUDA

The GPU is especially well-suited to address problems that can be expressed as data-parallel computation (the same program is executed on many data elements in parallel). Hence, multi-thread programs can be speed up the computation when running on GPU.

The relation between CPU and GPU (processing flow on CUDA) has shown in Fig. 1. There are four main steps: 1) Copy data from main memory to GPU memory. 2) CPU instructs the process to GPU (running a GPU kernel program). 3) GPU executes program in parallel in each core also read data from and write data back to GPU memory. 4) Copy the result from GPU memory to main memory.

In our implementation, each individual (chromosome) is map on each GPU core. We also setup the layout of GPU core to 2 dimension like the layout of a 2 dimensional cellular genetic algorithm. Our cellular genetic algorithm has 4 steps: population initialization, crossover, mutation, and selection.

3.1 Population initialization

In this step, the individual solutions are randomly generated to form an initial population by CPU. Each chromosome contains the order and position of cities in the tour.

Then, CPU copies the generated data from main memory to GPU memory and instructs the process to GPU via kernel program. Thus, each GPU core will map on a different chromosome data in GPU memory.

3.2 Crossover

The crossover is a process of taking more than one chromosome (parent solutions) and producing offspring (child solutions) from them. Because we use cellular genetic algorithm, so each GPU core interacts with only its four neighbors (in Fig 2.) to do the crossover operation.

In our implementation, we use the partially matched crossover (PMX) operator [8] which has two crossover points selected randomly from the parent's chromosomes to produce the offspring. The two crossover points give a matching selection which is used to affect a cross through position by position exchange operations. The algorithm can be realized through four steps:

For example, consider two parents (P_1 and P_2):

P_1: 1 2 3 | 4 5 6 | 7 8 9

P_2: 1 5 6 | 8 2 9 | 3 7 4

First step: Generate two crossover points randomly. Here are at the 4th and 7th position.

Second step: Exchange genes in crossover region. The result are I_1 and I_2

I_1: *1 5 6* | 4 5 6 | *3 7 4*

I_2: *1 2 3* | 8 2 9 | *7 8 9*

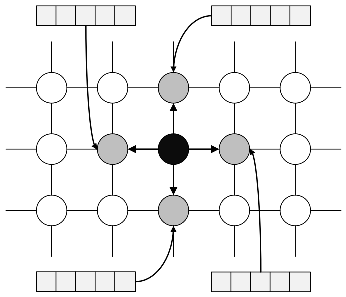


Fig. 2. Interaction between a GPU core and its neighbors in cellular genetic algorithm

Third step: Find the corresponding exchange rules in match region. Here are 4 ↔ 8, 5 ↔ 2, and 6 ↔ 9.

Fourth step: According to corresponding exchange rules in match region, genes in crossover region are exchanged. The result are C_1 and C_2 offspring.

C_1: 1 *2* *9* | 4 5 6 | 3 7 *8*

C_2: 1 *5* 3 | 8 2 9 | 7 *4* *6*

Thus, in the end of this step, each GPU core produces eight offspring.

3.3 Mutation

Mutation is a genetic operator used to make genetic diversity from one generation of a population of genetic algorithm chromosome to the next. The probability of mutation should be set low, so it means that not all of offspring has been mutated.

In our implementation, if the mutation occurs in a chromosome, we simply random two positions and then swap two genes of the chromosome in that position.

For example, the chromosome C

C: 1 2 *9* 4 5 6 *3* 7 8

Generate two positions randomly. Here are at the 3rd and 7th position. Then we obtain a new offspring C'

C': 1 2 *3* 4 5 6 *9* 7 8

3.4 Selection

After each GPU core applies genetic operators on its chromosome and obtains new eight offspring, we need to choose the best chromosome to survive for the next generation. We use *the distance of the tour* (Euclidian distance) as the fitness function to measure the quality of chromosomes. It means that the best chromosome has the shortest distance of the tour. We compare this fitness between the parent chromosome and its eight offspring and keep only the best of nine chromosomes to survive to the next generation.

Then, the genetic algorithm repeats the process of crossover, mutation, and selection until it reaches to the given number of generation.

4. Experiments

We do the experiments by solving TSP problems based on traditional genetic algorithm using CPU and cellular genetic algorithm based on GPU. Our test bed machine has Core i5-750 (8MB of cache, 2.66 GHz) CPU, 4096 MB of RAM, and NVIDIA Geforce GTX560. In our experiments, we measure the computation speed and the TSP's solution of both genetic algorithm running on CPU and GPU by adjusting the number of cities, the number of initial population and the number of generations.

4.1 Experiment 1

In this experiment, we fix the number of generations to 1000 and the number of initial population to 1024. Then, we vary the number of cities to 10, 20, 30, 40, and 50 to measure the computation speed of both genetic algorithm on CPU and GPU. From the experiment, the computation time (in Fig. 3) of the genetic algorithm run on GPU takes almost 4 times less to solve the problem than genetic algorithm run on CPU. However, the genetic algorithm on CPU has better solution than our genetic algorithm on GPU (in Fig. 4).

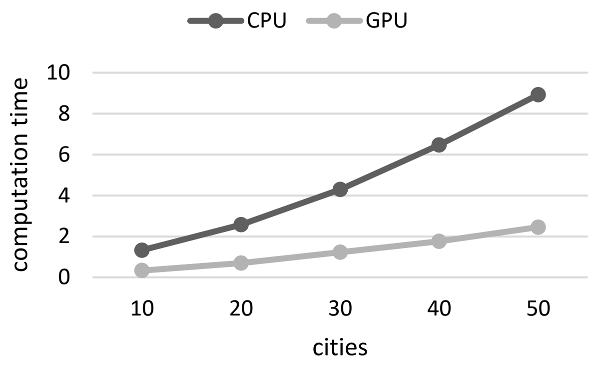


Fig. 3. The computation time of genetic algorithm run on CPU and GPU while adjusting the number of cities.

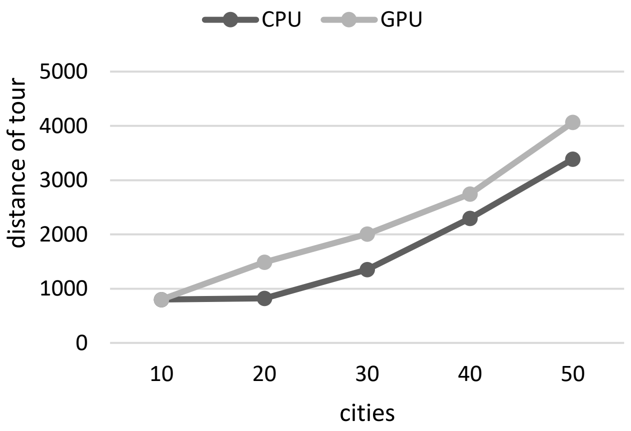


Fig. 4. The result of genetic algorithm run on CPU and GPU while adjusting the number of cities.

4.2 Experiment 2

In this experiment, we fix the number of generations to 1000 and the number of cities to 30. Then, we vary the number of initial population to 64, 128, 256, 512, 1024, and 2048 to measure the computation speed of both genetic algorithm on CPU and GPU. From the experiment, the computation time (in Fig. 5) of the genetic algorithm run on GPU takes almost 5 times less to solve the problem than the one on CPU. However, the genetic algorithm on CPU has better solution than the one on GPU (in Fig. 6).

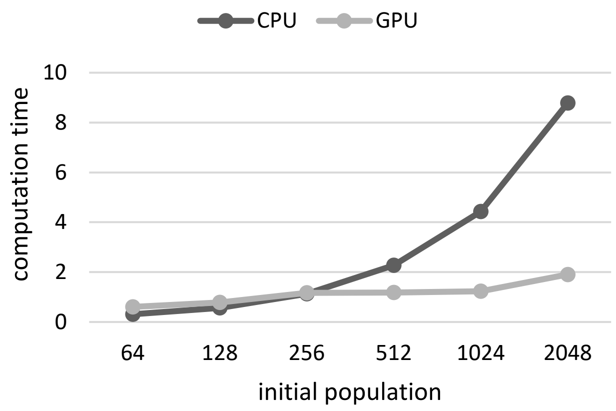


Fig. 5. The computation time of genetic algorithm run on CPU and GPU while adjusting the number of initial population.

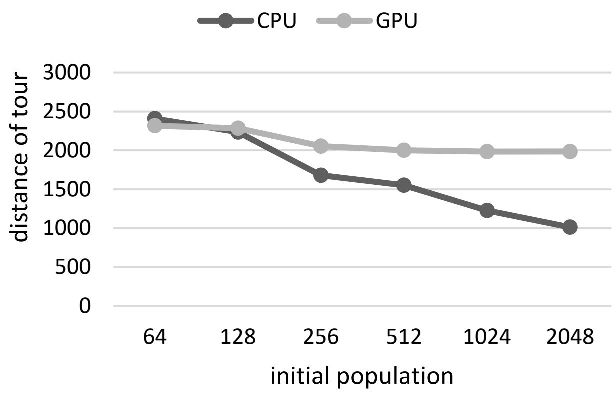


Fig. 6. The result of genetic algorithm run on CPU and GPU while adjusting the number of initial population.

4.3 Experiment 3

In this experiment, we fix the number cities to 30 and the number of initial population to 1024. Then, we vary the number of generations to 250, 500, 750, 1250, and 1500 to measure the computation speed of both genetic algorithm on CPU and GPU. From the experiment, the computation time (in Fig. 7) of the genetic algorithm run on GPU takes almost 4 times less to solve the problem than the one on CPU. However, the genetic algorithm on CPU has better solution than the one on GPU (in Fig. 8).

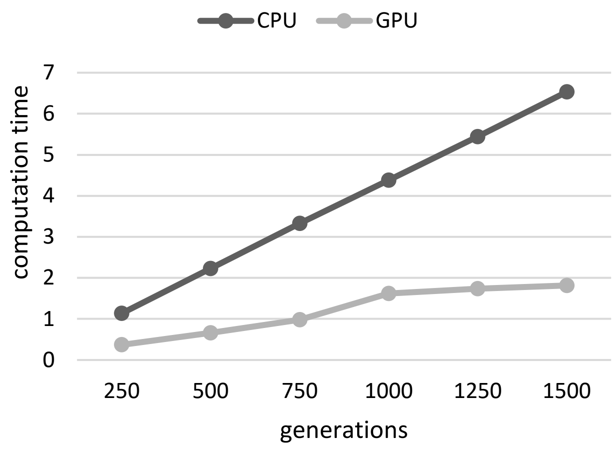


Fig. 7. The computation time of genetic algorithm run on CPU and GPU while adjusting the number of generations.

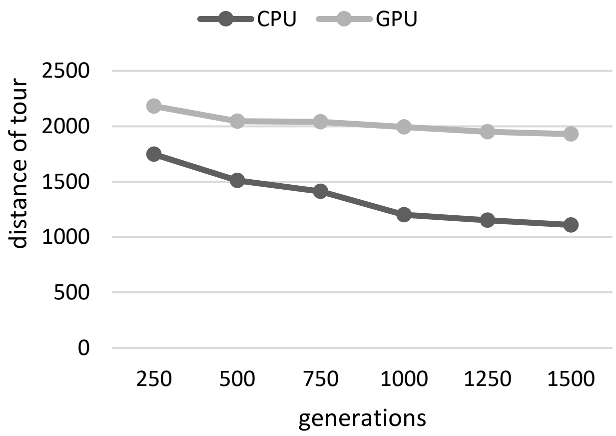


Fig. 8. The result of genetic algorithm run on CPU and GPU while adjusting the number of generations.

5. Conclusion

This paper presents a design and implementation of a traveling salesman problem solver by using cellular genetic algorithm model running on GPU via jCUDA, a java bindings for CUDA. From the experiments, we found that the computation time to solve the problem on GPU takes around four times less than the CPU's one. Especially, when we have a large number of initial population. The parallelism structure in GPU helps a lot in gaining speed of the computation. However, the result obtained from

GPU is not as good as the result obtained from CPU. However, we think that the result can be better by 1) replacing our random function which is used a lot in genetic algorithm with the one providing in CUDA and 2) using other crossover operators such as 2-Opt, and 3-Opt.

6. Acknowledgements

The author would like to thank Krittin Raethong, Weerapong Kaewchooduang, and Jarong Vutthiyotin, students in the department of Electronics Engineering Technology, King Monkut's University of Technology North Bangkok that help implementing the program.

7. Reference

[1] Sangit Chatterjee, Cecilia Carrera and Lucy A. Lynch, "Genetic algorithms and traveling salesman problems", European Journal of Operational Research, vol. 93, 1996, pp. 490-510.

[2] M. Fatih Tasgetiren, P.N. Suganthan, Quan-ke Pan and Yun-Chia Liang, "A genetic algorithm for the generalized traveling salesman problem", IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp. 2382-2389.

[3] Xue Shengjun, Guo Shaoyong and Bai Dongling, "The Analysis and Research of Parallel Genetic Algorithm", Proceeding of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, China, 2008, pp 1-4.

[4] S. Yussof, R.A. Razali, Ong Hang See, A.A. Ghaper and M.M. Din, "A Coarse-Grained Parallel Genetic Algorithm with Migration for Shortest Path Routing Problem", Proceeding of the 11th IEEE International Conference on High Performance Computing and Communications, South Korea, 2009, pp. 615-621.

[5] N.P. Karunadasa and D.N. Ranasinghe, "Accelerating high performance applications with CUDA and MPI", Proceeding of the 4th IEEE International Conference on Industrial and Information Systems, Sri Lanka, 2009, pp. 331-336

[6] Haiyang Zhang and F. Martin, "CUDA Accelerated Robot Localization and Mapping", Proceeding of the 5th IEEE International Conference on Technologies for Practical Robot Applications, USA, 2013, pp. 1-6.

[7] Erick Cantú-Paz, “A Survey of Parallel Genetic Algorithms”, Calculateurs Paralleles, vol. 10, 1998.
[8] D.E. Goldberg and R. Lingle, “Alleles, Loci and the TSP”, Proceeding of the 1st International Conference on Genetic Algorithms, 1985.