

ระบบผู้เชี่ยวชาญแบบฐานกฎ Rule-Based Expert System

ย้อนไปในทศวรรษที่ 70 เป็นที่ยอมรับกันว่า การที่จะสร้างเครื่องจักรที่สามารถแก้ปัญหาภูมิปัญญา จำเป็นจะต้องมีความรู้หรือวิธี (know-how) ในขอบเขตเฉพาะของปัญหานั้นๆ ความรู้คือความเข้าใจทั้งเชิงทฤษฎีและปฏิบัติในหัวข้อหรือภายในขอบเขต ผู้ที่มีความรู้เรียกว่าเป็นผู้เชี่ยวชาญ (expert) ผู้ซึ่งมีพลังอำนาจและความสำคัญมากที่สุดใ้องค์กรใดๆ ก็ตาม บริษัทที่จะประสบความสำเร็จได้จะต้องมีผู้เชี่ยวชาญอยู่ ผู้ที่จะเรียกว่าเป็นผู้เชี่ยวชาญในด้านหนึ่ง จะต้องมีความรู้เชิงลึก อันประกอบไปด้วยทั้งข้อเท็จจริง (fact) และกฎ (rule) ในด้านนั้นๆ และจะต้องมีประสบการณ์เชิงปฏิบัติเพียงพอในด้านนั้นๆ ด้วย ขอบเขตที่ผู้เชี่ยวชาญรู้ อาจจะจำกัด เช่นผู้เชี่ยวชาญทางด้านไฟฟ้าจะมีความรู้ทั่วไปทางด้านหม้อแปลงไฟฟ้าหรือวงจรไฟฟ้า ในขณะที่ผู้เชี่ยวชาญทางด้านฟุตบอลจะมีความรู้ (และ/หรือทักษะ) ในด้านการเล่นฟุตบอลหรือแผนการเล่นฟุตบอล เป็นต้น โดยทั่วไปแล้ว ผู้เชี่ยวชาญก็คือผู้ที่มีทักษะและสามารถทำในสิ่งที่คนอื่นทำไม่ได้นั่นเอง

มนุษย์มีขั้นตอนในการคิดอยู่ภายในหัวสมอง และมันเป็นการยากที่จะจำลองกระบวนการคิดดังกล่าว ให้อยู่ในรูปของอัลกอริทึม อย่างไรก็ตามผู้เชี่ยวชาญส่วนมากจะมีความสามารถในการแสดงออกของความรู้ ให้อยู่ในรูปของกฎที่ใช้สำหรับแก้ปัญหา พิจารณาตัวอย่างง่าย ๆ จินตนาการว่าเราพบกับมนุษย์ต่างดาว ผู้ซึ่งต้องการที่จะข้ามถนน เราต้องการช่วยมนุษย์ต่างดาวคนนี้ โดยคิดว่าเราเป็นผู้เชี่ยวชาญในการข้ามถนน ที่ซึ่งเราได้มีประสบการณ์ในการข้ามถนนมาหลายปี ดังนั้นเราสามารถที่จะสอนมนุษย์ต่างดาวให้ข้ามถนนได้ เราจะทำอย่างไรดี?

เราสามารถอธิบายให้กับมนุษย์ต่างดาวฟังว่า เขาจะสามารถข้ามถนนได้อย่างปลอดภัยก็ต่อเมื่อไฟสัญญาณข้ามถนนเป็นไฟเขียว และจะต้องหยุด (ไม่ข้ามถนน) เมื่อไฟสัญญาณข้ามถนนเป็นไฟแดง สิ่งเหล่านี้คือกฎพื้นฐานนั่นเอง ความรู้ในการข้ามถนนนี้สามารถเขียนแสดงเป็นข้อความได้ดังนี้

IF the 'traffic light' is green
THEN the action is go

IF the 'traffic light' is red
THEN the action is stop

ข้อความข้างต้นแสดงอยู่ในรูป IF-THEN เรียกว่ากฎการสร้าง (production rule) หรือเรียกง่าย ๆ เพียงว่ากฎ (rule) คำว่ากฎในปัญญาประดิษฐ์ ที่ซึ่งมักจะใช้แทนข้อมูลความรู้ สามารถนิยามได้ด้วยโครงสร้าง IF-THEN ที่ซึ่ง

มีข้อมูลหรือข้อเท็จจริงในส่วนของ IF และมีการกระทำ (action) ในส่วนของ THEN กฎดังกล่าวใช้ในการอธิบายถึงวิธีการในการแก้ปัญหา โดยปกติแล้วกฎจะเป็นอะไรที่สร้างและเข้าใจได้ง่าย

17.1 เทคนิคการแทนข้อมูลความรู้ด้วยกฎ

กฎประกอบไปด้วย 2 ส่วนคือส่วน IF เรียกว่าเป็นเงื่อนไข (condition หรือ antecedent หรือ premise) และส่วน THEN เรียกว่าเป็นผลลัพธ์ (consequent หรือ conclusion หรือ action) ไวยากรณ์พื้นฐานของกฎเขียนได้ดังนี้

```
IF      <condition>
THEN    <action>
```

โดยทั่วไปแล้วกฎสามารถมีได้หลายเงื่อนไข โดยจะเชื่อมต่อกันด้วยคำเชื่อม AND (conjunction) หรือ OR (disjunction) หรือทั้งสองอย่าง อย่างไรก็ตาม การดีที่จะไม่ควรที่ให้มีการใช้ทั้ง AND และ OR ในกฎเดียวกัน

```
IF      <condition 1>
AND     <condition 2>
      ⋮
AND     <condition N>
THEN    <consequent>
```

```
IF      <condition 1>
OR      <condition 2>
      ⋮
OR      <condition N>
THEN    <consequent>
```

ในทำนองเดียวกัน ผลลัพธ์ของกฎสามารถมีได้หลายข้อความดังนี้

```
IF      <condition>
THEN    <consequent 1>
        <consequent 2>
      ⋮
        <consequent M>
```

ส่วนเงื่อนไขของกฎประกอบไปด้วย 2 ส่วนคืออ็อบเจกต์เชิงภาษา (linguistic object) และค่าของอ็อบเจกต์ (object value) นั้นๆ ในตัวอย่างของการเดินข้ามถนน อ็อบเจกต์เชิงภาษาคือ ‘traffic light’ ซึ่งสามารถมีค่าเป็นได้ทั้ง ‘green’ หรือ ‘red’ อ็อบเจกต์และค่าของอ็อบเจกต์ถูกเชื่อมต่อกันด้วยตัวปฏิบัติการ (operator) ตัวปฏิบัติการจะทำการระบุอ็อบเจกต์แล้วทำการกำหนดของค่าให้กับอ็อบเจกต์ ตัวปฏิบัติการเช่น ‘is’ ‘are’ ‘is not’ หรือ ‘are not’ ใช้ในการให้ค่าเชิงสัญลักษณ์กับอ็อบเจกต์เชิงภาษา ระบบผู้เชี่ยวชาญทั่วไปสามารถใช้ตัวปฏิบัติการทางคณิตศาสตร์ ในการกำหนดอ็อบเจกต์ให้เป็นตัวเลขและให้ค่าเชิงตัวเลขกับอ็อบเจกต์นั้นๆ ยกตัวอย่างเช่น

```
IF      ‘temperature’ > 30
AND     ‘humidity’ > 60%
THEN    ‘air conditioner’ is on
```

เช่นเดียวกันกับประโยคเงื่อนไข ประโยคผลลัพธ์ประกอบไปด้วยอ็อบเจกต์ และค่าของอ็อบเจกต์ เชื่อมต่อกันด้วยตัวปฏิบัติการ ในตัวอย่างการเดินข้ามถนน ถ้าค่าของ 'traffic light' เป็น 'green' กฎแรกจะทำการให้ค่าอ็อบเจกต์ 'action' เป็น 'go' อ็อบเจกต์เชิงตัวเลขและตัวปฏิบัติการคณิตศาสตร์สามารถใช้ในประโยคผลลัพธ์ได้เช่นกัน

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

กฎสามารถใช้แทน 'ความสัมพันธ์' (relation) 'การแนะนำ' (recommendation) 'การชี้แนะ' (directive) 'กลยุทธ์' (strategy) และ 'สำนึก' (heuristic) [Durkin, 1994]

ความสัมพันธ์

```
IF      the fuel tank is empty
THEN    the car is dead
```

การแนะนำ

```
IF      the season is autumn
AND     the sky is cloudy
AND     the forecast is drizzle
THEN    the advice is 'take an umbrella'
```

การชี้แนะ

```
IF      the car is dead
AND     the 'fuel tank' is empty
THEN    the action is 'refuel the car'
```

กลยุทธ์

```
IF      the car is dead
THEN    the action is 'check the fuel tank';
        step 1 is complete

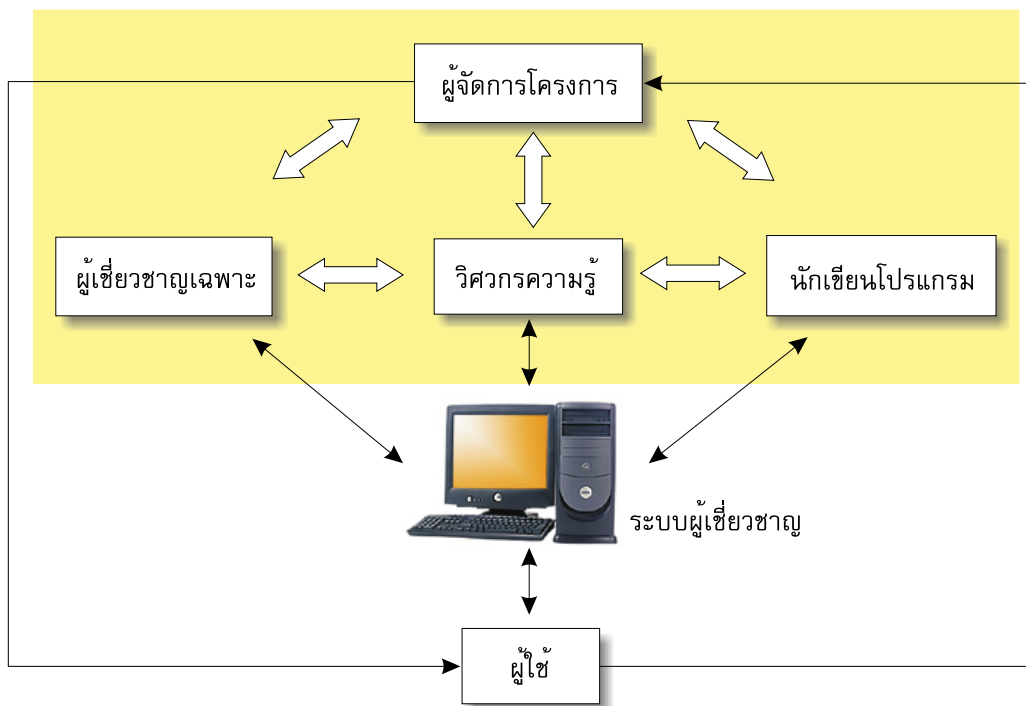
IF      step 1 is complete
AND     the 'fuel tank' is full
THEN    the action is 'check the battery';
        step 2 is complete
```

สำนึก

```
IF      the spill is liquid
AND     the 'spill pH' < 6
AND     the 'spill smell' is vinegar
THEN    the 'spill material' is 'acetic acid'
```

17.2 องค์ประกอบหลักของระบบผู้เชี่ยวชาญ

ภายหลังจากที่ได้ข้อมูลความรู้จากผู้เชี่ยวชาญที่เป็นมนุษย์แล้ว ขั้นตอนต่อไปคือการนำเอาข้อมูลดังกล่าวใส่ลงในคอมพิวเตอร์ จุดประสงค์ก็เพื่อให้คอมพิวเตอร์สามารถแก้ปัญหา ที่ซึ่งปกติจะทำโดยผู้เชี่ยวชาญ นอกไปจากนั้น



รูปที่ 17.1: ทีมงานพัฒนาระบบผู้เชี่ยวชาญ

แล้ว สิ่งหนึ่งที่เป็นที่ต้องการในการพัฒนาระบบผู้เชี่ยวชาญก็คือความสามารถในการรวบรวมข้อมูลความรู้ใหม่ๆ พร้อมทั้งสามารถแสดงข้อมูลความรู้ในรูปที่อ่านและเข้าใจง่าย รวมไปถึงการโต้ตอบด้วยภาษาธรรมชาติ (ในที่นี้คือ ภาษาอังกฤษ) แทนที่จะเป็นภาษาคอมพิวเตอร์ สิ่งสุดท้ายที่ต้องการจากระบบนี้ก็คือให้คอมพิวเตอร์สามารถอธิบายถึงเหตุผลหรือวิธีในการได้มาซึ่งคำตอบ ระบบผู้เชี่ยวชาญจึงได้แก่คอมพิวเตอร์โปรแกรม ที่สามารถทำงานในระดับของมนุษย์ผู้เชี่ยวชาญได้ในขอบเขตเฉพาะของปัญหานั้นๆ

ระบบผู้เชี่ยวชาญที่เป็นที่นิยมมากที่สุดก็คือระบบผู้เชี่ยวชาญแบบฐานกฎ (rule-based expert system) ที่ได้มีการนำเอามาประยุกต์ใช้และประสบความสำเร็จอย่างมากมาย ทั้งทางด้านธุรกิจ วิศวกรรม การแพทย์ ธรณีวิทยา ไฟฟ้ากำลัง หรือแม้กระทั่งทางด้านเหมือง ได้มีผู้พัฒนาซอฟต์แวร์จำนวนมาก ที่ใช้สำหรับสร้างระบบผู้เชี่ยวชาญแบบฐานกฎ เรียกว่าเป็นเปลือกกระบบผู้เชี่ยวชาญ (expert system shell) ข้อดีของการใช้เปลือกกระบบผู้เชี่ยวชาญคือผู้สร้างระบบเน้นเพียงขั้นตอนการสกัดข้อมูลความรู้จากผู้เชี่ยวชาญ โดยไม่ต้องเสียเวลาในการพัฒนาโปรแกรม ทำเปลือกของระบบขึ้นมาอีก

เปลือกของระบบผู้เชี่ยวชาญก็คือระบบผู้เชี่ยวชาญที่ยังไม่มีข้อมูลความรู้ (มีแต่เปลือก) ดังนั้นแล้ว สิ่งที่คุณพัฒนาระบบจะต้องทำก็คือทำการเพิ่มข้อมูลความรู้ในรูปของกฎให้กับเปลือกของระบบ และให้ข้อมูลเพื่อใช้สำหรับแก้ปัญหาที่ต้องการ รูปที่ 17.1 แสดงกลุ่มคนที่เกี่ยวข้องในการพัฒนาสร้างระบบผู้เชี่ยวชาญ ดังรายละเอียดต่อไปนี้

- **ผู้เชี่ยวชาญเฉพาะ (domain expert)** คือผู้ที่มีความรู้และทักษะในการแก้ปัญหาในขอบเขตเฉพาะด้าน ผู้เชี่ยวชาญเฉพาะจะต้องมีความชำนาญในงานเฉพาะนั้นๆ เป็นอย่างดี ความชำนาญดังกล่าวจะถูกสกัดให้กับระบบผู้เชี่ยวชาญ ดังนั้นแล้วผู้เชี่ยวชาญเฉพาะนี้ จะต้องสามารถที่จะถ่ายทอดข้อมูลความรู้ได้ดี และจะต้องทำงานและใช้เวลาร่วมกันกับทีมสร้างระบบผู้เชี่ยวชาญค่อนข้างมากที่สุด ถือเป็นบุคคลที่สำคัญที่สุดของทีมพัฒนาระบบผู้เชี่ยวชาญ
- **วิศวกรความรู้ (knowledge engineer)** คือผู้ที่สามารถออกแบบ สร้างและทดสอบระบบผู้เชี่ยวชาญได้ วิศวกรความรู้จะเป็นผู้ที่คุยกับผู้เชี่ยวชาญ เพื่อที่จะเรียนรู้วิธีการในการแก้ปัญหาที่ต้องการ มีการใช้วิธีทางเหตุผล

ในการประมวลผลข้อเท็จจริง (fact) และกฎ (rule) และเป็นผู้ตัดสินใจเลือกวิธีในการนำเอาข้อเท็จจริงและกฎต่างๆ เข้าสู่ระบบผู้เชี่ยวชาญ นอกจากนั้นแล้ววิศวกรความรู้ยังเป็นผู้ที่เลือกซอฟต์แวร์สำหรับสร้างระบบหรือเลือกเปลี่ยนระบบผู้เชี่ยวชาญที่เหมาะสม หรือเลือกแพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์เพื่อสร้างระบบผู้เชี่ยวชาญ ท้ายสุดแล้ว วิศวกรความรู้จะเป็นผู้ที่ทดสอบ ตรวจสอบและนำเอาระบบที่ได้ไปสู่การใช้งานจริง จะเห็นได้ว่าวิศวกรความรู้เป็นผู้ที่อยู่กับทีมพัฒนาระบบตั้งแต่ต้นจนจบ อาจจะมีอีกหลายๆ โอกาสที่วิศวกรความรู้ยังต้องทำการดูแลระบบ หลังจากทีระบบสร้างเสร็จเรียบร้อยแล้ว

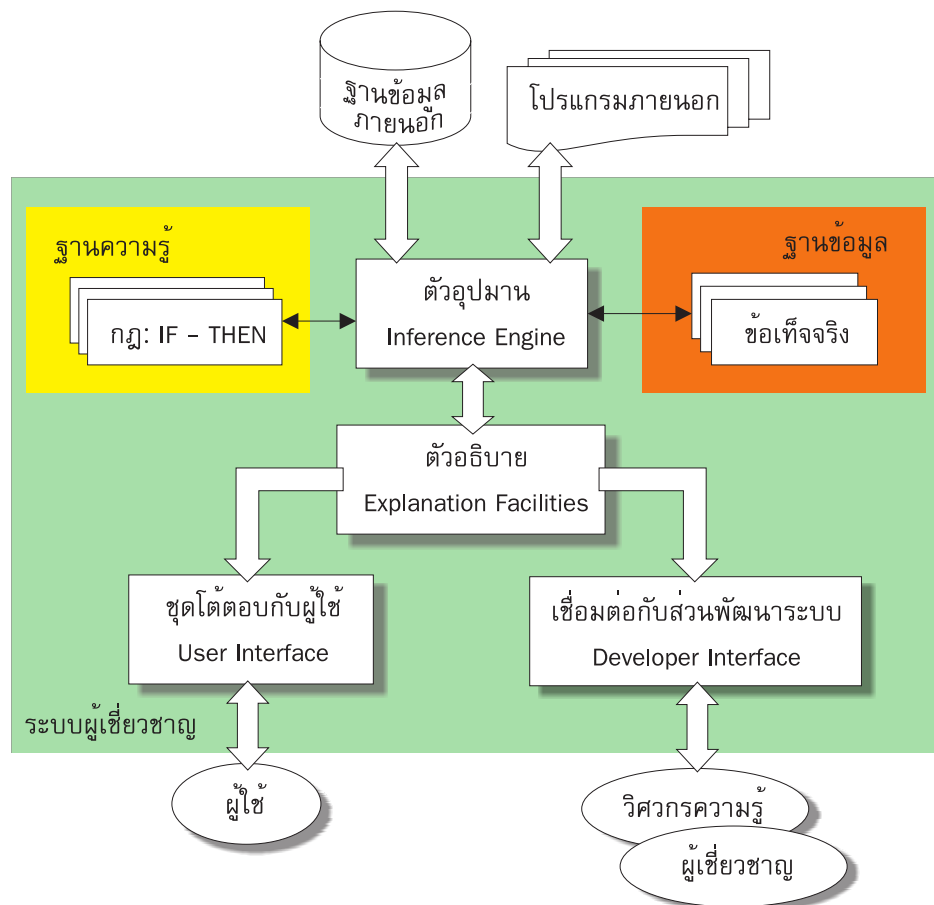
- **นักเขียนโปรแกรม (programmer)** คือผู้ที่ทำการเขียนโปรแกรมคอมพิวเตอร์ กล่าวคือเป็นผู้ที่มีทักษะของภาษาคอมพิวเตอร์ทางปัญญาระดับสูง เช่น LISP หรือ Prolog เป็นต้น รวมไปถึงภาษาคอมพิวเตอร์ทั่วไป เช่น C/C++ Pascal FORTRAN หรือ Basic ถ้ามีการใช้เปลี่ยนระบบผู้เชี่ยวชาญ วิศวกรความรู้จะสามารถป้อนข้อมูลความรู้ให้กับระบบได้โดยตรง โดยไม่ต้องพึ่งพานักเขียนโปรแกรมแต่อย่างใด แต่ถ้าไม่มีการใช้เปลี่ยนระบบผู้เชี่ยวชาญ นักเขียนโปรแกรมจะต้องพัฒนาทั้งข้อมูลความรู้ในรูปแบบที่ภาษาคอมพิวเตอร์เข้าใจ โครงสร้างการควบคุมการทำงานของระบบ รวมไปถึงการโต้ตอบกับผู้ใช้ (user interface) นักเขียนโปรแกรมอาจจะต้องเกี่ยวข้องกับขั้นตอนการทดสอบระบบอีกด้วย
- **ผู้จัดการโครงการ (project manager)** คือหัวหน้าทีมพัฒนา ผู้ซึ่งรับผิดชอบในการทำให้โครงการดำเนินไปได้ เป็นผู้ที่ต้องผลักดันให้โครงการเป็นไปตามเป้าหมาย กำหนดเวลาที่ตั้งไว้ และจะต้องเป็นผู้ที่ติดต่oprสานงานกับทุกๆ ฝ่ายในทีมพัฒนา
- **ผู้ใช้ (end-user)** คือผู้ใช้ระบบผู้เชี่ยวชาญในการแก้ปัญหาต่างๆ ผู้ใช้แต่ละคนอาจจะมีความต้องการและความพอใจจากตัวระบบผู้เชี่ยวชาญที่แตกต่างกัน นอกเหนือไปจากความมั่นใจในการใช้ระบบแล้ว ความสะดวกสบายในการใช้งานก็เป็นอีกสิ่งหนึ่งที่ผู้ใช้มักจะต้องการเสมอ การออกแบบการโต้ตอบกับผู้ใช้จึงถือเป็นสิ่งสำคัญที่จะทำให้ระบบผู้เชี่ยวชาญสามารถนำไปใช้งานได้มีประสิทธิภาพ

17.2.1 โครงสร้างของระบบผู้เชี่ยวชาญแบบฐานกฎ

ระบบผู้เชี่ยวชาญแบบฐานกฎมีองค์ประกอบที่สำคัญๆ อยู่ 5 องค์ประกอบดังรายละเอียดต่อไปนี้ (ดูรูปที่ 17.2)

- **ฐานความรู้ (knowledge base)** ประกอบไปด้วยข้อมูลความรู้เฉพาะด้านที่ใช้ในการแก้ปัญหา ของระบบผู้เชี่ยวชาญแบบฐานกฎ ข้อมูลความรู้จะอยู่ในรูปของกฎ IF - THEN
- **ฐานข้อมูล (database)** ประกอบไปด้วยเซตของข้อเท็จจริง สำหรับใช้ในการตรวจสอบกับเงื่อนไขในส่วน IF ของกฎในฐานความรู้
- **ตัวอุปมาน (inference engine)** เป็นส่วนการคำนวณเชิงเหตุผลเพื่อนำไปสู่คำตอบ ทำหน้าที่ในการเชื่อมกฎจากฐานความรู้ กับข้อเท็จจริงจากฐานข้อมูล
- **ตัวอธิบาย (explanation facility)** เป็นส่วนที่ช่วยให้ผู้ใช้ระบบเข้าใจว่าคำตอบได้มาอย่างไร และทำไมจึงต้องใช้ข้อเท็จจริงนั้นๆ ระบบผู้เชี่ยวชาญจะต้องสามารถที่จะอธิบายเหตุผลในการทำงานของระบบ และจัดคำแนะนำ การวิเคราะห์หรือข้อสรุปได้
- **ชุดโต้ตอบกับผู้ใช้ (user interface)** เป็นส่วนโต้ตอบระหว่างผู้ใช้กับระบบผู้เชี่ยวชาญ การโต้ตอบควรมีประสิทธิภาพและง่ายต่อการใช้งานให้มากที่สุด

องค์ประกอบข้างต้นถือเป็นองค์ประกอบหลักของระบบผู้เชี่ยวชาญ อย่างไรก็ตาม ระบบสามารถมีองค์ประกอบอื่นๆ เพิ่มเติมได้ตามความจำเป็น



รูปที่ 17.2: โครงสร้างระบบผู้เชี่ยวชาญแบบฐานกฎ

17.2.2 คุณลักษณะของระบบผู้เชี่ยวชาญ

คุณลักษณะของระบบผู้เชี่ยวชาญมีความเหมือนและแตกต่างไปจากผู้เชี่ยวชาญที่เป็นมนุษย์ในหลายๆ ประเด็น ตารางที่ 17.1 แสดงการเปรียบเทียบคุณลักษณะของระบบผู้เชี่ยวชาญกับผู้เชี่ยวชาญที่เป็นมนุษย์ พร้อมทั้งแสดงการเปรียบเทียบกับระบบการแก้ปัญหาแบบดั้งเดิมด้วย

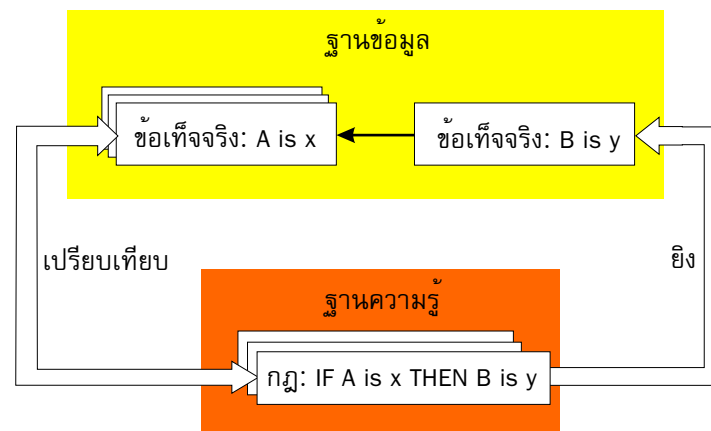
17.3 เทคนิคการอนุมาน Inference Techniques

ในระบบผู้เชี่ยวชาญแบบฐานกฎ ข้อมูลความรู้ถูกแทนด้วยกฎ IF - THEN และข้อมูลถูกแทนด้วยข้อเท็จจริงที่เกี่ยวข้องกับระบบหรือปัญหาที่ต้องการจะแก้ไข ตัวอุปมานจะทำการเปรียบเทียบแต่ละกฎที่เก็บไว้ในฐานความรู้ด้วยข้อเท็จจริงที่เก็บไว้ในฐานข้อมูล เมื่อเงื่อนไขในส่วน IF ตรงกับข้อเท็จจริง กฎที่สอดคล้องกับเงื่อนไขนั้นจะถูกเรียกใช้หรือถูกยิง (fired) และส่วน THEN จะถูกกระทำ กฎที่ถูกเรียกใช้จะทำให้เกิดการเปลี่ยนแปลงของข้อเท็จจริง โดยการเพิ่มข้อเท็จจริงใหม่เข้าไปดังแสดงในรูปที่ 17.3

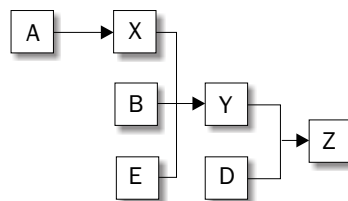
กฎในส่วน IF ที่ตรงกับการเปรียบเทียบกับข้อเท็จจริงจะให้ลูกโซ่อุปมาน (inference chain) ที่ซึ่งเป็นส่วนที่

ตารางที่ 17.1: เปรียบเทียบระหว่างผู้เชี่ยวชาญมนุษย์ ระบบผู้เชี่ยวชาญและระบบแก้ปัญหาแบบดั้งเดิม

ผู้เชี่ยวชาญมนุษย์	ระบบผู้เชี่ยวชาญ	ระบบดั้งเดิม
ใช้ข้อมูลความรู้ในรูปของสำนึกในการแก้ปัญหาภายในขอบเขตที่จำกัด	ประมวลผลข้อมูลความรู้ในรูปของกฎและความ เป็นเหตุเป็นผล ในการ แก้ ปัญหา ในขอบเขตที่จำกัด	ประมวลผลข้อมูลและใช้อัลกอริทึม และตัวปฏิบัติการในการแก้ปัญหาเชิงตัวเลข
ในสมองมนุษย์ ข้อมูลความรู้ปรากฏในรูปแบบที่แปลไว้แล้ว	มีการแยกส่วนของข้อมูลความรู้จากการประมวลผล	ไม่มีการแยกส่วนของข้อมูลความรู้ออกจากโปรแกรม
สามารถอธิบายเหตุผลและรายละเอียดของการทำงานได้	สามารถแกะรอยกฎที่ถูกใช้งานในขณะแก้ปัญหา และสามารถอธิบายว่าคำตอบของปัญหามาได้อย่างไรและทำไมจึงเลือกข้อมูลที่ใช้	ไม่สามารถอธิบายว่าผลลัพธ์ที่ได้มาอย่างไรและทำไมจึงใช้ข้อมูลนั้นๆ
ใช้เหตุผลที่ไม่แม่นยำและสามารถประมวลผลข้อมูลที่ไม่สมบูรณ์ ไม่แน่นอนและไม่ชัดเจนได้	สามารถใช้เหตุผลที่ไม่แม่นยำได้และสามารถประมวลผลข้อมูลที่ไม่สมบูรณ์ ไม่แน่นอนและไม่ชัดเจนได้	ใช้เฉพาะปัญหาที่ต้องมีข้อมูลครบถ้วนและแม่นยำ
สามารถทำงานผิดพลาดได้ เมื่อ ข้อมูล ไม่สมบูรณ์หรือไม่ชัดเจน	สามารถทำงานผิดพลาดได้ เมื่อ ข้อมูล ไม่สมบูรณ์หรือไม่ชัดเจน	ไม่สามารถให้คำตอบได้เลยหรือให้คำตอบที่ผิดถ้าข้อมูลไม่สมบูรณ์หรือไม่ชัดเจน
เพิ่มประสิทธิภาพในการแก้ไขปัญหาได้จากการเรียนรู้และฝึกฝน การเรียนรู้เป็นไปอย่างช้าๆ ไม่มีประสิทธิภาพและแพง	เพิ่มประสิทธิภาพในการแก้ไขปัญหได้โดยการเพิ่มกฎ หรือ ทำการปรับปรุงกฎเก่าในฐานความรู้ เมื่อมีข้อมูลความรู้ใหม่ๆ ระบบสามารถปรับแก้ด้วยข้อมูลใหม่จนได้อย่างไม่ยุ่งยาก	เพิ่ม ประสิทธิภาพ ในการ แก้ไข ปัญหา ได้ โดย การปรับ เปลี่ยน ต้น รหัส โปรแกรม ซึ่งมีผลกระทบต่อการประมวลผล การปรับแก้มีความยุ่งยาก



รูปที่ 17.3: วงจรตัวอย่าง



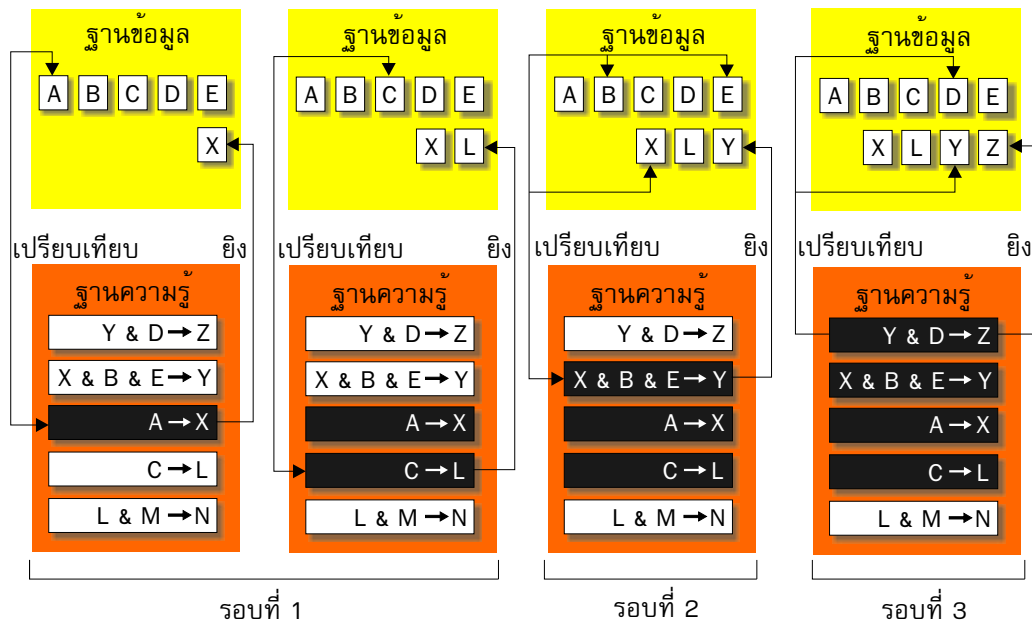
รูปที่ 17.4: ตัวอย่างลูกโซ่อนุมาน

บอกระบบผู้เชี่ยวชาญในการเลือกกฎที่จะทำให้ไปสู่คำตอบ พิจารณาตัวอย่างของลูกโซ่อนุมานต่อไปนี้
สมมติว่าในฐานข้อมูลเริ่มต้นมีข้อเท็จจริงคือ A B C D และ E และฐานความรู้มีอยู่เพียง 3 กฎคือ

- กฎ 1: IF Y is true
AND D is true
THEN Z is true
- กฎ 2: IF X is true
AND B is true
AND E is true
THEN Y is true
- กฎ 3: IF A is true
THEN X is true

รูปที่ 17.4 แสดงตัวอย่างตัวลูกโซ่อนุมานของระบบผู้เชี่ยวชาญในการใช้กฎเพื่ออนุมานข้อเท็จจริง Z โดยแรกสุด กฎ 3 จะถูกเรียกใช้ก่อนเพื่อให้ได้ข้อเท็จจริง X จากที่กำหนด A แล้วกฎ 2 จะถูกเลือกเพื่อทำการอนุมาน Y จากข้อเท็จจริงเริ่มต้น B และ E พร้อมทั้งข้อเท็จจริงที่ได้จากกฎ 3 นั่นคือ X ทำที่สุด กฎ 1 จะถูกเรียกใช้ด้วยข้อเท็จจริงเริ่มต้น D และข้อเท็จจริง Y จากกฎ 2 เพื่อทำการอนุมาน Z ระบบผู้เชี่ยวชาญสามารถแสดงตัวลูกโซ่อนุมานนี้เพื่อทำการอธิบายว่าข้อสรุป Z ได้มาอย่างไร

ตัวอนุมานเป็นส่วนสำคัญที่จะต้องตัดสินใจว่ากฎไหนควรจะถูกต้องถูกเรียกใช้ โดยปกติแล้วมีวิธีการเลือกกฎอยู่ 2 วิธีคือวิธีลูกโซ่ไปข้างหน้า (forward chaining) และวิธีลูกโซ่ย้อนกลับ (backward chaining) [Waterman and Hayes-Roth, 1986]



รูปที่ 17.5: ตัวอย่างวิธีลูกโซ่ไปข้างหน้า

17.3.1 วิธีการลูกโซ่ไปข้างหน้า Forward Chaining

ในตัวอย่างข้างต้นแสดงการอุปมานแบบวิธีลูกโซ่ไปข้างหน้า พิจารณาในรายละเอียดเพิ่มเติมโดยการเขียนกฎในรูปแบบดังต่อไปนี้ (เพิ่มกฎ 4 และกฎ 5)

- กฎ 1: $Y \& D \rightarrow Z$
- กฎ 2: $X \& B \& E \rightarrow Y$
- กฎ 3: $A \rightarrow X$
- กฎ 4: $C \rightarrow L$
- กฎ 5: $L \& M \rightarrow N$

รูปที่ 17.5 แสดงตัวอย่างวิธีการลูกโซ่ไปข้างหน้าสำหรับเซตของกฎง่ายๆ ข้างต้น วิธีการลูกโซ่ไปข้างหน้าเป็นระบบเหตุผลแบบขับเคลื่อนด้วยข้อมูล (data-driven) กล่าวคือระบบเริ่มต้นด้วยข้อมูลที่มีอยู่ แล้วทำดำเนินการไปข้างหน้ากับข้อมูลนั้นๆ แต่ละครั้งกฎที่อยู่ก่อนจะถูกพิจารณาเรียกใช้ก่อน เมื่อกฎนั้นๆ ถูกเรียกใช้ ข้อเท็จจริงที่ได้จากกฎนั้นจะถูกเพิ่มเข้าไปในฐานข้อมูล แต่ละกฎสามารถถูกเรียกใช้ได้เพียงครั้งเดียว

ในรอบแรก มีอยู่ 2 กฎที่ถูกเรียกใช้ได้แก่ กฎ 3: $A \rightarrow X$ และกฎ 4: $C \rightarrow L$ ซึ่งเป็นกฎที่ตรงกับในฐานข้อมูล กฎ 3 จะถูกเรียกใช้ก่อนเนื่องจากอยู่ในตำแหน่งสูงกว่า ส่วน IF ของกฎ 3 ตรงกับข้อเท็จจริง A ในฐานข้อมูล ส่วน THEN จึงถูกกระทำและได้ข้อเท็จจริง X ใหม่เข้าสู่ฐานข้อมูล เช่นเดียวกันกับกฎ 4: $C \rightarrow L$ ที่ซึ่งทำการเพิ่มข้อเท็จจริง L เข้าสู่ฐานข้อมูล

ในรอบที่สอง กฎ 2: $X \& B \& E \rightarrow Y$ ถูกเรียกใช้เพราะข้อเท็จจริง B E และ X มีอยู่ในฐานข้อมูลเรียบร้อยแล้ว ผลลัพธ์ Y ที่ได้จะถูกอุปมานและเพิ่มเข้าสู่ฐานข้อมูล ทำให้ในรอบที่สาม กฎ 1: $Y \& D \rightarrow Z$ ถูกเรียกใช้ และทำการเพิ่มข้อเท็จจริง Z เข้าสู่ฐานข้อมูล ระบบจะหยุดที่จุดนี้เนื่องจากไม่มีกฎไหนที่เหลืออยู่ที่ตรงกับข้อเท็จจริงในฐานข้อมูล (นั่นคือไม่มี L และ M จากกฎ 5 ในฐานข้อมูลที่จะทำให้กฎ 5: $L \& M \rightarrow N$ ถูกเรียกใช้ได้)

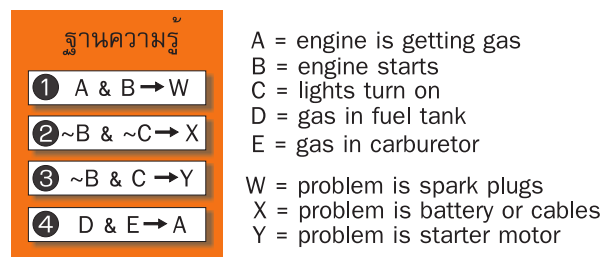
วิธีการลูกโซ่ไปข้างหน้าเป็นเทคนิคในการรวบรวมข้อมูลและทำการอุปมานจากข้อมูลนั้นๆ อย่างไรก็ดี จะมีอยู่หลายๆ กฎที่ถูกเรียกใช้ แต่เป็นกฎที่ไม่ได้มีส่วนเกี่ยวข้องกับเป้าหมายเลย จากตัวอย่างข้างต้น เป้าหมาย

คือทำการหาข้อเท็จจริง Z จากกฎทั้ง 5 กฎ มีอยู่ถึง 4 กฎที่ถูกเรียกใช้ แต่เมื่อพิจารณากฎ 4: $C \rightarrow L$ ไม่ได้มีส่วนเกี่ยวข้องกับ Z แต่อย่างใด ในระบบผู้เชี่ยวชาญจริงๆ จึงอาจจะมีกฎอยู่ในฐานข้อมูลมากเป็นร้อยๆ กฎ ดังนั้นวิธีการลูกโซ่ไปข้างหน้าอาจจะไม่เหมาะสมกับสถานการณ์ดังกล่าว ตัวอย่างต่อไปนี้แสดงการทำงานของระบบผู้เชี่ยวชาญช่างซ่อมรถแบบฐานกฎ โดยมีกลไกการอนุมานเป็นแบบวิธีการลูกโซ่ไปข้างหน้า เพื่อใช้ในการสรุปหาผลการวินิจฉัยอาการเสียของรถ วิธีการลูกโซ่ไปข้างหน้าดังกล่าวเป็นลักษณะการขับเคลื่อนของข้อมูล (data-driven) ตัวอย่างที่นำเสนอเป็นเพียงระบบเล็กๆ เพื่อสาธิตการทำงานของระบบผู้เชี่ยวชาญ โดยเฉพาะการเชื่อมต่อกฎเป็นลูกโซ่ การรวบรวมข้อมูลใหม่เข้าสู่ระบบ และการอธิบายผลลัพธ์หรือคำตอบที่ได้จากระบบ

■ ตัวอย่างที่ 17.1 ระบบผู้เชี่ยวชาญช่างซ่อมรถ - วิธีการอุปมานแบบลูกโซ่ไปข้างหน้า

พิจารณาระบบผู้เชี่ยวชาญช่างซ่อมรถ ที่ซึ่งมีกฎในการวินิจฉัยอาการเสียของรถอยู่ 4 กฎดังแสดงในรูปที่ 17.6 (A หมายถึง not A) ระบบทำการเก็บกฎทั้ง 4 ไว้ในฐานความรู้เพื่อใช้ในการอนุมานต่อไป

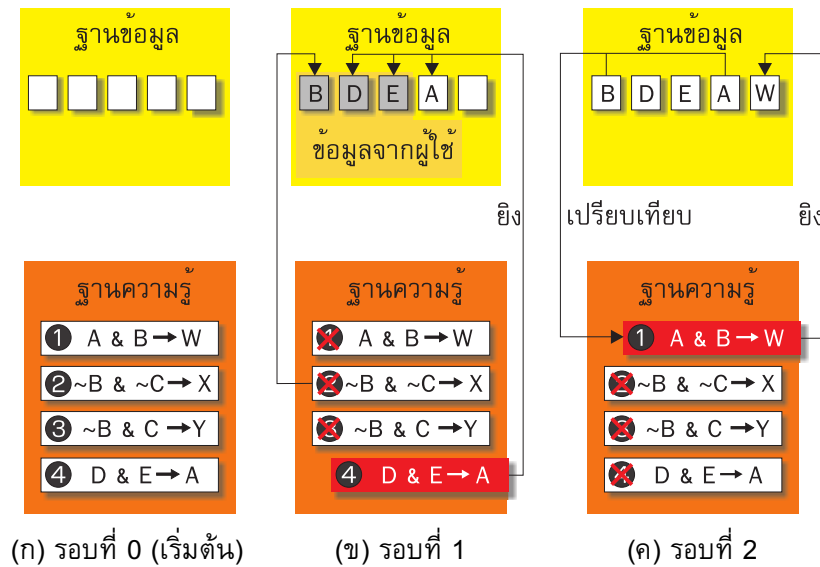
พิจารณาสถานะเริ่มต้นของระบบแสดงในรูปที่ 17.7-(ก) ที่ซึ่งไม่มีข้อมูลใดๆ ในฐานข้อมูล ในกรณีนี้ระบบอาจจะถามคำถามผู้ใช้เพื่อให้ได้มาซึ่งข้อมูลเพิ่มเติมสำหรับการอุปมาน เราสามารถให้ข้อมูลด้วยการตอบคำถามใดๆ ก็ได้ ตราบเท่าที่ข้อมูลหรือข้อเท็จจริง (ซึ่งอยู่ในส่วนของ IF) ไม่เป็นส่วนของ THEN ในกฎใดๆ ในฐานความรู้ ภายในระบบ ยกตัวอย่างเช่นเราไม่สามารถตอบคำถาม “engine is getting gas?” หรือ Z ได้เนื่องจาก Z เป็นส่วน THEN ของกฎ 4 ตัวอย่างขั้นตอนการทำงานของระบบมีดังต่อไปนี้



รูปที่ 17.6: กฎ 1 - 4 ของการวินิจฉัยอาการเสียของรถ

• รอบที่ 1 (รูปที่ 17.7-(ข))

- ระบบเริ่มต้นจากกฎ 1 ที่ซึ่งข้อเท็จจริงหรือข้อมูลในฐานข้อมูลไม่ตรงกับส่วน IF ของกฎ 1 โดยเฉพาะข้อเท็จจริง A ที่ซึ่งระบบไม่สามารถถามข้อมูลเพิ่มเติมจากผู้ใช้ได้ ดังเหตุผลที่กล่าวมาแล้วข้างต้น ดังนั้นกฎ 1 จะไม่ถูกเรียกใช้
- ระบบจะดำเนินการต่อไปยังกฎ 2 ที่ซึ่งข้อมูล $\sim B$ สามารถถูกถามจากผู้ใช้ได้ (แน่นอนว่าผู้ใช้จะต้องรู้ว่ารถสตาร์ทติดหรือไม่) ถ้าผู้ใช้ตอบระบบว่ารถสตาร์ทติด ซึ่งก็คือ B นั่นเอง ข้อมูลหรือข้อเท็จจริง B นี้จะถูกป้อนเก็บไว้ในฐานข้อมูลและทำให้กฎ 2 ไม่ถูกเรียกใช้ (เพราะ $\sim B$ ไม่เป็นจริง)
- เช่นเดียวกันกับกฎ 2 ที่ซึ่ง $\sim B$ ไม่เป็นจริง ทำให้กฎ 3 ไม่ถูกเรียกใช้ด้วย
- พิจารณากฎ 4 ระบบทำการถามข้อมูลเพิ่มเติมจากผู้ใช้ สมมุติว่าทั้ง D และ E เป็นจริง กล่าวคือผู้ใช้ตอบว่า “gas in fuel tank” (เติมน้ำมัน) และ “gas in carburetor” (น้ำมันเข้าสู่คาร์บูเรเตอร์) ดังนั้นกฎ 4 จะถูกเรียกใช้ ทั้ง D และ E รวมไปถึงข้อเท็จจริง A ซึ่งเป็นการสรุปอาการรถที่ว่า “problem is getting gas” จะถูกป้อนเก็บไว้ในฐานข้อมูล



รูปที่ 17.7: ตัวอย่างการอุปมานแบบลูกโซ่ไปข้างหน้าของระบบผู้เชี่ยวชาญช่างซ่อมรถ

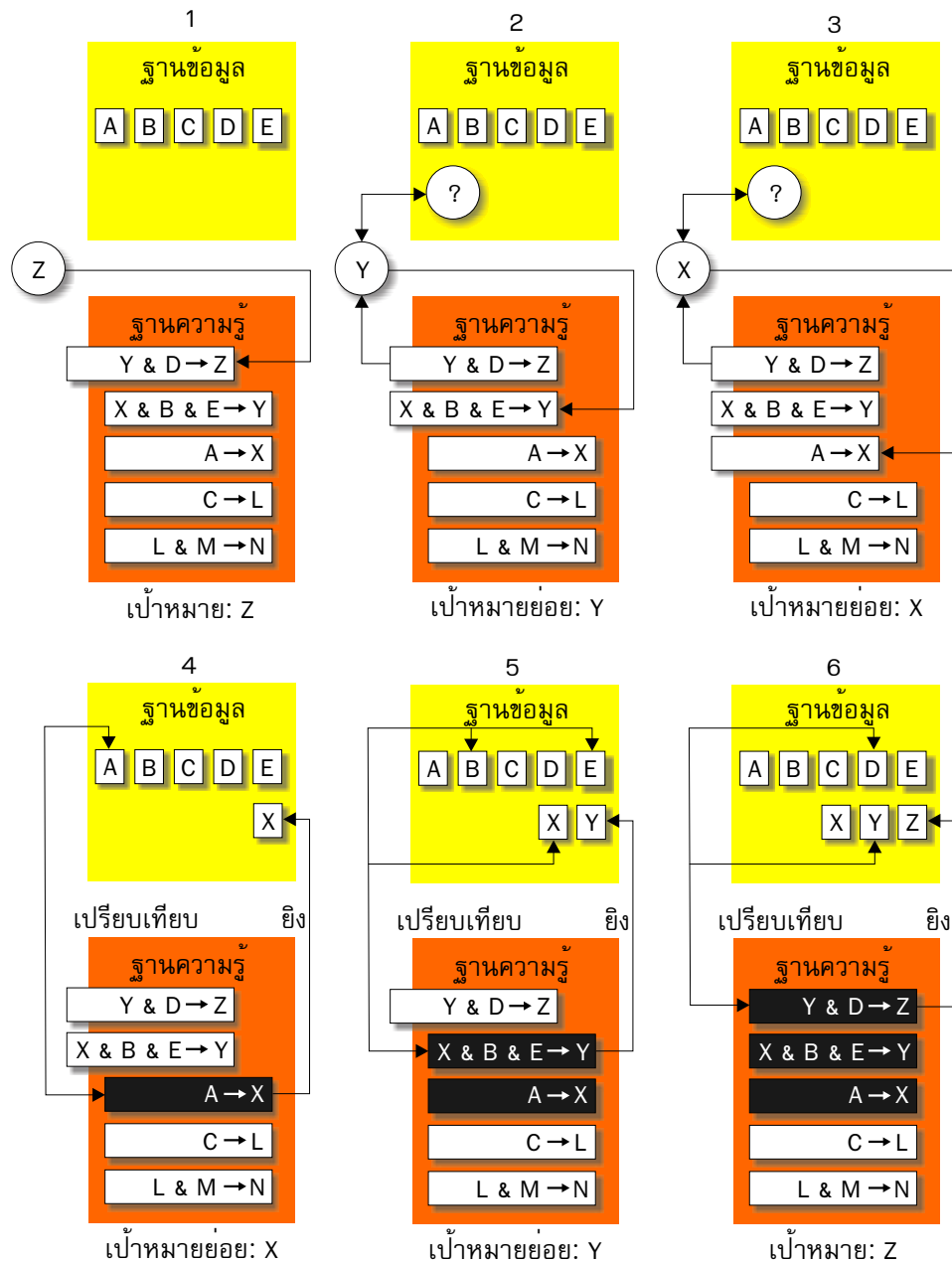
- **รอบที่ 2** (รูปที่ 17.7-(ค)) ระบบเริ่มต้นที่กฎ 1 ใหม่ โดยในรอบนี้ ฐานข้อมูลมีข้อเท็จจริงเพิ่มเติมมาจากรอบที่ 1 จะเห็นได้ว่ากฎ 1 จะถูกเรียกใช้เนื่องจากข้อเท็จจริงทั้ง A และ B ปรากฏอยู่ในฐานข้อมูลเรียบร้อยแล้ว ระบบจะทำการแทรกข้อเท็จจริง W ซึ่งเป็นผลจากกฎ 1 เข้าสู่ฐานข้อมูล ระบบทำการตรวจสอบกฎที่เหลือทั้งหมดและพบว่าไม่มีกฎใดถูกเรียกใช้อีก ดังนั้นระบบสามารถอนุมานอาการเสียของรถได้คือ “problem is spark plugs” (จากข้อเท็จจริง W)

ตัวอย่างข้างต้นแสดงลักษณะการอนุมานที่ขับเคลื่อนด้วยข้อมูล (data-driven) ในหัวข้อต่อไปจะได้นำเสนอการอนุมานอีกแบบหนึ่ง ที่ซึ่งใช้หลักการขับเคลื่อนด้วยเป้าหมาย (goal-driven) ตัวอย่างข้างต้นจะถูกเปรียบเทียบในรายละเอียดอีกครั้ง

17.3.2 วิธีการลูกโซ่ย้อนกลับ Backward Chaining

วิธีการลูกโซ่ย้อนกลับเป็นวิธีขับเคลื่อนด้วยเป้าหมาย (goal-driven) กล่าวคือระบบจะมีเป้าหมายเป็นสมมติฐานเอาไว้ แล้วตัวอุปมานจะทำการค้นหาหลักฐานเพื่อพิสูจน์เป้าหมายนั้น โดยอันดับแรก ฐานข้อมูลจะถูกค้นเพื่อหากฎที่อาจจะเกี่ยวข้องกับเป้าหมาย กฎที่ได้ในขั้นตอนนี้นั้นจะต้องมีเป้าหมายอยู่ในส่วน THEN ถ้ากฎดังกล่าวถูกค้นพบและส่วน IF ของกฎนั้นอยู่ในฐานข้อมูล กฎนั้นจะถูกเรียกใช้และเป้าหมายถือว่าได้รับการพิสูจน์ อย่างไรก็ตาม ในความเป็นจริง เป้าหมายจะไม่ถูกค้นพบในครั้งแรกที่เจอกฎที่เกี่ยวข้อง ตัวอุปมานจะทำการสร้างเป้าหมายย่อย (subgoal) ขึ้นใหม่จากกฎที่มีอยู่เพื่อทำการพิสูจน์เป้าหมายย่อยนี้ ระบบจะทำขั้นตอนนี้ไปเรื่อยๆ จนกระทั่งไม่มีกฎใดที่เกี่ยวข้องเหลืออยู่ รูปที่ 17.8 แสดงตัวอย่างการทำงานของวิธีการลูกโซ่ย้อนกลับ

1. ตัวอุปมานพยายามจะทำการอุปมานข้อเท็จจริง Z โดยการค้นหาในฐานความรู้ที่มีเป้าหมาย Z อยู่ในส่วน THEN ตัวอุปมานเจอกฎ 1: $Y \& D \rightarrow Z$ ที่ซึ่งส่วน IF มีข้อเท็จจริง Y และ D อยู่



รูปที่ 17.8: ตัวอย่างวิธีลูกโซ่ย้อนกลับ

- เนื่องจากข้อเท็จจริง Y ไม่มีปรากฏในฐานข้อมูล ตัวอุปมานะทำการตั้ง Y เป็นเป้าหมายย่อย และทำการค้นหา Y ในส่วน THEN จากในฐานความรู้ ซึ่งได้เจอกกฎ 2: $X \& B \& E \rightarrow Y$ โดยมีข้อเท็จจริง X B และ E ในส่วน IF
- เนื่องจากข้อเท็จจริง X ไม่มีในฐานข้อมูล ตัวอุปมานะจึงทำการตั้งเป้าหมายย่อย X ขึ้นมา ซึ่งตัวอุปมานะทำการค้นหาในฐานความรู้และเจอกกฎ 3: $A \rightarrow X$ ดังนั้นตัวอุปมานะจะต้องทำการค้นหาข้อเท็จจริง A ต่อไป
- ตัวอุปมานะเจอ A ในฐานข้อมูล กฎ 3: $A \rightarrow X$ ถูกเรียกใช้และข้อเท็จจริง X ถูกเพิ่มเข้าในฐานข้อมูล
- ตัวอุปมานะกลับไปพิจารณาเป้าหมายย่อย Y และค้นพบกฎ 2: $X \& B \& E \rightarrow Y$ ที่ซึ่งข้อเท็จจริง X B และ E มีอยู่ในฐานข้อมูลเรียบร้อยแล้ว ดังนั้นกฎ 2 จึงถูกเรียกใช้และข้อเท็จจริง Y ถูกเพิ่มใหม่เข้าไปในฐานข้อมูล

6. ตัวอุปมาวนกลับไปยังกฎ 1: $Y \ \& \ D \rightarrow Z$ เพื่อทำการพิจารณาเป้าหมาย Z ซึ่งส่วนมี Y และ D ในส่วน IF และพบว่ากฎ 1 มีเงื่อนไข IF ที่ตรงกับที่ต้องการ กฎ 1 จึงถูกเรียกใช้และสุดท้ายเป้าหมายได้ถูกพิสูจน์

เมื่อเปรียบเทียบวิธีลู่ไขย้อนกลับกับวิธีลู่ไขไปข้างหน้า จะเห็นได้ว่าวิธีนี้มีการเรียกใช้กฎเพียง 3 กฎ แสดงถึงประสิทธิภาพที่ดีกว่าของวิธีลู่ไขย้อนกลับ ในวิธีลู่ไขไปข้างหน้า เรารู้ข้อเท็จจริงเริ่มต้นและผู้ใช้จะไม่มีส่วนเกี่ยวข้องกับการให้ข้อเท็จจริงที่ต้องการเพิ่มเติม ในขณะที่วิธีการลู่ไขย้อนกลับ เป้าหมายจะถูกตั้งขึ้นและใช้เป็นข้อเท็จจริงที่ต้องการในการสนับสนุนเป้าหมายนั้นๆ ในกรณีนี้ผู้ใช้จะต้องมีส่วนร่วมในการให้ข้อเท็จจริงเพิ่มเติมในกรณีที่ข้อเท็จจริงนั้นๆ ไม่ปรากฏอยู่ในฐานข้อมูล

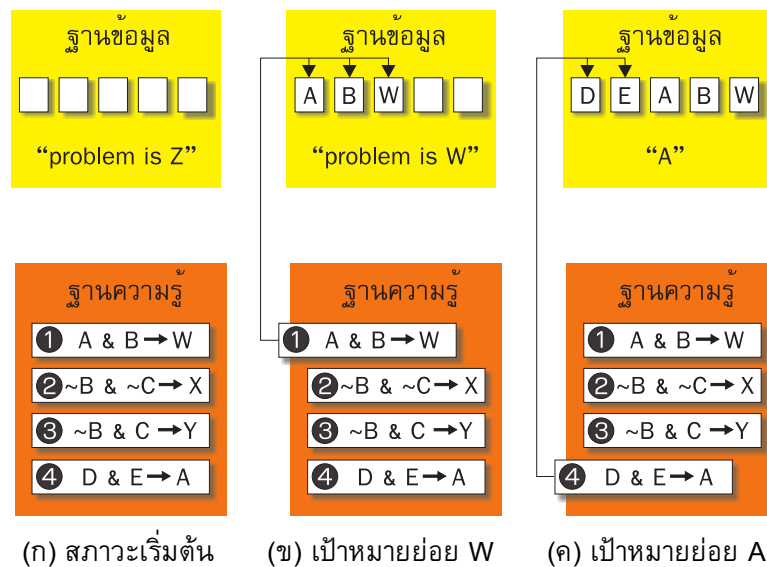
■ ตัวอย่างที่ 17.2 ระบบผู้เชี่ยวชาญช่างซ่อมรถ - วิธีการอุปมาวนแบบลู่ไขย้อนกลับ

พิจารณาระบบผู้เชี่ยวชาญช่างซ่อมรถ ที่มีกฎดังแสดงในรูปที่ 17.6 จากตัวอย่างที่แล้ว ในหลักการของการขับเคลื่อนด้วยเป้าหมาย (goal-driven) ผู้ใช้จะทำการตั้งเป้าหมายให้กับระบบ แล้วระบบจะทำการอนุมานข้อเท็จจริงต่างๆ เพื่อพิสูจน์ให้ได้ว่าเป้าหมายที่ตั้งไว้นั้นถูกหรือผิดอย่างไร สมมติว่าระบบเริ่มตั้งเป้าหมายไว้ว่า “problem is Z” โดยที่ Z เป็นตัวแปรที่สามารถเท่ากับข้อเท็จจริงใดๆ ในฐานข้อมูลได้ ดังนั้นระบบจะเริ่มดำเนินการพิจารณาจากกฎ 1 2 และ 3 (“problem is spark plugs” “problem is battery or cables” และ “problem is starter motor” ตามลำดับ) รูปที่ 17.9-(ก) แสดงสถานะเริ่มต้นของระบบ รายละเอียดการทำงานแต่ละขั้นตอนของระบบมีดังต่อไปนี้

- **เป้าหมายย่อย W** กฎ 1 2 และ 3 จะถูกพิจารณา โดยระบบเริ่มต้นจากกฎที่มีลำดับน้อยกว่า นั่นคือกฎ 1 (ตามหลักการลดข้อขัดแย้ง รายละเอียดจะได้กล่าวถึงในหัวข้อต่อไป) ดังนั้นข้อเท็จจริง A B และ W จะถูกป้อนเข้าสู่ฐานข้อมูลของระบบ ดังแสดงในรูปที่ 17.9-(ข) ขณะนี้ระบบมีเป้าหมายย่อย (subgoal) เป็น W นั่นคือ “problem is spark plugs” (ตัวแปร Z ในขณะนี้ถูกแทนด้วย ‘spark plugs’) ระบบจะทำการอนุมานว่า ‘spark plugs’ นั้นเสียจริงหรือไม่เป็นลำดับต่อไป
- **เป้าหมายย่อย A** พิจารณาข้อเท็จจริงในฐานข้อมูล ซึ่งประกอบไปด้วย A B และ W ระบบจะทำการอนุมานข้อเท็จจริง A (“engine is getting gas”) ในกรณีนี้เป้าหมายย่อยของระบบคือ A เมื่อระบบทำการตรวจสอบกฎที่มีอยู่ จะได้ว่ากฎ 4 มีเป้าหมายตรงกับที่ต้องการ ดังนั้นกฎ 4 จะถูกเรียกใช้ พร้อมทั้งข้อเท็จจริง D E ถูกป้อนเข้าสู่ฐานข้อมูล (เป้าหมายย่อย A อยู่ในฐานข้อมูลอยู่แล้ว) พิจารณากฎ 4 ซึ่งจะเป็นจริงได้ก็ต่อเมื่อเงื่อนไขหรือข้อเท็จจริงในส่วน IF ซึ่งได้แก่ D และ E ต้องเป็นจริง สถานะฐานข้อมูลของระบบแสดงในรูปที่ 17.9-(ค)
- พิจารณาข้อเท็จจริงที่อยู่ในฐานข้อมูลของระบบ จะเห็นได้ว่ามีข้อเท็จจริงอยู่ 3 ข้อที่ไม่สามารถอนุมานจากกฎใดๆ ในฐานความรู้ได้ นั่นคือ D E และ B โดยระบบสามารถสอบถามข้อเท็จจริงทั้ง 3 จากผู้ใช้ได้ กล่าวคือ
 - “Gas in fuel tank?”
 - “Gas in carburetor?”
 - “Engine starts?”

ถ้าผู้ใช้ยืนยันคำตอบทั้งหมดว่าเป็นจริง ระบบจะสามารถอนุมานได้ว่าเป้าหมายย่อย A เป็นจริง และนำไปสู่เป้าหมาย W ที่เป็นจริงด้วย ข้อเท็จจริงทั้งหมดในฐานข้อมูลได้รับการพิสูจน์ว่าเป็นจริง ระบบจะตอบผู้ใช้ได้ว่า “problem is W” หรือ “problem is spark plugs” นั่นเอง

ตัวอย่างการอนุมานแบบลู่ไขย้อนกลับข้างต้น เป็นระบบที่เล็กและไม่ยุ่งยาก ในการใช้งานจริงนั้นปัญหาจะมีเงื่อนไขความซับซ้อนมากยิ่งขึ้น ในกรณีตัวอย่างนี้ ถ้าระบบไม่สามารถไปสู่เป้าหมายย่อย W ได้สำเร็จ ระบบจะทำการพิจารณากฎถัดไปแทน จะเห็นได้ว่าลำดับของการวางกฎจะมีผลต่อการทำงาน เมื่อกฎเหล่านี้มีจำนวนมาก



รูปที่ 17.9: ตัวอย่างการอุปมานแบบลูกโซ่ย้อนกลับของระบบผู้เชี่ยวชาญช่วงข้อมรด

ระบบอาจต้องการเทคนิคการค้นหา (searching) ที่มีประสิทธิภาพมากกว่านี้

วิธีการอุปมานทั้งสองแบบสะท้อนแนวคิดการแก้ปัญหาที่แตกต่างกัน ถ้าผู้เชี่ยวชาญต้องการข้อมูลในการตัดสินใจก่อน กล่าวคือทำการวิเคราะห์แล้วแปลความหมาย วิธีลูกโซ่ไปข้างหน้าจะเป็นวิธีที่เหมาะสม แต่ถ้าผู้เชี่ยวชาญเริ่มต้นจากการตั้งสมมุติฐานและพยายามที่จะหาข้อเท็จจริงเพื่อมาพิสูจน์สมมุติฐานนั้นๆ วิธีการลูกโซ่ย้อนกลับจะเหมาะสมที่สุด ตัวอย่างของระบบที่ใช้วิธีการลูกโซ่ไปข้างหน้าก็คือ DENDRAL ซึ่งเป็นระบบผู้เชี่ยวชาญที่ใช้ในการวิเคราะห์โครงสร้างโมเลกุลของดินที่ไม่รู้จักมาก่อน โดยการวิเคราะห์ข้อมูลสเปกตรัมของมวลดิน ระบบที่ใช้วิธีลูกโซ่ย้อนกลับมักจะใช้กับระบบที่ทำการตรวจอาการ เช่น MYCIN เป็นระบบผู้เชี่ยวชาญทางด้านแพทย์สำหรับตรวจสอบการติดเชื้อโรคของเม็ดเลือด

โดยปกติแล้วตัวเลือกระบบผู้เชี่ยวชาญจะใช้วิธีการอุปมานทั้งสองวิธีร่วมกัน อย่างไรก็ตาม ตัวอย่างหลักที่ใช้มักจะเป็นวิธีการลูกโซ่ย้อนกลับ วิธีการลูกโซ่ไปข้างหน้าจะถูกใช้ก็ต่อเมื่อมีการเพิ่มข้อเท็จจริงใหม่เข้าสู่ระบบ

17.4 การแก้ปัญหาข้อขัดแย้ง

มีอยู่หลายๆ โอกาสที่มีข้อขัดแย้งเกิดขึ้นในระบบผู้เชี่ยวชาญ ถ้าตัวอุปมานเจอเงื่อนไข IF ที่ตรงตามข้อเท็จจริงในฐานข้อมูล แต่ปรากฏว่ากฎที่ถูกเรียกใช้เกิดความขัดแย้งกับกฎอื่นๆ โดยที่ตัวอุปมานจะต้องเลือกใช้กฎได้เพียงกฎเดียวเท่านั้น ในวิธีลูกโซ่ไปข้างหน้า กฎที่ขัดแย้งกันจะถูกเรียกใช้ทั้งหมด เนื่องจากข้อเท็จจริงที่ได้จากเงื่อนไข IF จะปรากฏทั้งหมดในฐานข้อมูล การเรียกใช้กฎจะมีลำดับก่อนหลังตามที่กฎถูกจัดวางตำแหน่งอยู่ วิธีแก้ไขที่ง่ายที่สุดก็คือการจัดตั้งเป้าหมายขึ้นมา โดยถ้าเป้าหมายบรรลุ จะมีผลให้ยุติการเรียกใช้กฎใดๆ ที่เหลือทั้งหมด อย่างไรก็ตาม วิธีดังกล่าวจะให้ผลที่แตกต่างกันถ้าลำดับของกฎที่ไม่เหมือนกัน อาจจะไปสู่ข้อสรุปสุดท้ายที่ผิดพลาดได้ กล่าวคือลำดับในการจัดวางกฎยังมีผลกระทบต่อระบบผู้เชี่ยวชาญ วิธีการอื่นๆ ในการแก้ปัญหาข้อขัดแย้งมีเช่น [Shirai and Tsuji, 1982]

- **ทำการเรียกใช้กฎที่มีความสำคัญสูงสุด** การจัดลำดับความสำคัญอาจทำได้โดยการจัดลำดับตำแหน่งของ

กฎในฐานความรู้ เทคนิควิธีนี้ใช้ได้กับระบบผู้เชี่ยวชาญที่มีประมาณ 100 กฎ อย่างไรก็ตาม ในบางระบบ ข้อมูลควรจะต้องมีการประมวลผลล่วงหน้าถึงลำดับความสำคัญ ยกตัวอย่างเช่นระบบที่ปรึกษาทางการแพทย์ ซึ่งมีการจัดลำดับความสำคัญดังนี้ [Durkin, 1994]

Goal 1.	Prescription is? Prescription
RULE 1	Meningitis Prescription 1
(Priority 100)	
IF	Infection is Meningitis
AND	The Patient is a Child
THEN	Prescription is Number_1
AND	Drug Recommendation is Ampicillin
AND	Drug Recommendation is Gentamicin
AND	Display Meningitis Prescription 1
RULE 2	Meningitis Prescription 2
(Priority 90)	
IF	Infection is Meningitis
AND	The Patient is a Adult
THEN	Prescription is Number_2
AND	Drug Recommendation is Penicillin
AND	Display Meningitis Prescription 2

ตัวอย่างข้างต้นแสดงการขัดแย้งกันของกฎ ว่าด้วยโรคเยื่อหุ้มสมอง (และไขสันหลัง) อักเสบหรือ meningitis ในเงื่อนไขของโรคดังกล่าวมีกฎ 2 ข้อที่แตกต่างกันตรงที่ กฎ 1 ไว้ใช้สำหรับกรณีที่ผู้ป่วยเป็นเด็กและกฎ 2 ไว้ใช้สำหรับผู้ป่วยที่เป็นผู้ใหญ่ ดังนั้นระบบจะทำการอนุมานผลของแต่ละกฎต่างกัน โดยกำหนดให้กฎ 1 มีความสำคัญสูงกว่ากฎ 2 เมื่อเงื่อนไขว่าด้วยโรคเยื่อหุ้มสมองอักเสบเกิดขึ้น ระบบจะพิจารณากฎ 1 (ผู้ป่วยเป็นเด็ก) ว่าสำคัญกว่าเสมอ

- **ทำการเรียกใช้กฎที่เป็นเฉพาะอย่าง** วิธีนี้เป็นที่รู้จักในนามของกลยุทธ์การเปรียบเทียบที่ยาวที่สุด (longest matching strategy) หลักการก็คือกฎที่เป็นเฉพาะอย่างจะให้ข้อเท็จจริงที่มากกว่ากฎอื่นๆ ไป ยกตัวอย่างเช่น

Rule 1:
IF the season is autumn
AND the sky is cloudy
AND the forecast is rain
THEN the advice is 'stay home'

Rule 2:
IF the season is autumn
THEN the advice is 'take an umbrella'

กฎ 1 เป็นกฎที่มีความเฉพาะมากกว่ากฎ 2 เพราะมีเงื่อนไขที่ยาวกว่า ดังนั้นถ้ามีข้อเท็จจริงแต่เพียงว่า “season is autumn” กฎ 2 จะถูกเรียกใช้ แต่ถ้ามีข้อเท็จจริง “the season is autumn” “the sky is cloudy” และ “the forecast is rain” กฎ 1 จะถูกเรียกใช้

- **ทำการเรียกใช้กฎที่มีการใช้ข้อเท็จจริงล่าสุดจากฐานข้อมูล** วิธีการนี้จะต้องมีการบันทึกเวลาของข้อเท็จจริงในฐานข้อมูล และเลือกเรียกใช้กฎล่าสุด ยกตัวอย่างเช่น

Rule 1:

IF the forecast is rain [08:16 PM 11/25/96]
THEN the advice is 'take an umbrella'

IF the weather is wet [10:18 AM 11/26/96]
THEN the advice is 'stay home'

สมมติว่าทั้งกฎ 1 และ 2 ถูกเลือกจากฐานข้อมูล ในกรณีนี้กฎ 2 จะถูกเรียกใช้เนื่องจากข้อเท็จจริง 'the weather is wet' ถูกบันทึกไว้หลังจาก 'the forecast is rain' เทคนิควิธีนี้มีประโยชน์มากในระบบผู้เชี่ยวชาญแบบเวลาจริง ที่ซึ่งข้อเท็จจริงต่างๆ ถูกปรับเปลี่ยนอยู่ตลอดเวลา

การแก้ปัญหาข้อขัดแย้งของระบบผู้เชี่ยวชาญข้างต้นเป็นวิธีการที่ง่ายและสร้างจริงได้อย่างไม่ยุ่งยาก วิธีดังกล่าวสามารถแก้ปัญหาได้ในระดับหนึ่ง อย่างไรก็ตาม เมื่อระบบมีขนาดใหญ่และซับซ้อนมากขึ้น การจัดการกฎต่างๆ จะมีความยุ่งยากขึ้นตามลำดับ การปรับปรุงประสิทธิภาพของระบบผู้เชี่ยวชาญสามารถทำได้ด้วยการใช้อภิปความรู้ (metaknowledge) หรือความรู้เกี่ยวกับข้อมูลความรู้ที่มีอยู่ (knowledge about knowledge) ในระบบผู้เชี่ยวชาญแบบฐานกฎ อภิปความรู้จะอยู่ในรูปอภิปกฎ (metarule) อภิปกฎจะเป็นการใช้กลยุทธ์ในการเลือกกฎในระบบผู้เชี่ยวชาญ ยกตัวอย่างเช่น

Metarule 1:

Rules supplied by experts have higher priorities than rules supplied by novices.

กฎที่ได้จากผู้เชี่ยวชาญมีความสำคัญกว่ากฎที่ได้จากมือใหม่

Metarule 2:

Rules governing the rescue of human lives have higher priorities than rules concerned with clearing overloads on power system equipment.

กฎที่เกี่ยวข้องกับการช่วยชีวิตมนุษย์มีความสำคัญสูงกว่ากฎที่จัดการกับโหลดเกินในเครื่องมืกระบบไฟฟ้า

กล่าวคืออภิปกฎจะระบุเงื่อนไขที่จะต้องพิจารณาเมื่อเกิดข้อขัดแย้งขึ้นนั่นเอง

17.5 สรุป

ระบบผู้เชี่ยวชาญแบบฐานกฎมีข้อดีและข้อเสียสรุปได้ดังนี้

ข้อดี

- มีการนำเสนอข้อมูลความรู้อย่างธรรมชาติ - เป็นไปตามหลักการแก้ปัญหาของผู้เชี่ยวชาญมนุษย์
- มีโครงสร้างเพียงรูปแบบเดียว - นั่นคือโครงสร้าง IF - THEN ทำให้ง่ายต่อการเข้าใจและการสร้าง
- มีการแยกข้อมูลความรู้จากการประมวลผล - ทำให้สะดวกในการสร้างตัวเลือกระบบผู้เชี่ยวชาญ รวมไปถึงลดความยุ่งยากในการขยายระบบอีกด้วย
- มีความสามารถในการจัดการข้อมูลความรู้ที่ไม่สมบูรณ์หรือไม่แน่นอนได้ - ยกตัวอย่างเช่น

IF season is autumn
AND sky is 'cloudy'
AND wind is low
THEN forecast is clear {cf 0.1}
forecast is drizzle {cf 1.0}
forecast is rain {cf 0.9}

จะเห็นได้ว่าการกำหนดความไม่แน่นอนให้กับแต่ละข้อความในส่วน THEN กฎแต่ละข้อถูกกำหนดเพิ่มเติมด้วยตัวประกอบความแน่นอน (certainty factors หรือ cf) ระบบผู้เชี่ยวชาญใช้ค่าตัวประกอบความแน่นอนแทนระดับความมั่นใจว่าข้อสรุปจากกฎนั้นๆ เป็นจริงเท่าใด

ข้อเสีย

- ไม่สามารถอธิบายความสัมพันธ์ระหว่างกฎได้อย่างชัดเจน - ถึงแม้ว่าแต่ละกฎจะมีความเข้าใจง่ายและมีการอธิบายตัวมันเอง ความสัมพันธ์ระหว่างกฎหลายๆ กฎยังไม่มี ความชัดเจน
- ไม่มีกลวิธีการค้นหาคำตอบที่มีประสิทธิภาพ - ในขั้นตอนการค้นหาข้อเท็จจริง ตัวอุปมาทำการค้นหากฎทุกๆ กฎ ถ้าระบบมีกฎเป็นจำนวนมาก ขั้นตอนดังกล่าวจะกินเวลามาก ทำให้ระบบทำงานได้ช้า ไม่เหมาะแก่การใช้งานในระบบเวลาจริง
- ไม่มีความสามารถในการเรียนรู้ - โดยปกติแล้ว ระบบผู้เชี่ยวชาญไม่มีกลไกในการเรียนรู้จากประสบการณ์ ซึ่งแตกต่างไปจากผู้เชี่ยวชาญมนุษย์ ระบบผู้เชี่ยวชาญไม่สามารถทำการปรับแต่งข้อมูลความรู้ได้ด้วยตนเอง ซึ่งส่วนนี้ยังคงต้องใช้วิศวกรความรู้ในการตรวจสอบและบำรุงรักษาระบบ อย่างไรก็ตาม ได้มีการศึกษาและพัฒนาหลักการการเรียนรู้แบบอัตโนมัติ จากปัญญาเชิงคำนวณที่มีการเรียนรู้แบบต่างๆ เช่น เครือข่ายประสาทเทียม เป็นต้น ทำให้การเรียนรู้ของระบบผู้เชี่ยวชาญเป็นไปอย่างมีประสิทธิภาพมากยิ่งขึ้น รายละเอียดของระบบผสมดังกล่าวจะได้กล่าวถึงในหัวข้อที่เกี่ยวข้องต่อไป



โจทย์คำถาม

- 17.1. ให้ทำการออกแบบสร้างระบบผู้เชี่ยวชาญแบบฐานกฎ สำหรับนักเรียนเพื่อใช้ตัดสินใจว่าจะไปเข้าชั้นเรียนในตอนเช้าหรือไม่ ดังรายละเอียดต่อไปนี้
- วิเคราะห์และเลือกเงื่อนไขต่างๆ ที่เกี่ยวข้อง ยกตัวอย่างเช่น สภาพอากาศ ระยะทาง เวลา ความสำคัญของวิชาเรียน ความยากง่ายของวิชา เวลาเข้าเรียนเพียงพอหรือไม่ เป็นต้น
 - ใช้รูปแบบกฎแบบ IF - THEN สำหรับระบบที่ออกแบบ
 - ทดลองสร้างเงื่อนไขแล้วใช้วิธีลูกโซ่ไปข้างหน้าเพื่อทดสอบหาคำตอบจากระบบ
- 17.2. พิจารณางานซ่อมรถ เมื่อช่างจะทำการซ่อมรถ ช่างจะต้องวิเคราะห์อาการเสียของรถเพื่อหาวิธีการแก้ไข รวมไปถึงอะไหล่ที่ต้องซ่อมหรือเปลี่ยน และเปลี่ยนอย่างไรด้วย ชิ้นส่วนที่ซ่อมหรือเปลี่ยนจะต้องเหมาะสมกับชนิดและรุ่นของรถ ขนาดของเครื่องยนต์ ฯลฯ ให้ยกตัวอย่างการออกแบบระบบที่สามารถตอบคำถามอาการเสียของรถ แล้วทำการสรุปวิธีการซ่อมดังรายละเอียดต่อไปนี้
- ออกแบบฐานความรู้สำหรับซ่อมรถ (ยกตัวอย่างข้อมูลความรู้พอประมาณ)
 - วิเคราะห์และออกแบบเงื่อนไขอาการเสียของรถ (ปกติจะถามโดยช่าง) ในรูปแบบคำถาม-ตอบ
 - แสดงความสามารถของระบบในการเพิ่มข้อเท็จจริงเข้าสู่ฐานข้อมูลได้
 - ทดสอบระบบโดยป้อนอาการของรถ เช่น เครื่องยนต์สตาร์ทติด? แบตเตอรี่มีไฟ? เครื่องยนต์เดินเรียบ? แล้วทำการสรุปอาการและวิธีการซ่อม
- 17.3. พิจารณาออกแบบระบบผู้เชี่ยวชาญในการแยกประเภทสิ่งมีชีวิตดังรายละเอียดต่อไปนี้
- ประเภทของสิ่งมีชีวิตแบ่งเป็นสัตว์บก สัตว์น้ำ สัตว์ครึ่งบกครึ่งน้ำ สัตว์ปีกและสัตว์เลื้อยคลาน
 - ให้ออกแบบข้อเท็จจริงของสัตว์แต่ละประเภท บางข้อเท็จจริงสามารถมีได้ในสัตว์มากกว่า 1 ประเภทขึ้นไป
 - ทดลองระบบโดยสมมุติอินพุตเป็นเช่น นก แมว น้ำ ปลา ตีน ฯลฯ เพื่อดูว่าระบบจะสามารถสรุปประเภทของสิ่งมีชีวิตดังกล่าวได้ถูกต้องหรือไม่ (สังเกตอินพุตที่ทำให้เกิดข้อขัดแย้งเช่นปลาตีน ที่ซึ่งเป็นปลาที่หายใจอยู่บนบกได้ เป็นต้น)
- 17.4. พิจารณาออกแบบระบบตรวจสอบความถูกต้องของประโยค (ภาษาอังกฤษ) ดังรายละเอียดต่อไปนี้
- เลือกชนิดของประโยคเช่น simple tense/past tense/past perfect tense ฯลฯ พร้อมทั้งระบุรายละเอียดไวยากรณ์ของประโยคแต่ละชนิด
 - ออกแบบส่วนประกอบของประโยค เช่น ประธาน กริยา กรรม ฯลฯ รวมถึงรายละเอียดต่างๆ ที่เกี่ยวข้อง เช่นความเป็นเอกพจน์และพหูพจน์ ฯลฯ เพื่อใช้ในการสร้างประโยคให้ถูกต้อง
 - กำหนดตัวอย่างของข้อเท็จจริงสำหรับเก็บในฐานความรู้ เช่น he/she/it ในส่วนประธาน is/am/are does/do was/were และ did ในส่วนของกริยา เป็นต้น
 - ออกแบบกฎสำหรับเขียนประโยคตามชนิดของประโยค เพื่อใช้ในตัวอย่าง
 - ทดสอบระบบด้วยประโยคแบบต่างๆ แล้วพิจารณาว่าระบบจะสามารถตรวจจับว่าประโยคที่เป็นอินพุตนั้นถูกหรือผิดไวยากรณ์ และถ้าผิดแล้วผิดอย่างไร

J. Durkin. *Expert Systems Design and Development*. Prentice Hall, 1994.

Y. Shirai and J. Tsuji. *Artificial Intelligence: Concepts, Technologies and Applications*. John Wiley, New York, 1982.

D.A. Waterman and F. Hayes-Roth. *An Overview of Pattern-Directed Inference Systems*. Academic Press, New York, d.a. waterman and f. hayes-roth edition, 1986.



