

บทที่ 5

เอเจนต์ตรรกะ (Logical Agents)

เอเจนต์ที่อาศัยความรู้ช่วยแก้ปัญหาหรือ Knowledge-based agent มีสิ่งสำคัญที่ต้องพิจารณาก่อนการออกแบบอยู่ 2 สิ่ง คือ การแทนความรู้ (Representation) และกระบวนการให้เหตุผล (Reasoning process) ทั้งสองสิ่งนี้เป็นแนวคิดสำคัญที่ทำให้เอเจนต์นำความรู้มาใช้แก้ปัญหาได้ และเป็นเรื่องหลักของงานในสาขาปัญญาประดิษฐ์ทั้งหมด

เอเจนต์ที่ฉลาดต้องมีความรู้ และใช้เหตุผลเพื่อให้พฤติกรรมของตนประสบความสำเร็จ แต่ความรู้ของเอเจนต์เหล่านี้เป็นความรู้ที่เฉพาะเจาะจง และไม่สามารถพลิกแพลง แต่ก็ยังคงเป็นตัวช่วยที่มีประโยชน์มากในการสรุปให้ได้รับความรู้เกี่ยวกับสถานะของโลกที่เอเจนต์อยู่ การแทนความรู้ต้องใช้ตรรกะเป็นตัวช่วย เอเจนต์ประเภทนี้จึงเรียกอีกอย่างหนึ่งว่า เอเจนต์ตรรกะ (Logical agent)

5.1 เอเจนต์ที่ใช้ความรู้แก้ปัญหา (Knowledge-based agents)

เอเจนต์ที่ต้องอาศัยความรู้ในการแก้ปัญหา มีองค์ประกอบหลักคือ ฐานความรู้ (Knowledge based) เรียกสั้น ๆ ว่า KB ความหมายโดยทั่วไปของฐานความรู้หมายถึงที่รวมของประโยค (Sentence) ต่าง ๆ (คำว่าประโยคในที่นี้เป็นศัพท์เทคนิค ซึ่งมีส่วนคล้ายกับประโยคในภาษาธรรมชาติทั่วไป แต่ก็ไม่เหมือนกัน) โดยแต่ละประโยคอยู่ในรูปของภาษาพิเศษที่ใช้แทนความรู้เหล่านี้ได้ เรียกแบบทั่วไปว่าภาษาตัวแทนความรู้ (Knowledge representation language) ประโยคเหล่านี้เป็นตัวบอกเรื่องราวความรู้ต่าง ๆ เกี่ยวกับโลกที่เอเจนต์นั้นดำรงอยู่

การทำงานกับฐานความรู้มีหลักปฏิบัติพื้นฐาน 2 เรื่อง คือ

1. การเพิ่มความรู้ใหม่ ๆ ลงไปในฐานความรู้ มีชื่อมาตรฐานว่า TELL
2. การถามคำถามเพื่อนำคำตอบมาจากฐานความรู้ มีชื่อมาตรฐานว่า ASK

ทั้ง 2 งานนี้ต้องทำโดยอาศัยการสรุปความ (Inference) ซึ่งหมายถึงการสร้างความรู้ใหม่ขึ้นจากความรู้เดิม สำหรับเอเจนต์ตรรกะแล้ว การสรุปความต้องเป็นไปตามกฎเกณฑ์เบื้องต้นว่าคำตอบของคำถามใด ๆ ที่ถามมานั้น ต้องตอบตามข้อมูลเท่าที่มีอยู่ในฐานความรู้เท่านั้น

Knowledge-based agent มีลักษณะเช่นเดียวกับเอเจนต์ทั่วไป นั่นคือ รับเพอร์เซ็ปเป็นอินพุต แล้วส่งการกระทำ (Action) ออกมาเป็นเอาต์พุต เอเจนต์เป็นผู้ดูแลฐานความรู้ ซึ่งเดิมมีความรู้บรรจุอยู่เรียกว่า Background knowledge เมื่อเอเจนต์รันโปรแกรม จะมีงานที่ต้องกระทำ 2 เรื่องดังนี้

1. บอกฐานความรู้ว่ามีเพอร์เซ็ปอะไร (โดยการ TELL)
2. ถามฐานความรู้ว่าเอเจนต์ควรมีการกระทำอะไร (โดยการ ASK)

โปรแกรมของ Knowledge-based agent เขียนโครงร่างได้ดังนี้

function KB-AGENT(percept) **returns** an action

static: KB, a knowledge base

t, a counter, initially 0, indicating time

TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))

Action \leftarrow ASK(KB, MAKE-ACTION-QUERY(t))

TELL(KB, MAKE-ACTION-SENTENCE(action, t))

t \leftarrow t + 1

return action

จากโปรแกรมโครงร่าง กระบวนการตอบคำถามของโปรแกรมจะต้องมีการใช้เหตุผล โดยอาศัยสถานะปัจจุบันของโลก (สถานะปัจจุบันของเอเจนต์) และใช้ผลลัพธ์ที่จะเกิดจากลำดับการกระทำที่เป็นไปได้หลาย ๆ ทางมาพิจารณาร่วมกัน เมื่อเลือกการกระทำได้แล้ว เอเจนต์จะบันทึกการกระทำนั้นไว้ (TELL) แล้วดำเนินการกระทำ (Execute) การ TELL ครั้งที่สองในโปรแกรมเป็นสิ่งที่จำเป็น เพราะเป็นการบอกให้ฐานความรู้ทราบว่าเอเจนต์ได้ดำเนินการกระทำสำหรับการกระทำนั้นแล้ว

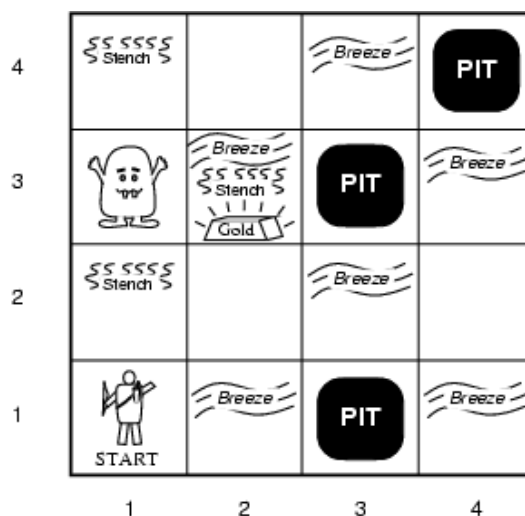
ฟังก์ชันที่ใช้กับกระบวนการ TELL และ ASK ในโปรแกรมเป็นฟังก์ชันการทำงานของเซ็นเซอร์และแอคชูเอเตอร์ที่ผ่านระบบการใช้เหตุผล MAKE-PERCEPT SENTENCE มีหน้าที่รับเพอร์เซ็ป และเวลา แล้วคืนค่าเป็นประโยคที่บอกเกี่ยวกับเพอร์เซ็ปที่เอเจนต์รับมา ณ เวลานั้น เก็บไว้ในฐานความรู้ ส่วน MAKE-ACTION-QUERY รับเวลา t เข้ามาเป็นอินพุต แล้วคืนค่าเป็นประโยคที่ถามว่าต้องใช้ในการกระทำอะไรในขณะนั้น

เอเจนต์ตามโปรแกรมดังกล่าวมีลักษณะคล้ายกับเอเจนต์ชนิด Model-based reflex agent ในบทที่ 2 เพราะมีการทำงานของ TELL และ ASK แปลว่าต้องการรู้สถานะของโลกตลอดเวลา แต่ Knowledge-based agent ไม่ใช่โปรแกรมที่ใช้เพื่อต้องการหาการกระทำธรรมดาเท่านั้น โปรแกรมนี้ทำงานกับความรู้ อาจจะแบ่งตามระดับความรู้ออกเป็น 3 ระดับ ทำให้เกิดมุมมอง Knowledge-based agent ได้ 3 แบบ ดังนี้

1. ระดับความรู้ (Knowledge level) ในระดับนี้มองว่าเอเจนต์รู้อะไรบ้าง กำหนดได้ว่าเอเจนต์รู้อะไร และมีเป้าหมายอะไร เพื่อจะให้เอเจนต์ได้มีการกระทำที่เหมาะสม ตัวอย่างเช่น รถแท็กซี่อัตโนมัติ ขับรถจากวงจรไปวงเวียนใหญ่ และรู้ว่าสะพานพุทธเป็นเส้นทางเดียวที่เชื่อมต่อไปยังวงเวียนใหญ่ได้ง่ายที่สุด จึงใช้เส้นทางนั้น
2. ระดับตรรกะ (Logical level) พิจารณาเอเจนต์ในบทบาทที่แทนความเป็นไปต่าง ๆ ของโลกในรูปของประโยค (ในภาษา Knowledge representation language)
3. ระดับการใช้งาน (Implementation level) ทำงานในระดับสถาปัตยกรรมของเอเจนต์ เช่น ดูว่าเอเจนต์แทนประโยคในระดับกายภาพโดยใช้ตาราง (Table) หรือลิงก์ลิสต์ (Linked list) แทนความรู้ภูมิศาสตร์ด้วยลิงก์ลิสต์ หรือ Pixel maps หรือมีวิธีให้เหตุผลโดยใช้สายอักขระ (Character string) เก็บลงในรีจิสเตอร์ หรือจะใช้สัญญาณผ่านเข้าเครือข่ายเส้นประสาท

5.2 กรณีศึกษา : โลกของ Wumpus (The Wumpus world)

Wumpus world เป็นเกมชนิดหนึ่ง เป็นที่นิยมตั้งแต่สมัยที่ยังไม่มีงานกราฟิกส์แพร่หลาย โลกของ Wumpus คือถ้ำที่ประกอบด้วยห้องจำนวนหลายห้อง แต่ละห้องมีทางเดินเชื่อมถึงกัน ภายในถ้ำมีสัตว์ประหลาดเรียกว่า Wumpus ซ่อนอยู่ในห้องใดห้องหนึ่งคอยกินผู้ที่เข้าไปในห้องนั้น เอเจนต์คือผู้เล่น สามารถยิง Wumpus ตายได้ แต่เอเจนต์มีลูกธนูเพียงดอกเดียวจึงยิงได้ครั้งเดียวเท่านั้นตลอดเกม ห้องบางห้องมีหลุมลึก เรียกว่า Pit เป็นกับดัก คนที่หลงเข้าไปในห้องนี้จะตกลงไปตาย (ยกเว้นตัว Wumpus) ภายในถ้ำมีทองคำ เป้าหมายคือต้องการค้นหาและนำทองออกมาจากถ้ำให้ได้อย่างปลอดภัย เกมนี้จะดูง่ายมาก เมื่อเทียบกับเกมคอมพิวเตอร์สมัยใหม่ แต่จะเป็นสภาพแวดล้อมทดสอบอย่างดีเยี่ยมสำหรับ Intelligent agent รูปที่ 5.1 แสดงตัวอย่างของ Wumpus world ส่วน Task environment ของปัญหา แสดงได้โดยใช้ PEAS ดังนี้



รูปที่ 5.1 Wumpus world

1. เกณฑ์การวัดสมรรถนะ หาเกณฑ์พิจารณาความสำเร็จ โดยให้คะแนนแก่เอเจนต์ตามระดับต่าง ๆ ได้แก่

- ก. +1000 เมื่อเก็บทองได้
- ข. -1000 เมื่อตกหลุม หรือถูก Wumpus กิน
- ค. -1 เมื่อมีการกระทำแต่ละแอคชัน
- ง. +1 เมื่อยิงธนู

2. สภาพแวดล้อมได้แก่ ห้องต่าง ๆ 16 ห้องภายในถ้ำ แทนด้วยตารางขนาด 4x4 ช่อง เอเจนต์จะเริ่มต้นที่ช่อง [1,1] เสมอ แล้วหันหน้าไปทางขวา ตำแหน่งของห้องที่มีทองและ Wumpus เปลี่ยนไปในการเล่นแต่ละครั้งแบบสุ่ม โดยมีการกระจายสม่ำเสมอ (Uniform distribution) ยกเว้นช่อง [1,1] ซึ่งเป็นจุดเริ่มต้นที่ไม่นับ นอกจากนี้ แต่ละช่อง ยกเว้นช่อง [1,1] มีโอกาสที่จะมีหลุมอยู่ด้วยความน่าจะเป็นเท่ากันคือ 0.2

3. แอคชูเอเตอร์ (Actuators) เอเจนต์สามารถเดินตรงไปข้างหน้า เลี้ยวขวา ซ้าย หรืออาจตายได้หากตกลงไปในหลุม หรือถูก Wumpus กิน (แต่ถ้า Wumpus ตายแล้วก็สามารถผ่านห้องนั้นได้อย่างปลอดภัย) เอเจนต์เดินหน้าไม่ได้ถ้าข้างหน้าเป็นผนังถ้ำขวางอยู่ การกระทำอื่นของเอเจนต์ได้แก่ หยิบทอง ยิงธนู (ยิงได้ครั้งเดียวโดยยิงตรงไปข้างหน้า ดังนั้นถ้ายังไม่ถูก Wumpus ลูกธนูก็จะกระทบผนัง)

4. เซ็นเซอร์ เอเจนต์มีเซ็นเซอร์ 5 ชนิดด้วยกัน แต่ละชนิดจะทำให้ได้รับข้อมูลแต่ละเรื่องดังนี้

- ก. ช่องใดมี Wumpus อยู่ ในช่องที่ติดกัน (เฉพาะในแนวฉาก ไม่นับแนวทแยง) เอเจนต์จะได้กลิ่นเหม็น (Stench)

- ข. ช่องที่อยู่ติดกับช่องที่มีหลุม (ไม่นับแนวทแยง) เอเจนต์จะรู้สึกว่ามีลมพัด (Breeze)
- ค. ช่องที่มีทองอยู่ เอเจนต์จะมองเห็นประกาย (Glitter)
- ง. เมื่อเอเจนต์เดินไปพบผนัง จะรับเพอร์เซ็ป เป็น Bump
- จ. เมื่อ Wumpus ถูกยิงตาย เอเจนต์จะได้ยินเสียงกรีดร้อง (Scream) ไม่ว่าเอเจนต์อยู่ในช่องใดก็ตาม

เพอร์เซ็ปต่าง ๆ ที่เอเจนต์ได้รับจะอยู่ในรูปของรายการที่ประกอบด้วยสมาชิก 5 จำนวน ตัวอย่างเช่น ถ้าเอเจนต์ได้รับเพอร์เซ็ปเป็นกลิ่นเหม็น และลมพัด แต่ไม่เห็นประกายทอง ไม่ชนผนัง และไม่ได้ยินเสียงกรีดร้อง เพอร์เซ็ปของเอเจนต์จะเขียนได้ว่า

[Stench, Breeze, None, None, None]

สภาพแวดล้อมในงาน (Task environment) ของเอเจนต์ (ที่บรรยายโดยใช้ PEAS) เป็นสิ่งสำคัญที่ไม่ควรละเลย การละเลยลักษณะของสภาพแวดล้อมตั้งแต่เริ่มต้น จะทำให้เอเจนต์ทำงานได้ลำบาก หลังจากพิจารณาสภาพแวดล้อมแล้ว จึงอาศัยการใช้เหตุผลทางตรรกะมาช่วยแก้ปัญหาให้เอเจนต์ จากตัวอย่างการเล่นเกม Wumpus เป็นจำนวนหลาย ๆ ครั้ง พบว่าส่วนใหญ่แล้ว มีความเป็นไปได้ที่เอเจนต์จะสามารถนำทองกลับออกไปได้อย่างปลอดภัย แต่บางกรณีเอเจนต์ต้องเลือกเอาระหว่างการออกจากถ้ำมือเปล่า กับการเสี่ยงตายค้นหาทอง และมีอยู่ 21% ของเกม ที่สถานการณ์ไม่อำนวยเป็นอย่างยิ่งเนื่องจากทองอยู่ในห้องเดียวกันกับหลุม หรือล้อมรอบด้วยหลุม (ในกรณีนี้เอเจนต์ต้องล้มเหลวแน่นอน แต่ตำแหน่งของทอง และหลุมเกิดจากการสุ่มของเกมที่เอเจนต์ควบคุมไม่ได้)

Knowledge-based agent ที่อยู่ใน Wumpus world มีวิธีสำรวจสภาพแวดล้อมตามตัวอย่างในรูปที่ 5.2 โดยแทนสภาพแวดล้อมด้วยตารางสี่เหลี่ยมและมีหมายเลขกำกับไว้ในช่องแต่ละช่องเพื่อใช้เรียกช่องเหล่านั้น ก่อนอื่น เอเจนต์มีฐานความรู้เบื้องต้นประกอบด้วยกฎ (Rule) ต่าง ๆ เกี่ยวกับสภาพแวดล้อมตามที่กล่าวไว้แล้วในเรื่องของ PEAS โดยเฉพาะอย่างยิ่ง เอเจนต์รู้ว่าตัวเองอยู่ที่ช่อง [1,1] และเป็นช่องที่ปลอดภัย ต่อไปจะพิจารณาวิธีที่เอเจนต์ใช้ความรู้ ผสมกับการรับเพอร์เซ็ปใหม่ ๆ ประกอบกับแอคชันที่เลือกกระทำในขณะที่ดำเนินไปตามเกม

เพอร์เซ็ปแรกของเอเจนต์คือ [None, None, None, None, None] เอเจนต์สามารถสรุปได้ว่าช่องข้างเคียงปลอดภัย รูปที่ 5.2 (a) แสดงความรู้ของเอเจนต์ที่มีต่อสถานะขณะนั้น ในที่นี้จะแทนประโยคในฐานความรู้ด้วยตัวอักษร เช่น B (Breeze) และ OK (ปลอดภัยจากหลุมและ Wumpus) แล้วนำตัวอักษรไปกำกับไว้ในช่องตามรูปที่ 5.2

1,4	2,4	3,4	4,4	A = Agent
				B = Breeze
1,3	2,3	3,3	4,3	G = Glitter, Gold
				Ok = Safe square
1,2	2,2	3,2	4,2	P = Pit
Ok				S = Stench
1,1	2,1	3,1	4,1	V = Visited
Ok A	Ok			W = Wumpus

(a) สถานะเริ่มต้น ได้รับเพอร์เซ็ปเป็น [None, None, None, None, None]

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
Ok	P?		
1,1	2,1 A	3,1	4,1
Ok V	Ok B	P?	

(b) หลังจากเดิน 1 ตา ไปที่ช่อง [2,1] ได้รับเพอร์เซ็ปเป็น [None, Breeze, None, None, None]

รูปที่ 5.2 เอเจนต์เดินตาแรกในเกม Wumpus

จากข้อมูลที่ทราบมาคือ ในช่อง [1,1] ไม่มีทั้งกลิ่นและลม เอเจนต์จึงสรุปความได้ว่า [1,2] และ [2,1] ปลอดภัยไม่มีอันตรายทั้งจาก Wumpus และหลุม จึงใส่สัญลักษณ์ OK ไว้ในช่องดังกล่าว เอเจนต์ที่เดินด้วยความระมัดระวังย่อมเดินไปตามช่องที่รู้ว่า OK เท่านั้น สมมุติว่าเอเจนต์เลือกเดินไปยังช่อง [2,1] ตามรูปที่ 5.2 (b)

ในช่อง [2,1] เอเจนต์พบว่ามีลม แสดงว่าช่องข้างเคียงต้องมีหลุมอยู่ แต่หลุมไม่สามารถอยู่ในช่อง [1,1] ได้จากกฎของเกม ดังนั้นหลุมจึงต้องอยู่ในช่อง [2,2] หรือ [3, 1] หรือทั้งสองช่อง

สัญลักษณ์ P? ที่ลงไว้ในช่องทั้ง 3 ตามรูปที่ 5.2 (b) แสดงความเป็นไปได้ว่าในช่องเหล่านั้นจะมีหลุมอยู่ ขณะนี้จึงเหลือเพียงช่องเดียวเท่านั้นที่แน่ใจว่าปลอดภัย และยังไม่ได้เดินไป จึงตัดสินใจเดินย้อนกลับไป [1,1] แล้วเดินไปยัง [1,2]

1,4	2,4	3,4	4,4	A = Agent
				B = Breeze
1,3	2,3	3,3	4,3	G = Glitter, Gold
				Ok = Safe square
1,2 A Ok S	2,2 Ok	3,2	4,2	P = Pit
				S = Stench
1,1 Ok V	2,1 B Ok V	3,1 P!	4,1	V = Visited
				W = Wumpus

(a) หลังจากเดินตาที่ 3 ได้รับเพอร์เซ็ปต์ [Stench, None, None, None, None]

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 V Ok S	2,2 V Ok	3,2	4,2
1,1 Ok V	2,1 B Ok V	3,1 P!	4,1

(b) หลังจากเดินตาที่ 5 ได้รับเพอร์เซ็ปต์ [Stench, Breeze, Glitter, None, None]

รูปที่ 5.3 เหตุการณ์เมื่อเอเจนต์เดินต่ออีก 2 ขั้นตอน

ในช่อง [1,2] เอเจนต์พบว่ามีกลิ้น ดังนั้นเพอร์เซ็ปต์ขณะนี้คือ [Stench, None, None, None, None] ดังรูปที่ 5.3 (a) กลิ่นที่พบบอกให้รู้ว่าต้องมี Wumpus อยู่ในช่องข้างเคียง แต่ต้องไม่ใช่ [1,1] ตามกฎของเกม และไม่ใช่ [2,2] เพราะถ้า Wumpus อยู่ที่ [2,2] เอเจนต์จะต้องได้

กลืนขณะที่อยู่ในช่อง [2,1] ดังนั้น เอเจนต์จึงสรุปได้ว่า Wumpus ต้องอยู่ในช่อง [1,3] จึงใส่สัญลักษณ์ W! ลงไปเพื่อระบุตำแหน่งของ Wumpus ไว้

นอกจากนี้ การที่ในช่อง [1,2] ไม่มีลม ก็แสดงว่าไม่มีหลุมอยู่ใน [2,2] แต่จากการสรุปครั้งแรกว่าอาจจะมีหลุมอยู่ใน [2,2] หรือ [3,1] ดังนั้นจึงแปลว่าหลุมต้องอยู่ที่ [3,1] ทั้งหมดนี้เป็นการสรุปความที่เรียกว่า Inference เป็นสิ่งที่ทำได้ยาก เนื่องจากต้องรวมเอาความรู้ที่รวบรวมมาหลายช่วงเวลา ในหลายสถานที่เข้าด้วยกัน และสังเกตจากเพอร์เซ็ปต์ที่มี และที่หายไปในแต่ละขั้นตอน การพิจารณา การสรุปความนี้เป็นสิ่งที่เกินความสามารถของสัตว์หลายประเภท แต่เน่เป็นการใช้เหตุผลขั้นพื้นฐานของเอเจนต์ตรรกะ เมื่อทำการสรุปความนี้แล้ว จะได้ข้อสรุป หรือ Conclusion

ขณะนี้เอเจนต์รู้ว่าช่อง [2,2] ปลอดภัยจากหลุม และ Wumpus จึงสามารถเดินไปได้ ตัวอย่างจะไม่แสดงความรู้ของเอเจนต์ที่มีต่อสถานะ [2,2] แต่จะสมมุติว่าเอเจนต์เดินไปพบทองใน [2,3] และจบเกมเพียงเท่านี้

แต่ละครั้งที่เอเจนต์สามารถดึงข้อสรุปมาจากข้อมูลที่มีอยู่ ถ้าข้อมูลที่นำมาใช้นั้นถูกต้อง ข้อสรุปนั้นจะต้องถูกต้องเสมอ นี่เป็นคุณสมบัติพื้นฐานของการใช้เหตุผลทางตรรกะ

5.3 ตรรกะ (Logic)

ฐานความรู้ประกอบด้วยประโยคจำนวนหนึ่ง การเขียนประโยคเหล่านี้ต้องมีรูปแบบไวยากรณ์ (Syntax) ที่ถูกต้องแน่นอนตามแบบของแต่ละภาษาที่นำมาใช้ การเขียนนี้จะต้องมีความหมายชัดเจนในเชิงคณิตศาสตร์ทั่วไปได้ด้วย ตัวอย่างเช่น ถ้าเขียนประโยคในรูป

$$x + y = 4$$

จะเป็นประโยคที่ถูกต้อง แต่ถ้าเขียนดังนี้

$$X4y+=$$

เป็นประโยคที่เขียนไม่ถูกต้อง สื่อความหมายทางคณิตศาสตร์ไม่ได้ รูปแบบไวยากรณ์ของภาษาทางตรรกะ (และคณิตศาสตร์) เหล่านี้ มักจะออกแบบมาเพื่อใช้ในตำรา และงานวิจัย จึงมีหลากหลายรูปแบบต่างกันไป อาจจะใช้ตัวอักษรกรีกทั้งตัวเล็กและตัวใหญ่ รวมถึงสัญลักษณ์ทางคณิตศาสตร์อื่น ๆ บางครั้งเขียนอยู่ในรูปของแผนผัง (Diagram) ที่ต้องใช้ลูกศร และวงกลมมาช่วย แต่ไม่ว่าจะใช้แบบใดก็ตาม ประโยคต่าง ๆ ที่อยู่ในฐานความรู้ของเอเจนต์นั้นก็คือคุณสมบัติทางกายภาพของเอเจนต์จริง ๆ การใช้เหตุผลของเอเจนต์เกิดขึ้นได้โดยการสร้างและจัดการกับคุณสมบัติเหล่านี้

ตรรกะจะให้นิยามความหมาย (Semantic) ของภาษาด้วย และนิยามเหล่านี้มีความเที่ยงตรง ไม่พลิกแพลง ความหมายของภาษาเป็นตัวกำหนดนิยามค่าความจริงให้กับประโยค ตัวอย่างเช่น ถ้าความหมายกำหนดไว้ว่าประโยคคณิตศาสตร์ $x+y=4$ จะเป็นจริงในโลกแห่งหนึ่ง เมื่อ x เป็น 2 และ y เป็น 2 และจะเป็นเท็จเมื่อ x เป็น 1 และ y เป็น 1 สำหรับตรรกะมาตรฐานแล้ว แต่ละประโยคจะให้ค่าเป็นจริงหรือเท็จได้เพียง 2 ค่าไม่ว่าอย่างไรอย่างหนึ่งเท่านั้น ไม่มีการแบ่งรับแบ่งสู้ อยู่กลาง ๆ ระหว่างจริงกับเท็จ

เพื่อให้คำกล่าวเป็นไปด้วยความเที่ยงตรงแล้ว ในที่นี้จะใช้คำว่า โมเดล (Model) แทนคำว่า โลกโลกหนึ่ง หรือกล่าวว่า m เป็นโมเดลของ α หมายความว่า ประโยค α เป็นจริงในโมเดล m จะเห็นว่า โมเดลเป็นโลกที่มีความจำกัดมากกว่าโลกจริง มีความเป็นนามธรรมในแบบของคณิตศาสตร์ ซึ่งตัดสินความเป็นจริงหรือเท็จของประโยคได้แบบง่าย ๆ ไม่ซับซ้อน ตัวอย่างเช่น ให้โมเดลเป็นโลกของการเล่นไพ่บริดจ์ อาจมองว่า x และ y แทนจำนวนชายและหญิงที่นั่งรอบวงไพ่ ดังนั้น ประโยค $x+y = 4$ จะเป็นจริงเมื่อมีจำนวนคนเล่นไพ่ครบ 4 คน แต่สำหรับโลกจริงแล้ว x, y อาจจะเป็นจำนวนจริงใด ๆ ก็ได้

การใช้เหตุผลทางตรรกะจะมีการตกทอดประโยคทางตรรกะ (Entailment) นั่นคือเมื่อมีประโยคหนึ่งแล้ว จะเกิดอีกประโยคหนึ่งตามมาโดยใช้ตรรกะเข้ามาช่วยทำให้เกิดประโยคใหม่นี้ขึ้น เมื่อใช้วิธีการเขียนแบบคณิตศาสตร์ จะเขียนสัญลักษณ์ว่าประโยค α ตกทอดไปทำให้เกิดประโยค β ได้ดังนี้ $\alpha \models \beta$

นิยามของ $\alpha \models \beta$ กำหนดไว้ว่า $\alpha \models \beta$ ก็ต่อเมื่อในโมเดลใด ๆ ซึ่งมีประโยค α เป็นจริงแล้วจะมี β เป็นจริง

อาจกล่าวอีกอย่างหนึ่งว่า ถ้า α เป็นจริงแล้ว β ต้องเป็นจริง ความสัมพันธ์ของการตกทอดเป็นเช่นเดียวกันกับความสัมพันธ์ทางคณิตศาสตร์ เช่น ประโยค $x+y = 4$ ตกทอดไปสู่ประโยค $4 = x+y$ และหากจะพิจารณาง่าย ๆ โดยทั่วไปว่า ฐานความรู้คือประโยคใด ๆ ก็อาจกล่าวได้ว่า ฐานความรู้ตกทอดไปเป็นประโยค

ค่าความจริงจากฐานความรู้เป็นเท็จเมื่อเรื่องนั้นขัดแย้งกับสิ่งที่เอเยนต์รับรู้มา เช่นจากตัวอย่างโมเดล Wumpus world ในรูปที่ 5.3 ฐานความรู้บอกค่าความจริงเป็นเท็จในกรณีถามว่าช่อง [1,2] มีหลุมหรือไม่ (เป็นเท็จเพราะไม่มีหลุม) เนื่องจากช่อง [1,1] ไม่มีลมพัด

การตกทอดนี้เมื่อดำเนินไปเรื่อย ๆ สามารถทำให้เกิดผลสรุปขึ้นมาได้ นั่นคือเกิดการสรุปความขึ้น เขียนเป็นสัญลักษณ์ว่า

$$KB \vdash \alpha$$

หมายความว่าอัลกอริทึมในการสรุปความ i สามารถทำให้เกิดประโยค α ออกมาจากฐานความรู้ (อ่านว่า i Derives α จาก KB หรือ α ถูก Derive จาก KB โดยอัลกอริทึม i)
เครื่องหมาย |- หมายถึงการ Derive

อัลกอริทึมในการสรุปความที่ Derive เฉพาะประโยคที่ตกทอดมาทางตรรกะ เรียกว่า Sound preserving หรือ Truth preserving นั่นคือจะได้ข้อสรุปที่ถูกต้องแน่นอนอย่างสมบูรณ์แบบ

จะเห็นว่ากระบวนการใช้เหตุผลซึ่งสรุปได้ข้อสรุปที่ถูกต้องแน่นอนต้องมาจากพื้นฐานว่าในโลก (หรือโมเดล) นั้น มีคำกล่าวอ้าง (Premise) ที่ถูกต้องเป็นจริงอยู่ก่อน ดังนั้นถ้าฐานความรู้เป็นจริงแล้ว ประโยคใด ๆ ที่ Derive มาจากฐานความรู้โดยผ่านกระบวนการสรุปความที่สมบูรณ์แบบย่อมเป็นจริงแน่นอน

5.4 ตรรกศาสตร์ประพจน์ (Propositional Logic)

ตรรกศาสตร์ประพจน์เป็นตรรกะที่ง่ายที่สุด บางครั้งเรียกว่า Boolean logic สามารถนำมาเขียนแทนความเป็นไปในโลกได้ แม้ไม่ละเอียดมากนัก แต่ก็ใช้เป็นพื้นฐานขั้นแรกของการศึกษาการใช้เหตุผลทางตรรกะที่ดีได้

5.4.1 ไวยากรณ์ (Syntax)

ไวยากรณ์ของตรรกศาสตร์ประพจน์กำหนดรูปแบบของประโยคไว้ดังนี้

5.4.1.1 ประโยคเดี่ยว (Atomic sentence) เป็นหน่วยเดี่ยวทางไวยากรณ์ที่แบ่งแยกไม่ได้ ประกอบด้วยสัญลักษณ์เดี่ยว เรียกว่าประพจน์ (Proposition symbol) ใช้แทนสิ่งต่าง ๆ ที่ต้องการบอกว่าเป็นจริง หรือเท็จ เขียนด้วยตัวอักษรตัวใหญ่ เช่น P, Q, R เป็นต้น ตัวอย่างเช่น $W_{1,3}$ แทนเหตุการณ์ที่มี Wumpus อยู่ในช่อง $[1,3]$ สัญลักษณ์ประพจน์ที่มีความหมายเฉพาะและไม่เปลี่ยนแปลงมี 2 ตัวคือ True และ False

5.4.1.2 ประโยคเชิงซ้อน (Complex sentence) คือประโยคที่สร้างขึ้นมาจากประโยคเดี่ยวหลายประโยคนำมาเชื่อมกันด้วยตัวเชื่อมทางตรรกะ (Logical connectives) ซึ่งได้แก่

1) \neg (Not) บอกความหมายเชิงปฏิเสธของประพจน์ที่ประกอบอยู่ข้างหลัง เช่น $\neg W_{1,3}$ หมายความว่าไม่มี Wumpus อยู่ในช่อง $[1,3]$ กล่าวว่าเป็นนิเสธ (Negation) ของ $W_{1,3}$ การเขียนค่าคงที่ที่เป็นข้อความซึ่งเรียกว่า Literal อาจจะเป็นประโยคเดี่ยวบวก (Positive literal) หรือประโยคเดี่ยวลบ (Negative literal) ก็ได้

2) \wedge (And) ประโยคที่เชื่อมด้วยเครื่องหมาย \wedge เช่น $W_{1,3} \wedge P_{3,1}$ เรียกว่า Conjunction แต่ละส่วนใน Conjunction เรียกว่า Conjunct

3) \vee (Or) ประโยคที่เชื่อมด้วยเครื่องหมาย \vee เช่น $(W_{1,3} \wedge P_{3,1}) \vee W_{2,2}$ เรียกว่า Disjunction ส่วนของการเชื่อมเช่น $(W_{1,3} \wedge P_{3,1})$ และ $W_{2,2}$ เรียกว่า Disjunct

4) \Rightarrow (Imply) ประโยคที่เชื่อมด้วยเครื่องหมาย \Rightarrow เช่น $(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$ เรียกว่า Implication ประกอบด้วยส่วนทางซ้ายมือของเครื่องหมาย \Rightarrow เรียกว่า Premise หรือข้อกล่าวอ้าง ในที่นี้คือ $(W_{1,3} \wedge P_{3,1})$ ส่วนทางขวามือเรียกว่าข้อสรุป (Conclusion) ในที่นี้คือ $\neg W_{2,2}$ Implication เรียกอีกอย่างหนึ่งว่ากฎ (Rule) หรือ if-then statement

5) \Leftrightarrow (If and only if) เช่น ประโยค $W_{1,3} \Leftrightarrow \neg W_{2,2}$ เรียกว่า Biconditional หรือประโยคเงื่อนไขสองทาง นั่นคือส่วนซ้ายมือจะเป็นจริงก็ต่อเมื่อส่วนขวามือเป็นความจริงด้วย

รูปแบบไวยากรณ์ของประโยคตรรกศาสตร์ประพจน์ที่เชื่อมกันด้วยตัวเชื่อมทางตรรกะต่าง ๆ แสดงไว้ในรูปที่ 5.4 จะเห็นว่าไวยากรณ์เข้มงวดเรื่องการใช้วงเล็บเล็ก () เป็นอย่างมาก

<i>Sentence</i>	\rightarrow	<i>AtomicSentence</i> <i>ComplexSentence</i>
<i>AtomicSentence</i>	\rightarrow	True False Symbol
<i>Symbol</i>	\rightarrow	P Q R ...
<i>ComplexSentence</i>	\rightarrow	\neg <i>Sentence</i> (<i>Sentence</i> \wedge <i>Sentence</i>) (<i>Sentence</i> \vee <i>Sentence</i>) (<i>Sentence</i> \Rightarrow <i>Sentence</i>) (<i>Sentence</i> \Leftrightarrow <i>Sentence</i>)

รูปที่ 5.4 ไวยากรณ์ของประโยคตรรกศาสตร์ประพจน์ในรูปแบบ BNF (Backus-Naur Form)

ทุกประโยคที่เชื่อมกันโดยใช้ตัวเชื่อมทางตรรกะแบบ Binary connective จะต้องอยู่ภายในวงเล็บเพื่อกันความสับสนคลุมเครือ เช่นเขียน $(A \wedge B) \Rightarrow C$ แทนที่จะเขียน $A \wedge B \Rightarrow C$

เป็นต้น แต่เพื่อให้อ่านเข้าใจง่ายขึ้น ในที่นี้จึงขอตัดวงเล็บออกไป แล้วเน้นที่ลำดับของการทำงาน ก่อนหลัง (Precedence) ของตัวเชื่อมแทน ลำดับการทำงานก่อนหลังของตัวเชื่อมสำหรับ ตรรกศาสตร์ประพจน์เรียงจากสูงสุดไปต่ำสุดตามลำดับซ้ายไปขวาดังนี้

$$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$$

ดังนั้น ประโยค

$$\neg P \vee Q \wedge R \Rightarrow S$$

จึงมีความหมายเท่ากับประโยค

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

5.4.2 ความหมาย (Semantic)

ความหมายของประโยคในทางคอมพิวเตอร์ใช้คำว่า Semantic เป็นตัวให้นิยาม กฎเกณฑ์การตัดสินใจหาค่าความจริงของประโยคที่อยู่ในโมเดล สำหรับตรรกศาสตร์ประพจน์ โมเดลเป็นผู้กำหนดและปรับเปลี่ยนค่าของสัญลักษณ์ประพจน์ที่ใช้ว่าจะเป็นจริงหรือเท็จ โดยตัดสินใจตามสภาพเหตุการณ์ที่เกิดขึ้นของโลก ตัวอย่างเช่น ถ้าประโยคในฐานความรู้ใช้สัญลักษณ์ ประพจน์แทนการมีลมอยู่ในช่องต่าง ๆ ดังนี้ $P_{1,2}$, $P_{2,2}$, และ $P_{3,1}$ โมเดลหนึ่งที่สามารถเป็นไปได้ก็คือ

$$M_1 = \{ P_{1,2} = \text{False}, P_{2,2} = \text{False}, P_{3,1} = \text{True} \}$$

ทั้งนี้การใช้สัญลักษณ์ประพจน์ 3 ตัวตามตัวอย่าง ทำให้โมเดลที่อาจเป็นไปได้ทั้งหมดมีอยู่เป็นจำนวน $2^3 = 8$ โมเดล (คิดจากหลักการเรียงลำดับ)

เมื่อกำหนดโมเดลมาให้ จากความหมายของตรรกศาสตร์ประพจน์ จะทำให้รู้ว่าจะหาค่าความจริงได้อย่างไร ประโยคทั้งหมดในฐานความรู้ที่แสดงความเป็นไปของโมเดลจะประกอบขึ้นจากประโยคเดียว และตัวเชื่อมทางตรรกะทั้ง 5 ตัว สำหรับประโยคเชิงซ้อนจะต้องมีกฎการหาค่าความจริงมาใช้ร่วมด้วย เช่นกฎกล่าวไว้ว่า สำหรับประโยค s ใด ๆ ที่อยู่ในโมเดล m แล้ว ประโยค $\neg s$ เป็นจริง ก็ต่อเมื่อประโยค s เป็นเท็จ เป็นต้น

กฎที่นำมาใช้กับประโยคเชิงซ้อน เมื่อมีตัวเชื่อมต่าง ๆ เขียนสรุปไว้ในตารางค่าความจริง (Truth table) ซึ่งจะแจกแจงค่าความจริงของสัญลักษณ์ที่ใช้แต่ละตัว และค่าหลังจากที่เชื่อมกันด้วยตัวเชื่อมแล้ว ดังตารางที่ 5.1 ตารางนี้ใช้หาค่าความจริงของประโยคต่าง ๆ ในโมเดล ได้โดยทำงานวนซ้ำแบบ Recursive ตัวอย่างเช่น ประโยค $\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$ ในโมเดล m_1 หาค่าความจริงได้ว่า $\text{True} \wedge (\text{False} \vee \text{True}) = \text{True} \wedge \text{True} = \text{True}$

เนื่องจากฐานความรู้ประกอบด้วยประโยคจำนวนหนึ่ง จึงอาจมองฐานความรู้ว่าเป็นการนำประโยคมารวมกันแบบ Conjunction (คือการเชื่อมด้วย \wedge) ถ้าฐานความรู้ KB เริ่มต้น

จากที่ว่างเปล่าไม่มีอะไร แล้วเพิ่มความรู้ให้ทีละประโยคโดยใช้วิธี $TELL(KB, s_1), \dots, TELL(KB, s_n)$ แล้ว จะได้

$$KB = s_1 \wedge \dots \wedge s_n$$

ดังนั้น ฐานความรู้กับคำว่าประโยค (Sentence) จึงถูกนำมาใช้แทนกันได้ และมีความหมายเดียวกัน

ตารางที่ 5.1 ตารางค่าความจริงสำหรับการใช้ตัวเชื่อมทางตรรกะ

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

5.4.3 ฐานความรู้อย่างง่าย

ต่อไปจะทดลองสร้างส่วนหนึ่งของฐานความรู้ของ Wumpus world เพื่อให้เข้าใจง่ายขึ้น จะกล่าวถึงเฉพาะกรณีการค้นหาหลุมในเกมเท่านั้น

กำหนดให้ i, j แสดงหมายเลขช่อง สัญลักษณ์ที่ใช้จะมีค่าดังนี้

ให้ $P_{i,j}$ เป็นจริง ถ้ามีหลุม (Pit) อยู่ในช่อง $[i, j]$

ให้ $B_{i,j}$ เป็นจริง ถ้ามีลมพัด (Breeze) อยู่ในช่อง $[i, j]$

ประโยคต่าง ๆ ในฐานความรู้มีดังนี้ (แต่ละประโยคมีหมายเลขกำกับไว้เพื่อความสะดวกในการอ้างถึงประโยคนั้น ๆ)

1. ไม่มีหลุมในช่อง $[1,1]$

$$R1: \quad \neg P_{1,1}$$

2. ในช่อง ๆ หนึ่งจะมีลมก็ต่อเมื่อมีหลุมอยู่ในช่องข้าง ๆ กรณีนี้ต้องแจกแจงกฎของแต่ละช่อง

$$R2: \quad B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$$R3: \quad B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$

กฎ R1 ถึง R3 เป็นจริงสำหรับ Wumpus world ทุกโมเดล ต่อไปสร้างความรู้ตามที่เอเจนต์รับเพอร์เซ็ปต์เข้ามาในการเดิน 2 ช่องแรก โดยใช้ตัวอย่างจากรูปที่ 5.2 (b)

3. ในช่อง [1,1] ไม่มีลม แต่มีลมในช่อง [2,1] ได้กฎใหม่ 2 ข้อดังนี้

$$R4: \neg B_{1,1}$$

$$R5: B_{2,1}$$

ฐานความรู้ประกอบด้วยประโยค R1 ถึง R5 และอาจมองว่าเป็นประโยคเพียงประโยคเดียวที่เกิดจากการรวมประโยคเดียวแบบ Conjunction คือ $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$ ที่มองเช่นนี้ได้เพราะประโยคทั้ง 5 ประโยคนั้นเป็นจริงทั้งสิ้น

ตารางที่ 5.2 ตารางค่าความจริงของ Wumpus world

B _{1,1}	B _{2,1}	P _{1,1}	P _{1,2}	P _{2,1}	P _{2,2}	P _{3,1}	R ₁	R ₂	R ₃	R ₄	R ₅	KB
False	False	False	False	False	False	False	True	True	True	True	False	False
False	False	False	False	False	False	True	True	True	False	True	False	False
.
.
False	True	False	False	False	False	False	True	True	False	True	True	False
False	True	False	False	False	False	True	True	True	True	True	True	<u>True</u>
False	True	False	False	False	True	False	True	True	True	True	True	<u>True</u>
False	True	False	False	False	True	True	True	True	True	True	True	<u>True</u>
False	True	False	False	True	False	False	True	False	False	True	True	False
.
True	True	True	True	True	True	True	False	True	True	False	True	False

5.4.4 การสรุปความ (Inference)

กระบวนการสรุปความต้องการตัดสินใจว่า เมื่อมีประโยค α แล้ว ประโยคนั้นตกทอดมาจาก KB จริงหรือไม่ ($KB \models \alpha$) เช่น $P_{2,2}$ จะเป็นประโยคความรู้ที่ตกทอดมาจากฐานความรู้หรือไม่ วิธีสรุปความอย่างหนึ่งที่น่ามาใช้ได้ก็คือ นำนิยามของการตกทอดมาใช้โดยตรงไปตรงมา แจกแจงโมเดลออกไปให้ละเอียด แล้วตรวจสอบว่า α เป็นจริงในทุกโมเดลที่ KB เป็นจริง สำหรับตรรกศาสตร์ประพจน์ โมเดลจะกำหนดค่าจริง / เท็จ ให้กับสัญลักษณ์ทุกตัว เช่น ใน Wumpus world สัญลักษณ์ที่ใช้คือ $B_{1,1}, B_{2,1}, P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}$ และ $P_{3,1}$ ทั้งหมด

7 ตัว ทำให้เกิดโมเดลที่เป็นไปได้ต่าง ๆ กันจำนวน $2^7 = 128$ โมเดล และมีอยู่ 3 โมเดลจากทั้งหมดที่ KB เป็นจริง (ดูตารางที่ 5.2) เนื่องจาก R1 ถึง R5 เป็นจริง นอกนั้น KB จะเป็นเท็จ ใน 3 โมเดลนี้ $\neg P_{1,2}$ เป็นจริง ดังนั้นแปลว่า ช่อง [1,2] ไม่มีหลุม แต่ตรงกันข้าม ใน 3 โมเดลนี้มีอยู่ 2 โมเดลที่ $P_{2,2}$ เป็นจริง ส่วนในอีกโมเดลหนึ่งเป็นเท็จ ดังนั้นจึงยังบอกไม่ได้ว่าช่อง [2,2] มีหลุมหรือไม่

5.4.5 การสมมูล (Equivalence), Validity และ Satisfiability

แนวคิดที่สำคัญที่นำมาใช้กับการตกทอดมีอีก 3 เรื่อง ซึ่งต่างก็นำมาใช้กับตรรกะได้ทุกรูปแบบ แต่จะมองเห็นภาพพจน์ได้ดี และเข้าใจง่ายที่สุดเมื่อนำมาอธิบายกับตรรกศาสตร์ประพจน์

5.4.5.1 การสมมูล (Equivalence)

ประโยค α และ β สมมูลกันถ้าทั้ง 2 ประโยคเป็นจริงในโมเดลเดียวกัน เขียนว่า $\alpha \equiv \beta$ ตัวอย่างเช่น $P \wedge Q$ และ $Q \wedge P$ เป็นประโยคที่สมมูลกัน (พิสูจน์ได้โดยใช้ตารางค่าความจริง) อาจเขียนนิยามของการสมมูลกันได้อีกแบบหนึ่งดังนี้

$$\alpha \equiv \beta \text{ ก็ต่อเมื่อ } \alpha \models \beta \text{ และ } \beta \models \alpha$$

5.4.5.2 Validity

ประโยคจะ Valid ถ้าประโยคนั้นมีค่าเป็นจริงในทุกโมเดล ตัวอย่างเช่น $P \vee \neg P$ เป็นประโยคที่ Valid ประโยคที่ Valid เรียกอีกอย่างหนึ่งว่าสัจนิรันดร์ (Tautology) สิ่งที่สำคัญอย่างยิ่งสำหรับ Validity คือ การมีค่าความจริงเป็นจริง เนื่องจากประโยค True มีค่าเป็นจริงเสมอในทุกโมเดล ดังนั้น ทุกประโยคที่ Valid จะสมมูลกับ True

5.4.5.3 Satisfiability

ประโยคจะ Satisfiable ถ้ามีค่าเป็นจริงในบางโมเดล เช่นจากตัวอย่างฐานความรู้ที่ให้ไว้ ($R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$) เป็นประโยคที่ Satisfiable เพราะมีอยู่ 3 โมเดลที่ประโยคนั้นเป็นจริง (ตามตารางที่ 5.2)

คุณสมบัติ Validity กับ Satisfiability มีความเกี่ยวข้องกันคือ

α Valid ก็ต่อเมื่อ $\neg\alpha$ ไม่ Satisfiability และ

α Satisfiability ก็ต่อเมื่อ $\neg\alpha$ ไม่ Valid

เขียนเป็นผลลัพธ์ได้ว่า

$\alpha \models \beta$ ก็ต่อเมื่อประโยค $(\alpha \wedge \neg\beta)$ ไม่ Satisfiable

การพิสูจน์ว่า β ตกทอดมาจาก α โดยการตรวจสอบความไม่ Satisfiable ของ $(\alpha \wedge \neg \beta)$ สอดคล้องกันกับเทคนิคการพิสูจน์โดยทฤษฎีทางคณิตศาสตร์ Reduction to an absurd thing หรือพิสูจน์โดยข้อขัดแย้ง (Proof by refutation หรือ Proof by contradiction) โดยสมมติให้ β เป็นเท็จ และพยายามแสดงให้เห็นการขัดแย้งกับสัจพจน์ (Axiom) α การขัดแย้งนี้คือการกล่าวหาว่าประโยค $(\alpha \wedge \neg \beta)$ ไม่ Satisfiable นั่นเอง

5.5 แบบแผนการใช้เหตุผลในตรรกศาสตร์ประพจน์ (Reasoning Patterns in Propositional Logic)

การสรุปความมีแบบแผน (Pattern) ที่เป็นมาตรฐานซึ่งสามารถนำมาประยุกต์ใช้ และก่อให้เกิดผลสรุป (Conclusion) ออกมาได้ ผลสรุปนี้นำไปสู่เป้าหมายที่ต้องการ แบบแผนเหล่านี้เรียกว่า กฎการสรุปความ (Inference rule) ใช้กับตรรกะต่าง ๆ ได้

5.5.1 กฎการสรุปความ มีกฎที่นิยมใช้ และมักจะพบเห็นอยู่เสมอในศาสตร์หลายด้าน ในที่นี้จะกล่าวถึงกฎที่มีประโยชน์และจำเป็นต่อการพิสูจน์ในบทนี้ 2 ข้อ

5.5.1.1 Modus Ponens เป็นกฎที่รู้จักกันดีทั่วไป เขียนเป็นรูปแบบดังนี้

$$\frac{A \Rightarrow B, \quad A}{B}$$

หมายความว่า ถ้าให้ประโยคมา 2 ประโยค คือ $A \Rightarrow B$ และ A จะได้ข้อสรุปคือประโยค B ตัวอย่างเช่น ถ้ามีประโยค

$(WumpusAhead \wedge WumpusAlive) \Rightarrow Shoot$

และอีกประโยคหนึ่งคือ $(WumpusAhead \wedge WumpusAlive)$

จะสรุปความได้ประโยค $Shoot$

หมายเหตุ ลักษณะการเขียนรูปแบบของกฎการสรุปความเช่นนี้ ส่วนที่อยู่เหนือเส้นคือ Premise ส่วนที่อยู่ใต้เส้นคือ Conclusion

5.5.1.2 And-Elimination เมื่อมี Conjunction สามารถสรุปได้ Conjunct ใด ๆ

$$\frac{A \wedge B}{A}$$

ตัวอย่างเช่น $(WumpusAhead \wedge WumpusAlive)$ สามารถสรุปได้ว่า $WumpusAlive$ เป็นต้น หรืออาจสรุปว่า $WumpusAhead$ ก็ได้เช่นกัน

นอกจากนี้ตรรกศาสตร์ยังได้กล่าวถึงความสมมูลกันทางตรรกะ ซึ่งเกิดกับประโยคมากกว่าหนึ่งประโยคขึ้นไปมีความหมายตรงกัน (Logical equivalence) ความสมมูลเหล่านี้ได้แก่

1. $(A \wedge B) \equiv (B \wedge A)$ การสลับที่ของ \wedge
2. $(A \vee B) \equiv (B \vee A)$ การสลับที่ของ \vee
3. $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$ การเปลี่ยนหมู่ของ \wedge
4. $((A \vee B) \vee C) \equiv (A \vee (B \vee C))$ การเปลี่ยนหมู่ของ \vee
5. $\neg(\neg A) \equiv A$ นิเสธซ้อนนิเสธ
6. $(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$ Contraposition
7. $(A \Rightarrow B) \equiv \neg A \vee B$ Implication elimination
8. $(A \Leftrightarrow B) \equiv ((A \Rightarrow B) \wedge (B \Rightarrow A))$ Biconditional elimination
9. $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$ de Morgan
10. $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$ de Morgan
11. $(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee (A \wedge C))$ การกระจายของ \wedge ไปบน \vee
12. $(A \vee (B \wedge C)) \equiv ((A \vee B) \wedge (A \vee C))$ การกระจายของ \vee ไปบน \wedge

หมายเหตุ A, B, C เป็นประโยคใด ๆ ที่เป็นตรรกศาสตร์ประพจน์

ความสมมูลทางตรรกะเหล่านี้นำมาประยุกต์ใช้กับกฎการสรุปความได้เช่นกัน ตัวอย่างเช่น การนำข้อ 8 (Biconditional elimination) มาใช้ ทำให้ได้กฎเพิ่ม 2 ข้อคือ

$$\frac{A \Leftrightarrow B}{(A \Rightarrow B) \wedge (B \Rightarrow A)} \quad \text{และ} \quad \frac{(A \Rightarrow B) \wedge (B \Rightarrow A)}{A \Leftrightarrow B}$$

แต่กฎบางข้อก็ไม่สามารถทำงานในทิศทางย้อนกลับได้ เช่น Modus Ponens ถ้าให้มาแค่ B จะไม่สามารถหาข้อสรุปเป็น $(A \Rightarrow B)$ และ A ได้

5.5.2 วิธีใช้กฎการสรุปความใน Wumpus world

พิจารณาการสรุปความใน Wumpus world สมมติว่าฐานความรู้ประกอบด้วย R1 ถึง R5 (ตามที่กล่าวไว้ในหน้า 90-91) ได้แก่

- R1: $\neg P_{1,1}$
 R2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
 R3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
 R4: $\neg B_{1,1}$

$$R5: \quad B_{2,1}$$

การพิสูจน์ว่าไม่มีหลุมอยู่ในช่อง [1,2] หรือ $\neg P_{1,2}$ มีขั้นตอนดังนี้

1. ใช้ Biconditional elimination กับ R2 ทำให้ได้กฎใหม่

$$R6: \quad (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

2. ใช้ And-elimination กับ R6 ทำให้ได้กฎใหม่

$$R7: \quad ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

3. ใช้ Contraposition กับ R7 ทำให้ได้กฎใหม่

$$R8: \quad \neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})$$

4. ใช้ Modus Ponens กับ R8 และ R4 ทำให้ได้กฎใหม่

$$R9: \quad \neg(P_{1,2} \vee P_{2,1})$$

5. ใช้กฎของ de Morgan ได้ข้อสรุป

$$\neg P_{1,2} \wedge \neg P_{2,1}$$

นั่นคือ ไม่มีหลุมอยู่ในช่อง [1,2] และช่อง [2,1]

การประยุกต์ใช้กฎการสรุปความหรือการทำ Derivation ดังที่กล่าวมานี้ เรียกว่า การพิสูจน์ (Proof) การหาข้อพิสูจน์มีลักษณะเช่นเดียวกันกับการหาคำตอบในปัญหาการค้นหา จะเห็นว่า ถ้า Successor function คือการประยุกต์ใช้กฎการสรุปความทุกกฎที่เป็นไปได้แล้ว อัลกอริทึมของการค้นหาในบทที่ 3 และ 4 ก็สามารถนำมาใช้กับการหาข้อพิสูจน์ได้เช่นกัน

การหาข้อพิสูจน์ในตรรกศาสตร์ประพจน์ไม่ใช่วิธีที่มีประสิทธิภาพดีที่สุดสำหรับทุกโมเดล แต่ในบางโมเดลก็ทำได้ง่ายและเร็วมาก เช่นตัวอย่าง Wumpus world ที่กล่าวถึงข้างต้น การนำไปสู่ข้อสรุปว่า $\neg P_{1,2} \wedge \neg P_{2,1}$ ไม่ได้กล่าวถึงประพจน์ตัวอื่น เช่น $B_{2,1}$, $P_{1,1}$, $P_{2,2}$ หรือ $P_{3,1}$ เนื่องจากเป้าหมายคือ $P_{1,2}$ นั้นพบอยู่ในกฎข้อ R2 ส่วนประพจน์อื่นที่อยู่ใน R2 คือ $P_{2,1}$ และ $B_{1,1}$ ก็พบอยู่แต่ใน R4 และ R2 เท่านั้น ดังนั้น R1, R3, และ R5 จึงไม่ได้มีส่วนเกี่ยวข้องกับการพิสูจน์เลย แม้ว่าจะมีกฎเพิ่มอีกเป็นล้านกฎในฐานความรู้ แต่เหตุการณ์ก็ยังคงเป็นเช่นเดิม คือใช้กฎเพียงไม่กี่ข้อหาข้อสรุปได้ แต่ถ้าใช้ตารางค่าความจริงแล้ว การมีกฎจำนวนมากทำให้จำนวนโมเดลเพิ่มมากขึ้นได้เป็นจำนวนมหาศาลเนื่องจากการแจกแจงค่าความจริงของประพจน์ในกฎเหล่านั้น ทุกตัว ค่าความจริงที่แตกต่างกันทำให้โมเดลเกิดขึ้นได้ต่าง ๆ กัน

5.5.3 รีโซลูชัน (Resolution)

การหาข้อพิสูจน์โดยใช้แต่กฎการสรุปความเท่านั้นไม่อาจรับรองจะพบเป้าหมายที่ต้องการ กฎการสมมูลที่กล่าวถึงในหัวข้อที่แล้วมีอยู่หลายข้อ แต่ถ้ายังไม่เพียงพอ หรือขาดหายไป

บางข้อ ก็อาจจะทำให้การพิสูจน์ล้มเหลวได้ วิธีพิสูจน์เป็นกฎการสรุปความอีกข้อหนึ่งซึ่งได้มาจากการใช้อัลกอริทึมของการสรุปความ จึงเป็นทางเลือกที่นำมาใช้ได้อย่างดีเพราะทำให้ใช้กฎทางตรรกะต่าง ๆ ลดน้อยลง ก่อนที่จะกล่าวถึงเนื้อหาของวิธีพิสูจน์ จะยกตัวอย่างการใช้งานวิธีพิสูจน์มาให้ดูก่อนว่าทำให้กระบวนการพิสูจน์เร็วขึ้นเท่าใด

จากตัวอย่าง Wumpus world ในรูปที่ 5.3 (a) เอเจนต์เดินกลับจาก [2,1] ไปยัง [1,1] แล้วเดินไปยัง [1,2] ซึ่งที่นี้เอเจนต์ได้กลิ่น (Stench) แต่ไม่มีลมพัด (Breeze) เราจะเพิ่มข้อเท็จจริงลงในฐานความรู้ดังนี้

$$R11: \neg B_{1,2}$$

$$R12: B_{1,2} \Rightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$$

โดยกระบวนการเดิมที่ใช้สรุปหากฎ R10 ก่อนหน้านี้ สามารถหาได้ว่าไม่มีหลุมอยู่ในช่อง [2,2] และ [1,3]

$$R13: \neg P_{2,2}$$

$$R14: \neg P_{1,3}$$

ใช้ Biconditional elimination กับ R3 ตามด้วย Modus Ponens กับ R5 จะได้ว่ามีหลุมอยู่ในช่อง [1,1] หรือ [2,2] หรือ [3,1]

$$R15: P_{1,1} \vee P_{2,2} \vee P_{3,1}$$

ต่อไปทำวิธีพิสูจน์ครั้งแรก โดยนำ $\neg P_{2,2}$ จาก R13 มาสลายซ้ำกับ R15 จะได้

$$R16: P_{1,1} \vee P_{3,1}$$

ใช้วิธีพิสูจน์อีกครั้งหนึ่ง โดยนำ $\neg P_{1,1}$ (ข้อเท็จจริงคือในช่อง [1,1] ไม่มีหลุมแน่นอน เพราะเอเจนต์มีจุดเริ่มต้นอยู่ที่ช่องนี้) มาสลายซ้ำกับ R16 จะได้

$$R17: P_{3,1}$$

กระบวนการทำงานดังที่กล่าวมานี้ ใช้กฎการสรุปความใหม่ 2 ข้อ คือ

1. Unit resolution

$$\frac{I_1 \vee \dots \vee I_k, m}{I_1 \vee \dots \vee I_{i-1} \vee I_{i+1} \vee \dots \vee I_k}$$

ซึ่ง I แต่ละตัวคือค่าข้อความ (Literal) ส่วน I_i และ m เป็นค่าข้อความที่เป็นนิเสธของกันและกัน (Complementary literal)

กฎ Unit resolution ต้องมีวลีเป็น Disjunction (การเชื่อมด้วย \vee) ของค่าข้อความ และมีค่าข้อความอีกค่าหนึ่ง จากนั้นจะสร้างวลีใหม่ขึ้นมา

ค่าข้อความที่อยู่เดี่ยวเพียงข้อความเดียว อาจมองว่าเป็น Disjunction ของค่าข้อความก็ได้ เรียกอีกอย่างหนึ่งว่า Unit clause

2. Resolution

$$\frac{I_1 \vee \dots \vee I_k, \quad m_1 \vee \dots \vee m_n}{I_1 \vee \dots \vee I_{i-1} \vee I_{i+1} \vee \dots \vee I_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n}$$

โดยที่ I_i และ m_j เป็นค่าข้อความที่เป็นนิเสธของกันและกัน เพื่อให้ง่ายขึ้น สมมติว่าวลีมีเพียง 2 ค่าเท่านั้น อาจเขียนได้ว่า

$$\frac{I_1 \vee I_2, \quad \neg I_2 \vee I_3}{I_1 \vee I_3}$$

นั่นคือ ริโซลูชันจะรับวลีมา 2 วลี แล้วสร้างใหม่ กลายเป็นวลีเดี่ยวที่ประกอบด้วย 2 วลีเดิม แต่ตัดค่าข้อความที่เป็นนิเสธของกันและกันออกไป ตัวอย่างอีกตัวอย่างหนึ่งเช่น

$$\frac{P_{1,1} \vee P_{3,1}, \quad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

5.5.4 คอนจังก์ทีฟนอร์มัลฟอร์ม (Conjunctive normal form)

เนื่องจากกฎริโซลูชันใช้กับ Disjunction เท่านั้น จึงดูเหมือนว่าจะเป็นวิธีที่เหมาะสมกับฐานความรู้ที่ประกอบด้วย Disjunction เป็นส่วนใหญ่ จึงต้องหาวิธีดัดแปลงรูปแบบของประโยคในฐานความรู้เพื่อให้ใช้กฎนี้ได้ กระบวนการสรุปความจึงจะทำได้สมบูรณ์ทุกรูปแบบ และเนื่องจากทฤษฎีทางตรรกศาสตร์กล่าวว่า ทุกประโยคของตรรกศาสตร์ประพจน์ มีความสมมูลทางตรรกะกับ Conjunction ของ Disjunction ของค่าข้อความ หมายความว่า ประโยคใด ๆ ก็ตาม สามารถเขียนอยู่ในรูปของ Conjunction ของ Disjunction ของค่าข้อความได้ เรียกว่า Conjunctive normal form หรือ CNF เพื่อให้ง่ายขึ้นในที่นี้จะศึกษาประโยคที่เป็น k-CNF ซึ่งมีลักษณะจำกัด กล่าวคือประโยคใน k-CNF ประกอบด้วยค่าข้อความจำนวน k ค่าต่อหนึ่งวลี มีรูปแบบการเขียนทั่วไปดังนี้

$$(I_{1,1} \vee \dots \vee I_{1,k}) \wedge \dots \wedge (I_{n,1} \vee \dots \vee I_{n,k})$$

ตัวอย่างการแปลงประโยคแบบง่าย เพื่อให้เห็นได้ชัดเจน ให้นึกกฎ R2 ที่เคยกล่าวถึงไว้แล้วมาพิจารณา

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

สามารถแปลงเป็น CNF โดยขั้นตอนดังนี้

1. ตัด \Leftrightarrow ออก แทนที่ด้วย \Rightarrow (เพราะว่า $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$)

$$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

2. ตัด \Rightarrow ออกแล้วแทนที่ $A \Rightarrow B$ ด้วย $\neg A \vee B$

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg (P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$

3. CNF ต้องมี \neg กำกับที่ค่าข้อความ จึงย้าย \neg เข้าไปในวงเล็บ ในที่นี้ต้อง

อาศัยกฎการสมมูล 3 ข้อ

$$\neg(\neg A) \equiv A \quad \text{นิเสธซ้อนนิเสธ}$$

$$\neg(A \wedge B) \equiv (\neg A \vee \neg B) \quad \text{de Morgan}$$

$$\neg(A \vee B) \equiv (\neg A \wedge \neg B) \quad \text{de Morgan}$$

จากข้อ 2 ได้

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$$

4. ขณะนี้ประโยคจะประกอบด้วยค่าข้อความที่เชื่อมด้วย \wedge และ \vee เท่านั้น

ใช้กฎการกระจาย

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

ประโยคในข้อ 4 นี้เป็นประโยคที่อยู่ในรูป CNF ประกอบด้วย Conjunction ของวลี 3 วลี แต่ละวลีเป็น Disjunction สามารถนำไปใช้เป็นอินพุตของกระบวนการรีโซลูชันได้

5.5.5 ฟอว์เวิร์ดและแบ็คเวิร์ดเชนนิ่ง (Forward และ Backward chaining)

การสรุปความโดยใช้รีโซลูชันเป็นวิธีการที่ดีมาก แต่ในบางสถานการณ์อาจจะไม่จำเป็นต้องใช้วิธีที่ซับซ้อนเช่นนี้ ในฐานความรู้มักพบวลีจำนวนมากมีลักษณะเป็น Horn clause กล่าวคือ เป็น Disjunction ของค่าข้อความ โดยที่ค่าข้อความเหล่านี้มีค่าเป็นบวก (Positive หรือไม่ใช่ลบ) ได้อย่างมากที่สุดเพียงค่าเดียวเท่านั้น เช่น $(\neg L_{1,1} \vee \neg \text{Breeze} \vee B_{1,1})$ เมื่อ $L_{1,1}$ หมายถึงตำแหน่งของเอเจนต์ในช่อง [1,1] เป็น Horn clause แต่ $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1})$ ไม่ใช่ Horn clause เงื่อนไขของการที่จำเป็นต้องมีค่าข้อความเป็นบวกเพียงค่าเดียวเท่านั้นมีความจำเป็นและสำคัญ เนื่องจากเหตุผล 3 ข้อ ได้แก่

1. Horn clause สามารถเขียนใหม่ในรูป Implication ได้ โดยให้ Premise เป็น Conjunction ของค่าข้อความบวก และมีผลสรุป เป็นค่าข้อความบวกเดียว ตัวอย่างเช่น Horn clause $(\neg L_{1,1} \vee \neg \text{Breeze} \vee B_{1,1})$ เขียนเป็นรูป Implication ได้ดังนี้

$$(L_{1,1} \wedge \text{Breeze}) \Rightarrow B_{1,1}$$

รูปแบบ Implication อ่านง่ายกว่ารูปแบบเดิม จากตัวอย่าง อ่านได้ว่า ถ้าเอเยนต์อยู่ในช่อง [1,1] และมีลม สรุปว่า มีลมพัดในช่อง [1,1] การอ่านประโยคเช่นนี้ใกล้เคียงกับภาษาธรรมชาติมากขึ้น และเข้าใจได้ง่ายขึ้น

2. การสรุปความกับ Horn clause ทำได้โดยอาศัยอัลกอริทึมแบบฟอร์เวิร์ดเชนนิ่ง (Forward chaining) และ แบ็คเวิร์ดเชนนิ่ง (Backward chaining) ซึ่งอัลกอริทึมทั้งสองนี้เป็นวิธีใช้เหตุผลที่ชัดเจน และเข้าใจง่ายเหมือนเป็นไปตามธรรมชาติของความเข้าใจของมนุษย์

3. การทำงานกับ Horn clause ใช้เวลาไม่มากนัก ขึ้นอยู่กับขนาดของฐานความรู้ ข้อนี้จึงแสดงให้เห็นว่า การสรุปความทางตรรกะในฐานความรู้ที่เป็นตรรกศาสตร์ประพจน์มีค่าใช้จ่ายต่ำ

5.5.4.1 อัลกอริทึมแบบฟอร์เวิร์ดเชนนิ่ง

อัลกอริทึมจะหาว่าตัวสัญลักษณ์ประพจน์เดี่ยว q (จากคำถาม Query) ตกทอดจากฐานความรู้ (ที่ประกอบด้วย Horn clause) หรือไม่ โดย

- 1) เริ่มต้นจากค่าข้อความบวกเดียว (Fact) ที่มีในฐานความรู้
- 2) ถ้ารู้ค่าความจริงของส่วน Premise ทั้งหมดของ Implication จะสามารถนำผลสรุปไปเพิ่มลงในฐานความรู้ ตัวอย่างเช่น ถ้ารู้ค่า $L_{1,1}$ และ Breeze แล้ว และ $(L_{1,1} \wedge \text{Breeze}) \Rightarrow B_{1,1}$ อยู่ในฐานความรู้ (แสดงว่าเป็นจริง) แล้ว สามารถเติม $B_{1,1}$ ลงไปในฐานความรู้ได้

- 3) ทำตามขั้นตอนในข้อ 2 ไปเรื่อย ๆ จนกระทั่ง q ที่ถามมาถูกเติมลงในฐานความรู้ หรือจนกว่าจะไม่มีสรุปอะไรใหม่อีกต่อไป

เพื่อให้เข้าใจอัลกอริทึมของฟอร์เวิร์ดเชนนิ่งจะยกตัวอย่างโดยกำหนดให้ฐานความรู้ประกอบด้วย Horn clause และข้อเท็จจริงดังนี้ (ตัวอักษรตัวใหญ่ใช้เขียนแทนประพจน์แต่ละเทอม)

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

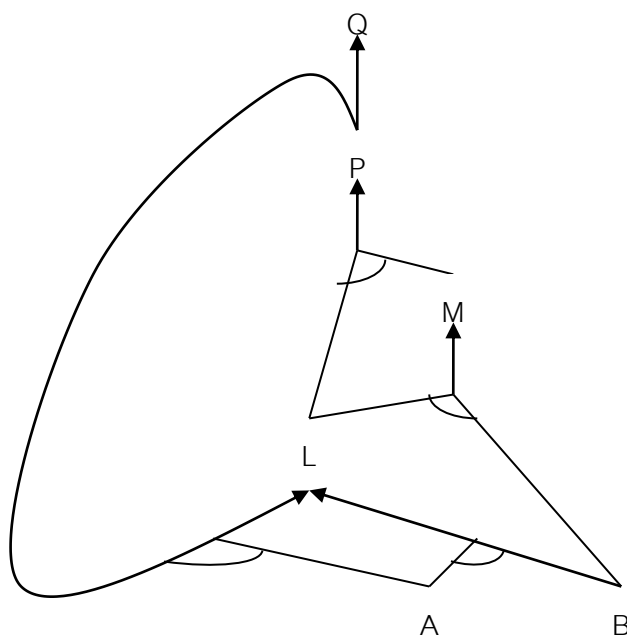
$$A \wedge B \Rightarrow L$$

A

B

จากประโยคดังกล่าวนำมาเขียนเป็น AND-OR graph ได้ดังแผนภาพใน

รูปที่ 5.5



รูปที่ 5.5 AND-OR graph

ฐานความรู้ที่ประกอบด้วย Horn clause สามารถเขียนเป็นแผนภาพ เรียกว่ากราฟ AND-OR เส้นเชื่อมหลายเส้นที่มีโค้งเชื่อมติดอยู่แสดงถึง Conjunction เส้นเชื่อมทุกเส้นที่มาจากสัญลักษณ์ประพจน์ต้องพิสูจน์ได้ว่ามีค่าเป็นจริง ส่วนเส้นเชื่อมหลายเส้นที่ไม่มีโค้งเชื่อมติดอยู่แสดงถึง Disjunction มีเส้นใดเส้นหนึ่งเป็นจริงก็ได้ การดูจากกราฟทำให้เข้าใจฟอร์เวิร์ดเชนได้ง่ายขึ้น ในที่นี้ โหนดที่รู้ว่ามีอยู่ คือ A และ B ถูกกำหนดไว้ให้ จากนั้น การสรุปความดำเนินไปตามเส้นเชื่อม เมื่อใดที่พบ Conjunction การดำเนินการจะต้องรอจนกว่าจะรู้ค่าเส้นเชื่อมทุกเส้นให้หมดก่อนจึงจะทำต่อไปได้

ฟอร์เวิร์ดเชนหนึ่งเป็นการแสดงแนวคิดของการใช้เหตุผลแบบ Data-driven นั่นคือ การใช้เหตุผลที่มุ่งให้ความสนใจเริ่มต้นที่ตัวข้อมูลที่อยู่แล้ว หรือมีอยู่แล้วในฐานความรู้ และยัง

สามารถใช้ในเอนต์ที่ต้องการค้นหาผลสรุปจากเพอร์เซ็ปที่รับมา โดยไม่ได้มีคำถามอะไรเป็นพิเศษ แค่อยากรู้ว่ามีความรู้อะไรอยู่บ้าง และจะเกิดอะไรขึ้นอีกเท่าที่จะหาได้ ตัวอย่างเช่น ถ้าเราอยู่ในบ้าน และได้ยินเสียงฝนเริ่มตกลงมา เราจะรู้ต่อไปว่า การปิกนิกต้องยกเลิก แต่เนื่องจากสมองคนเราใช้ฟอร์เวิร์ดเซนนิ่งภายใต้การควบคุมอย่างระมัดระวัง จึงไม่คิดไกลออกไปถึงรายละเอียดอื่น เช่นว่ากลีบกุหลาบกลีบที่ 17 ของดอกกุหลาบดอกใหญ่ที่สุดในสวนของเพื่อนบ้านจะต้องเปียกฝน แต่ถ้าอยู่ในฐานความรู้แล้ว เอนต์จะสามารถค้นหาข้อสรุปได้จนหมดเท่าที่มีอยู่

5.5.4.2 อัลกอริทึมแบบแบ็คเวิร์ดเซนนิ่ง

มีวิธีการทำงานเช่นเดียวกับข้อ คือทำงานย้อนหลังจากตัวคำถาม (Query) โดยมีขั้นตอนดังนี้

- 1) ถ้า Query q เป็นจริง หยุดทำงาน
- 2) ถ้าไม่เช่นนั้น หา Implication จากฐานความรู้ โดยที่ในประโยค Implication นั้นมีผลสรุปเป็น q
- 3) ถ้ามี Implication ประโยคใดประโยคหนึ่งในข้อ 2 ที่มี Premise ทั้งหมดเป็นจริง หรือพิสูจน์ได้ว่าจริง โดยการพิสูจน์ต้องใช้วิธีแบ็คเวิร์ดเซนนิ่งเช่นเดียวกันแล้ว จะได้ q เป็นจริง

ถ้าดูจากตัวอย่างกราฟ AND-OR ในรูปที่ 5.5 เมื่อต้องการหา Query Q จะมีการทำงานย้อนศร จาก Q ไปตามกราฟ จนกระทั่งถึง Fact ที่รู้ค่าความจริงซึ่งอยู่ที่ฐานของการพิสูจน์ (A และ B)

แบ็คเวิร์ดเซนนิ่งเป็นรูปแบบของการใช้เหตุผลแบบมุ่งตรงสู่เป้าหมาย (Goal-direct reasoning) เหมาะสมกับคำถามเฉพาะเจาะจง เช่น ตอนนี้ควรจะทำอะไรดี หรือถามว่า กุญแจอยู่ที่ไหน เป็นต้น ค่าใช้จ่ายในวิธีการแบบนี้ค่อนข้างต่ำเมื่อเทียบกับขนาดของฐานความรู้ เพราะกระบวนการทำงานเลือกเจาะจงใช้งาน Fact เฉพาะในส่วนที่เกี่ยวข้องเท่านั้น

แบบฝึกหัดบทที่ 5

1. จงอธิบายความหมายของ Knowledge-based agent
2. จงพิสูจน์คํากล่าวต่อไปนี้
 - 2.1 α Valid ก็ต่อเมื่อ $\text{True} \models \alpha$
 - 2.2 $\alpha \models \beta$ ก็ต่อเมื่อ ประโยค $(\alpha \Rightarrow \beta)$ Valid
3. พิจารณาการใช้ประพจน์ 4 ตัวคือ A, B, C, D ประโยคในข้อต่อไป่นี้จะมีโมเดลได้ต่าง ๆ กัน เป็นจำนวนเท่าใด
 - 3.1 $(A \wedge B) \vee (B \wedge C)$
 - 3.2 $A \vee B$
 - 3.3 $A \Leftrightarrow B \Leftrightarrow C$
4. จากข้อความต่อไปนี้

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

จงพิสูจน์ว่า The unicorn is mythical. และจะพิสูจน์เรื่อง magical และ horned ได้หรือไม่ ถ้าได้ให้พิสูจน์ด้วย
5. จงหาว่าประโยคในข้อต่อไป่นี้ ข้อใด Valid, ไม่ Satisfiable หรือไม่ทั้งสองอย่าง (ใช้ตารางค่าความจริง หรือกฎการสมมูลกันก็ได้)
 - 5.1 $\text{Smoke} \Rightarrow \text{Smoke}$
 - 5.2 $\text{Smoke} \Rightarrow \text{Fire}$
 - 5.3 $\text{Smoke} \vee \text{Fire} \wedge \neg \text{Fire}$