บทที่ 6

ตรรกศาสตร์ภาคแสดง

บทนี้แสดงถึงการแทนค่าความจริงโดยวิธีหนึ่งคือ ใช้ตรรกศาสตร์ภาคแสดงอันดับที่หนึ่ง (first order predicate logic: FOPL) ซึ่งเหมาะแก่การสรุปหาความรู้ใหม่ที่ทราบความจริงมาก่อนหรือการใช้คณิตศาสตร์นิรนัย (mathematic deduction) วิธีการนี้เป็นการนำความคิดในเรื่องการพิสูจน์หรือการแสดงข้อความจริงโดยการอนุมาน ซึ่งสามารถตอบคำถามที่ต้องการได้จากความรู้เดิมที่มี งานส่วนหนึ่งที่สำคัญทางปัญญาประดิษฐ์คือการหาขั้นตอนวิธีในการพิสูจน์ทฤษฎีทางคณิตศาสตร์ แต่ในบทนี้จะแสดงถึงการใช้หลักการของคณิตศาสตร์ที่ค่อนข้างจะมีประโยชน์มากกว่าคณิตศาสตร์พื้นฐานทั่วไป

การแทนความรู้ที่ใช้ตรรกศาสตร์ประพจน์

จากที่กล่าวมาแล้วในบทที่ 5 เราสามารถแทนค่าความจริงโดยใช้ประพจน์แทนความจริงของทั้งโลกนี้ได้ ซึ่งต่อไปจะกล่าวถึงตรรกศาสตร์เชิงประพจน์โดยละเอียดดังนี้

ประโยคที่ใช้แทนความจริงในโลกที่เขียนในรูปสูตรที่จัดดีแล้ว (well-formed formulas: WFF's) หมายความว่า กำหนดให้ P, Q, R, S, T , ... แทนสัญลักษณ์ของประพจน์และ true, false แทนสัญลักษณ์ที่มีค่าความจริงเป็นจริง และเท็จ ตามลำดับ และ $\wedge, \vee, \neg, \rightarrow$ และ $\equiv$ แทนสัญลักษณ์ตัวเชื่อมทางคณิตศาสตร์

เรานิยามรูปสูตรที่จัดดีแล้ว ได้ดังนี้

1) สัญลักษณ์ของประพจน์ true และ false เป็นรูปสูตรที่จัดดีแล้ว

2) นิเสธของรูปแบบของสูตรที่ดีเป็นจะต้องเป็นรูปสูตรที่จัดดีแล้ว เช่น กำหนดให้ $\neg$ p เป็น รูปสูตรที่จัดดีแล้ว เมื่อ p เป็นรูปสูตรที่จัดดีแล้ว

3) การเชื่อมด้วย “และ” ของรูปสูตรที่จัดดีแล้ว เป็นรูปสูตรที่จัดดีแล้ว เช่น p $\wedge$ q เป็นรูปสูตรที่จัดดีแล้ว เมื่อ p และ q เป็นรูปสูตรที่จัดดีแล้ว

4) การเชื่อมด้วย “หรือ” ของรูปสูตรที่จัดดีแล้ว เป็นรูปสูตรที่จัดดีแล้ว เช่น p $\vee$ q เป็น รูปสูตรที่จัดดีแล้ว เมื่อ p และ q เป็นรูปสูตรที่จัดดีแล้ว

5) ประพจน์แบบมีเงื่อนไขของรูปสูตรที่จัดดีแล้ว เป็นรูปสูตรที่จัดดีแล้ว เช่น p $\rightarrow$ q เป็นรูปสูตรที่จัดดีแล้วเมื่อ p และ q เป็นรูปสูตรที่จัดดีแล้ว

6) ประพจน์สมมูลกันของรูปสูตรที่จัดดีแล้วเป็นรูปสูตรที่จัดดีแล้ว p = q เป็นรูปสูตรที่จัดดีแล้ว เมื่อ p และ q เป็นรูปสูตรที่จัดดีแล้ว

ตัวอย่างเช่น

วันนี้ฝนตก แทนด้วย RAINING

วันนี้แดดออก แทนด้วย SUNNY

วันนี้ลมแรง แทนด้วย WINDY

ถ้าวันนี้ฝนตกแล้วแดดไม่ออก แทนด้วย $RAINING \rightarrow \neg SUNNY$

ถ้าเราทราบว่า วันนี้ฝนตกแทนด้วย RAINING จากข้อความ 4 เราได้ว่า วันนี้แดดไม่ออก ¬ SUNNY ซึ่งเราพบว่าการเขียนแทนข้อความด้วยประพจน์มีข้อจำกัด คือ ถ้าเราต้องการแทนความจริง Socrates is a man เราต้องแทนด้วย socreates_is_a_man และเมื่อเราต้องการแทนว่า Plato is a man เราต้องแทนด้วย plato_is_a_man

จะเห็นได้ว่า ต้องสร้างประพจน์ใหม่ทุกครั้งที่เปลี่ยนชื่อ ดังนั้น จึงไม่สะดวกต่อการนำไปใช้ สิ่งที่ต้องการคือการแทนข้อความดังกล่าวโดยใช้ตัวแปรที่สามารถเปลี่ยนไปตามชื่อได้ เช่น

man(socreates)

man(plato)

จากข้างต้นจะได้ตัวแทนความรู้ซึ่งสามารถสรุปข้อความได้โดยการใช้อาร์กิวเมนต์ (argument) ช่วย โดยเฉพาะข้อความที่บอกว่า

คนทุกคนต้องตาย

ถ้าเราแทนข้อความนี้ด้วยประพจน์ เราสามารถเขียนได้ดังนี้

โซครีตต้องตาย

พลาโตต้องตาย

นาย...ต้องตาย

จะเห็นได้ว่า ไม่สามารถหาความสัมพันธ์ระหว่างการเป็นมนุษย์ กับการตาย ซึ่งวิธีการเช่นนี้ไม่สะดวกต่อการอนุมานหรือนำไปใช้หาข้อสรุปทางตรรกศาสตร์ การแทนค่าจึงจำเป็นต้องใช้ตัวแปรแทนตรรกศาสตร์ภาคแสดง การทดสอบว่าประโยคใดเป็นจริงสามารถทำได้โดยการใช้กฎการอนุมาน (inference) หรือ กฎการนิรนัย (deduction) แต่มักจะมีปัญหาทางเลือกเพื่อไปสู่ข้อสรุปนั้นยาก ในกรณีที่ข้อความที่ต้องการพิสูจน์นั้นเป็นเท็จ ขั้นตอนวิธีที่ใช้อาจเป็นวงวน (loop) นั่นคือขั้นตอนวิธีใดๆ ก็

ตามที่พยายามจะสรุปความจริงจากประโยคที่ไม่จริง ไม่สามารถจะรับประกันได้ว่า ขั้นตอนวิธีนั้นจะหยุดทำงานเสมอ หรือกล่าวอีกนัยหนึ่งคือ ปัญหาประเภทนี้เป็นปัญหาที่ไม่สามารถจะอธิบายได้และเป็นวงวนที่ไม่รู้จบ อย่างไรก็ตามอาจแสดงขั้นตอนวิธีที่ใช้งานสำหรับการพิสูจน์ประโยคส่วนใหญ่ที่เราสนใจได้พิจารณาการแก้ปัญหาต่อไปนี้

1) Marcus was a man.
2) Marcus was a Pompeian.
3) All Pompeians were Rpmans.
4) Cacsar was a ruler.
5) All Romans were either loyal to Ceasar or hated him.
6) Everyone is loyal to someone.
7) People only try to assassinate rulers they are not loyal to.
8) Marcus tried to assassinate Ceasar.

สามารถเขียนตรรกศาสตร์ภาคแสดงอันดับหนึ่ง ได้ดังนี้

1) Marcus was a man.= man(marcus)

man แทนข้อความที่มีตัวแปรคือ marcus ซึ่งจะละข้อมูลบางอย่างไว้ เช่น was a เวลาที่อ้างถึงจากประโยค แต่สำหรับการตอบคำถามที่เราสนใจ การแทนข้อความด้วย man(marcus) ก็เพียงพอต่อการตอบคำถามแล้ว

เรียก man ว่าชื่อประพจน์(predicate name) และ marcus คือ อากิวเมนต์ที่หนึ่งของภาคแสดง man

2) Marcus was a Pompeian.= pompeian(marcus)

จะเห็นได้ว่า pompeian เป็นชื่อประพจน์ในลักษณะเดียวกับ man ข้างต้น ประโยคนี้เพื่อบอกว่า marcus เป็นชาวปอมเปเอียน

3) All Pompeian were Romans. = $\forall$X:pompeian(X) $\rightarrow$roman(X)

ประโยคนี้เพิ่มตัวแปร X เข้ามาเพื่อใช้แทนคนทุกคน และเพื่ออธิบายว่าคนทุกคนที่เป็นชาวปอมเปอีจะต้องเป็นชาวโรมันด้วยเสมอ

4) Caesar was a ruler. = ruler(ceasar)

ข้อควรระวังอย่างหนึ่งของการใช้ตรรกศาสตร์ภาคแสดงคือ ชื่อเฉพาะ เพราะการใช้นั้นไม่จำเป็นต้องหมายถึงบุคคล คนเดียวกัน แต่ในที่นี้เราจะใช้ความหมายที่หมายถึงบุคคลคนเดียวกันตลอด

5) All Romans were either loyal to Ceasar or hated him.

$=\forall$X:roman(X)$\rightarrow$loyalto(X,ceasar) $\vee$ hate(X,ceasar).

โดยปกติ "หรือ" เราจะหมายถึงอินคลูซีพออร์ (inclusive – or) แต่ในที่นี้เราหมายถึง เอ็กซ์คลูซีพออร์ (exclusive-or) นั่นคือค่าความจริงทั้งสองแบบเป็นจริงพร้อมกันไม่ได้ ถ้าเราต้องการแทนเอ็กซ์คลูซีพออร์จริงๆ เราต้องเขียนประโยคนี้ใหม่ว่า

$\forall$X:roman(X)$\rightarrow$[(loyalto(X,ceasar)$\wedge$hate(X,ceasar))$\wedge\neg$(loyalto(X,ceasar)$\wedge$ hate(X,ceasar))]

6) Every is loyal to someone. $=\forall$X $\exists$Y:loyalto(X,Y)

ปัญหาอย่างหนึ่งที่พบในการแปลงประโยคภาษาอังกฤษให้เป็นรูปประโยคที่ต้องการคือ ปัญหาขอบข่าย (scope) ของตัวแปร นั่นคือประโยคที่เราเขียนนี้หมายความว่า สำหรับทุกคนจะต้องมีคนคนหนึ่งซึ่งเขานั้นเลือกที่จะซื่อสัตย์ หรือ ทุกคนจะซื่อสัตย์ต่อคนนี้เพียงคนเดียว ซึ่งเป็นความหมายที่เราต้องการใช้

7) People only try to assassinate rules that they are not loyal to.

$=\forall$X $\forall$Y:person(X)$\wedge$ruler(Y)$\wedge$tryassassinate(X,Y) $\rightarrow\neg$ loyalto(X,Y)

ประโยคนี้เขียน tryassassinate ด้วยประพจน์เพียงตัวเดียวเพราะเราไม่ได้สนใจความหมายอื่นของการพยายามลอบฆ่า ในกรณีอื่นที่เราอาจต้องแทนข้อความนี้ด้วยสัญลักษณ์ของประพจน์ที่มากกว่าหนึ่งตัว

8) Marcus tried to assassinate Ceasar. =tryassassinate(marcus,ceasar).

การหาคำตอบจากตรรกศาสตร์ภาคแสดง

เมื่อได้ตรรกศาสตร์ภาคแสดงและประพจน์ทั้งหมดในรูปแบบสูตรที่จัดดีแล้ว สมมุติว่าต้องการตอบคำถาม

Was Marcus loyal to Ceasar? = loyalto(marcus,ceasar) ?

จากการใช้ข้อความที่ 7) และ 8) เราสามารถสรุปได้ว่า Marcus was not loyal to Ceasar โดยพิจารณาจากการให้เหตุผลย้อนกลับดังต่อไปนี้

$\neg$ loyalto(marcus,ceasar)

เราใช้กฎการอนุมานโดยแปลงจากเป้าหมายหลัก ให้เป็นเป้าหมายรอง อื่นๆ ที่สามารถแปลงต่อได้จนกระทั่งเราไม่สามารถหาเป้าหมายรองได้อีกต่อไป เป้าหมายที่ไม่ทราบค่าความจริง เราสามารถใช้การค้นที่ได้ศึกษา ในเรื่องกราฟแอนออร์ (AND-OR graph) ได้ว่า

$\neg$ loyalto(marcus,ceasar)

โดยที่ (7,X=marcus,Y=ceasar)

จะได้ person(marcus) $\wedge$ ruler(ceasar)$\wedge$tryassassinate(marcus,ceasar)

(4) เหลือ person(marcus)$\wedge$tryassassinate(marcus,ceasar)

(8) เหลือ person(marcus)

แต่ไม่สามารถสรุปต่อไปได้ เนื่องจากไม่ทราบข้อความของ person(marcus) จึงต้องเพิ่มค่าความจริงที่ว่า

All men are people. $\forall$X:man(X) $\rightarrow$person(X)

จะได้ X=marcus จากประโยคข้างบน ทำให้เราสรุปได้ว่า Marcus was not loyal to Ceasar

จะได้ว่า

1) ประโยคภาษาอังกฤษส่วนใหญ่คลุมเครือ ถ้าเราพิจารณาประโยคนั้นเพียงประโยคเดียว การเลือกความหมายที่ต้องการอาจยุ่งยาก และต้องใช้ความรู้รอบข้างจำนวนมาก เพื่อสรุปประโยคที่ต้องการ

2) ส่วนใหญ่แล้วมักมีทางเลือกหลายทางในการแทนความรู้ การใช้ตัวแทนที่ง่าย กะทัดรัด อาจทำให้ความหมายบางอย่างสูญเสียไป

3) ความจริงบางอย่างชัดเจนมาก จนกระทั่งคนส่วนใหญ่มักละเลยในการเขียนข้อความลงไป แต่ข้อความนี้อาจเป็นกุญแจสำคัญ ในการสรุปข้อความที่ต้องการและการแทนข้อความจริงที่ชัดเจนทุกข้อความอาจทำให้ได้จำนวนประพจน์ มากเกินความจำเป็น

ปัญหาอีกอย่างหนึ่งคือ เราทราบได้อย่างไรว่าเราควรเริ่มต้นด้วยประโยค

$\neg$ loyalto(marcus,ceasar) แทนที่จะเป็น loyalto(marcus,ceasar)

โปรแกรมหรือขั้นตอนวิธีที่อาจใช้การให้เหตุผลแบบย้อนหลัง (backward reasoning) หรืออาจต้องการให้เหตุผลไปข้างหน้า (forward reasoning)โดยเริ่มจากสมมติฐาน ที่กำหนดแล้วพยายามหากฎที่จำเป็นต้องใช้เพื่อสรุปข้อความที่ต้องการได้ แต่วิธีการนี้มีจำนวนของการเลือกมากจนทำให้เราไม่สามารถหาคำตอบได้ในระยะเวลาจำกัด หรือเราอาจใช้วิทยาการศึกษาสำนึก (heuristic) เลือกกฎ ในกรณีที่เราไม่สามารถสรุป ได้ในเวลาที่จำกัดเราอาจทำการหาทางเลือกอื่น หรือ เราอาจใช้การให้เหตุผลไปข้างหน้าและย้อนกลับพร้อมๆ กัน หรือวิธีอีกวิธีหนึ่งคือพยายามพิสูจน์ ข้อความนั้นพร้อมนิเสธของข้อความ

1. การแทนกรณีตัวอย่างและความสัมพันธ์แบบเป็น-อยู่-คือ

เราทราบว่าความสัมพันธ์ สองแบบที่สำคัญต่อการแทนความรู้คือการแทนกรณีตัวอย่าง (instance relationship) ซึ่งเป็นสมาชิกของคลาส (class membership) และความสัมพันธ์แบบเป็นอยู่คือ (is a relationship) ในเชิงของตรรกศาสตร์ภาคแสดง เราแทนความสัมพันธ์สองแบบนี้โดย

1.1 ใช้ประพจน์ที่มีตัวแปร P(X) ให้ X เป็นสมาชิกของ P เช่น man(marcus) หมายความว่า marcus เป็น สมาชิกของ man

1.2 ใช้กฎการอนุมาน ในการแทนที่ความสัมพันธ์แบบเป็น อยู่ คือ เช่น $\forall$X:pompeian(X)$\rightarrow$roman(X) หมายความว่า สำหรับทุกๆ คนที่เป็น pompeian จะเป็น roman ด้วย

เราอาจแสดงการแทนข้อความที่ต้องการแทนดังตัวอย่างต่อไปนี้

1) man(marcus).
2) pompeian(marcus).
3) $\forall$X:pompeian(X) $\rightarrow$Roman(X).
4) ruler(ceasar).
5) $\forall$X :roman(X)$\rightarrow$loyalto(X,ceasar) $\vee$ hate(X,ceasar).
6) instance(marcus,man).
7) instance(marcus,pompeian).
8) $\forall$X:instance(X,pompeian) $\rightarrow$instance(X,roman).
9) instance(ceasar,ruler).
10) $\forall$X:instance(x,Roman) $\rightarrow$loyalto(x,Ceasar) $\vee$ hate(x,Ceasar).
11) $\forall$X: .$\forall$Y: .$\forall$Z:instance(X,Y) $\wedge$ isa(y,z) $\rightarrow$ instance(x,z)

การเลือกใช้รูปแบบของตรรกศาสตร์ภาคแสดงทั้งสามขึ้นอยู่กับความพอใจของแต่ละบุคคล แต่โดยทั่วๆ ไปแล้วเราจะเลือกตัวแทนที่ง่ายต่อการนำไปใช้ ที่สำคัญที่สุดในการแทนความรู้ประเภทนี้คือการแทนค่าอย่างคงเส้นคงวา นั่นคือหลังจากเราเลือกรูปแบบหนึ่งแล้ว เราต้องแทนความจริงทั้งหมดโดยใช้ รูปแบบนั้นตลอด อย่างไรก็ดีการใช้ตรรกศาสตร์ภาคแสดง ก็มีข้อจำกัดคือ ถ้าเราเปรียบเทียบการแทนความรู้นี้กับโครงข่ายความหมาย (semantic network) ถ้ากรณีตัวอย่างใดไม่มีลักษณะประจำ (attribute) ที่ต้องการเราสามารถอ้างถึงค่าของ ลักษณะประจำกับเซตที่สมาชิกตัวนั้นอยู่ได้ แต่สำหรับตรรกศาสตร์ภาคแสดง ประพจน์ที่พยายามเลียนแบบสมบัติดังกล่าว ไม่สามารถ

เพิ่มเติมได้โดยตรง เช่น ถ้าเราต้องการเพิ่มข้อยกเว้นที่ว่ามี paulus ที่ไม่ loyal to Ceasar และไม่ hate Ceasar

pompeian(paulus)

¬ [loyalto(paulus,ceasar) ∨ hate(paulus,ceasar)]

เราไม่สามารถเพิ่มกฎข้อนี้ไปตรงๆ ได้เพราะกฎจะทำให้เกิดการขัดแย้งกับข้อความที่ว่า

$\forall$X :roman(X) $\longrightarrow$ loyalto(x,Ceasar) ∨ hate(x,Ceasar)

ถ้าเราต้องการใช้ข้อยกเว้นนี้จริงๆ เราต้องไปแก้ไขข้อความดังนี้

$\forall$X : roman(X) ¬ eq(X,paulus) $\longrightarrow$ loyalto(X,ceasar) ∨ hate(X,ceasar)

eq(X,paulus) หมายถึง X เท่ากับ paulus

2. การใช้ฟังก์ชันที่คำนวณได้กับตรรกศาสตร์ภาคแสดง

ประพจน์ของตรรกศาสตร์โดยปกติเราพบว่า การแทนความจริงของข้อความเรามักใช้วิธีการเขียนข้อความออกมาทีละประโยค เช่น

tryassassinate(marcus,ceasar)

ซึ่งการทำเช่นนี้จะใช้ได้กับข้อความที่มีจำนวนตัวแปรที่จำกัด แต่ถ้าต้องการแทนความจริงของมากกว่าหรือน้อยกว่าโดยวิธีนี้ ทำได้โดยการกำหนดให้ gt แทน มากกว่า และ lt แทนน้อยกว่า จะได้ว่า

gt(1,0),gt(2,1),gt(3,2),…

lt(0,1),lt(1,2),lt(2,3),…

อย่างไรก็ตามเราไม่สามารถเขียนข้อความที่เป็นไปได้ทั้งหมด ประพจน์ประเภทนี้เรียกว่าประพจน์ภาคแสดงที่คำนวณได้ โดยสามารถจะทราบค่าความจริงโดยการคำนวณค่าเพื่อให้ได้ค่าความจริงว่าจริงหรือเท็จ เพื่อขยายการใช้งานออกไป โดยจะใช้ฟังก์ชันคณนา (computation function) แทนประพจน์คณนา (computation predicate) เช่นในกรณี

gt(2+3,1)

โดยจะต้องคำนวณค่า 2+3 ก่อน ได้ค่าเป็น 5 แล้วจึงค่อยหาค่าความจริงของ gt(5,1) ว่าเป็นจริงหรือเท็จ ตัวอย่างต่อไปนี้จะแสดงประโยชน์ของการใช้ ประพจน์ภาคแสดงที่คำนวณได้

1) Marcus was a man.=man(marcus)
2) Marcus was a Pompeian.=pompeian(marcus)
3) Marcus was born in 40 A.D.=born (marcus,40)

ในกรณีนี้เราละ AD ไว้เนื่องจากเหตุการณ์ที่เกิดขึ้นอยู่ในช่วงเวลา หลังคริสต์ศักราชโดยตลอด

4) All men are mortal. $\forall$x:man(X) $\rightarrow$ mortal(X)

5) All Pompeian died when the volcano erupted in 79 A.D. =erupted(volcano,79) $\wedge$ $\forall$X:[pompeian(X) $\rightarrow$died(X,79)]

ข้อความข้างต้นประกอบด้วยความจริงสองส่วน คือ การระเบิดของภูเขาไฟ ซึ่งไม่ทราบชื่อ แต่ถ้าค้นจากหนังสือประวัติศาสตร์ จะพบว่า ภูเขาไฟที่กล่าวถึงคือภูเขาไฟวิซุเวียส (Vesuvius)) และการตายของประชาชนชาวปอมเปอี เนื่องจากการระเบิดของภูเขาไปนี้ในครั้งนั้น

6) No mortal lives loger than 150 years. $\forall$X: $\forall$t1: $\forall$t2:mortal(X) $\wedge$ born(X,t1) $\wedge$ gt(t2-t1,150) $\rightarrow$dead(x,t2).

เราอาจแทนประพจน์ภาคแสดงนี้ ด้วยประโยคในลักษณะอื่น คือเราอาจนิยามอายุ age(X) เพื่อหาข้อสรุป แต่ประโยคที่กำหนดให้ก็เพียงพอ ต่อการสรุปที่เราต้องการ

7) It is now 2001.now=2001.

จากคำถาม Is Marcus alive ? พิจารณาประโยคทั้งหมดที่กำหนดให้เราพบว่าสามารถหาข้อสรุปได้สองแบบ คือ Marcus ตายไปแล้วเนื่องจาก เขาตายในเหตุการณ์ของการระเบิดของภูเขาไฟ เมื่อปี ค.ศ. 79 หรือเขาตายไปแล้วเนื่องจาก มนุษย์มีอายุไม่เกิน 150 ปี และเขาเกิดในปี 40 แต่เราพบว่า เรายังขาดความจริงบางอย่างที่ชัดเจนคือ

การมีชีวิตอยู่หมายถึงยังไม่ตาย (alive means not dead) จะได้

$\forall$x : $\forall$t:[alive(x,t) $\rightarrow\neg$ dead(x,t)] $\wedge$ [$\neg$ dead(x,t) $\rightarrow$ alive(x,t)]

หากใครคนหนึ่งตาย แสดงว่าเขาไม่มีชีวิตอยู่หลังจากเขาตาย จะได้

$\forall$x: $\forall$t1: $\forall$t2:died(x,t1) $\wedge$ gt(t2,t1) $\rightarrow$ dead(x,t2)

กล่าวสรุปได้ว่า

1) man(Marcus)
2) Pompeian(Marcus)
3) born(Marcus,40)
4) $\forall$x :man(x) $\rightarrow$ mortal(x)
5) $\forall$x :[Pompeian(x) $\rightarrow$ died(x,79)]

6) erupted(volcano,79)

7) $\forall$x : $\forall$t1: $\forall$t2:mortal(x) $\wedge$ born(x,t1) $\wedge$ gt(t2-t1,150) $\rightarrow$ dead(x,t2).

8) now =2001.

9) $\forall$x : $\forall$t:[alive(x,t) $\rightarrow \neg$ dead(x,t)] $\wedge$ [$\neg$ dead(x,t) $\rightarrow$ alive(x,t)]

$\forall$x : $\forall$t1 : $\forall$t2:died(x,t1) $\wedge$ gt(t2,t2) $\rightarrow$ dead(x,t2)

เราต้องตอบคำถาม Is Marcus alive ? โดยพิจารณา

$\neg$ alive(Marcus,now)

ในการพิจารณาการพิสูจน์ของประโยค

a $\wedge$ b $\rightarrow$ c

การแสดงว่า c เป็นจริงจะใช้เปลี่ยนเป็นการแสดงว่า a และ b จริง ในกรณีที่ a และ b เป็นประพจน์ สามารถแสดงได้โดยไม่มีปัญหา แต่ในกรณีที่ a และ b เป็นประพจน์ภาคแสดงที่มีตัวแปรร่วมกัน จำเป็นต้องแทนค่าตัวแปรด้วยค่าที่เหมือนกัน เช่นในประโยคที่ 3

born(Marcus,t1)

ถ้าต้องการสรุปโดยการใช้ประโยคที่ 7 จำเป็นต้องกำหนดตัวแปรใน gt ให้ t1 = 40 คือ

gt(now-t1,150)

ดังนั้นหากต้องการพิสูจน์โดยการคำนวณที่มีการจับคู่และประกันการแทนค่าที่เหมือนกันตลอดการพิสูจน์ จะได้ว่า การพิสูจน์ข้อความต่างๆ อาจต้องใช้หลายขั้นตอนในการประมวลผล หรือจะต้องใช้ความรู้ในบทที่ 5 เช่นการจับคู่ การแทนที่ และกฎโมดัสโพเนนส์ในการสรุปผลการพิสูจน์

สรุปการพิสูจน์ของ Is Marcus alive? จะได้ว่า

$\neg$ alive(Marcus,now)

(9,x=Marcus,t=now) ได้ dead(Marcus,now)

(10,x=Marcus,t2=now) ได้ died(Marcus,t1) $\wedge$ gt()now,t1

(5,x=Marcus,t1=79) ได้ Pompeian(Marcus) $\wedge$ gt(now,79)

(2) เหลือ gt(now,79)

(8) เหลือ gt(2001,79)

(คำนวณ gt) ได้ True

$\neg$ alive(Marcus,now)

(9,x=Marcus,t=now) ได้ dead(Marcus,now)

(7,x=Marcus,t2=now) ได้ mortal(Marcus,t1) $\wedge$ born(Marcus,t1) $\wedge$ gt(now-t1,150)

(4,x=Marcus) ได้ man(Marcus) $\wedge$ born(Marcus,t1) $\wedge$ gt(now-t1,150)

(1)เหลือ born(Marcus,t1) ∧ gt(now-t1,150)

(3,t1=40) เหลือ gt(now-40,150)

(8) เหลือ gt(2001-40,79)

(คำนวณ gt) ได้ true

รีโซลูชัน

รีโซลูชัน (resolution) เป็นกระบวนการหรือเทคนิคที่ใช้ในการแก้ไขปัญหาเพื่อให้เกิดข้อสรุป เมื่อมีการให้เหตุผลขั้นตอนสุดท้ายของกระบวนการ กระบวนการจะเกี่ยวข้องกับค่าความจริงและความรู้ในฐานความรู้ ซึ่งหมายถึงการรีโซลูชัน จะต้องใช้เหตุผลประกอบเพื่อพิสูจน์ข้อสรุปก่อนที่จะสรุปผลที่ได้

รีโซลูชันเป็นทั้งตรรกศาสตร์ภาคแสดงและประพจน์แคลคูลัส อย่างไรก็ตาม การรีโซลูชันจะต้องทำตามขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 ตั้งสิ่งที่ต้องการจะพิสูจน์ให้เป็นเท็จ ตัวอย่างเช่น ถ้าต้องการจะพิสูจน์ P จะต้องเริ่มพิสูจน์โดยการให้ P เป็นเท็จ นั่นคือ ¬P เรียกวิธีนี้ว่าการพิสูจน์ "Proof by refutation"

ขั้นตอนที่ 2 เปลี่ยนกฎที่มีเครื่องหมาย "ถ้า A แล้ว B" หรือ "A→B" ให้เป็น "¬A∨B" จะกล่าวถึงทีหลัง

ขั้นตอนที่ 3 ใช้กฎของการจับคู่รูปแบบ (pattern matching) และการรวมกฎ (unification) สำหรับการรวมกฎที่เป็นส่วนกลับกัน เข้าด้วยกัน เช่น P และ ¬ P สามารถรวมกันแล้วตัดทิ้งได้ เรียกวิธีการนี้ว่ารีโซเวนต์ (resolvent) และขั้นตอนนี้จะต้องทำหลายๆ ครั้งเพื่อให้เหลือข้อสรุปเพียงข้อเดียว

ขั้นตอนที่ 4 จะได้เงื่อนไขแรกเป็น positive เช่น P ส่วนข้อสรุป คือ ¬P ซึ่งสามารถมาเชื่อมกันได้และสรุปได้ว่า ถ้า P จริง ¬P ก็ต้องเป็นเท็จ

ขั้นตอนที่ 5 สรุปว่า P เป็นจริง เมื่อ ¬P เป็นเท็จ

การจับคู่รูปแบบหมายถึงการที่ประโยคประพจน์ของตรรกศาตร์มีลักษณะการเขียนเหมือนกัน ตัวอย่างเช่น tweety และ tweety แต่ tweety และ tweaty ไม่เหมือนกันเราสามารถใช้สัญลักษณ์เพื่อแสดงกฎ ตัวอย่างเช่น A→B และเราอนุมานว่า A เป็นจริง ดังนั้นประโยค A→B จะเป็นประโยคสมบูรณ์

การรวมกฎเป็นที่เกี่ยวกับรีโซลูชันของประพจน์แคลคูลัสโดยประพจน์ จะต้องประกอบด้วยชื่อประพจน์และอาร์กิวเมนต์หนึ่งตัวหรือมากกว่าดังนั้นเราสามารถแทนค่าของสิ่งต่าง โดยใช้ตัวแปรตัวเดียวได้เช่นถ้าต้องการแทนค่าผู้ชายที่มีอยู่ในโลกนี้ทั้งหมดแทนที่เราจะเขียนว่า man(adam)

$\wedge$man(alex) $\wedge$man(aumnat) $\wedge$... เราสามารถเขียน man(X) โดยที่ X สามารถแทนผู้ชายทั้งโลกนี้ได้ กระบวนการในการจับคู่รูปแบบและการแทนค่าในอาร์กิวเมนต์เรียกว่าการรวมกฎ ตัวอย่างเช่น $\forall$X:man(X)$\rightarrow$mortal(X) และประพจน์ man(adam) ถ้าชื่อประพจน์ "man" เหมือนกันกับโจทย์ adam ก็จะคือ X นั่นเอง กฎที่สมมูลกันแสดงในตารางที่ 6.1

ตารางที่ 6.1 *กฎที่สมมูลกัน*

$P\wedge(Q\wedge R)\Leftrightarrow (P\wedge Q)\wedge R$	Associativity of conjunction
$P\vee(Q\vee R) \Leftrightarrow(P\vee Q)\vee R$	Associativity of disjunction
$P\wedge Q\Leftrightarrow Q\wedge P$	Commutativity of conjunction
$P\vee Q \Leftrightarrow Q\vee P$	Commutativity of disjunction
$P\wedge(Q\vee R) \Leftrightarrow(P\wedge Q)\vee(P\wedge R)$	Distributivity of $\wedge$over$\vee$
$P\vee(Q\wedge R)\Leftrightarrow (P\vee Q)\wedge(P\vee R)$	Distributivity of $\vee$over$\wedge$
$\neg(P\wedge Q)\Leftrightarrow \neg P\vee\neg Q$	de Morgan's Law
$\neg(P\vee Q)\Leftrightarrow \neg P\wedge\neg Q$	de Morgan's Law
$P\Rightarrow Q\Leftrightarrow\neg Q\Rightarrow\neg P$	Contraposition
$\neg\neg P \Leftrightarrow P$	Double negation
$P\Rightarrow Q \Leftrightarrow\neg P\vee Q$	
$P\Leftrightarrow Q\Leftrightarrow (P\Rightarrow Q) \wedge(Q\Rightarrow P)$	
$P\Leftrightarrow Q\ (P\wedge Q)\vee(\neg P\wedge\neg Q)$	

1. รีโซลูชันในตรรกศาสตร์ประพจน์

เพื่อให้เข้าใจถึงกระบวนการ รีโซลูชัน ให้พิจารณาตัวอย่างต่อไปนี้

temperature>100 $\rightarrow$ patient has high temperature

patient has high temperature $\rightarrow$advise two paracet

กำหนดให้

A แทน temperature>100

B แทน patient has high temperature

C แทน advise two paracet

สามารถพิสูจน์ได้ตามลำดับขั้นตอนต่อไปนี้

ขั้นตอนที่ 1

$A \rightarrow B$

$B \rightarrow C$

สมมติว่า A จริง เราสามารถเขียนเงื่อนไขได้เป็น

ขั้นตอนที่ 2

$A \rightarrow B$

$B \rightarrow C$

A

และเราสามารถสรุปได้ว่า B จริง

ขั้นตอนที่ 3

$A \rightarrow B$

$B \rightarrow C$

A

B

สุดท้ายเราจะได้ว่า C จริง

ขั้นตอนที่ 4

$A \rightarrow B$

$B \rightarrow C$

A

B

C

การพิสูจน์ จะต้องพิสูจน์ให้ได้ว่า C จริง ขั้นตอนที่ 1 รีโซลูชัน ต้องสมมติว่า ¬C จริง ขั้นตอนที่ 2 จะต้องเปลี่ยนเงื่อนไขโดยใช้หลักของ กฎเดอมอร์แกน (de Morgan's Law) ดังนั้น State 0 จะได้ดังนี้

ขั้นตอนที่ 1

1. $\neg A \vee B$
2. $\neg B \vee C$
3. A
4. $\neg C$

ขั้นตอนที่ 2

$\neg A \vee \cancel{B}$

$\underline{\cancel{\neg B} \vee C}$

$\neg A \vee C$

เขียนผลที่ได้ลงใน state 1

ขั้นตอนที่ 3

1. $\neg A \vee B$
2. $\neg B \vee C$
3. A
4. $\neg C$
5. $\neg A \vee C$

รวมข้อ3 และ 5 เข้าด้วยกัน จะได้

$\cancel{A}$

$\underline{\cancel{\neg A} \vee C}$

C

ขั้นตอนที่ 4

1. $\neg A \vee B$
2. $\neg B \vee C$
3. A
4. $\neg C$
5. $\neg A \vee C$
6. C

สามารถสรุปได้ว่า เมื่อ C จริง $\neg C$ จะเป็นเท็จ ดังนั้นสามารถสรุปได้ว่า Advise two paracet จริง

2. รีโซลูชันในแคลคูลัสภาคแสดง

รีโซลูชันในแคลคูลัสภาคแสดงแตกต่างจากรีโซลูชันในประพจน์แคลคูลัส เพราะรีโซลูชันในประพจน์แคลคูลัสจะต้องระมัดระวังเรื่องการเทียบค่าของการรวมกฎ ที่ตรงกันข้ามของในแต่ละประพจน์ด้วย ตัวอย่างเช่น man(X) และ $\neg$man(spot) เป็นประโยคตรงข้าม เช่น ถ้าบอกว่า for all X ถ้าเราสามารถบอกได้ว่ามี spot อยู่คนหนึ่งที่ไม่ได้เป็นผู้ชาย ประโยคนี้จะผิดทันที

1) man(marcus)

2) $\neg$man(X)$\vee$mortal(X)

ดูตัวอย่างการ รีโซลูชัน ดังต่อไปนี้

ให้พิจารณาประโยคภาษาอังกฤษต่อไปนี้

1) Marcus was a man.
2) Marcus was a Pompeian.
3) All Pompeians were Romans.
4) Caesar was a ruler.
5) All Romans were either loyal to Caesar or hated him.
6) Everyone is loyal to someone.
7) People only try to assassinate rules they are not loyal to.
8) Marcus tried to assassinate Caesar.

ประโยคสามารถเปลี่ยนเป็นประพจน์แคลคูลัสได้ดังนี้

1) man(marcus)
2) pompeian(marcus)
3) $\forall$X: pompeian(X) $\rightarrow$ roman(X)
4) ruler(caesar)
5) $\forall$X: roman(X) $\rightarrow$ loyalto(X,caesar)$\vee$ hate(X,caesar)
6) $\forall$X:$\exists$Y: loyalto(X,Y)
7) $\forall$X: $\forall$Y: person(X) $\wedge$ruler(Y) $\wedge$ tryassassinate(X,Y) $\rightarrow$ $\neg$loyalto(X,Y)
8) tryassassinate(marcus,caesar)

ต้องพิสูจน์ว่า Marcus hates Caesar. นั่นคือ พิสูจน์ว่า hate(marcus,caesar) โดย วิธีรีโซลูชัน โดยจะต้องกำหนดให้เป้าหมายคือ ประโยคปฏิเสธดังนั้นจะได้เป้าหมาย คือ $\neg$hate(marcus,caesar) เป็นจริง ก่อนที่จะทำรีโซลูชัน จะต้องเปลี่ยนประโยคที่มีเครื่องหมาย $\rightarrow$ ให้เป็นประโยคที่สมมูลกัน เสียก่อนซึ่งจะเรียกว่า รูปแบบอนุประโยคธรรมดา (clause normal form: CNF) ดังนี้

1) man(marcus)
2) pompeian(marcus)
3) $\neg$pompeian(X1) $\vee$ roman(X1)
4) ruler(caesar)
5) $\neg$ roman(X2) $\vee$ loyalto(X2,caesar) $\vee$ hate(X2,caesar)
6) loyalto(X3,X3)
7) $\neg$man(X4) $\vee$ $\neg$ruler(Y1) $\vee$ $\neg$tryassassinate(X4,Y1) $\vee$$\neg$loyalto(X4,Y1)
8) tryassassinate(marcus,caesar)
9) $\neg$hate(marcus,caesar)

ขั้นตอนการทำรีโซลูชันแสดงได้ดังนี้

¬hate(marcus,caesar) 5

3 ¬ roman(X2) ∨ loyalto(X2,caesar)

2 ¬pompeian(marcus) ∨ loyalto(marcus,caesar)

loyalto(marcus,caesar) 7

1 ¬man(marcus) ∨ ¬ruler(caesar) ∨ ¬tryassassinate(marcus,caesar)

¬ruler(caesar) ∨ ¬tryassassinate(marcus,caesar) 4

¬tryassassinate(marcus,caesar) 8

nil

แต่ถ้าต้องการพิสูจน์ข้อความ ¬hate(marcus,ceasar) อาจเติม hate(marcus,ceasar) ซึ่งจะพบว่าประโยคใดที่มี ¬ hate ทำให้ทราบว่าไม่สามารถใช้ รีโซลูชัน สรุปได้ แต่ถ้าเติมประโยคสองประโยคต่อไปนี้เข้าไป

persecute(x,y) → hate(y,x)

hate(x,y) →persecute(y,x)

จะสามารถแปลงประโยคดังกล่าวให้อยู่ในรูปของ clause form ได้ดังนี้

9. ¬ persecute(x5,y2) ∨ hate(y2,x5)

10. ¬ hate(x6,y3) ∨ persecute(y3,x6)

การพิสูจน์ว่า loyalto(marcus,ceasar)

¬ loyalto(marcus,ceasar)   5

marcus/x2

3   ¬ roman(marcus) ∨ hate(marcus,ceasar)

marcus/x1

¬ pompeian(marcus) ∨ hate(marcus,ceasar)   2

hate(marcus,ceasar)

ในกรณีนี้จะไม่พบการขัดแย้ง วิธีที่จะทำให้หยุดการพิจารณาได้ มีเพียงแบบเดียวคือถ้าไม่สามารถสร้างประโยคใหม่ได้อีก เช่น

¬ loyalto(marcus,ceasar)   5

marcus/x2

3   ¬ roman(marcus) ∨ hate(marcus,ceasar)

marcus/x1

¬ Pompeian(marcus) ∨ hate(marcus,ceasar)   2

10   hate(marcus.ceasar)

marcus/x6,ceasar/y2

persecute(ceasar,marcus)   9

marcus/x5,ceasar/y2

hate(marcus,ceasar)

นอกจากนี้ยังสามารถสรุปกฎ นอกเหนือจากการใช้รีโซลูชัน โดยการแทนค่าของตัวแปร หรือค่าคงที่มีค่าเดียวกัน และการลดรูปของประโยคเมื่อพบว่าประพจน์ มีค่าเป็นเท็จเราสามารถตัดการพิจารณาของ ประพจน์นั้นได้เพราะค่าความจริงของประพจน์ที่เป็นเท็จจะไม่มีผลต่อการใช้ OR หรือในกรณีที่เราพบว่า ประพจน์ มีค่าเป็นจริง เราจะไม่สามารถพิสูจน์โดยการขัดแย้ง โดยใช้ประพจน์นี้ได้เพราะประพจน์ที่เป็นจริงกับการใช้ OR จะได้สัจนิรันดร์ (tautology)

3. การตอบคำถามโดยการใช้รีโซลูชัน

เราพบว่าการใช้รีโซลูชันนอกจากจะตอบคำถามว่าประโยคที่กำหนดให้เป็นจริงหรือไม่ อาจยังใช้ตอบคำถามที่ต้องการคำตอบที่เติมในช่องว่างเช่น “When did Marcus die?” หรือ “Who tried to assassinate a ruler?” เช่น ถ้าต้องการหาคำตอบของ

died(marcus,??)

เมื่อ ?? แทนค่าที่ยังไม่ทราบแต่ต้องการรู้ ถ้าสามารถสรุปได้ว่า

Died(marcus,79)

จะทราบได้ทันทีว่าคำตอบคือ 79

ดังนั้นในการใช้รีโซลูชันเพื่อตอบคำถาม “When did Marcus die?” จะต้องใช้ประโยค $\exists$T : died(marcus,T)ซึ่งสามารถเขียนนิเสธได้คือ

$\neg \exists$T : died(marcus,T)

ประโยคดังกล่าวจะขัดแย้ง กับประโยคที่กล่าวว่า

$\forall$T: died(marcus,T)

หรืออาจเลือกประโยคที่ว่า

Died(marcus,DATE)

สำหรับค่าบางค่าของ DATE ให้พิจารณาดังนี้

1) man(Marcus)
2) Pompeian(Marcus)
3) born(Marcus,40)
4) $\forall$x : man(x) $\rightarrow$mortal(x)
5) $\wedge$ $\forall$x :[Pompeian(x) $\rightarrow$ died(x,79)]
6) erupted(volcano,79)
7) $\forall$x: $\forall$t1: $\forall$t2:mortal(x) $\wedge$ born(x,t1) $\wedge$gt(t2-t1,150) $\rightarrow$ dead(x,t2).
8) now = 2001.
9) $\forall$x: $\forall$t:[alive(x,t) $\rightarrow\neg$dead(x,t)] $\wedge$ [$\neg$ dead(x,t) $\rightarrow$ alive(x,t)] $\forall$x: $\forall$t1: $\forall$t2: died(x,t1) $\wedge$ gt(t2,t1) $\rightarrow$ dead(x,t2)

Prove : $\neg$died(marcus,T)

$\neg$ pompeian(x1)$\vee$ died(x1,79)　　　　$\neg$ died(marcus,T)

79/T,marcus/x1

Pompeian(marcus)　　　　$\neg$ Pompeian (marcus)

□

คำตอบที่ต้องการในกรณีนี้สามารถหาได้จากการรวมกฎ ที่เกิดขึ้นของค่า t ในระหว่างการหาข้อขัดแย้ง โดยที่จะต้องพิสูจน์จากประโยคที่ว่างอยู่ กลับมาที่ประโยคขัดแย้งเริ่มต้นและดูค่าที่ต้องการว่าถูกรวม ด้วยค่าใดในการลดการทำงานขั้นนี้ สามารถใช้เทคนิค การเติมนิพจน์ที่ต้องการหาคำตอบ ซึ่งมักจะเป็นนิเสธของข้อความที่ต้องการพิสูจน์ โดยที่เราจะคอยสังเกต การทำงานโดยไม่ให้มีการนำข้อความนี้ไปใช้กับรีโซลูชันดังแผนภาพ

Prove : died(marcus,T)

$\neg$ pompeian(x1) $\vee$ died(x1,79)　　　$\neg$ died(marcus,T) $\vee$ died(marcus,T)

79/T,marcus/x1

Pompeian(marcus)　　　$\neg$ pompeian(marcus) $\vee$ died(marcus,79)

died(marcus,79)

ข้อความพิเศษที่ว่าคือข้อความที่ถูกขีดเส้นใต้ไว้ ข้อความนี้จะถูกรวม เหมือนข้อความอื่นๆ แต่จะไม่ถูกใช้ในการทำการรวมกฎ การทำงานของรีโซลูชัน แบบนี้จะหยุดลงเมื่อเราเหลือข้อความพิเศษเพียงข้อความเดียว

บทสรุป

ตรรกศาสตร์ประพจน์สามารถแทนค่าความจริงทั้งโลกนี้ได้ ในบทนี้ได้กล่าวถึงการแทนความรู้ที่นำสัญลักษณ์ทางคณิตศาสตร์เข้ามาช่วยเพื่อแทนประโยคต่างๆ และการหาคำตอบจากตรรกศาสตร์ภาคแสดง การหาคำตอบโดยตรรกศาสตร์ภาคแสดงนั้นจะต้องมีการแทนกรณีตัวอย่างและแทนความสัมพันธ์แบบเป็นอยู่คือ เพื่อจะทำให้ประโยคสมบูรณ์ขึ้น นอกจากนี้ยังต้องใช้ฟังก์ชันในการคำนวณที่ใช้กับตรรกศาสตร์ภาคแสดง ตลอดจนวิธีการแก้ปัญหาที่เรียกว่า รีโซลูชัน ซึ่งสามารถหาคำตอบของตรรกศาสตร์ภาคแสดงได้

คำถามทบทวน

1. จงให้ความหมายของกรณีตัวอย่าง ว่าสัมพันธ์กับคลาสอย่างไร
2. จงยกตัวอย่างความสัมพันธ์แบบเป็นอยู่คือในคลาสเดียวกันมา 3 ชนิด
3. การใช้กฎการอนุมานในการแทนที่ความสัมพันธ์แบบเป็นอยู่คือทำได้อย่างไร จงอธิบาย
4. รีโซลูชันในตรรกศาสตร์ประพจน์มีหลักการทำงานอย่างไร จงอธิบาย
5. ลักษณะของรูปสูตรที่จัดดีแล้วเป็นอย่างไร
6. การตอบคำถามโดยใช้รีโซลูชันมีหลักการทำงานอย่างไร จงอธิบาย
7. รีโซลูชันในตรรกศาสตร์ประพจน์ และรีโซลูชันกับตรรกศาสตร์ภาคแสดงแตกต่างกันอย่างไร
8. การใช้ฟังก์ชันในการคำนวณกับตรรกศาสตร์ภาคแสดงทำได้อย่างไรจงอธิบาย
9. จงอธิบายวิธีรีโซลูชันของตรรกศาสตร์ประพจน์
10. กำหนดให้
    1) $A \wedge C \rightarrow E$
    2) $D \wedge C \rightarrow F$
    3) $B \wedge E \rightarrow F$
    4) $B \rightarrow C$
    5) $F \rightarrow G$

    ถ้า A และ B จริง จงแปลงข้อความทั้งหมดเป็นรูปแบบอนุประโยคธรรมดา (CNF) และพิสูจน์ว่า G จริง โดยวิธีรีโซลูชัน

บรรณานุกรม

ราชบัณฑิตยสถาน. (2545). **ศัพท์คณิตศาสตร์ฉบับราชบัณฑิตยสถาน.** พิมพ์ครั้งที่ 8 (แก้ไขเพิ่มเติม). กรุงเทพมหานคร :ราชบัณฑิตยสถาน.

ราชบัณฑิตยสถาน. (2540). **ศัพท์คอมพิวเตอร์ฉบับราชบัณฑิตยสถาน.** พิมพ์ครั้งที่ 4. กรุงเทพมหานคร :ราชบัณฑิตยสถาน.

ราชบัณฑิตยสถาน. (2546). **ศัพท์คอมพิวเตอร์และเทคโนโลยีสารสนเทศ ฉบับราชบัณฑิตยสถาน.** พิมพ์ครั้งที่ 6 (แก้ไขเพิ่มเติม). กรุงเทพมหานคร :ราชบัณฑิตยสถาน.

Durkin, J. (1998) **Expert Systems Design and Development.** London: Prentice-Hall International Edition.

Giarratano, Joseph C.and Riley, G. (1994). **Expert Systems Principles and Programming.** Second Edition. Boston:PWS Publishing Co.

Patterson, Dan W. (1990). **Introduction to Artificial Intelligence and Expert Systems.** New Jersey:Prentice-Hall Inc.

Rich, E. and Knight, K. (1991). **Artificial Intelligence. Second Edition.** New yYork:McGraw-Hill.

Russell, Stuart J. and Norving, P. (1999). **Artificial Intelligence: A Modern Approach.** New Jersey:Prentice-Hall Inc.