บทที่ 4
Predicate calculus

ในบทนี้ จะเป็นเรื่องของ formal system ที่เรียกว่า predicate calculus ซึ่งเป็นระบบที่เกี่ยวกับการให้เหตุผลของประโยคซึ่งมีรูปแบบที่ซับซ้อนกว่า proposition (แต่มีโครงสร้างที่คล้ายกัน) ระบบนี้ถูกสร้างขึ้นมาจาก propositional calculus โดยการเพิ่มตัวภาษาให้มีประสิทธิภาพสูงขึ้น และเพิ่มกฎให้กับ deductive apparatus ซึ่งทำให้ formal system นี้เพิ่มความหมายให้กับเรื่อง "equality"

4.1 Predicates

ในเรื่องของ proposition เราได้พิจารณาถึงรูปแบบ และระบบที่ทำให้เราพิจารณาถึงค่าความจริงของ proposition แต่เนื่องจาก proposition นั้นอยู่ในรูปของตัวอักษร และประโยคที่เกิดจากการใช้ตัว connective เชื่อมไว้ ทำให้ระบบนี้ถูกจำกัดลักษณะการใช้รูปประโยค เช่น ประโยค

โดนัลเป็นเป็ด , เป็ดชอบน้ำ ดังนั้น โดนัล ชอบน้ำ

ข้อโต้แย้ง (argument) ดังกล่าวประกอบด้วย proposition 3 ส่วน ซึ่งถ้าแสดงในรูปของ propositional logic อาจแสดงได้เป็น P,Q และ R ซึ่งเมื่อเป็นเช่นนี้เราไม่สามารถจัด ความสัมพันธ์ของ P,Q และ R ได้เนื่องจาก deductive apparatus ไม่มีกฎที่ทำให้เราสามารถอนุมาน R จาก P และ Q

เพื่อที่จะให้เราสามารถจัดข้อโต้แย้งดังกล่าวให้อยู่ในรูปแบบที่ดี เราจึงต้องการ formalism ที่ทำให้เราสามารถแสดงโครงสร้างของตัวอย่างดังกล่าวได้ ในขณะเดียวกันก็ทำให้เราเป็นอิสระจากการให้ความหมายของ expression นั่นคือที่มาของแนวคิดของ predicate

จากตัวอย่างที่กล่าวมาแล้ว เราต้องการสรุปถึง "ความเป็นเป็ด" ของโดนัล ดังนั้นระบบของเราจะต้องสามารถทำให้เราอ้างถึงสิ่งที่เราต้องการสรุปนี้ได้ โดยการกำหนดองค์ประกอบบางอย่างขึ้นมาเพื่อให้เรานำไปใช้แสดงถึงข้อสรุปที่เราต้องการได้ สิ่งนี้ก็คือที่เรียกว่า predicate จากตัวอย่างเราจะกำหนด predicate ที่สรุปถึง "ความเป็นเป็ด" ในรูปของ DUCK(X) โดย X เป็นตัวแปรอิสระ ซึ่งเราสามรถกำหนดให้เป็นอะไรก็ได้ตามความเหมาะสมเพื่อให้เป็น proposition เช่น

CT 488 41

อย่างเราจะกำหนด predicate ที่สรุปถึง "ความเป็นเป็ด" ในรูปของ DUCK(X) โดย X เป็นตัวแปรอิสระ ซึ่งเราสามรถกำหนดให้เป็นอะไรก็ได้ตามความเหมาะสมเพื่อให้เป็น proposition เช่น

DUCK(GROOFY) ถ้า GROOF เป็นเป็ดจริง DUCK(GROOFY) ก็จะเป็นจริง มิฉะนั้น DUCK(GROOFY) จะเป็นเท็จ

การใช้ predicate อาจเกิดปัญหาได้ถ้าเราใช้ชื่อที่ไม่เหมาะสมคือเกิดความสับสน เช่น DUCK(LOVE) ซึ่งจะหมายถึง ความเป็นเป็ดของความรัก ซึ่งเป็นข้อสรุปที่ไม่ให้ความหมายที่แน่ชัด predicate ในรูปของ DUCK(X) เรียกว่า unary predicate นั่นคือ predicate ที่มีที่ใส่ชื่อเพียงแค่ที่เดียว (บางครั้งเรียก one-place predicate) ส่วน n-ray หรือ n-place predicate ใช้ใน predicate ที่แสดงความสัมพันธ์ N อย่างตัวอย่างได้แก่ predicate ของ FATHER OF(X,Y) ซึ่งแสดงถึงข้อสรุปที่ว่า X เป็นพ่อของ Y หรือ predicate ของ SWIM TEAM(A,B,C,D) ซึ่งแสดงถึงข้อสรุปที่ว่า A,B,C,D เป็นนักว่ายน้ำของทีมว่ายน้ำ

4.2 Syntax of predicate logic

predicate logic คือ formal language ที่เราใช้แสดงค่าความจริงของประโยคที่เกี่ยวข้องกับ predicate และ proposition โดย syntax ของ predicate logic จะอยู่ในรูปของ standard syntactic metalanguage

sentence	= simple proposition | predicate
	| "$\neg$",sentence
	| "(",sentence, "$\wedge$",sentence ")"
	| "(",sentence, "$\vee$",sentence ")"
	| "(",sentence, "$\Longrightarrow$",sentence ")"
	| "(",sentence, "$\Leftrightarrow$",sentence ")"
	| "$\forall$",variablename "$\bullet$",sentence
	| "$\exists$",sentence, "$\bullet$",sentence;
simple proposition	= "P" | "Q" | "R"…;
predicate	= predicate name, "(",termlist, ")";
predicate name	= …(*These will be left free*);
term list	= term|term, ",",term list;

term	= proper name|arbitrary name|variable name;
proper name	= …(*These will be left free*);
arbitrary name	= "a" | "b" | "c"…;
variable name	= "x" | "y" | "z"…;

จะเห็นได้ว่ากฎไวยากรณ์นี้ จะให้ประโยคที่เป็นทั้ง proposition logic และ predicate logic ดังนั้น propositions สามารถแสดงได้ในรูปของ P(t) โดย P เป็น n-ray predicate และ t คือชุดของ n ชื่อ ซึ่ง proposition นี้ยังคงมีค่าเป็นจริง แต่สามารถให้รายละเอียดเพิ่มขึ้น

ส่วน predicate นั้น ไม่มีค่าเป็นจริง แต่จะแสดงคุณสมบัติ หรือ ความสัมพันธ์ในรูปของตัวแปร predicate สามารถทำให้เป็น proposition ได้ 2 วิธี คือ

1. โดยกำหนดชื่อให้กับ free variable (ดังตัวอย่าง DUCK(X))
2. โดยการใช้เทคนิคของ quantification

Quantification

เทคนิคนี้ เป็นการระบุความหมายให้กับตัวเอง ซึ่งมีการใช้สัญลักษณ์เพิ่มขึ้น 2 ตัว คือ $\forall$ และ $\exists$ สำหรับ unary predicate,P(X), โดยมี X เป็น free variable เราจะได้ว่า

$\forall X \bullet P(X)$ และ $\exists (X) \bullet P(X)$ เป็น proposition

ตัวแปร X จะถูกกำหนดขอบเขตโดยการ quantify และจะไม่สามารถกำหนดความหมายใหม่ได้

ส่วน predicate ที่มีมากกว่า 1 ตัวแปร เช่น Q(X,Y) การ quantify ตัวแปร ตัวหนึ่ง จะทำให้เกิด predicate ขึ้นมาอีกเนื่องจากตัวแปรอีกตัวหนึ่งยังคงเป็นอิสระ ดังนั้นการ quantify จะเป็นดังนี้

$\forall X \bullet Q(X,Y)$ หรือ $\exists (Y) \bullet Q(X,Y)$

ซึ่งเป็น predicate ในรูปใหม่ ดังนั้นเราสามารถทำการ quantify ตัวแปรที่เหลือผล ที่ได้จะเป็น

$\forall Y \forall X \bullet Q(X,Y)$ หรือ $\forall X \exists (Y) \bullet Q(X,Y)$

การกำหนดชื่อให้กับตัวแปรใน predicate ไม่เป็นสิ่งสำคัญ ดังนั้น

$\forall X \bullet P(X)$ และ $\forall Y \bullet P(X)$

จะแสดง proposition อย่างเดียวกัน แต่การกำหนดตัวแปรต้องระวังไม่ให้เกิดความสับสน เช่นกัน ดังนั้น การกำหนดในลักษณะนี้จึงถือว่าไม่ถูกต้อง

$\forall X \forall Y \bullet Q(X,X)$ หรือ $\forall X \bullet P(Y)$

4.3 Giving predicate logic a semantic

การกำหนดความหมายของ predicate logic ค่อนข้างจะซับซ้อนมากกว่า propositional logic เนื่องจากเป็นภาษามีประสิทธิภาพสูงกว่าใน propositional logic เราให้ความหมายแก่ propositional symbols ทุกตัว รวมทั้งบรรดาตัว connective ด้วย จากนั้นก็จะมีการกำหนดชื่อที่เหมาะสมตามความเป็นจริง อย่างไรก็ตามใน quantified expression นั้นตัวแปรจะไม่มีการกำหนดความหมาย

เราจะใช้ predicate symbol ในการแสดงคุณสมบัติของสิ่งที่เราสนใจ รวมทั้งแสดงความสัมพันธ์ระหว่างสิ่งเหล่านั้น โดยมีข้อจำกัดว่า predicate จะต้องมีจำนวน free variable ตรงตามความหมายที่เราต้องการ เช่น LIKE(X,Y) หมายถึง คุณสมบัติที่ว่า "X เป็นเป็ด" นั้นถือว่าไม่ถูกต้อง เพราะเราไม่ทราบว่า Y เป็นตัวแปรอิสระที่มีบทบาทอะไร

สัญลักษณ์ใหม่ 2 ตัว คือ $\forall$ และ $\exists$ นั้นใช้ในการแสดงความหมายของ universal และ existential ซึ่งความหมายก็คือ $\forall$ นั้นจะมีความหมายว่า "every object has this property" หรือ "all objects art related in this way" ซึ่งความหมายทั้ง 2 นี้ถือว่าเป็น proposition ด้วย เพราะเป็นความหมายที่แฝงค่าของความเป็นจริง และยังคลุมไปถึงกฎของ excluded middle และ contradiction ด้วย แต่เป็นการแสดงในรูปของ predicate

ประโยค $\forall X \bullet P(X)$ เราสามารถตีความในเชิงของ proposition ได้เป็น "เราสามารถกำหนด X ให้เป็นชื่อสิ่งใดๆ ก็ได้ และผลของ proposition ก็ยังคงจริง"

<u>ตัวอย่าง</u>

$\forall X \bullet DUCK(X)$ จะหมายถึง "every object is a duck" ซึ่งเป็นความหมายที่ทำให้เกิดข้อยุ่งยาก ดังนั้นจะเห็นได้ว่าบางครั้งสิ่งบางสิ่งอาจจะไม่สื่อความหมายนักใน predicate บางตัว

พิจารณา predicate ของ GREATER THAN(X,Y) ซึ่งเราจะใช้หมายถึงคุณสมบัติของ X เป็นตัวเลขที่มีค่ามากกว่า Y ดังนั้นในการให้ความหมาย GREATER THAN(7,4) จึงให้ผลที่เป็นจริง แต่ถ้าเป็น GREATER THAN(CUP OF COFFEE,BOB) ซึ่งทำให้เราไม่สามารถบอกได้ว่าเป็นจริง หรือ เท็จ และประโยคเช่นนี้ถือว่าไม่เป็น proposition เนื่องจากมันมีค่ากลาง (middle value) ดังนั้น ถ้าเราจะนำเอาสิ่งต่างๆ ที่มีอยู่ในโลกมาใช้กำหดนความหมาย ก็จะเกิดปัญหาเช่นนี้ เพื่อป้องกันไม่ให้เกิดปัญหาดังกล่าว จึงเกิดแนวคิดในเรื่อง domain ofinterest ขึ้น คือการกำหนดสิ่งที่แน่นอน ที่จะทำให้ predicate มีความหมายได้ โดยการพิจารณาสิ่งที่ สนใจนั้นภายใต้ขอบเขตที่กำหนด

ส่วนประโยค $\exists$(X)•P(X) นั้นจะหมายถึง "there is at least one value in the domain of interest for which the predicate P(X) is true" นั่นคือ predicate P(X) จะไม่เป็นจริงเสมอไปสำหรับทุกๆ ค่าของ X แต่จะเป็นจริง ก็เฉพาะบางค่าของ X เท่านั้น และ X ที่ทำให้ predicate เป็นจริงได้ จะต้องอยู่ในขอบเขตของ domain of interest ด้วย ตัวอย่าง

$\exists$(X)•DUCK(X) หมายถึง มีเป็ดอย่างน้อย 1 ตัว

$\exists$(X)•FATHER_OF(X,BILL) หมายถึง BILL มีพ่อ

สำหรับ $\exists$(X)•FATHER_OF(X,BILL) นั้นจริงๆแล้วไม่ได้หมายความว่า BILL มีพ่อ แต่ความหมายที่ถูกต้องของประโยคนี้คือ BILL มีพ่ออย่างน้อย 1 คน ทำให้เกิดการใช้สัญลักษณ์ตัวใหม่คือ $\exists!$ หรือ $\exists_1$ ซึ่งหมายถึง "there exists exactly one" นั่นก็คือมีเพียง 1 เดียวเท่านั้น

เราจะมาพิจารณาถึงการกำหนดความเป็นจริงในรูปประโยคภาษาอังกฤษ ที่จะแสดงไว้ในรูปของ predicate logic ตัวอย่างประโยค "ducks like the pond" โดยที่ POND ก็คือสิ่งที่เราสนใจใน domain of interest ของเรา สิ่งแรกที่เราต้องทำในการเปลี่ยนประโยคนี้ ให้เป็น predicate ก็คือการกำหนด object และ properties ให้อยู่ในรูปของ term และ predicate โดยเราจะกำหนด

DUCK(X)	หมายถึงobject บางอย่างคือ เป็ด
POND	หมายถึงสระน้ำ
LIKES(X,Y)	หมายถึงX LIKES Y

จากนั้นเราจะมาพิจารณาประโยคภาษาอังกฤษอีกครั้งเพื่อให้เราเปลี่ยนเป็น predicate ได้อย่างถูกต้องจริง ประโยค "ducks like the pond" นี้สามารถเรียบเรียงใหม่ได้เป็น "ไม่ว่าเราจะเลือกค่า X ใดๆ ก็ตาม ถ้า X เป็นเป็ดแล้ว X จะต้องชอบสระน้ำด้วย" นั่นคือประโยนี้ได้แฝงการใช้ประโยค IF X ..... THEN ..... ไว้ ดังนั้นในการทำให้เป็น predicate logic จะมีการใช้สัญลักษณ์ $\Longrightarrow$ ด้วย สรุปได้ว่า predicate logic คือ

$\forall$(X)• (DUCK(X) $\Longrightarrow$ LIKES(X,POND))

จาก predicate นี้ เราพบว่าถ้าไม่มีเป็ดอยู่ใน domain of interest ของเราดังนั้น DUCK(X) ก็จะไม่มีวันที่มีค่าเป็นจริง ซึ่งทำให้ expression เป็นจริง

มาพิจารณาประโยค "there is a duck that likes the pond." ซึ่งต่างจากประโยค "ducks like the pond" ก็คือประโยค "there is a duck that likes the pond" นั้นบอกให้เราทราบว่ามีเป็ดอย่างน้อย 1 ตัวอยู่ใน universe of discourse ดังนั้นเราจะต้องพิจารณาถึงความหมายควรจะเป็น "there is a duck and ...." แทนที่จะเป็น "if there is a duck then ....." ประโยค predicate ของเราก็จะเป็น

$$\exists(X)\bullet(DUCK(X) \wedge LIKES(X,POND))$$

ประโยคต่อมา "ducks like water"

ในการสร้าง predicate ให้กับประโยคนี้ เราต้องพิจารณาให้ละเอียดขึ้นเพราะประโยคนี้ไม่ได้หมายความว่าเป็ดชอบสระน้ำเหมือน "ducks like the pond" แต่กลายเป็นเป็ดชอบสิ่งที่มีคุณสมบัติเป็นน้ำนั่นคือเราต้องกำหนด predicate WATER(Y) ซึ่งหมายความว่า Y คือ สิ่งที่มีคุณสมบัติเป็นน้ำ และความหมายที่ถูกต้องของประโยคก็คือ ถ้ามี X เป็นเป็ดและมี Y เป็นน้ำแล้วจะได้ว่า X ชอบ Y ซึ่งความหมายนี้จะเป็นจริงเสมอไม่ว่าเราจะเลือกกำหนดสิ่งใดให้กับ X และ Y ดังนั้น predicate จะเป็น

$$\forall X \forall Y \bullet ((DUCK(X) \wedge WATER(Y)) \Longrightarrow LIKES(X,Y))$$

มาถึงประโยคสุดท้าย "there is a duck who likes every watery object" ประโยคนี้จัดว่าเป็นประโยคที่มีความหมายคลุมเครือ นั่นคือมันสามารถมีความหมายได้ 2 อย่าง

1. มีเป็ดอย่างน้อย 1 ตัว ที่ชอบน้ำ และชอบสิ่งทุกๆ สิ่งที่เป็นน้ำ
2. ในสิ่งที่เป็นน้ำทุกสิ่ง จะต้องมีเป็นอย่างน้อย 1 ตัวชอบ

ในความหมายที่1 เราสามารถเขียนใหม่ได้เป็น "there is at least one object which is a duck and likes every watery object."

หรือ $\exists X\bullet$ (X IS A DUCK AND X LIKES EVERY WATERY OBJECT)

ซึ่งประโยค "X likes every watery object"

$$\forall Y \bullet WATER(Y) \Longrightarrow LIKES(X,Y)$$

ดังนั้น ประโยคเต็มๆ จะกลายเป็น

$$\exists X\bullet (DUCK(X) \wedge (\forall Y \bullet WATER(Y) \Longrightarrow LIKES(X,Y)))$$

แนวความคิดเรื่องการให้ความหมายกับ propositional logic ถูกนำมาใช้ในการให้ความหมายแก่ predicate logic ซึ่งเราพบว่า การให้ความหมายของ predicate logic นั้นยุ่งยากกว่า เพราะเราต้องให้ความหมายที่จะเป็นไปได้ทั้งหมดกับ term และ predicate ทุกตัว ยิ่งกว่านั้นถ้าเราต้องการแสดงความถูกต้องของ universal quantifier เราจะต้องแสดงให้ได้ว่า predicate นั้นต้องเป็นจริงสำหรับทุกสิ่งที่อยู่ใน domain of discourse ซึ่งขอบเขตนี้อาจเป็น infinte domain ตาราง truth table และวิธีอื่นๆ ที่คล้ายๆ กันจะไม่สามารถนำมาใช้กับ predicate logic ได้

มีรูปแบบการใช้ equivalence ซึ่งมีประโยชน์ต่อ predicate logic ได้แก่

$\forall X \bullet P(X) \equiv \neg\exists X \bullet \neg P(X)$	all true – not false
$\forall X \bullet \neg P(X) \equiv \neg\exists X \bullet P(X)$	all false – not true
$\neg\forall X \bullet P(X) \equiv \exists X \bullet \neg P(X)$	not all true – at least one false
$\neg\forall X \bullet \neg P(X) \equiv \exists X \bullet P(X)$	not all false – at least one true

4.4 Predicate calculus

predicate calculus เป็น formal system โดยการเพิ่ม deductive apparatus ให้กับ predicate logic และ deductive apparatus ของ predicate calculus ถูกสร้างขึ้นมาโดยใช้หลักการของ proposition calculus โดยมีการเพิ่ม กฎใหม่ 4 ข้อ คือ การ introduction และ elimaination ให้กับตัวกำหนดปริมาณ (quantifier) ทั้ง 2 ตัวคือ $\forall$ และ $\exists$ และนอกจากจะพิจารณากฎเหล่านี้ในเชิง syntactic แล้วเรายังสามารถพิจารณากฎเหล่านี้ในด้านความหมายที่ใช้ได้ด้วย ทั้งนี้ก็เพราะเราทราบแล้วว่า กฎที่ใช้นี้มีคุณสมบัติของการเป็น consistency และ completeness แล้วนั่นเอง

มาพิจารณาประโยค $\forall$ X•(P) และกำหนดให้ FRED หมายถึงสิ่งหนึ่งที่อยู่ในขอบเขตของการสนทนา (domain of discourse) เนื่องจากประโยค $\forall$ X•P(X) หมายถึง "P(X)เป็นจริงสำหรับทุกสิ่ง" ทำให้เราสามรถอ้างได้ว่า P(FRED) เป็นจริงลักษณะดังกล่าวนี้ทำให้เราได้กฎของการ elimination ตัว $\forall$ (universal quantifier) ซึ่งเป็นกฎที่ค้อยลงเนื่องจากเรามีการกำหนดให้ FRED เป็นสิ่งเฉพาะเจาะจงลงไป ในทำนองเดียวกัน เราสามารถสร้างกฎที่แกร่งขึ้น โดยการที่เรากำหนดให้ $n_1, n_2, \ldots, n_k$ เป็นชื่อของสิ่งที่อยู่ในขอบเขตของการสนทนา ดังนั้นประโยค $\forall$ X•P(X) จะอนุมานได้เป็น

$$P(n_1) \land P(n_2) \land P(n_3) \land \ldots P(n_k)$$

ซึ่งจะกลายเป็นประโยคขนาดใหญ่ โดยเฉพาะอย่างยิ่งถ้าขอบเขตที่สนทนามีจำนวนไม่รู้จบ ดังนั้นเพื่อไม่ให้เกิดปัญหาดังกล่าว จึงได้มีการใช้แนวความคิดของการใช้ คำตัดสิน (arbitary term) ในการทำ derivation โดยที่คำที่ใช้จะหมายถึงสิ่งใดก็ได้ที่อยู่ในขอบเขตของการสนทนาคือ ไม่มีการใช้ข้อบังคับใดๆ ในการเลือกคำที่จะใช้นั่นเอง ในการทำ derivation เราจะเลือกใช้ตัวอักษร a,b,... แทนคำตัดสิน ผลที่ตามมาก็คือ การกำหนดกฎ

$\forall$ **- elimination**

$$\frac{\forall X \bullet P(X)}{P(a)}$$ โดยที่ a คือค่าตัดสิน

สิ่งตามมาของกฎ $\forall$ - elimination ก็คือ การสร้างกฎ $\forall$ -introduction นั่นคือ เมื่อเราทราบว่าคุณสมบัติบางอย่างถูกอ้างให้กับสิ่งที่เราเลือกให้เป็นค่าตัดสิน เราจะสามารถอ้างคุณสมบัติดังกล่าวให้กับสิ่งทุกสิ่งที่อยู่ในขอบเขตที่เราสนใจ นั่นคือ จะไม่มีการใช้ข้อควบคุมใดๆ กับค่าตัดสิน ทำให้เกิดกฎ

$\forall$ **- introduction**

$$\frac{P(a)}{\forall X \bullet P(a)}$$ โดย a คือค่าตัดสิน

จะเห็น ได้ว่า การ eliminate ตัว universe quatifier ($\forall$) เราไม่ต้องมีการใช้ค่าตัดสินใดๆ ก็ได้ นั่นคือ เราสามรถกำหนดสิ่งเฉพาะ เช่นคำว่า FRED ในตัวอย่างที่กล่าวมาแล้ว ส่วนการ introduce ตัว $\forall$ นั้นเราต้องมีการกำหนดเงื่อนไข (predicate) ซึ่งเกี่ยวข้องกับการใช้ค่าตัดสิน ดังนั้นจึงเป็นเรื่องสำคัญที่จะต้องจำให้ได้ว่าคำที่ใช้นั้นเป็น ค่าตัดสิน หรือ สิ่งเฉพาะ

ส่วนการทำ elimination ให้กับ $\exists$ นั้นจะคล้ายกับ $\forall$ - elimination เว้นแต่ว่า ประโยค $\exists X.P(X)$ จะอนุมานได้เป็น

$$P(n_1) \vee P(n_2) \vee P(n_3) \vee \ldots P(n_k)$$

ซึ่งเราไม่มีวิธีการที่แน่ชัดที่จะจัดการเกี่ยวกับตัว disjunctive ($\vee$) ที่มีจำนวนมากเช่นนี้ อย่างไรก็ตามเราพบว่าถ้าเรามีประโยค $\exists X \bullet P(X)$ และเราสามารถแสดงได้ว่า $(\exists X \bullet P(X)) \Rightarrow Q$ เราจะสามารถสรุปได้ว่าประโยค Q เป็นจริงซึ่งเป็นการ eliminate ตัว existential quantifite ($\exists$) แต่ว่าการแสดงว่า $(\exists X \bullet P(X)) \Rightarrow Q$ เป็นเรื่องยากแต่เราสามารถทำได้โดยการอาศัยกฎของ equivalence นั่นเอง

$\forall X \bullet (P(X)) \Rightarrow Q) \equiv (\exists X \bullet P(X)) \Rightarrow Q$ แล้วทำการ derive ตัว $\forall$ แทน จะทำให้เราได้กฎ

$\exists$ **- elimination**

$$\frac{\exists X \bullet P(X), \forall X \bullet (P(X) \Rightarrow Q)}{Q}$$

ถ้าเรามีประโยค P(t) เราจะได้ว่า P(X) เป็นจริงสำหรับค่า t โดย t อาจจะเป็นชื่อของสิ่งจริง หรือเป็นคำตัดสินที่นำมาใช้ในกฎ $\forall$ - elimination ก็ได้ แต่ไม่ว่าจะเป็นกรณีใดก็ตาม ก็จะเป็นการอ้างว่า มีอย่างน้อยหนึ่งสิ่งทำให้ P เป็นจริง ทำให้เราได้กฎ

$\exists$ - **introduction**

$$\frac{P(t)}{\exists X \bullet P(X)}$$ โดย t เป็นคำตัดสิน

สรุปได้ว่าเราได้กำหนด deductive apparatus ให้กับ predicate calculus แล้ว นั่นคือเราสามารถทำการ proof และ derivation ได้แล้วนั่นเอง

DERIVATION 4.1 แสดงว่า

$$P(m),\ \forall X \bullet (P(X) \Rightarrow Q(X)) \vdash Q(m)$$

<u>DERIVATION</u>

1	$\forall X \bullet (P(X) \Rightarrow Q(X))$	premise
2	$P(m) \Rightarrow Q(m)$	1, $\forall$ - elimination
3	$P(m)$	premise
4	$Q(m)$	2,3, $\Rightarrow$ - elimination

QED.

ถ้าเราให้ความหมายกับคำ และเงื่อนไขที่ใช้ข้างต้น ดังนี้

P(X) แสดงถึง "X เป็น เป็ดตัวหนึ่ง"

Q(X) แสดงถึง "X ชอบบ่อน้ำ"

m แสดงถึง "โดนัล"

ดังนั้นประโยค $\forall X \bullet (P(X) \Rightarrow Q(X))$ แสดงถึง "เป็ดทุกตัวชอบบ่อน้ำ" P(m) แสดงถึง "โดนัลเป็นเป็ดตัวหนึ่ง" และการ derive จะทำให้เราได้ว่า Q(m) ซึ่งแสดงถึง "โดนัลชอบบ่อน้ำ" นี่คือการจัดการกับข้อโต้แย้ง (argument) ที่ propositional calculus ไม่สามารถทำได้

DERIVATION 4.2 แสดงว่า

$$\forall X \bullet P(X) \vdash \forall X \bullet Q(X) \Rightarrow P(X))$$

<u>DERIVATION</u>

1.	$\forall X \bullet P(X)$	premise
2.	Q(a)	assumption, arbitary a
3.	P(a)	1, $\forall$ - elimination
4.	$Q(a) \Rightarrow P(a)$	2, 3, $\Rightarrow$ - introduction
5.	$\forall X \bullet Q(X) \Rightarrow P(X))$	4, $\forall$ - introduction

QED.

ตัวอย่างนี้ไม่จำเป็นต้องอธิบายโดยการให้ความหมายเหมือนกับ DERIVATION 3.1 ทั้งนี้ก็เพราะประโยคนี้แสดงให้เห็นว่า ถ้าส่วนขวามือของโครงสร้าง $A \Rightarrow B$ ถูกอ้างให้เป็นจริงแล้ว ประโยคนี้จะเป็นจริงไม่ว่าส่วนซ้ายมือจะเป็นจริงหรือเท็จก็ตาม

DERIVATION 4.3 แสดงว่า

$$\forall X \bullet \forall Y.P(X,Y) \Rightarrow \neg P(Y,X) \vdash \forall X \bullet \neg(P(X,X)$$

<u>DERIVATION</u>

1	$\forall X \bullet \forall Y.P(X,Y) \Rightarrow \neg P(Y,X)$	premise
2	$\forall Y.P(X,Y) \Rightarrow \neg P(Y,X)$	1, $\forall$ - elimination
3	$P(a,a) \Rightarrow \neg P(a,a)$	2, $\forall$ - elimination
4	P(a,a)	assumption
5	$\neg P(a,a)$	3,4, $\Rightarrow$ -elimination
6	$\neg P(a,a)$	4,5, $\neg$ - introduction
7	$\forall X \bullet \neg(P(X,X)$	6, $\forall$ - introduction

ในตัวอย่างนี้ สิ่งสำคัญที่เกิดขึ้น คือบรรทัดที่ 3 เราพบว่า เราเลือกใช้ คำตัดสิน โดยอาศัยสิ่งอ้างอิงจากโจทย์ นั่นคือ เราเลือกใช้คำตัดสิน a ทั้ง 2 $\forall$ - elimination ที่เกิดขึ้น(ซึ่งจริงๆ แล้วเราสามารถเลือกใช้คำตัดสินใดก็ได้) เพื่อที่จะทำให้เราสามารถสรุปผลได้ตามที่ต้องการ DERIVATION 3.3 เป็นลักษณะของการหาข้อสรุปโดยใช้แนวคณิตศาสตร์เป็นพื้นฐาน โดยเป็นการกล่าวถึงคุณสมบัติ

ของความสัมพันธ์ ตัวอย่างเช่น ถ้าเราให้ P(X,Y) แสดงถึง "การน้อยกว่า" ซึ่งแสดงได้ในรูปของ $X<Y$ ดังนั้น

$\forall X \bullet \forall Y \bullet (P(X,Y) \Rightarrow \neg P(Y,X)$ จะหมายถึง $(X<Y) \Rightarrow \neg (Y<X)$

และจากผลสรุปนี้ ทำให้การ derivation ของเราสรุปเป็นผลได้ว่า ไม่มีค่าของ X ใดๆ ที่จะ ทำให้ $X<X$ คุณสมบัตินี้ไม่เป็นจริงเฉพาะการให้ความหมายดังข้างต้นแต่เพียงอย่างเดียวแต่จะเป็นจริงเสมอสำหรับการตีความทุกๆ กรณีตราบเท่าที่การตีความนั้นจะให้ค่าของ premise เป็นจริง

DERIVATION 4.4 แสดงว่า

$$\forall X \bullet (P(X) \Rightarrow Q(X)), \exists Y \bullet P(Y) \vdash \exists Z \bullet Q(Z)$$

<u>**DERIVATION**</u>

1	$\forall X \bullet (P(X) \Rightarrow Q(X))$	premise
2	$P(a) \Rightarrow Q(a)$	1, $\forall$ - elimination
3	$\exists Y.P(Y)$	premise
4	$P(a)$	assumption
5	$Q(a)$	2,4, $\Rightarrow$ - elimination
6	$\exists Z \bullet Q(Z)$	5, $\exists$ - introduction
7	$P(a) \Rightarrow \exists Z \bullet Q(Z)$	4,6, $\Rightarrow$ - introduction
8	$\forall X \bullet (P(X) \Rightarrow \exists Z \bullet Q(Z)$	7, $\forall$ - introduction
9	$\exists Z \bullet Q(Z)$	3,8, $\exists$ - elimination

ตัวอย่างนี้เป็นการใช้กฎ $\exists$ - elimination เพื่อสรุปผล ดังนั้นเราจึงต้อง derive ประโยค $\forall$ ในบรรทัดที่ 8 เสียก่อน (ดูคำอธิบายในเรื่อง $\exists$ - elimination)

4.5 Equlity

ในส่วนนี้เราจะเพิ่มส่วนประกอบที่สำคัญอีกอย่างหนึ่งให้กับ predicate logic เพื่อสร้าง formal system ของเราให้มีประสิทธิภาพขึ้น และเพื่อให้ใช้ประโยชน์ทางด้าน software engineering ได้อย่างแท้จริง

CT 488

บางครั้ง เราต้องการที่จะให้ชื่อหลายๆ ชื่อกับสิ่งเดียวกัน เมื่อเราอ้างถึงสิ่งนั้นในหลายๆ กรณี โดยที่เรายังคงรู้ดีว่า สิ่งที่เราอ้างถึงเหล่านั้นคือสิ่งเดียวกัน เช่น เราต้องการอ้างถึงตัวเลข 6 รายการอ้างถึงในกรณีอื่นได้ดังนี้

- 3 + 3
- เลขจำนวนเต็มน้อยที่สุดที่มีค่ามากกว่า 5
- ค่ารากที่สองที่มีค่าบวกของ 36

ความต้องการเช่นนี้ ได้สร้างปัญหาให้กับ predicate logic นั่นคือ เราต้องการอ้างถึงชื่อที่เหมาะสมที่สุดให้กับสิ่งที่ใช้ใน predicate ขณะนั้น โดยที่ความของ predicate ดังกล่าวจะต้องเป็นอย่างเดียวกัน ไม่ว่าจะอ้างถึงชื่อใดก็ตาม การแก้ปัญหาทำได้ง่ายโดยการกำหนดกฎที่ซึ่งยอมให้มีการแทนที่ชื่อที่มีความหมายอย่างเดียวกันเข้าไปในประโยคโดยที่ชื่อใหม่ยังคงรักษาความหมายเดิมของประโยคนั้นไว้ได้ (ค่าความจริงของประโยค) ด้วยเหตุนี้เราจึงมีการขยาย formal language ของเราโดยการเพิ่มโครงสร้างลักษณะดังกล่าวเข้าไป เพื่อแสดงถึงเหตุผลที่ว่า ชื่อ 2 ชื่ออ้างถึงสิ่งเดียวกัน ตัวอย่างเช่น เราอาจใช้ predicate ชนิดพิเศษในรูปของ EQUAL(X,Y) ซึ่งตีความให้กับ X และ Y เพื่อแสดงถึงสิ่งเดียวกัน และเพื่อให้สะดวกขึ้น เราจะใช้สัญลักษณ์ "=" แทน ข้อควรระวังก็คือการใช้ "≡" แทน equivalence ซึ่งไม่ถูกต้อง ทั้งนี้ก็เพราะ equivalence เป็นประโยคที่ใช้ใน predicate logic ในขณะที่ equality เป็นความคิดที่เราเพิ่มเข้าไปใน formal system ของเรา นั่นคือ ประโยค $A \equiv B$ หมายความว่า ประโยค A มีค่าความเป็นจริงเหมือนกับประโยคB ส่วน การอ้างอิง $A = B$ จะเป็นประโยคใน formal system ซึ่งอาจเป็นจริงหรือ เท็จ เพื่อให้เราสามารถ proof และ derive เกี่ยวกับ predicate พิเศษตัวนี้ได้ (equal) เราจึงมีการเพิ่ม deductive apparatus ด้วยสิ่งต่อไปนี้

AXIOM OF REFLEXSIVE

$\forall$ **X.X=X**

ซึ่งเป็นประโยคที่แสดงเหตุผลให้เราทราบว่าเราไม่สามารถใช้ชื่อเดียวกันกำหนดสิ่งที่แตกต่างกัน 2 สิ่งได้ และด้วยเหตุนี้เอง เราจึงสามารถเขียนรูปแบบ n=n ได้โดยตรง (คือไม่ต้องใส่ $\forall$ ก็ได้นั่นเอง) และการกระทำเช่นนี้เรียกว่าเป็นการใช้ **axiom schema** เพราะเรามองตัว quantifier เป็น schema ที่ทำให้เราสามารถสร้าง axiom รูปแบบต่างๆ ขึ้นมาได้โดยการกำหนดชื่อเฉพาะลงไป

กฎข้อที่สองที่เราจะเพิ่มเข้าไปก็คือ substitution rule นั่นคือ ถ้าเรามีประโยค m=n แล้วเราจะสามารถแทน m เข้าไปในจุดทุกจุดที่มีการใช้ n ในประโยคที่อยู่ในการทำ derivation เพื่อเป็นการอธิบายรูปแบบนี้อย่างเป็นทางการเราจึงต้องเพิ่มรายละเอียดบางอย่างเข้าไปใน meta-language นั่นคือ เราจะใช้

S(n) เพื่อแสดงถึงประโยคใดๆ ใน predicate logic ที่เกี่ยวข้องกับ n ในลักษณะ ของคำที่ไม่ใช่ bound variable และ

S[m/n] เพื่อแสดงถึงประโยคเช่นเดียวกับ S แต่จะมีการแทน n ด้วย m ในประโยค

SUBSTITUION RULE

$$\frac{m = n, S(n)}{S[m/n]} \quad \text{และ} \quad \frac{m = n, S(m)}{S[n/m]}$$

DERIVATION 4.5 แสดงว่า

$$s=t \vdash t=s$$

<u>DERIVATION</u>

1	s=t	premise
2	s=s	reflexivity
3	t=s	1,2, substitution (replacing first s in 2 by t)

ซึ่งตัวอย่างนี้ เราสามารถนำไปใช้ในการ derivation อื่นๆ ได้โดยเราจะเรียกรูปแบบ s=t $\vdash$ t=s นี้ว่า "symmetry"

DERVIATION 4.6 แสดงว่า

$$s=t \vdash (s=u) \Rightarrow (t=u)$$

<u>DERIVATION</u>

1	s=t	premise
2	s=u	assumption
3	t=u	1,2, substitution
4	$(s=u) \Rightarrow (t=u)$	2,3,- introduction

เราสามารถใช้ equality ในการระบุความคิดได้ว่า

- predicate P(X) จะเก็บสิ่งได้มากที่สุดเพียงสิ่งเดียว

$$\forall X \forall Y \bullet ((P(X) \wedge P(Y)) \Rightarrow (X = Y)$$

หรือพูดได้อีกอย่างหนึ่งว่า P(X) จะเก็บสิ่งได้สิ่งเดียว

$$\exists X \bullet (P(X) \wedge \forall Y \bullet (P(Y) \Rightarrow (X = Y)))$$