Rule, Knowledge & Expert

Unit 2

You Will Know

- Rule & Knowledge
- Logic and First-order logic
- Rule-based knowledge representation
- Rule-based expert system
- Rule Inference

Basic

- Knowledge
  - fact + memory + inference + experiment
- Expert
  - focus on a special domain
- How to think
  - your opinion?
  - 科學人雜誌:通往專家心智之路
    - http://sa.ylib.com/read/readshow.asp?FDocNo=899&CL=4

Rule & Knowledge

- Rule is an import skill for knowledge representation
  - Are there other methods to represent knowledge?
  - What's the limit in using rules?
- An expert system collects a huge set of rules that focus on the interested issues
  - e.g., KingNet 國家網路醫院
  - http://hospital.kingnet.com.tw/

Logic

- Logics are formal languages for representing information such that conclusions can be drawn
- Three issues
  - Syntax
    - Define the sentences in the language
  - Semantic
    - Define the "meaning" of sentences
  - Proof theory
    - Generate new facts from old

Logic

1. Syntax: valid or invalid
2. Semantic: meaningful or not
3. inference: new information

1 $\boxed{x+2 \geq y}$ is sentence; $\boxed{x2+y>}$ is not

2 $x+2 \geq y$ is true iff
the number $x+2$ is no less than the number y

$x+2 \geq y$ is true in a world where $x=7,\ y=1$
$x+2 \geq y$ is false in a world where $x=0,\ y=6$

3 $x+2 \geq y$ and $y \geq c-1$
$\Rightarrow\ x+2 \geq c-1$

First-Order Logic: Syntax

	∧	∨	¬	⇒	⇔
symbol	and	or	not	implies if. . . then	equivalence (biconditional)

- First-order logic
  - Using the above symbols to represent the syntax
- 述詞演算中建立有很多推理規則，可用來從已知的公式集中推出新的公式，這些導出的公式稱爲定理。
- 寫出定理的推理程序及所使用的推理規則序列就構成了該定理的一個證明。
- 在證明定理的演繹過程中，經常要對量化的運算式進行匹配操作，因而需要對項作變數置換使運算式一致起來，這個程序稱作合一。

First-Order Logic: Syntax

- Examples

$$\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$$
$$\neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$$
$$\neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$$
$$S_{1,2} \Rightarrow W_{1,1} \vee W_{1,2} \vee W_{2,2} \vee W_{1,3}$$

➡ Whether the syntax of a rule is valid or not

First-Order Logic: Semantics

"Model" ≡ "completely specified possible world"

Every claim is either true or false

$m \models \varphi$ means
formula φ holds in model m
Otherwise $m \not\models \varphi$

Eg,

	A	B	C	D
m_1	+	0	0	+

$m_1 \models A$
"A is true in m_1"
"m_1 is a model of A"

Also. . .
$m_1 \models \neg B$ $m_1 \models \neg C$ $m_1 \models D$

First-Order Logic: Semantics

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*

Example

$\neg S$ is true iff S is false
$S_1 \wedge S_2$ is true iff S_1 is true and S_2 is true
$S_1 \vee S_2$ is true iff S_1 is true or S_2 is true
$S_1 \Rightarrow S_2$ is true iff S_1 is false or S_2 is true
" is false iff S_1 is true and S_2 is false

$m \models \neg S \equiv m \not\models S$
$m \models S_1 \wedge S_2 \equiv m \models S_1$ and $m \models S_2$
$m \models S_1 \vee S_2 \equiv m \models S_1$ or $m \models S_2$
$m \models S_1 \Rightarrow S_2 \equiv m \not\models S_1$ or $m \models S_2$

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

$p \Rightarrow q$ means $\neg p \vee q$

$p \Leftrightarrow q$ means $(p \Rightarrow q) \wedge (q \Rightarrow p)$

First-Order Logic: Inference

- $p \Rightarrow q$
  - If p is true then we can believe q also is true
  - q is new fact learned from p
- Rule inference
  - Searching rules for current facts
  - Backward and forward inferences

Normal Form of Logics

- Conjunctive Normal Form (CNF – universal)
  conjunction of *disjunctions of literals*
  clauses
  $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

- Disjunctive Normal Form (DNF – universal)
  disjunction of *conjunctions of literals*
  terms
  $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$

p 2-3

Rule

- Two parts in a rule
  - If part: condition of the rule
  - Then part: action of the rule
- Basic syntax of a rule

If <condition>
then <consequent>

<condition> ⇒ <consequent>

where <condition> is represented as CNF or DNF

p 2-4

Rule: Example

關係
If the 'fuel tank' is empty
Then the car is dead

建議
If the season is autumn
and the sky is cloudy
and the forecast is drizzle
Then the advice is 'take an umbrella'

策略
If the car is dead
Then the action is 'check the fuel tank'
If the 'fuel tank' is full
Then the action is 'check the battery'

指示
If the car is dead
and the 'fuel tank' is empty
Then the action is 'refuel the car'

啓發式
If the spill is liquid
and the 'spill pH' < 6
and the 'spill smell' is vinegar
Then the 'spill material' is 'acetic acid'

Rule: Example

If the "traffic light" is green
Then the action is "go"

If the "traffic light" is red
Then the action is "stop"

If the "traffic light" is yellow
Then the action is "be careful"

If the "fuel tank" is empty
Then car is dead

If the "fuel tank" is empty
and the car is dead
Then the action is "refuel"

Expert System

- Expert system contains
  - Expert knowledge
    - Domain knowledge (e.g. disease, help desk, …)
  - A system
    - Knowledge representation
    - Knowledge inference
    - User interface

p 2-6

Expert System

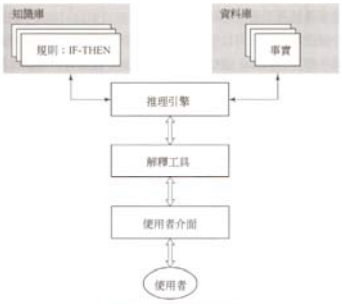
- The team members in the design of an expert system
  - Project manager
  - Domain expert
  - Knowledge engineer
  - Programmer

p 2-8

Rule-Based Expert System

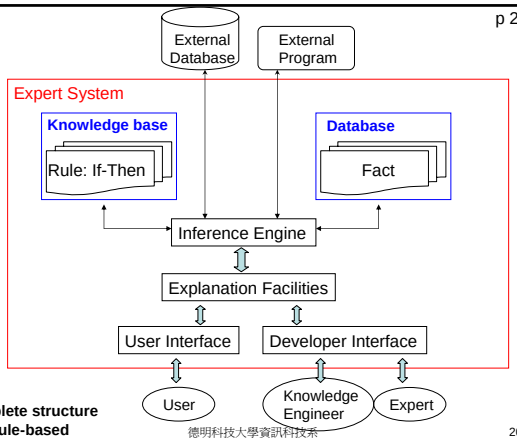
- Five basic components
  - Knowledge base: store rules
  - Database: store facts
  - Inference engine: link rules and facts
  - Explanation facilities: expression of rules to users
  - User interface: communication between a user and an expert system

Productive System



(b)基於規則的專家系統的基本結構

圖 2-2 生產系統模型和基於規則的專家系統的基本結構



complete structure of a rule-based expert system

表 2-1 專家系統、傳統的系統及人類專家的對比

人類專家	專家系統	傳統的程式
在有限的領域中，使用經驗式的或啓發式的知識解決問題	處理以規則表達的知識，在有限的領域中，使用符號推理解決問題	處理資料以及使用演算法和事先定義好的操作步驟來解決普通的數值型的問題
在人類的大腦中，知識以編輯的方式存在	在處理過程中，提供清晰、分離的知識	處理知識時，知識和控制結構不是分離的
能夠解釋推理串列並提供細節	在解決問題過程中跟蹤規則的激發，解釋如何得到具體的結論，爲什麼需要這些特定的資料	對於如何得到結論和爲什麼需要這些輸入資料都不作解釋
允許不確定的推理，可以處理不完整、不確定和模糊的資料	允許不確定的推理，並可處理不完整、不確定和模糊的資訊	僅在資料是完整且確切時，才能解決問題。
當資訊不完整或模糊時可能會出錯	當資料不完整或模糊時可能會出錯	如果資料不完善或模糊，則根本不提供解決方案，或者提供的方案是錯誤的
經過多年的學習和實際的訓練，可以提高解決問題的能力，但這個過程緩慢、效率低並且昂貴	在知識庫中增加新規則或改進原有規則可以改進解決問題的能力。在獲得新知識時，容易實作改變	透過改變程式碼來提高解決問題的能力，對知識及其處理都有影響，導致改變困難

Knowledge

- In a rule-based system, knowledge contains three parts
  - Fact: some terms are true or false
    - e.g. traffic light is green; fuel tank is empty…
  - Rule: mapping from percepts to actions
    - e.g. traffic light is green can imply the action is go
  - Inference: derive new information or facts
    - e.g. refuel the car if you find the fuel tank is empty

Knowledge

- Fact
  - Boolean expression in terms
- Rule
  - First-order logic
  - Syntax and semantics
- Inference
  - Forward inference
    - Search the valid rule and fire it to generate the new facts
  - Backward inference
    - Goal-driven reasoning. Check if a goal is valid in the current facts

Rule 1:	If	Y is true
	And	D is true
	Then	Z is true
Rule 2:	If	X is true
	And	B is true
	And	E is true
	Then	Y is true
Rule 3:	If	A is true
	Then	X is true
Rule 4:	If	C is true
	Then	L is true
Rule 5:	If	L is true
	And	M is true
	Then	N is true

Rule 1: $Y \wedge D \Rightarrow Z$

Rule 2: $X \wedge B \wedge E \Rightarrow Y$

Rule 3: $A \Rightarrow X$

Rule 4: $C \Rightarrow L$

Rule 5: $L \wedge M \Rightarrow N$

Rules in the expert database

partial

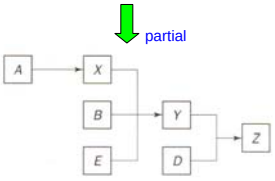


圖 2-5 推理鏈的例子

Rule 1:	If	Y is true
	And	D is true
	Then	Z is true
Rule 2:	If	X is true
	And	B is true
	And	E is true
	Then	Y is true
Rule 3:	If	A is true
	Then	X is true
Rule 4:	If	C is true
	Then	L is true
Rule 5:	If	L is true
	And	M is true
	Then	N is true

Rule 1: $Y \wedge D \Rightarrow Z$

Rule 2: $X \wedge B \wedge E \Rightarrow Y$

Rule 3: $A \Rightarrow X$

Rule 4: $C \Rightarrow L$

Rule 5: $L \wedge M \Rightarrow N$

Rules in the expert database

A, B, C, D, E

Facts in the expert database

Now, we want to derive all implicit facts using these facts and rules

Rule Match

- An expert system needs to search, match, and combine rules as well as facts
  - How to combine rules? **inference**

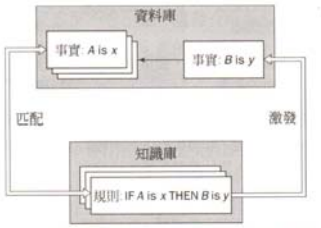


圖 2-4 藉由匹配－激發流程產生的推理引擎循環

Forward (前向) Inference

Initial facts A, B, C, D, E

1. Match rule 3
   Fire rule 3
   New fact: X — A, B, C, D, E, X
   Match rule 4
   Fire rule 4
   New fact: L — A, B, C, D, E, X, L

2. Match rule 2
   Fire rule 2
   New fact: Y — A, B, C, D, E, X, L, Y

3. Match rule 1
   Fire rule 1
   New fact: Z — A, B, C, D, E, X, L, Y, Z

Rule 1: $Y \wedge D \Rightarrow Z$

Rule 2: $X \wedge B \wedge E \Rightarrow Y$

Rule 3: $A \Rightarrow X$

Rule 4: $C \Rightarrow L$

Rule 5: $L \wedge M \Rightarrow N$

Rule base

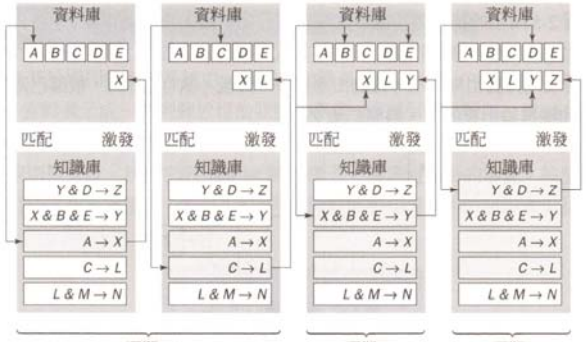


圖 2-6 前向連結的工作過程

Backward Inference

Initial facts A, B, C, D, E

1. Check rule 1: goal is Z
   Depend on Y
2. Check rule 2: sub-goal is Y
   Depend on X
3. Check rule 3: sub-goal is X
   Yes because of A
   New fact: X — A, B, C, D, E, X
   Return to step 2
   New fact: Y — A, B, C, D, E, X, Y
   Return to step 1
   New fact: Z — A, B, C, D, E, X, Y, Z
4. Check rule 4: goal is L
   Yes because of C
   New fact: X — A, B, C, D, E, X, Y, Z, L
5. Check rule 5: goal is N
   Depend on M
   Check 1-4 again
   and return False

Rule 1: $Y \wedge D \Rightarrow Z$

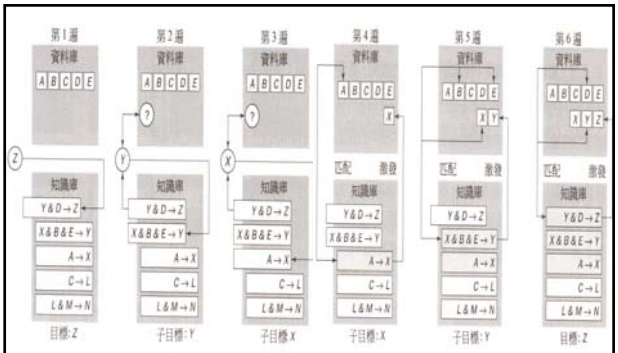
Rule 2: $X \wedge B \wedge E \Rightarrow Y$

Rule 3: $A \Rightarrow X$

Rule 4: $C \Rightarrow L$

Rule 5: $L \wedge M \Rightarrow N$

Rule base



Backward Inference: Step 1 to 3

p 2-19

Forward vs. Backward

- How does a domain expert solve a problem?
  - At the beginning: forward inference to analyze the collective information
  - At difficult problem: backward inference to seek a possible way to the goal facts
- Which is better
  - Forward inference is nature to design an expert system
  - Backward inference is often used for diagnostic purposes
- Combination
  - Most of expert systems use a combination of forward and backward techniques.

p 2-28

Conflict Resolution

- In general, we will have many rules in an expert system
  - Some rules may be conflict

If the "traffic light" is green
Then the action is "go"

If the "traffic light" is red
Then the action is "stop"

If the "traffic light" is red
Then the action is "go"

If "light" is red, what is the action?

⇩

Fire the first matching rule

⇩

Rule ordering is important

Other possible methods?

Conflict Resolution

- Possible methods to solve conflict resolution:
  1. Fire the first matching rule
  2. Fire the highest priority rule
  3. Fire the most informative rule
  4. Fire the newest rule
  5. Append metadata to rules
  6. …

Advantage of Rule-based Expert System

- Natural knowledge representation
  - An expert can understand rules
- Uniform structure
  - If-then structure
  - Each rule is an independent piece of knowledge
- Separation of knowledge from its processing
  - Knowledge and inference engine are independent
  - Flexible in knowledge maintenance

Disadvantage of Rule-based Expert System

- Opaque relations between rules
  - Rule-based systems make it difficult to observe how individual rules serve the overall strategy
- Ineffective search strategy
  - Searching matching rules is slow
- Inability to learn
  - The system only depends on rules
  - Rule-based expert systems do not have the ability to learn from the experience
  - A human expert knows when to "break the rules"