Problem Solving and Reasoning

Rule-Based System (Production System)

C_i : condition

A_i : Action

R_i : rule or production

- Loop continues until working patterns no longer matches the condition of any rules

- PS의 구성요소
  - rule base (production memory, long term memory)
  - working memory (shorter term memory)
  - interpreter (controller)

  < rule >

    If condition 1, then action 1

    If condition 1, then action 1

    …

  “ 각 rule의 조건부는 working memory에 의하여 만족되어야만 선택된다.”

- PS의 특징

1. Separation of knowledge & control
2. Modularity

   규칙의 첨가, 제거, 변화가 독립적으로 가능

   → 규칙간의 독립성

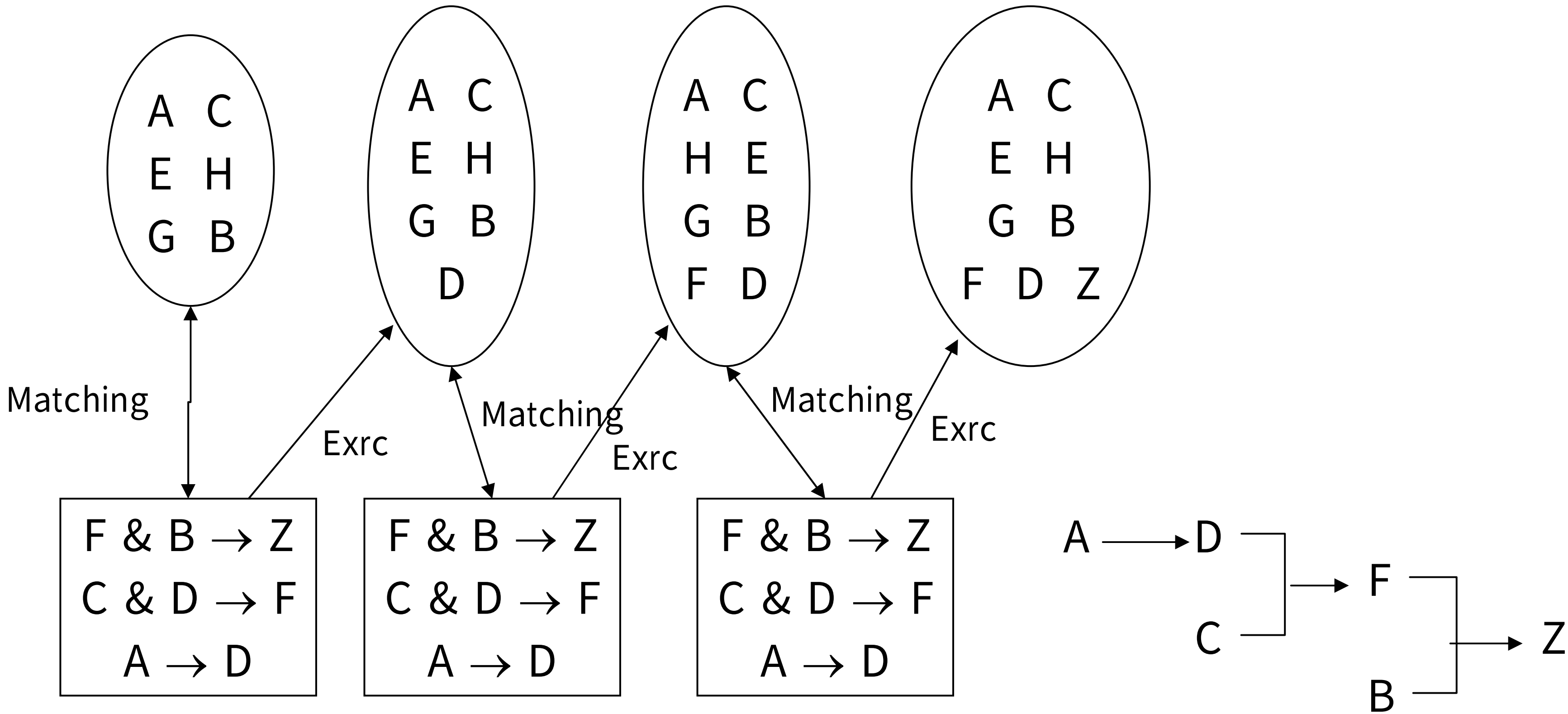
   * 규칙의 수가 증가할 수록 완전한 Modularity의 유지 힘듦
3. Uniformity

   지식의 균일한 구조
4. Naturalness

   지식을 자연스럽게(쉽게) 표현할 수 있음. 특히 " 주어진 situation에 대하여 무엇을 할 것인가" 형태의 지식

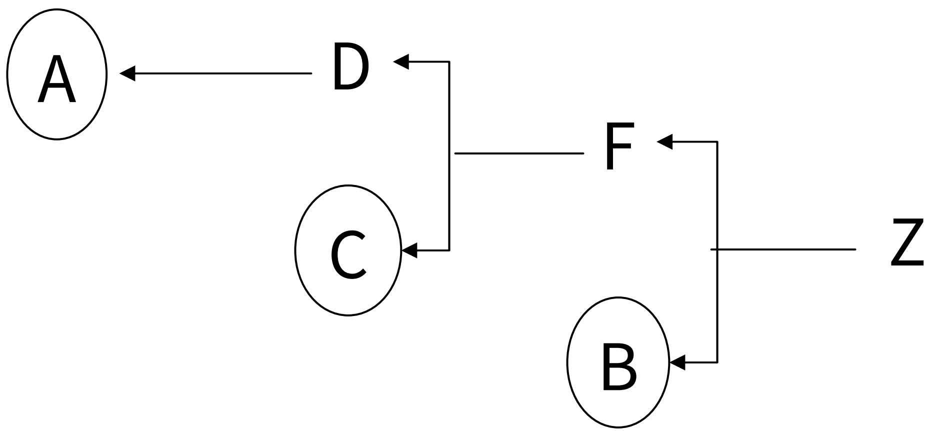
PS의 추론 방법

- forward chaining
- backward chaining



< Forward chaining example >

- If problem is to see “Z” is true or not,
  ← Backward chaining is better.
- Z가 존재하는지의 여부로부터 출발하여 Z를 얻기 위한 규칙만을 수행함. ← more efficient!

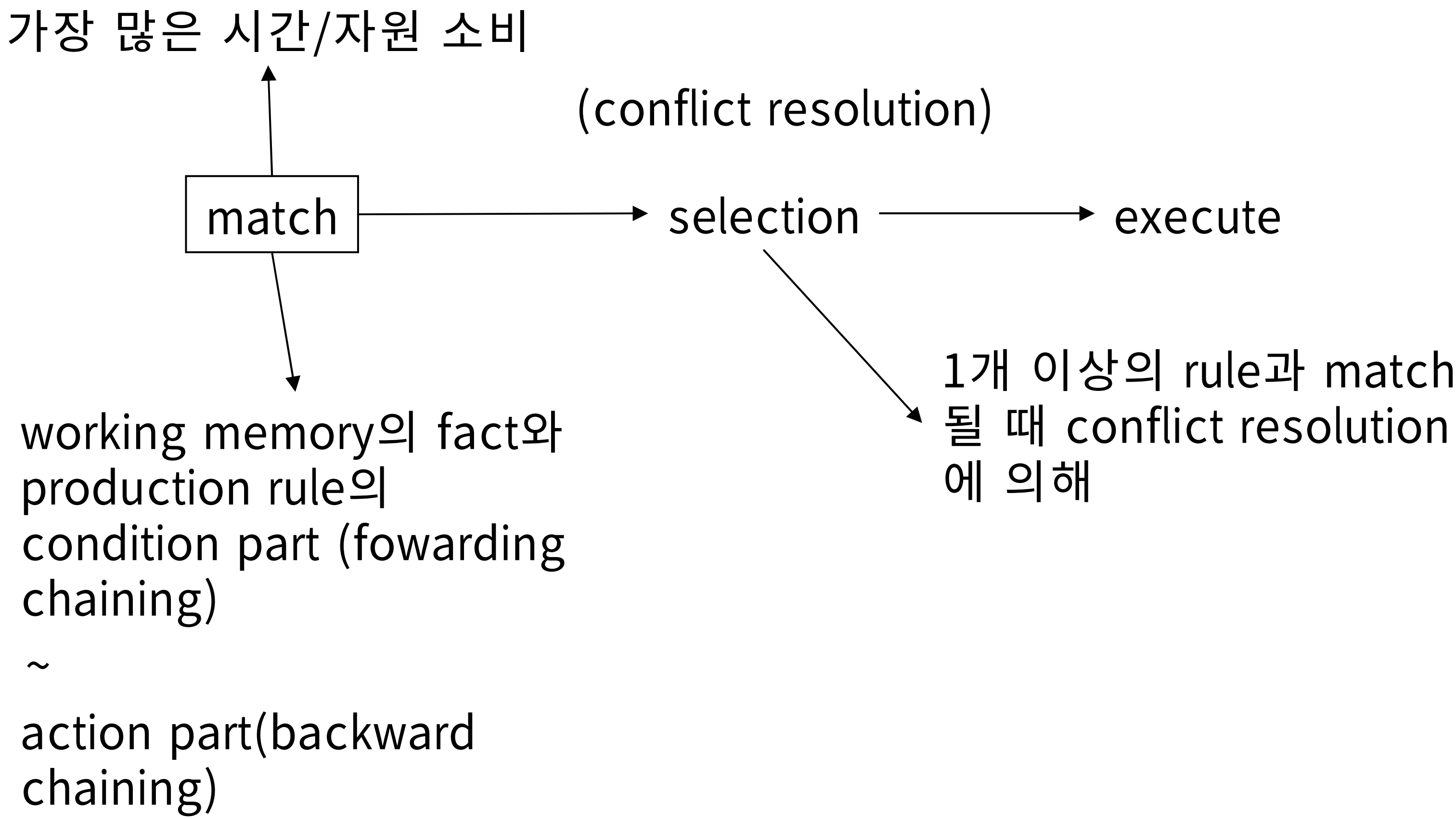


? Backward chaining vs. Abduction

- Backward chaining과 abduction의 차이점
  - backward chaining은 A→B에서 B가 true이기 위하여 필요한 조건 즉, A가 true인가를 찾아보는 것임. 그러므로 결국 (A →B & A) →B의 경우임
  - abduction : (A →B & B) →A
- Working memory의 size가 크면 memory내의 정보를 효율적으로 표현하는 것이 필요요함
  - 예> MYCIN의 context tree

    prospector의 semantic net
  - production rule도 조건부의 성질에 의해 grouping하는 것이 효율적임

< Production system >



- Conflict Resolution
  - 수행 가능한 규칙이 하나 이상인 경우 이들 중 하나를 선택하는 방법
  - Specify ordering : 가장 특수한 규칙부터
  - Rule ordering : rule base에서의 순서
  - Data ordering : priority 순서
  - Recency ordering : 가장 최근에 사용한 규칙(rule)부터(또는 가장 최근에 사용되지 않은 규칙부터)
  - Context limiting : 규칙을 유사한 것끼리 grouping함

- (ex)
  production rule :
  1. ba → ab
  2. ca → ac
  3. cb → bc

iteration	WM	conflict set	rule fired
0	cbaca	1,2,3	1
1	cabca	2	2
2	acbca	3,2	2
3	acbac	1,3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	0	stop

Matching → Conflict Resolution → execution
(interpreter의 수행 순서)

- Matching complexity
  - working memory : w elements
  - production rules : r
  - # of elements in condition part : n
  - # of cycles for solving problem : c
  - total # of match (unification) = wrnc
  - simpler expert system : w = 100
    r = 200
    n = 5
    c = 1000

    ⇒ 100M

- rule-based system을 효과적으로(신속, 정확) 동작 시키기 위해 rule의 구성(structuring)과 순서(ordering)를 조정할 수 있다.
  - p & q & r .... → S의 경우

    앞에 있는 조건(먼저 check되는 조건)들은 confirm이 쉽게 되거나 fail일 가능성이 큰 조건들을 우선적으로 놓는다.

    왜냐하면 fail할 경우 그 rule은 AND이므로 성립하지 않기 때문에 모든 조건들을 check할 필요가 없음

  → easier check first

    more likely fail first

Rule 1 : if

the engine is getting gas, and
the engine will turn over,
then
the problem is spark plugs.

Rule 2 : if

the engine does not tu
and
the lights do not come
then
the problem is battery

Rule 3 : if

the engine does not turn over,
and
the lights do come on
then
the problem is the starter motor.

Rule 4 : if

there is gas in the fue
and
there is gas in the car
then
the engine is getting

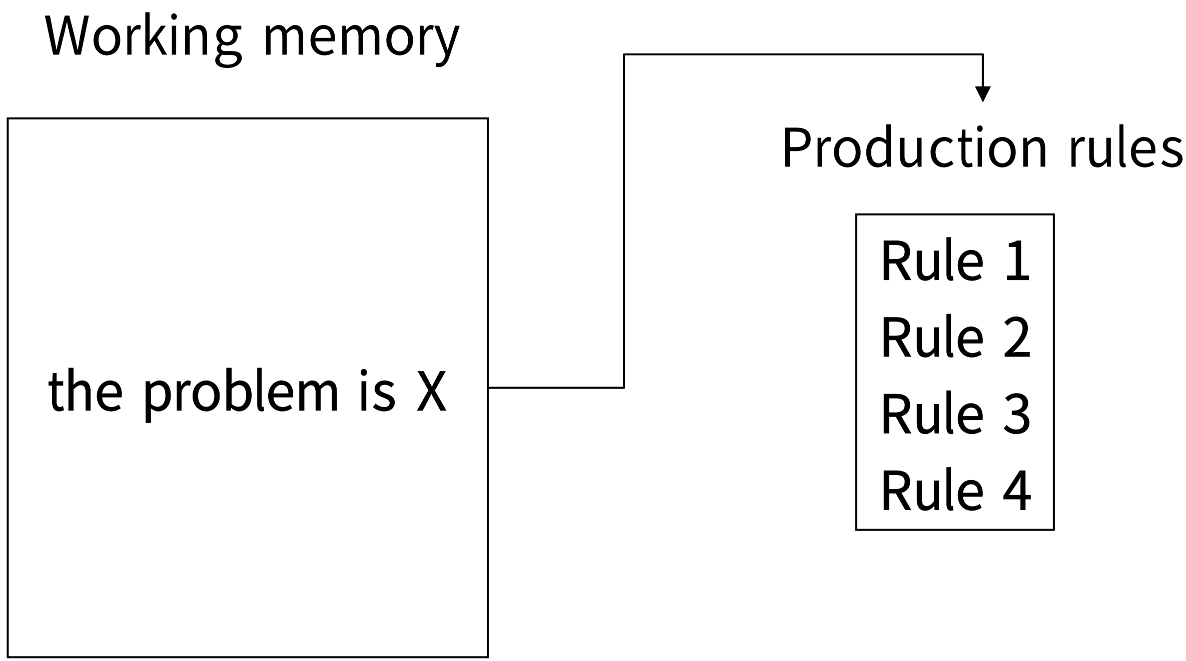


Figure 8.6 The production system at the start of a consultation in the car diagnostic example.

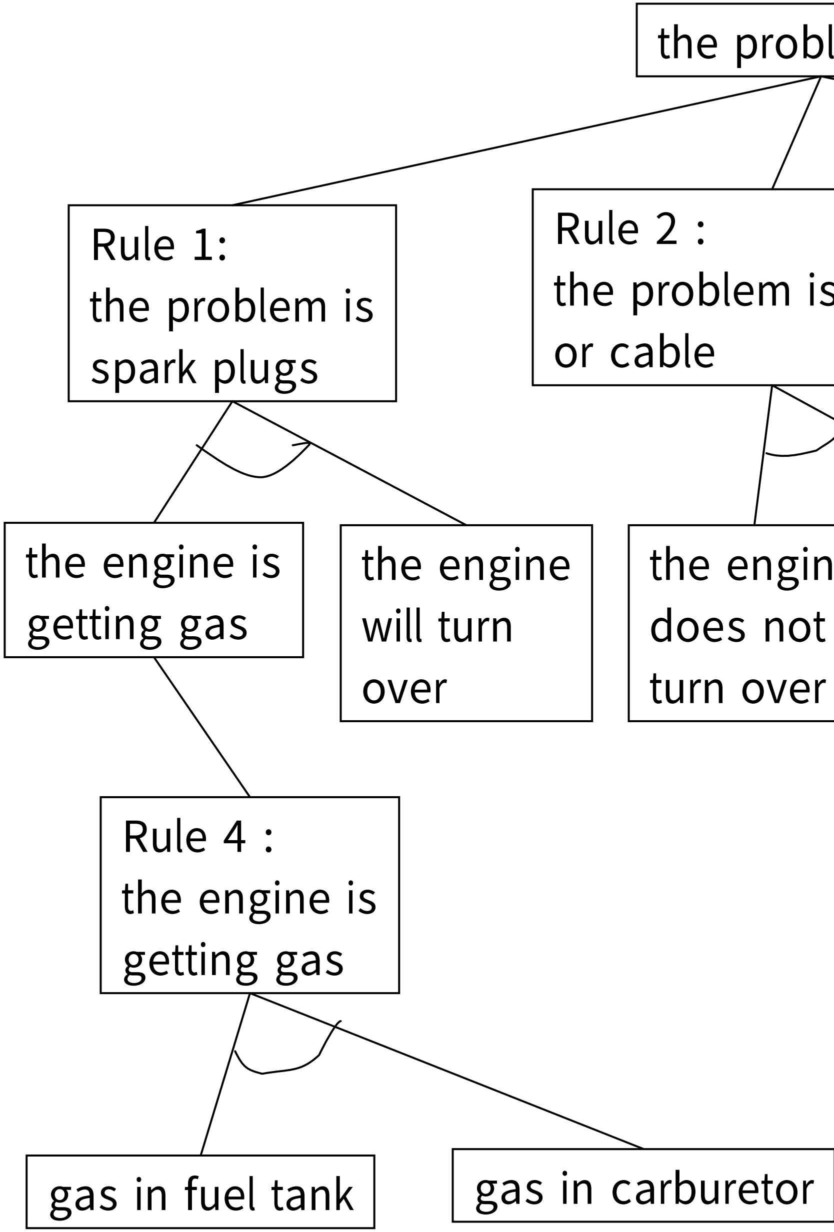


Figure 8.9 And/Or graph searched
in the car diagnostic

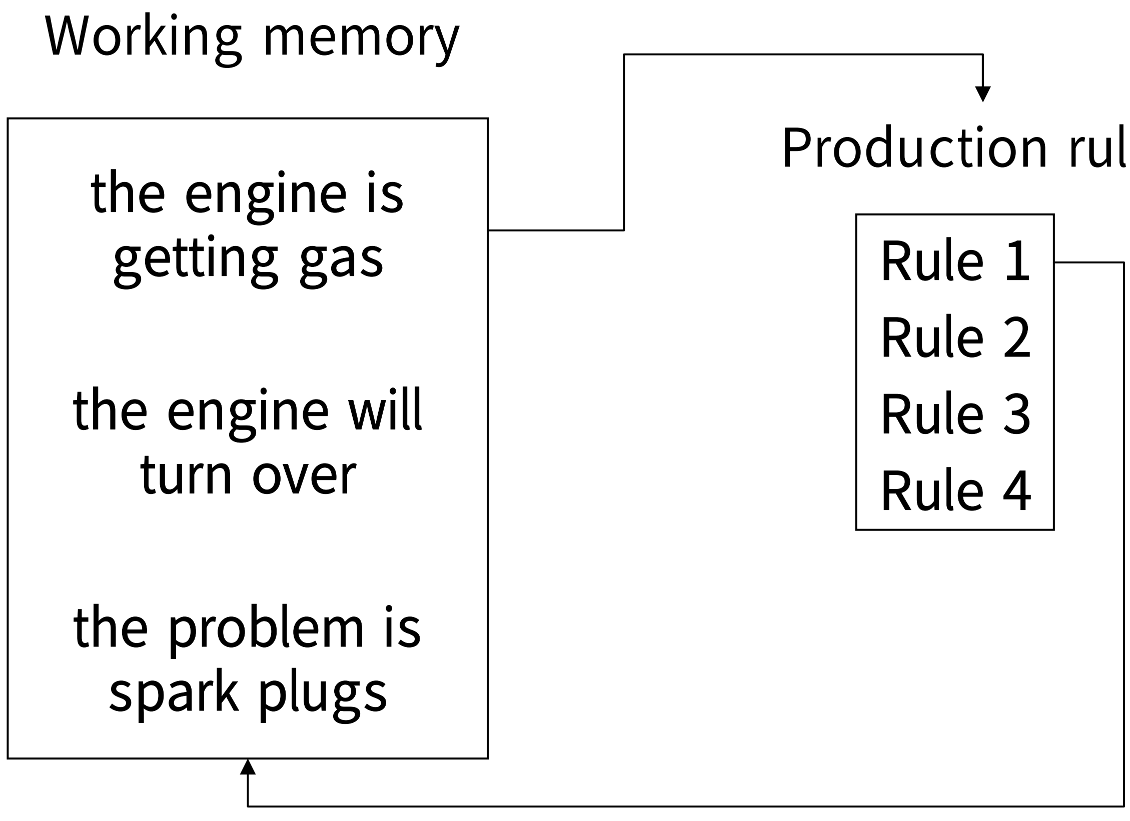


Figure 8.7 The production system after Rule 1 has fired.

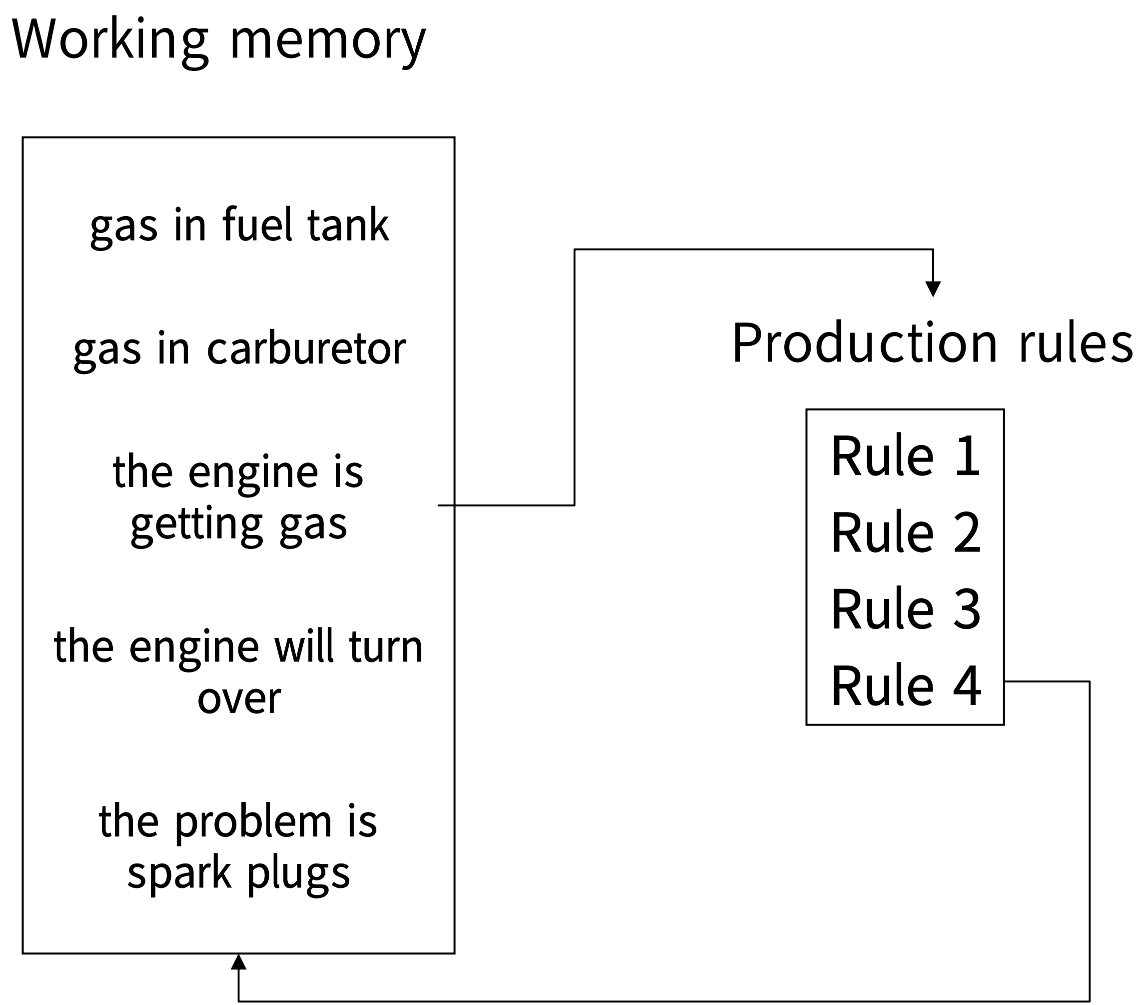


Figure 8.8 The production system after Rule 4 has fired.

(ex) “IDENTIFIER” 7마리의 동물 인식기)

(Cheetah, tiger, giraffe, zebra, ostrich, penguin, albatross)

R1 : If animal has hair
then it is a mammal

R2 : If animal gives milk
then it is a mammal

R3 : If animal has feathers
then it is a bird

R4 : If animal flies
and it lays egg
then it is a bird

R5 : If animal is a mammal
and it eats meat
then it is a carnivore

Mammal or bird

R6 : If animal is a mammal
and it has pointed teeth
and it has claws
and it eyes pointed forward
then it is a carnivore

Carnivore

R7 : If animal is a mammal
and it has hoofs
then it is an ungulate

R8 : If animal is a mammal
and it chew cud
then it is an ungulate
and it is even-toed

Ungulate

R9 : If animal is a carnivore
and it has a tawny color
and it has dark spots
then it is a cheetah

Carnivore

R10 : If animal is a carnivore
and it has a tawny color
and it has black stripes
then it is a tiger

R11 : If animal is an ungulate
and it has long legs
and it has a long neck
and it has a tawny color
and it has dark spots
then it is a giraffe

R12 : If animal is an ungulate
and it has a white color
and it has black stripes
then it is a zebra

Ungulate

R13 : If animal is a bird
and it does not fly
and it has long legs
and it has a long neck
and it is black and white
then it is an ostrich

R14 : If animal is a bird
and it does not fly
and it swims
and it is black and white
then it is a penguin

R15 : If animal is a bird
and it is a good flyer
then it is an albatross

bird

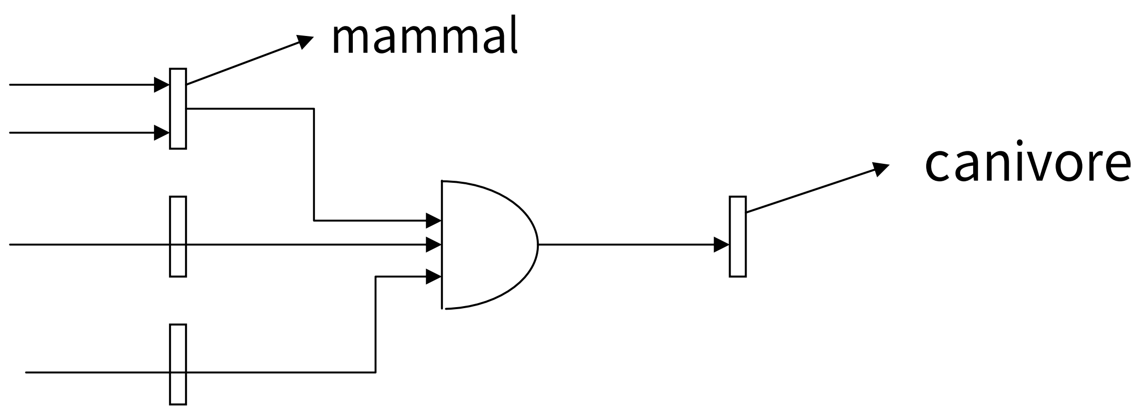
- Now the following has been observed.

  1. Tawny color & dark spots. $\rightarrow$?R_{10} ?R_{11}
  2. Gives a milk & chews its cud. R_2,R_8
  3. long legs & log neck. $\rightarrow$ R_{11}

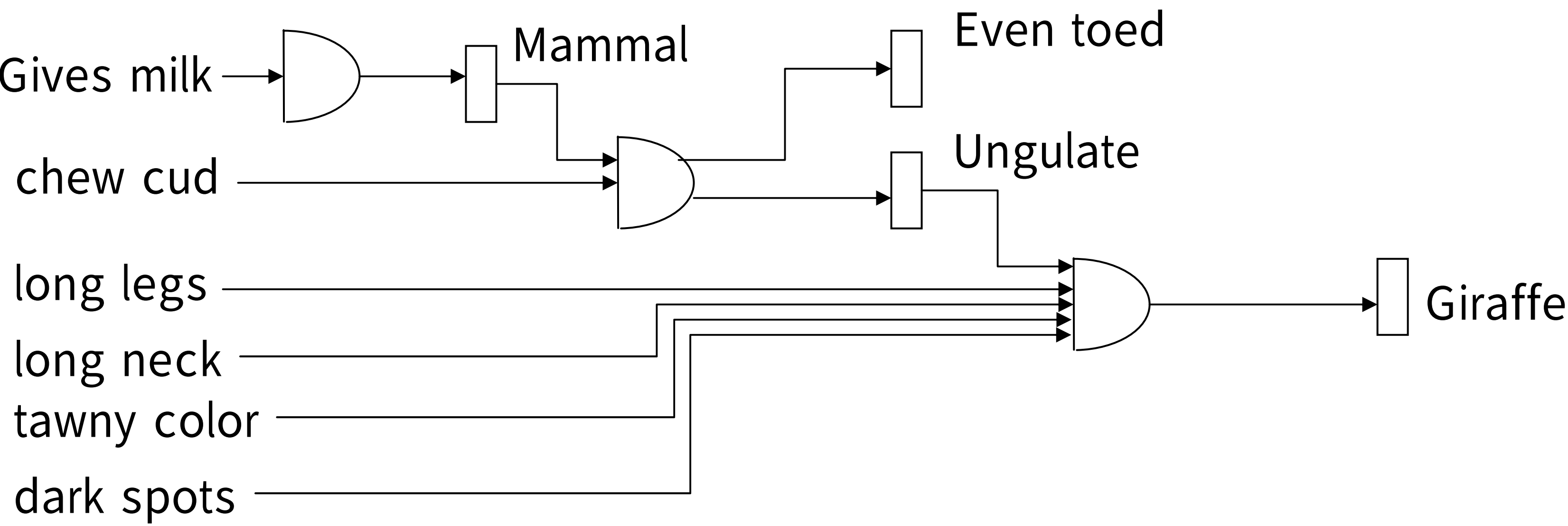
  The animal is giraffe.

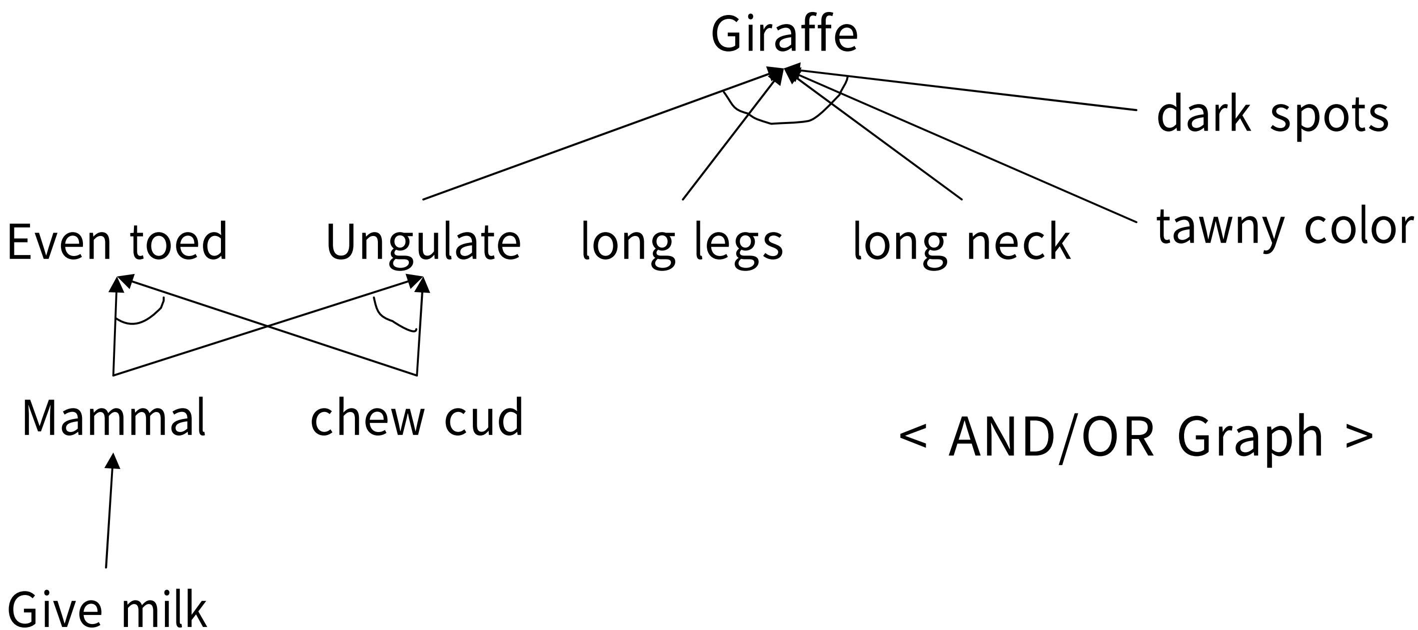
- 추론 과정에 대한 설명
  1. 왜 어떤 사실이 사용되는가? (why)
  2. 어떻게 어떤 사실을 얻어낼 수 있었는가? (how)

- Answer
  - for why : forward로 한단계 진행
  - for how : backward로 한단계 진행



< Inference Net >





< AND/OR Graph >

$$P(H_i \mid E) = \frac{P(E \mid H_i)P(H_i)}{\sum_{k=1}^{n} (P(E \mid H_k)P(H_k))}$$

- $P(H_i|E)$: Prob. that H_i true given evidence E
- $P(H_i)$: Prob. that H true overall
- $P(E|H_i)$: Prob. of observing evidence E when H_i true
- n : number of possible hypotheses

$$P(E) = \sum_{k=1}^{n} P(E \mid H_k)P(H_k)$$

$< Bayes\ theorem >$

$$P(A \mid B)P(B) = P(A, B)$$

$$P(B \mid A)P(A) = P(B, A)$$

$$\therefore P(A \mid B)P(B) = P(B \mid A)P(A)$$

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- Bayes' rule : known fact → unknown fact의 probability 계산

$$P(X|Y)P(Y) = P(Y|X)P(X)$$

$$P(X|Y) = \frac{P(Y|X)}{P(Y)}$$

P(X) : 환자가 감기에 걸릴 확률

P(Y) : 환자가 열이 날 확률

P(X|Y) : 열이 날 때 감기에 걸렸을 확률

P(Y|X) : 감기에 걸렸을 때 열이 날 확률

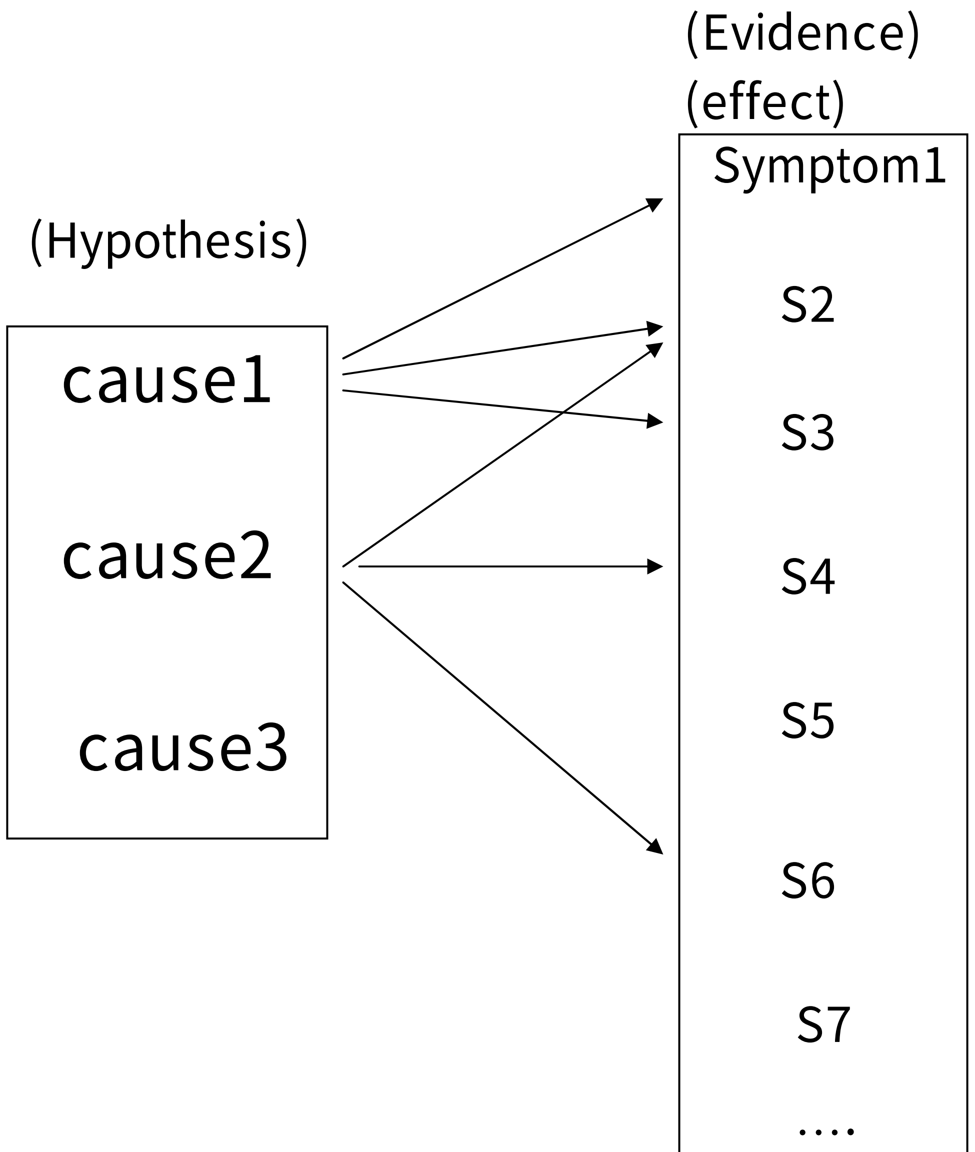
$$e.g.$$

$$P(X) = 0.1$$

$$P(Y) = 0.3 \qquad P(Y|X) = 0.9$$

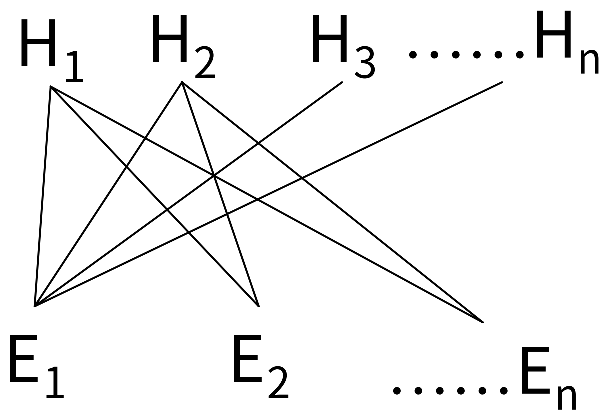
$$P(X|Y) = \frac{0.1 \times 0.9}{0.3} = \frac{0.09}{0.30} = \frac{3}{10} = 0.3$$

< Cause - Symptom(effect)의 relation >



Two major problems of Bayesian

- All relations of evidences are independent.
- Adding new hypotheses and evidences, it is necessary to rebuild probability tables.



$$P(H_i \mid E) = \frac{P(E \mid H_i) P(H_i)}{\sum_{k=1}^{n} (P(E \mid H_k) P(H_k))}$$

Bayes theorem
Multiple evidence case

$$P(Y \mid X, Y) = \frac{P(X \mid Y, E)P(Y \mid E)}{P(X \mid E)}$$

$$P(Z \mid X, Y) = \frac{P(X, Y \mid Z)P(Z)}{P(X, Y)} = \frac{P(X \mid Z)P(Y \mid Z)P(Z)}{P(X, Y)}$$

$$= \alpha(P(Z)P(X \mid Z)P(Y \mid Z)$$

If P(X|Z) & P(Y|Z) are independent

α : normalization constant

? Symptom2가 관측되었을 때 그 원인이 cause1인지 cause2이지 확실치 않음

cause $\rightarrow$ symptom known

symptom $\rightarrow$ cause unknown

rule r_1의 신뢰도를 Bayes rule로 계산한 결과

$$\boxed{P(C_1 | S_2)} = \frac{P(S_2 | C_1)P(C_1)}{P(S_2 | C_1)P(C_1) + P(S_2 | C_2)P(C_2)}$$

C_1 : 감기, C_2 : 배탈

S_1 : 콧물이 남

S_2 : 열이 남

$$P(C_1 | S_1) = \frac{P(S_1 | C_1)P(C_1)}{P(S_1 | C_1)P(C_1) + P(S_1 | C_3)P(C_3)}$$

rule r_3의 신뢰도

Certainty Factor

- Bayes' theorem의 문제점 해결을 위해 사용
- in traditional prob. theory the sum of confidence for a relationship and confidence against the same relationship must add to 1.
- But, often the expert may have confidence 0.7(e.g.) that some relationship is true & have no fealing about its being not true.

- Certainty Factor
  - 관계된 사실이 얼마나 확실한가?

0	~	1
completely wrong		completely right

-1	~	0	~	1
completely wrong		unknown		completely right

  - 조건부의 CF
  - 규칙의 CF
  - 행동부의 CF

확신율 c $r = c/(1-c)$

확신비율 r $c = r/(r+1)$

$c = 0.5 \rightarrow r = 1$

(어떤 사실에 대해 true or false를 말할 수 없는 중간값)

if $c \geq 0.5 \rightarrow r \geq 1$

$c < 0.5 \rightarrow r < 1$

if $c_1 \geq 0.5$, $c_2 < 0.5$ ⟹ $r \geq 1$, $r < 1$ ⟹ $r_1 \cdot r_2 \leq r_1$

$\rightarrow c = (r_1 r_2)/(r_1 r_2+1)$

e.g. $c_1 = 0.9$, $c_2 = 0.25$ ⟹ $r_1 = 9$, $r_2 = 0.25/0.75 = 1/3$

$r_1 r_2 = 3 \rightarrow c = 3/(1+3) = 0.75 < 0.9$

의미 : 다른 규칙에 의해 유도된 결론의 확신율이 0.25로서 결과에 대한 부정적 입장을 취하기 때문(0.5가 중간치)

- C가 dependent한 경우의 Input CF

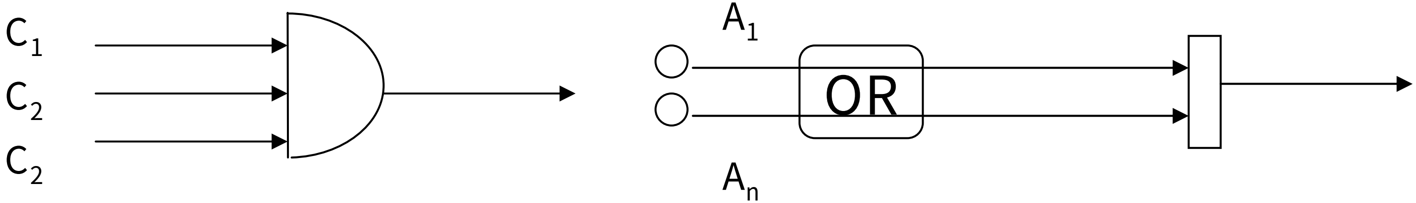
  $\Rightarrow C_1 \times C_2 \times \ldots \times C_n$

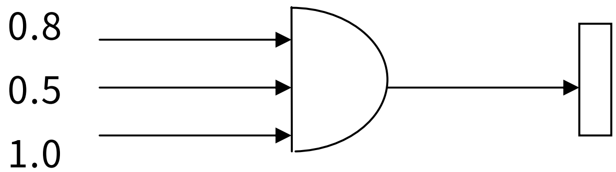
- A가 dependent한 경우의 Multiple argued certainty

  $\Rightarrow R_i = A_i/(1-A_i)$; Certainty Ratio

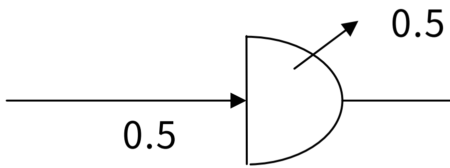
  $\rightarrow R_1 \times R_2 \times \ldots \times R_n = R$

  $\rightarrow A = R/(1+R)$

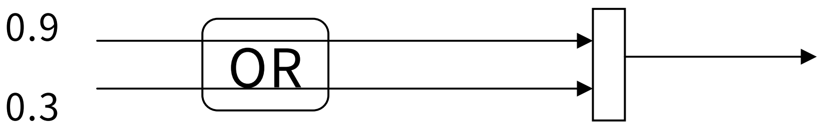




input certainty?



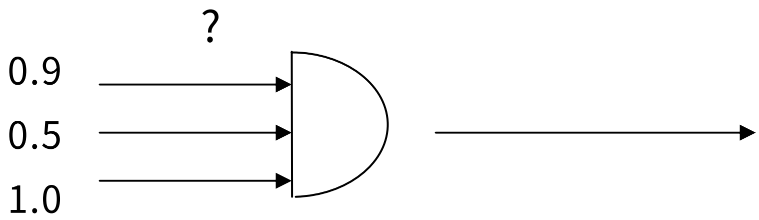
output certainty?

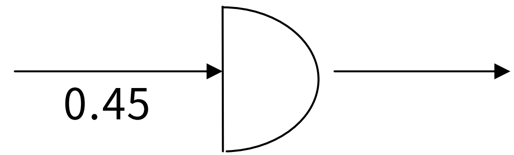


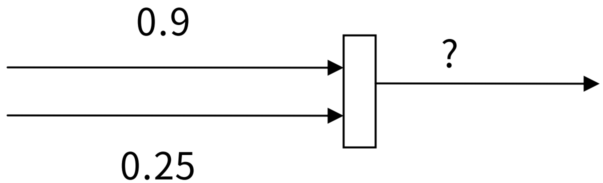
multiple argued certainty?

C_i, A_i : 모두 independent한 경우
(조건부) (결론부)

Multiple argued
⇒ some conclusion with different rules





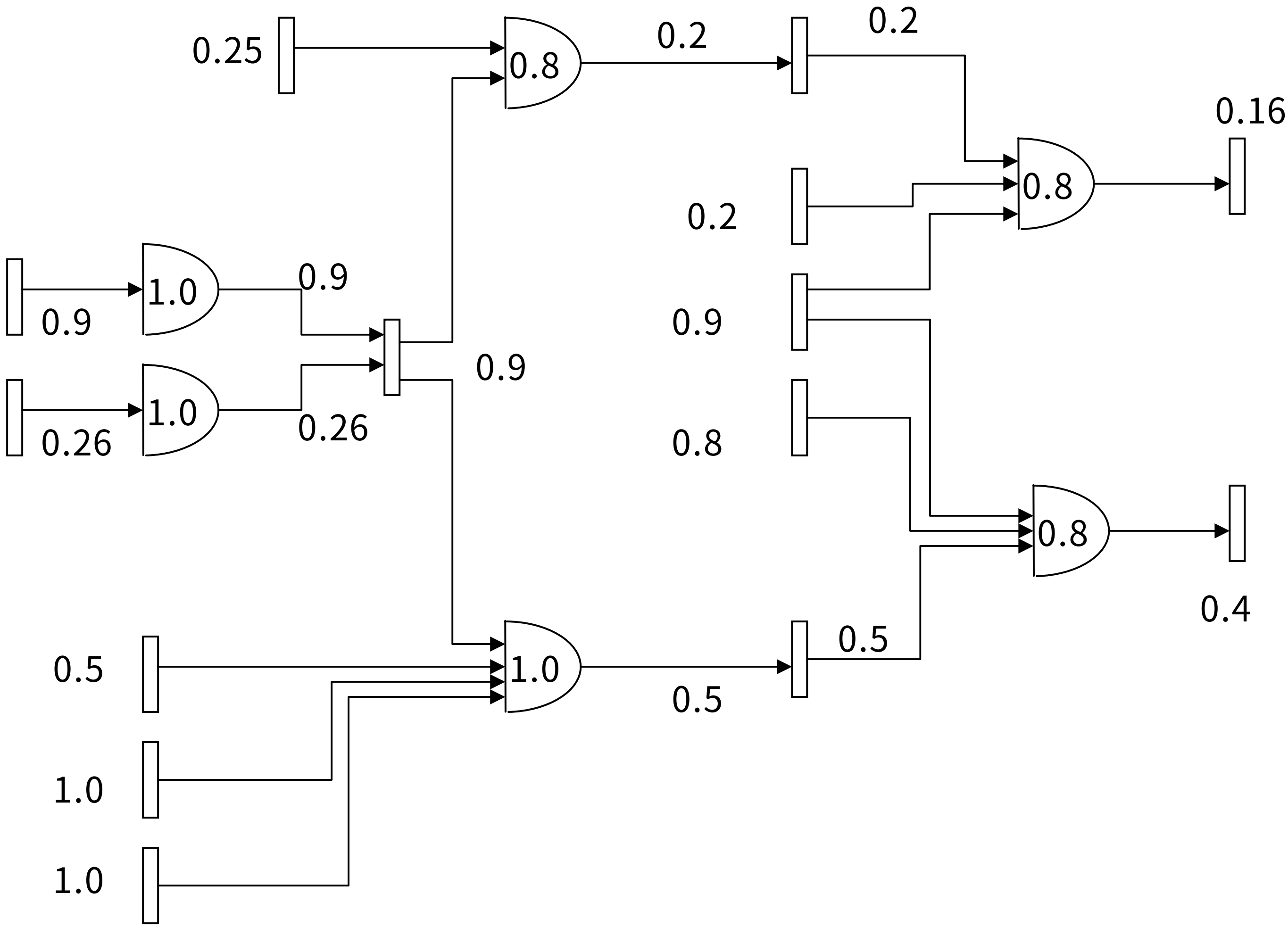


$$CF \quad 0.9 \qquad\qquad 0.25$$

$$CR \quad R_1 = \frac{0.9}{1-0.9} = 9 \quad \frac{0.25}{1-0.25} = \frac{1}{3} = R_2$$

$$R_1 \times R_2 = 9 \times \frac{1}{3} = 3 = R$$

$$C = \frac{R}{1+R} = 0.75$$



0.25
0.8
0.2
0.2
0.16
0.8
0.2
1.0
0.9
0.9
0.9
0.9
1.0
0.26
0.26
0.8
0.8
0.4
0.5
0.5
1.0
0.5
1.0
1.0

Conditional problem

	fever	$\neg$fever
cold	0.04	0.06
$\neg$cold	0.01	0.89

$$P(cold) = 0.04 + 0.06 = 0.1$$

$$P(cold \vee fever) = 0.04 + 0.01 + 0.06 = 0.11$$

$$= P(cold) + P(fever) - P(cold \wedge fever)$$

$$P(cold \mid fever) = \frac{P(fever \wedge cold)}{P(fever)}$$

$$= \frac{0.04}{0.04 + 0.01} = 0.8$$

Dempster - Shaper theory

- deal with the distinction between uncertainty & ignorance
- computes the probability that the evidence supports the proposition rather than computing the probability of a proposition

  → Belief function
- e.g. Coin flipping
  - 동전을 던져서 Head가 나오는 확률에 대한 belief

(1) since we don’t know coin is fair or not

Bel(Head) = 0 Bel(¬Head) = 0

(2) 만일 어떤 전문가가 그 동전이 fair할 certainty가 90%라고 하면

Bel(Head) = 0.9 x 0.5 = 0.45

Bel(¬Head) = 0.9 x 0.5 = 0.45

Dempster-Shaper theory(cont’d)

(3) ~ 100%

Bel(Head) = 1 x 0.5 = 0.5

Bel(¬Head) = 1 x 0.5 = 0.5

Plausibility = 1 – Bel()

(1) : Bel(Head) = 0, Plausibility = 1 – Bel(Head) = 1

< Probability interval >

(1) [0, 1]

(2) [0.45, 0.55]

(3) [0.5, 0.5]

Dempster-Shaper Theorem

- Belief
  - $b(\phi) = 0$
  - $b(\theta) = 1$
  - for all $A \subset \theta$, $0 \leq b(A) \leq 1$
- Support

$$spt(D_j) = \sum_{D_r \subset D_j} b(D_r)$$

- Plausibility
  - $pl(D_j) = 1 - spt(\neg D_j)$

Example 0

		Coin A	
		front(0.8)	back(0.2)
Coin B	front(0.6)	0.48	0.12
	back(0.4)	0.32	0.08

Belief(Coin A(front) and Coin B(front)) = 0.48
Belief(Coin A(front) and Coin B(back)) = 0.32
Belief(Coin A(back) and Coin B(front)) = 0.12
Belief(Coin A(back) and Coin B(back)) = 0.08

Support(same) = Belief(Coin A(front) and Coin B(front)) + Belief(Coin A(back) and Coin B(back)) = 0.56

Plausibility(same) = 1 - Belief(Coin A(front) and Coin B(back)) - Belief(Coin A(back) and Coin B(front)) = 0.56

Example 1

		Melissa	
		broken(0.9)	don’t know(0.1)
Bill	not broken(0.8)	0	(not broken) 0.08
	don’t know(0.2)	(broken) 0.18	(don’t know) 0.02

Belief(broken and not broken) = 0
Belief(broken) = 0.18/0.28 = 0.643
Belief(not broken) = 0.08/0.28 = 0.286
Belief(don’t know) = 0.02/0.28 = 0.071

Support(broken) = Belief (broken = 0.643
Plausibility(broken) = 1 – Belief(not broken) = 0.714

Support(not broken) = Belief(not broken) = 0.286
Plausibility(not broken)= 1 – Belief(broken) = 0.357

Resolution Theorem Proving (Refutation)

- Steps

1. Put the premises into clause form
2. Add the negation of what is to be proved in clause form to the set of premises.
3. Resolve these clauses together, producing new clauses that logically follows.
4. Produce a contradiction by generating the empty clause.

- Resolution example
  1. Whoever can read is literate.
  2. Dolphines are not literate.
  3. Some dolphines are intelligent cannot read?

  $\rightarrow$ Some who are intelligent cannot read?

  1. $(\forall X)[R(X) \Rightarrow L(X)]$
  2. $(\forall X)[D(X) \Rightarrow \neg L(X)]$
  3. $(\exists X)[D(X) \wedge I(X)]$
  4. $(\exists X)[I(X) \wedge \neg R(X)]$

  1; $\neg R(X) \vee L(X)$

  2; $\neg D(Y) \vee \neg L(Y)$

  3; $D(a) \wedge I(a)$

  $\neg$4; $\neg I(Z) \vee R(Z)$

  ※ $\neg\exists X[\sim] = \forall X \neg[\sim]$

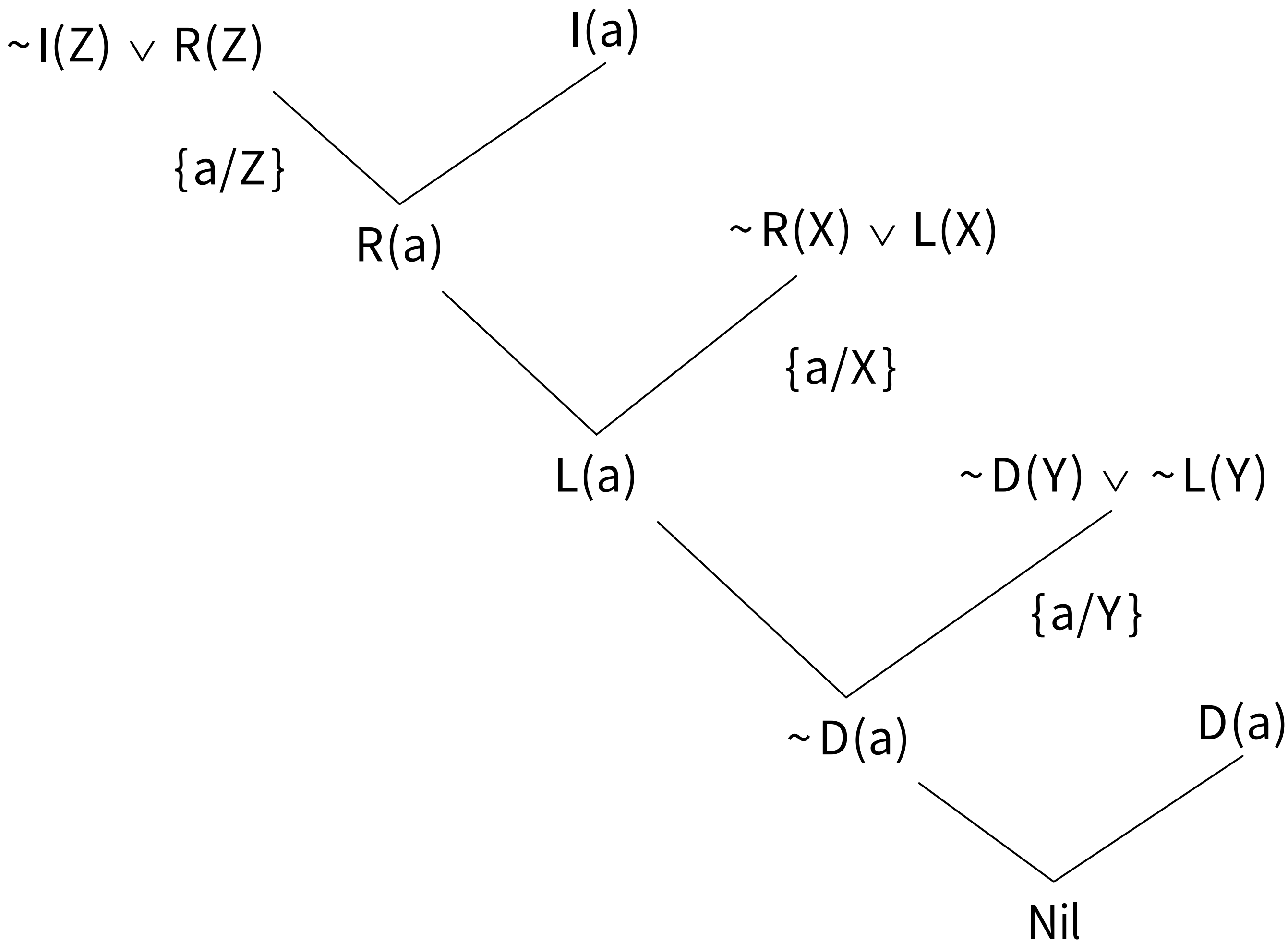
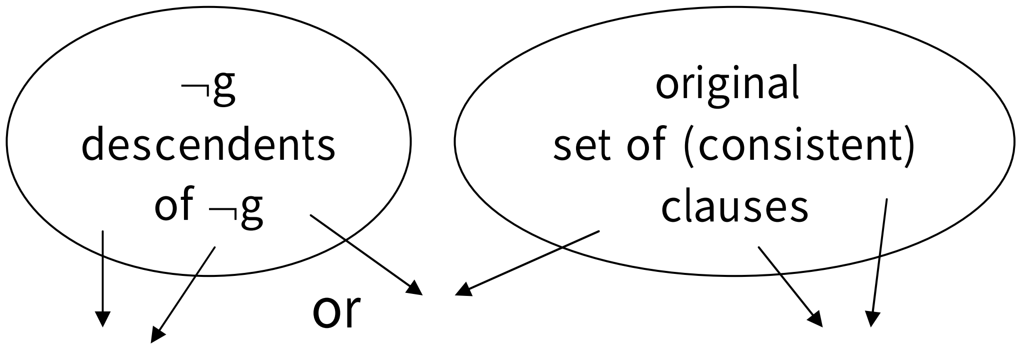


그림 5 : resolution example

therefore $I(a) \wedge \neg R(a)$

- Breadth First Strategy : complete
- Set of support strategy
  - at least one of the resolvents is either the negated goal clause or a clause produced by resolutions on the negated goal → complete



- Unit Preference strategy : complete

  ※ Unit resolution : not complete

  → one of the resolvents always be a unit clause.
- Linear Input Form Strategy : not complete

- Once the theorem prover shows that the negated goal is inconsistent with the given set of axioms, it follows that the original goal must be consistent $\rightarrow$ proof of the theorem
- Resolution is a sound inference rule
  - $(P \lor Q) \land (\neg G \lor R) \rightarrow P \lor R$

    $P \lor R$ logically follows from $(P \lor Q) \land (\neg G \lor R)$

    $\therefore$ sound
  - But it is not complete.

    i.e. given set of axiom으로 부터 logically follow하는 모든 fact를 생성할 수 없다.
  - refutation complete.

    i.e. the empty or null clause can always be generated wherever a contradiction in the set of clauses exists.
- Refutation
  - inference procedure using resolution, proof by contradiction

- Refutation
  - proof by contradiction
  - To prove P, assume P is false & prove a contradiction
  - $(S \wedge \neg P \Rightarrow \text{false}) \Leftrightarrow (S \rightarrow P)$

- resolution example

given,

$$\left\langle \begin{array}{l} 1.\ b \wedge c \Rightarrow a \\ 2.\ b \\ 3.\ d \wedge e \Rightarrow c \\ 4.\ e \vee f \\ 5.\ d \wedge \neg f \end{array} \right\rangle \Longrightarrow \text{“prove a”}$$

clause form

$$\left\langle \begin{array}{l} 1.\ \neg(b \wedge c) \vee a \Rightarrow \neg b \vee \neg c \vee a \\ 2.\ b \\ 3.\ \neg(d \wedge e) \vee c \Rightarrow \neg d \vee \neg e \vee c \\ 4.\ e \vee f \\ 5.\ d \\ 6.\ \neg f \\ 7.\ \neg a \end{array} \right.$$

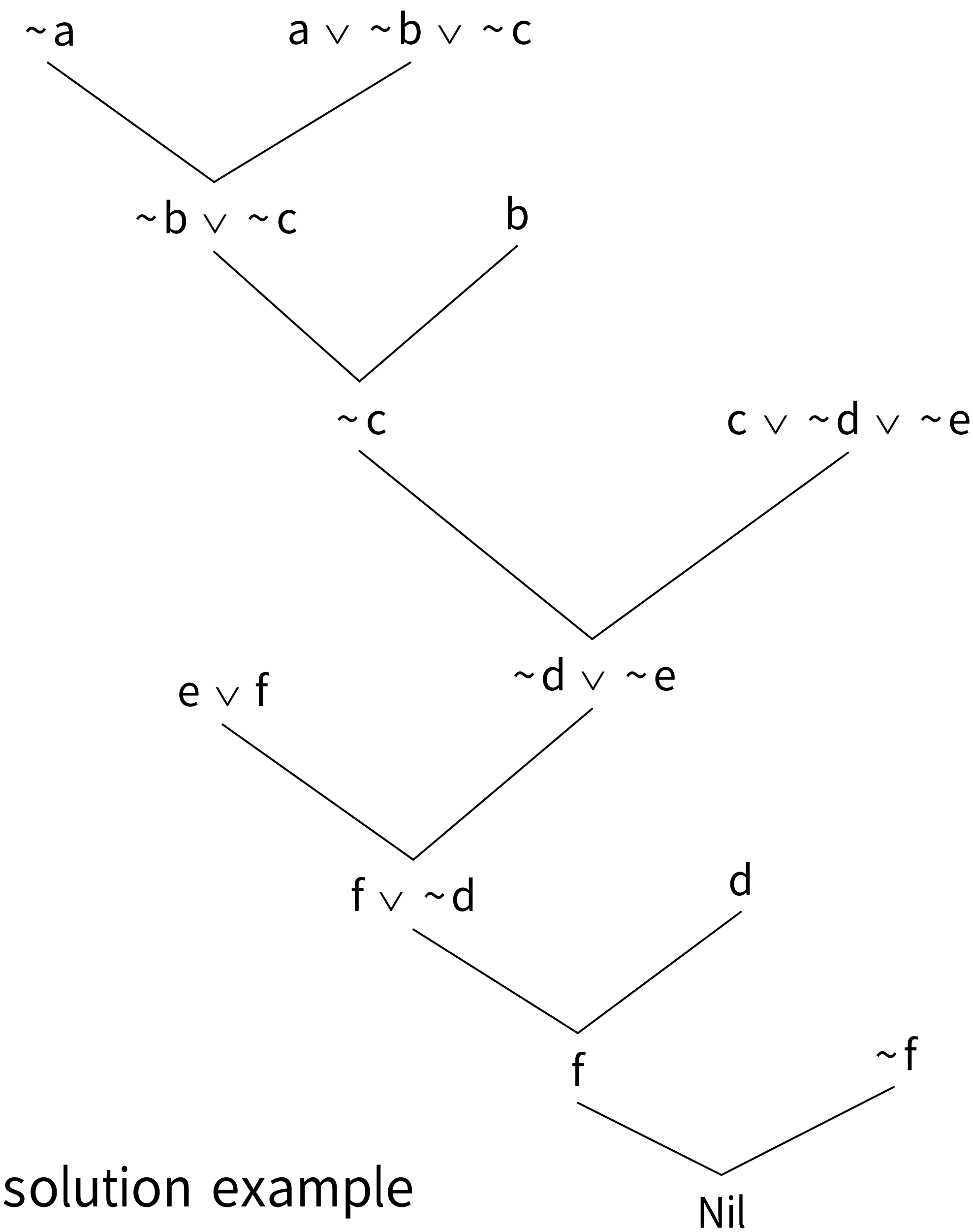


그림 3 : resolution example

- goal의 부정이 1개 이상의 clause로 구성되어 있는 경우
  - < Axioms >

    $\neg a(X) \lor f(X) \lor g(f(X))$

    $\neg f(X) \lor b(X)$

    $\neg f(X) \lor c(X)$

    $\neg g(X) \lor b(X)$

    $\neg g(X) \lor d(X)$

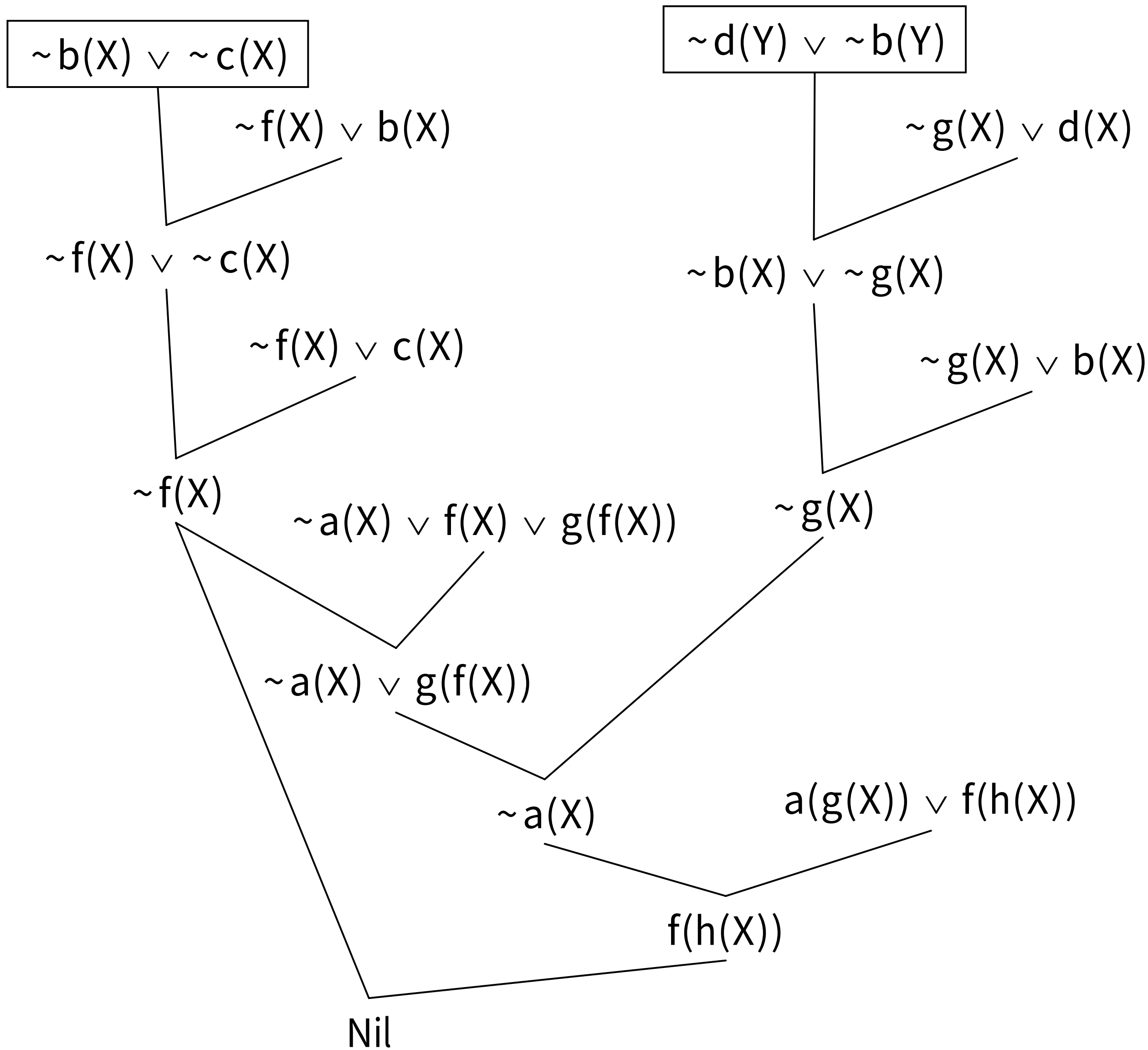
    $a(g(X)) \lor f(h(X))$
  - < goal >

    $(\exists X)(\exists Y)\{[b(X) \land c(X)] \lor [d(Y) \land b(Y)]\}$
  - < ¬goal >

    $\neg b(X) \lor \neg c(X)$

    $\neg b(Y) \lor \neg d(Y)$

Resolution refutation



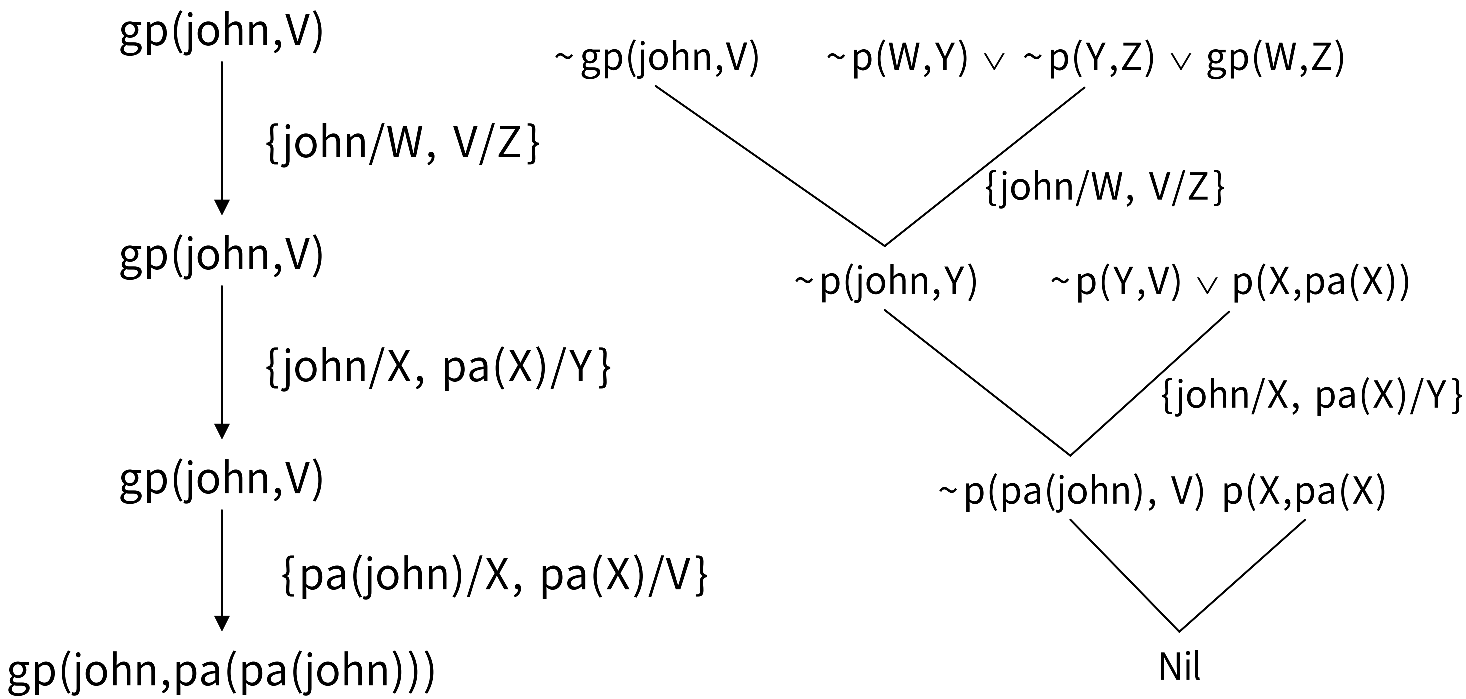
- example problem
  - Everyone has a parent.

    1. $(\forall X)(\exists Y)p(X,Y)$
  - A parent of a parent is a grandparent.

    2. $(\forall X)(\forall Y)(\forall Z)p(X,Y) \wedge p(Y,Z) \Rightarrow gp(X,Z)$
  - goal ; John has a grandparent?

    $(\exists W)(gp(john,W))$
  - negation of goal

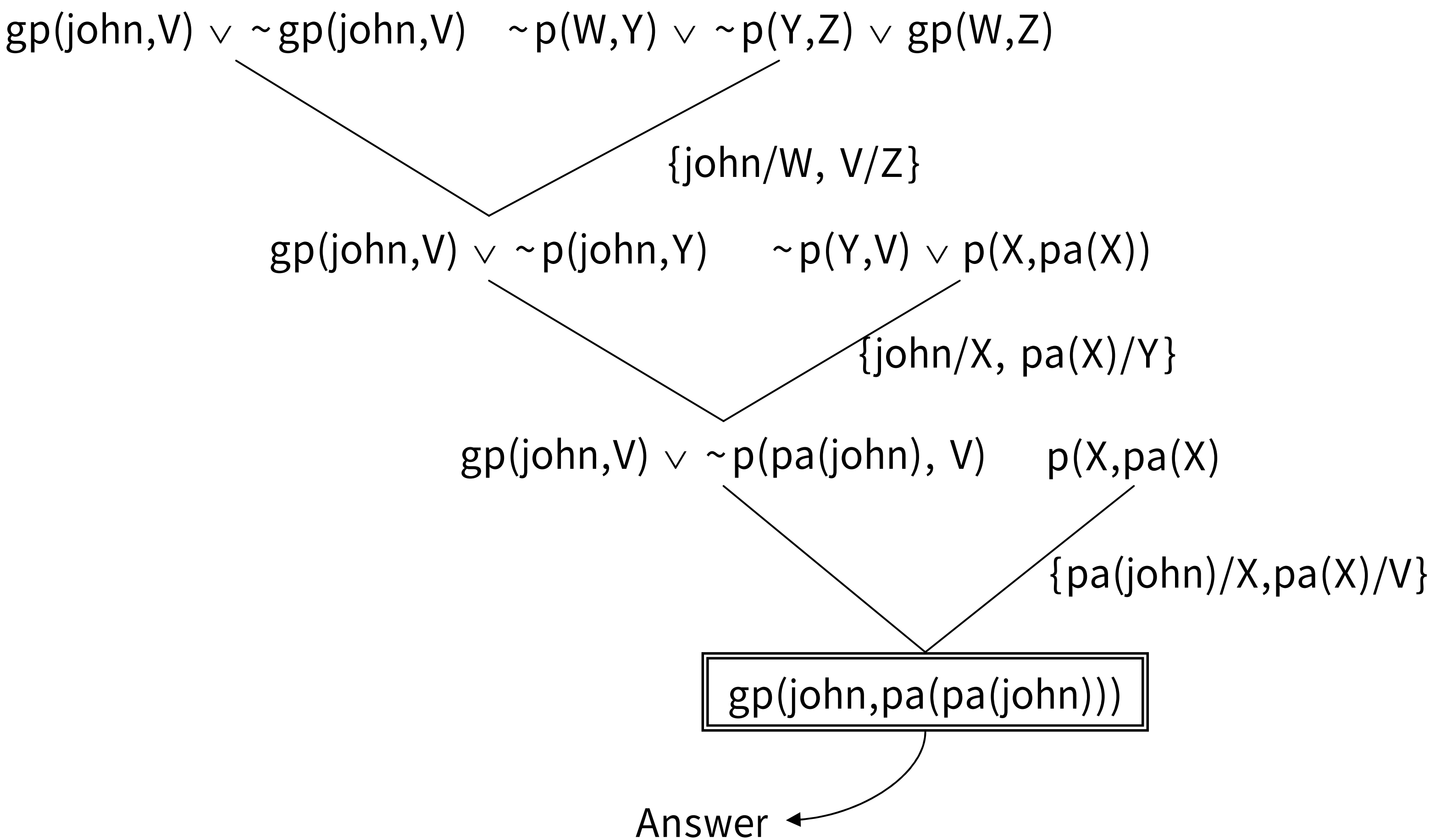
    3. $\neg gp(john,W)$
- Clause form
  1. $p(X,pa(X))$
  2. $\neg p(W,Y) \vee \neg p(Y,Z) \vee gp(W,Z)$
  3. $\neg gp(john,V)$



$V \rightarrow pa(X) \rightarrow pa(pa(john))$

* gp(john, pa(pa(john))) is proved

- tautology를 이용한 proof & answer extraction
  - goal statement의 negation을 tautology형태로 만든다.
    e.g.) $\neg gp(john,V) \rightarrow \neg gp(john,V) \vee gp(john,V)$



Resolution의 간소화 방법

- ?tolology의 제거
  - $p(X) \vee \neg p(X)$
- subsume clause의 제거
  - p(john) subsumes $\forall X\ p(X)$
  - $p(X)$ subsumes $p(X) \vee q(X)$
- procedure attachment

  T나 F로 evaluate 될 수 있는 literal은 evaluate하여
  - 임의의 점에서 하나의 literal이 T로 evaluate되면 그 절은 제거 가능하다.
  - 임의의 점에서 하나의 literal이 F로 evaluate되면 그 literal은 제거 가능하다.

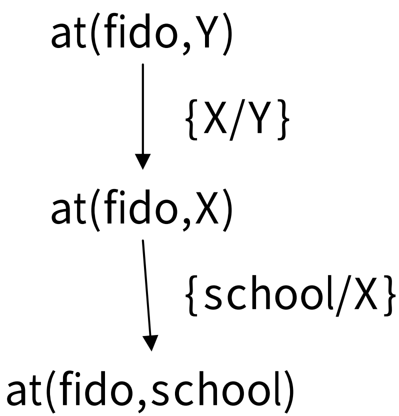
"If Fido goes where John goes & if John is at school, where is Fido?"

1. $(\forall X)[at(john,X) \Rightarrow at(fido,X)]$
2. at(john,school)
3. $(\exists X)at(fido,X)$? $\rightarrow$ goal

1; $\neg at(john,X) \vee at(fido,X)$

2; at(john,school)

$\neg$3; $\neg at(fido,X)$



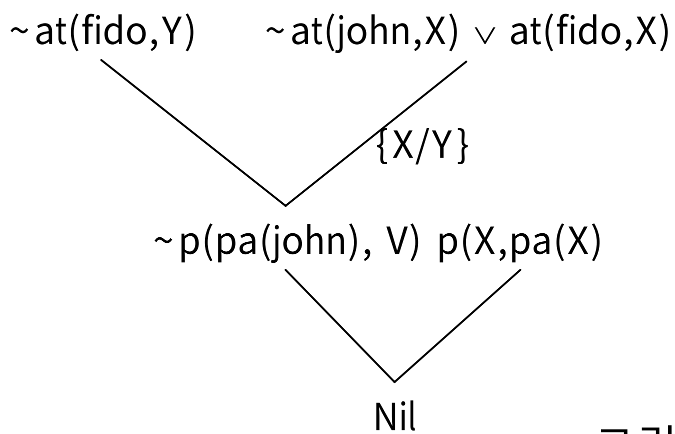


그림 4 : resolution example

•retain original goal & apply all the substitutions of the refutation to this clause.

we find the answer.

therefore at(fido,school)

fido는 school에 있다.

- Anyone passing his history exams and winning the lottery is happy

  $\forall X(pass(X, history) \wedge win(X, lottery) \rightarrow happy(X))$
- Anyone who studies or is lucky can pass all his exams

  $\forall X \forall Y(studies(X) \vee lucky(X) \rightarrow pass(X,Y))$
- John did not study but he is lucky

  $\neg study(john) \wedge lucky(john)$
- Anyone who is lucky wins the lottery

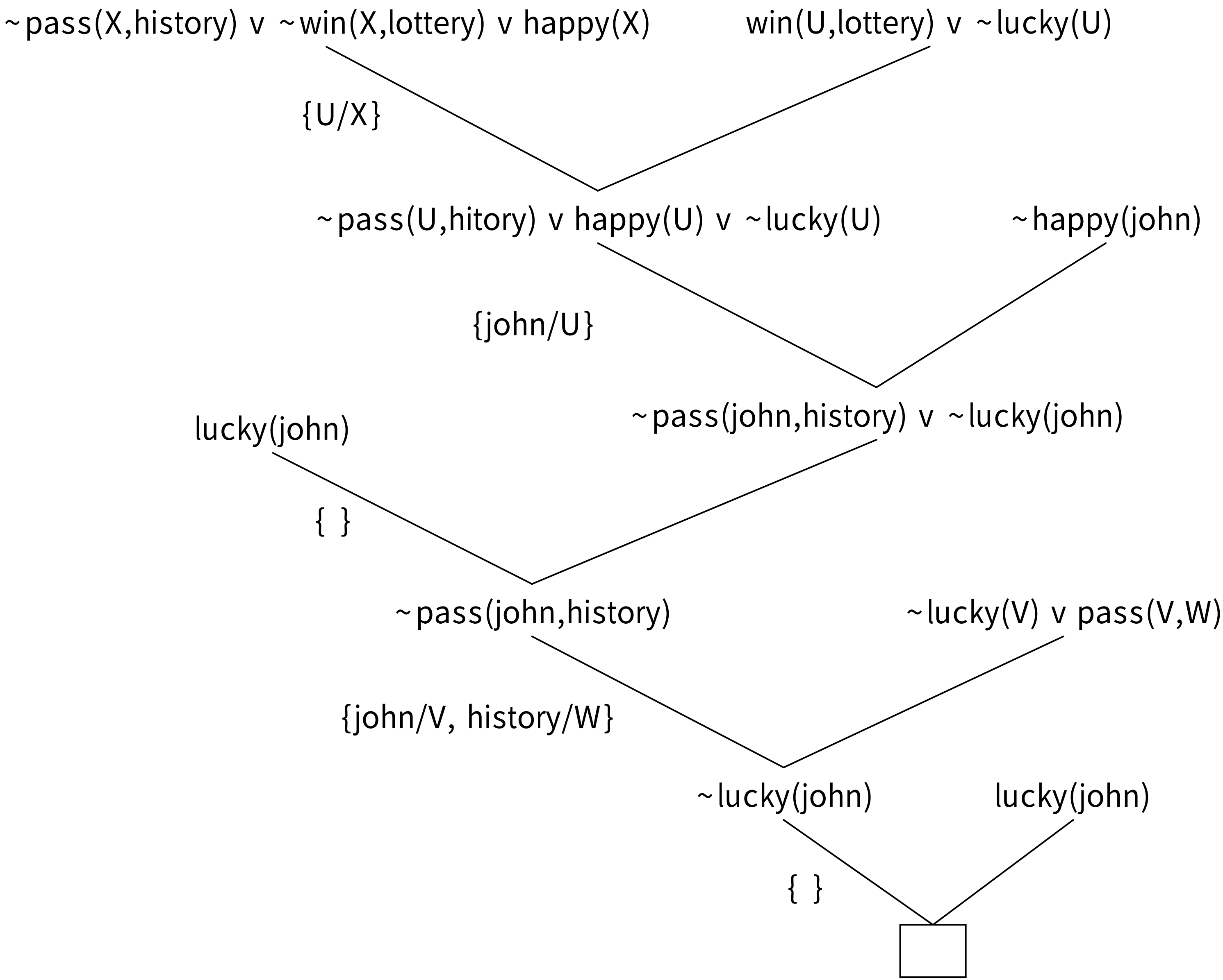
  $\forall X(lucky(X) \rightarrow win(X, lottery))$

  then
- Is John happy?

  happy(john)

< Clause form >

1. ¬pass(X,history) ∨ ¬win(X,lottery) ∨ happy(X)
2. ¬study(Y) ∨ pass(Y,Z)
   ¬lucky(W) ∨ pass(W,V)
3. ¬study(john)
   lucky(john)
4. ¬lucky(U) ∨ win(U,lottery)
5. ¬happy(john) ?


~pass(X,history) v ~win(X,lottery) v happy(X)
win(U,lottery) v ~lucky(U)
{U/X}
~pass(U,hitory) v happy(U) v ~lucky(U)
~happy(john)
{john/U}
lucky(john)
~pass(john,history) v ~lucky(john)
{ }
~pass(john,history)
~lucky(V) v pass(V,W)
{john/V, history/W}
~lucky(john)
lucky(john)
{ }

- Resolution strategy
  1. Breadth-First strategy

     N clause in the original clause set.

     1^{st} level : N^2 ways of combination

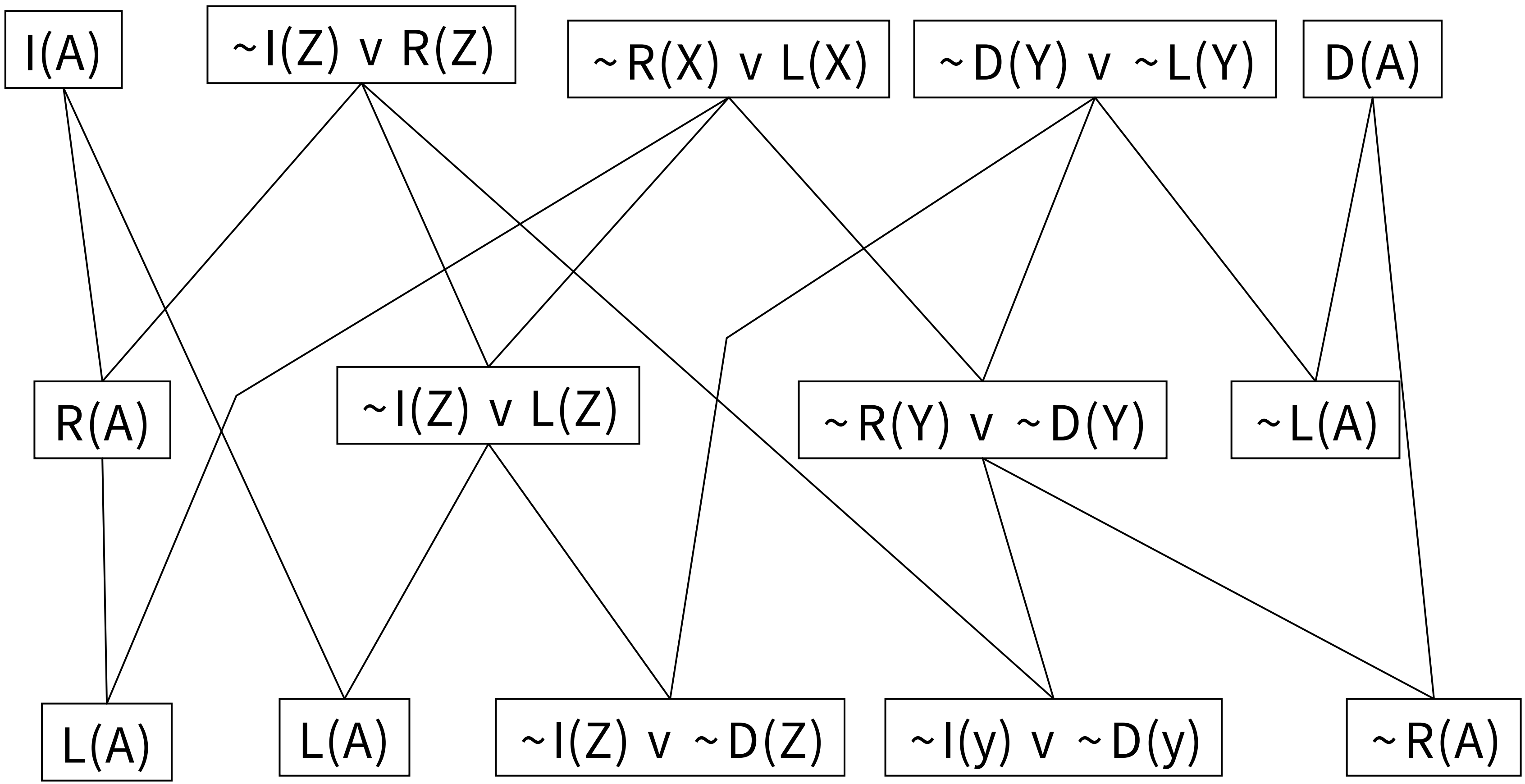
     2^{nd} level : Resolve the clauses produced at the 1^{st} level with all the original clauses.

     nth level : Resolve all clauses at the level n-1 against the elements of the original clause set & all causes previously produced.

     * large search space

     * find the shortest path solution

     * if refutation exist, if always finds. → complete.



< Breadth-First strategy >

1. The set of support strategy

Suppose a set of input clauses : S

subset of S that includes the negation of the goal : T

S is contradictory $\leftrightarrow$ T is contradictory

* The negation of what we want to prove true is responsible for causing the clause space to be contradictory

* Resolution 중에 적어도 parent의 하나를 goal의 부정 또는 이의 descendent 중에서 선택한다.

~I(Z) v R(Z)   I(A)   ~R(X) v L(X)   ~D(Y) v ~L(Y)

R(A)   ~I(Z) v L(Z)

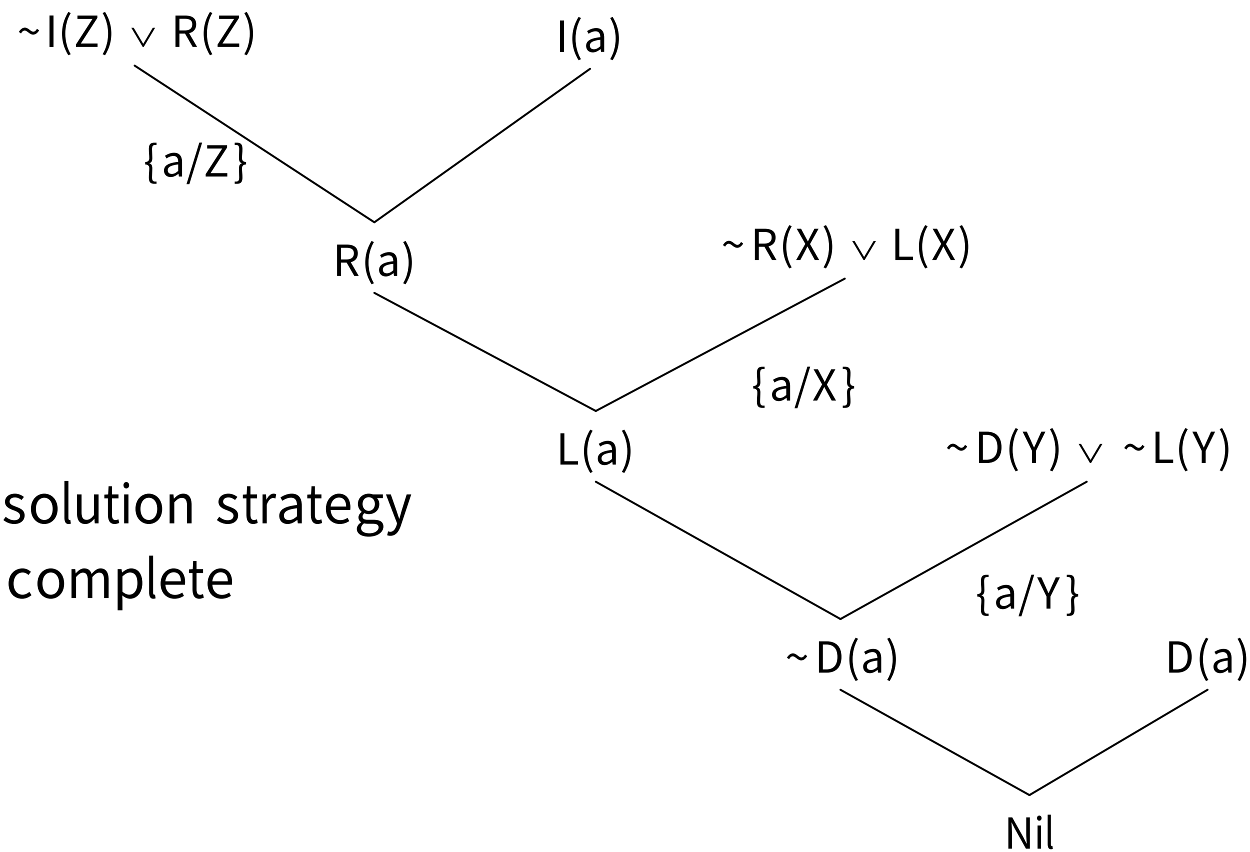
L(A)

~D(A)   ~I(A)   ~D(A)

- Unit Preference strategy(단위 우선 방법)
  - literal의 수가 적은 clause를 resolution에 사용할수록 Nil에 이를 가능성이 많음.

  → Resolution시 하나의 literal로 된 절을 선택하여 수행한다.

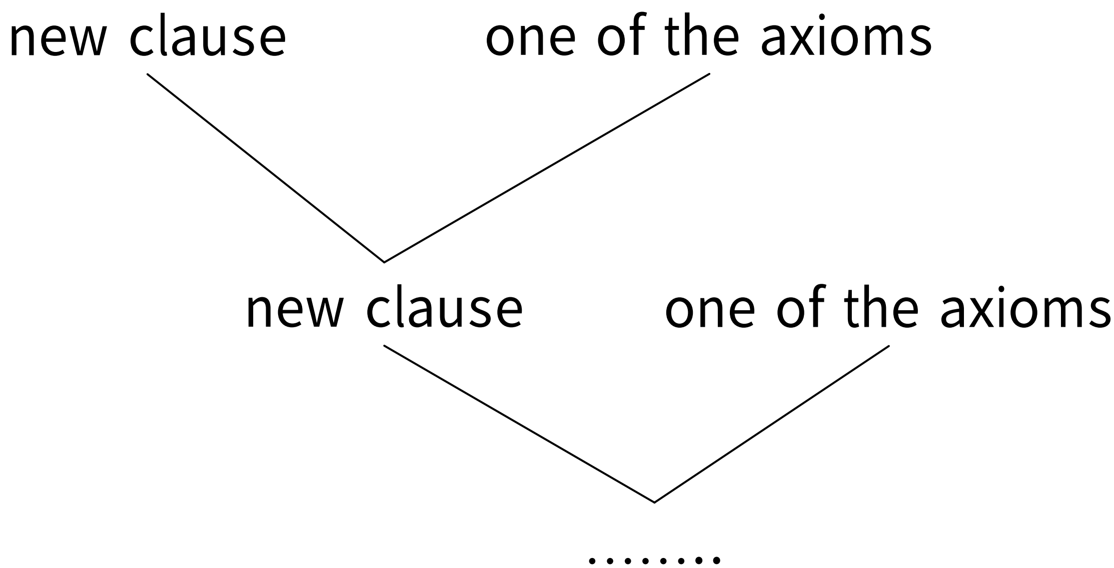
  → complete



Unit resolution strategy

→ not complete

- Linear Input Form strategy
  - negated goal & the original axioms
  - take the negated goal & resolve it with one of the axioms.



- No previously derived clause is used.
- No two axioms are used.
- not complete