
INSTITUT TEKNOLOGI BANDUNG
1920

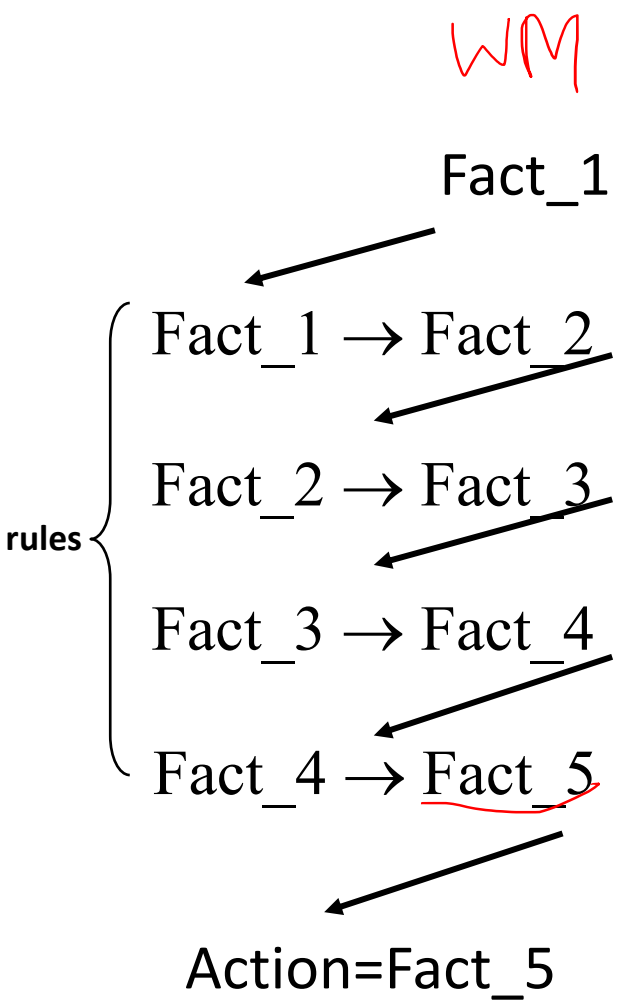
Modul 6: Rule-based System

Masayu Leylia Khodra
(masayu@informatika.org)

KK IF – Teknik Informatika – STEI ITB

03 Backward Chaining

Inteligensi Buatan
(*Artificial Intelligence*)



Forward Chaining

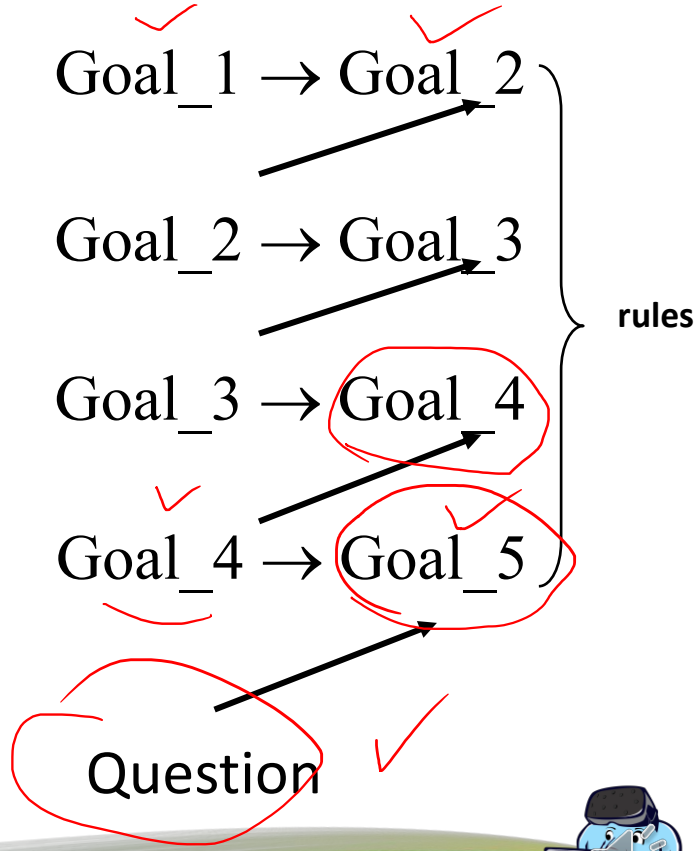
Backward Chaining

Match LHS

Match RHS

Data-driven reasoning

Goal-driven reasoning



Forward Chaining: Is Z true ?

Rule-base:

R1: Y, D → Z

R2: X, B, E → Y

R3: A → X

R4: C → L

R5: L, M → N

Facts:

A,B,C,D,E

Conflict resolution strategy:
refractoriness > fact recency > specificity > rule order

Iteration	Conflict set	Selected Rule	Working memory
1	{R3, R4}	R4	+ L
2	{R3, R4}	R3	+ X
3	{R2, R3, R4}	R2	+ Y
4	{R1, R2, R3, R4}	R1	+Z
5	{R1, R2, R3, R4}	-	stop

Answer: Yes

cek dulu sampai ga ada yg bisa dipilih baru stop, jgn langsung walau udah dapat z

Backward Chaining: Is Z true ?

Rule-base:

R1: Y, D → Z

R2: X, B, E → Y

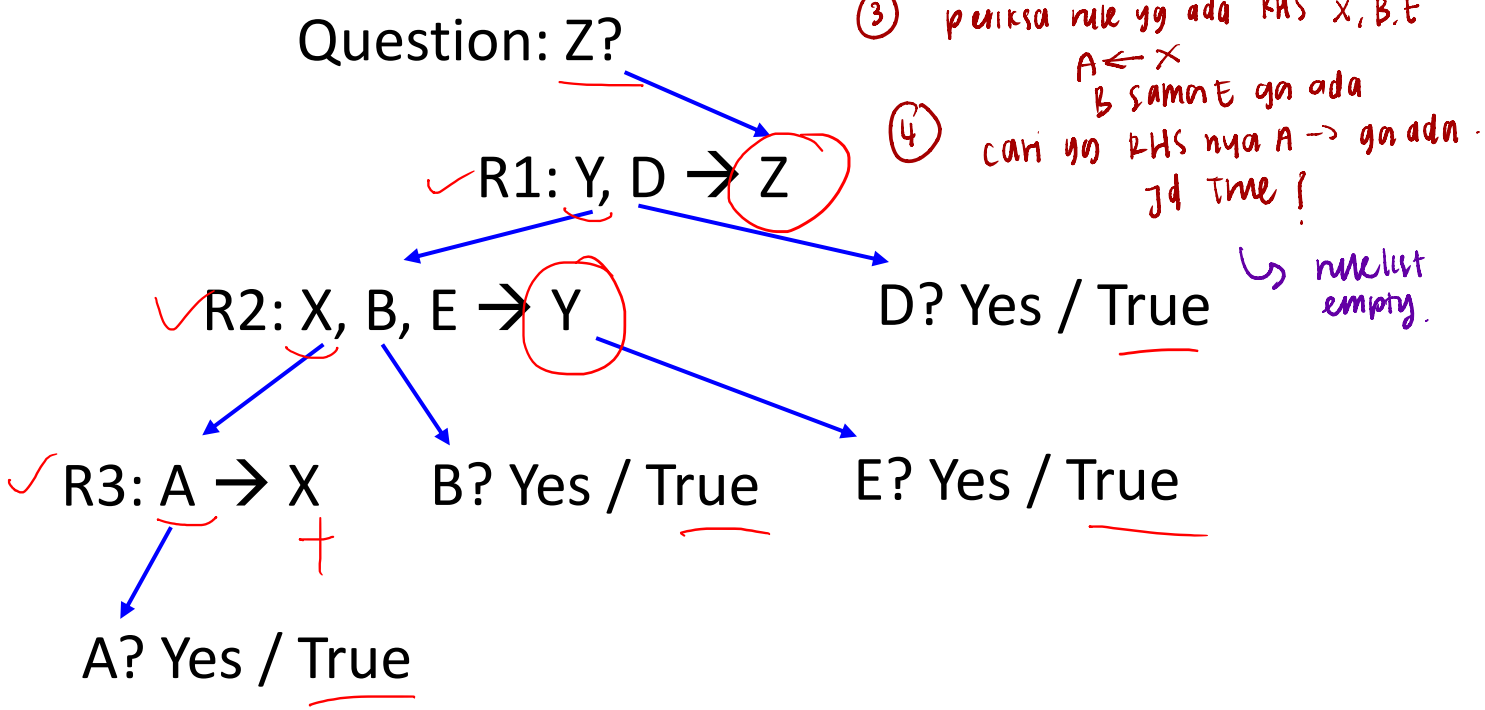
R3: A → X

R4: C → L

R5: L, M → N

Facts:

A,B,C,D,E

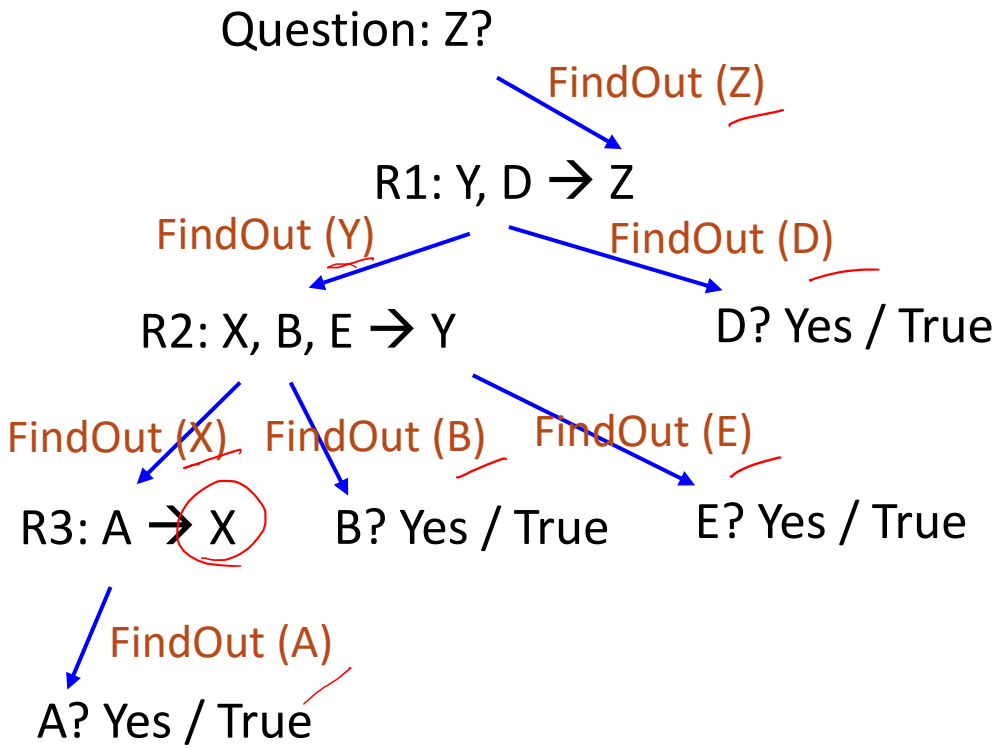


Answer: Yes

1. cari rule yg sisi kanannya Z
2. cari rule yg RHS nya mengandung Y atau D.
   * gd ada rule yg RHS nya D, tapi masih bisa lanjut karena gd ada yg nilainya false.
3. periksa rule yg ada RHS X, B, E
   A ← X
   B sama E gd ada
4. cari yg RHS nya A → gd ada.
   Jd True !

↳ rulelist empty.

Interpreter: FindOut (Goal)



Procedure FINDOUT (GOAL)

If (GOAL can be inferred)

then

set RULE-LIST = list all rules whose action part fulfills GOAL

until (RULE-LIST = empty) or (GOAL inferred) do

MONITOR(first or next rule from RULE-LIST)

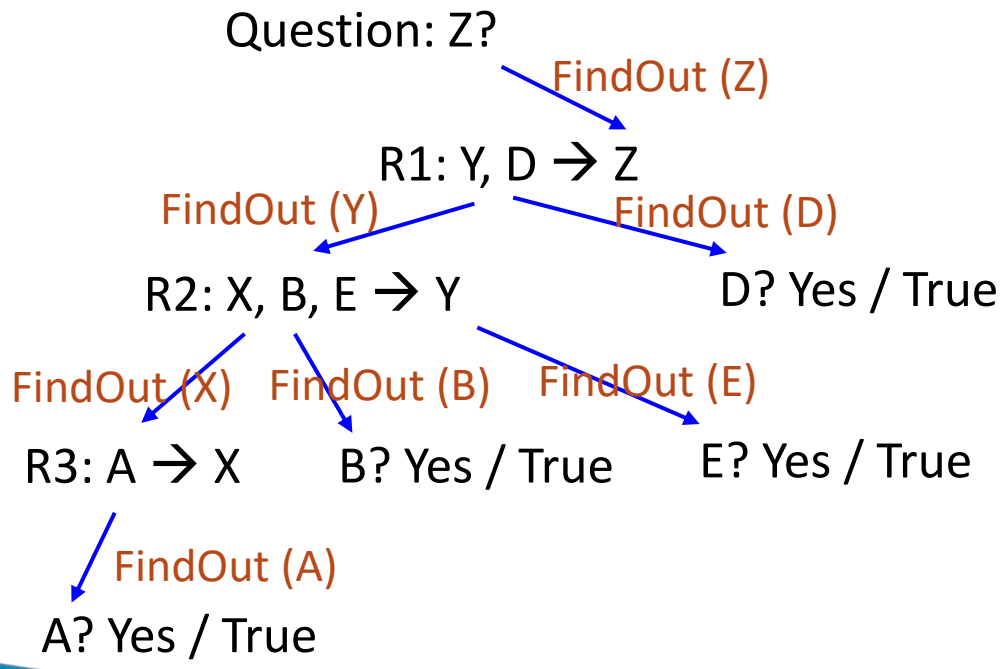
delete this rule from RULE-LIST

else (request GOAL)

Puppe, F. (1993). *Systematic Introduction to Expert Systems* . Springer, Berlin, Heidelberg.

Backward Chaining Process

Rule-base:
R1: Y, D → Z
R2: X, B, E → Y
R3: A → X
R4: C → L
R5: L, M → N

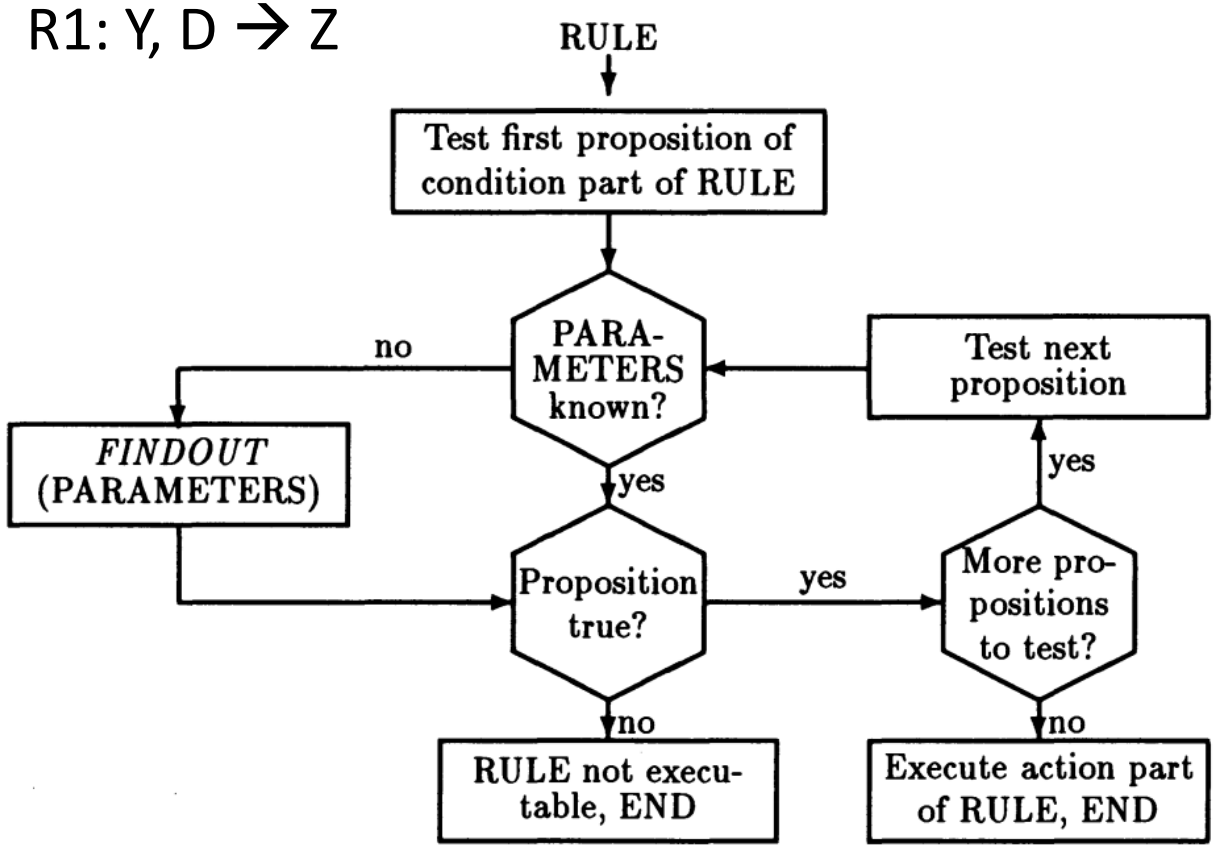


FindOut (Z) {R1}
Monitor(R1)

Procedure FINDOUT (GOAL)
If (GOAL can be inferred)
then
set RULE-LIST = list all rules whose action part fulfills GOAL
until (RULE-LIST = empty) or (GOAL inferred) do
MONITOR(first or next rule from RULE-LIST)
delete this rule from RULE-LIST
else (request GOAL)

Interpreter: Monitor (Rule)

R1: Y, D → Z



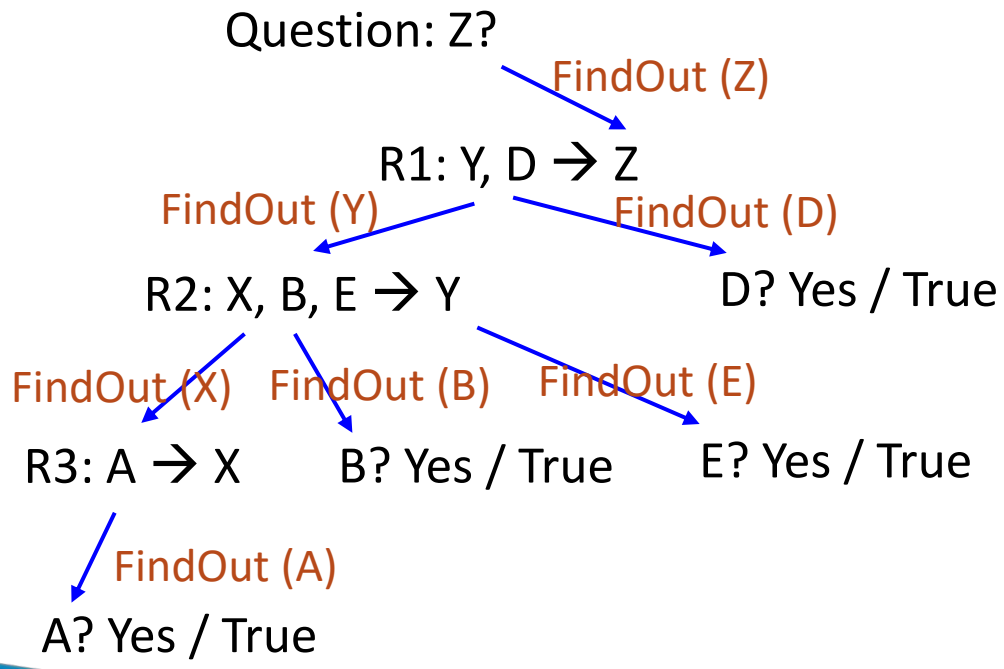
Procedure MONITOR (RULE)
Test first proposition of condition part of RULE
repeat
    If parameters known then
      if proposition true then
            proposition ← next proposition
      else RULE not executable
    else FINDOUT(PARAMETERS)
Until (no more propositions to test) or (RULE not executable)
If (no more propositions to test) then
    execute action part of RULE

Puppe, F. (1993). *Systematic Introduction to Expert Systems* . Springer, Berlin, Heidelberg.

Backward Chaining Process

Rule-base:
R1: Y, D → Z
R2: X, B, E → Y
R3: A → X
R4: C → L
R5: L, M → N

Facts:
A,B,C, E



FindOut (Z)	{R1}
Monitor(R1)	{Y,D}
FindOut(Y)	{R2}
Monitor(R2)	{X,B,E}
FindOut(X)	{R3}
Monitor(R3)	{A}
FindOut(A)	True
Execute(R3)	+X
Delete(R3)	
FindOut(B)	True
FindOut(E)	True
Execute(R2)	+Y
Delete (R2)	
FindOut(D)	request(D): True
Execute(R1)	+Z
Delete(R1)	

Answer: Yes

Inference pakai backward chaining.
↳ cari rule yg RHS nya action.

What action to take to get to a theatre

↳ di findoutnya dari rule yg mulai/muncul nya lebih awal

Inference using Backward Chaining to decide what action to take to get to a theatre. Start the process by FindOut(Action) until first action is inferred, and working memory is empty.

R	IF	THEN
1	Distance > 5 miles	Means is "~~drive~~ walk"
2	Distance > 1 mile, time < 15 minutes	Means is "drive"
3	Distance > 1 mile, time > 15 minutes	Means is "walk"
4	Means is "drive", location is "downtown"	Action is "take a cab"
5	Means is "drive", location is not "downtown"	Action is "drive your car"
6	Means is "walk", weather is "bad"	Action is "take a coat and walk"
7	Means is "walk", weather is "good"	Action is "walk"

Request facts:
Distance is about 6 miles;
Weather is "bad";
Location is downtown;
Time is about 20 minutes

What action to take to get to a theatre

Inference using Backward Chaining to decide what action to take to get to a theatre. Working memory is empty. Start the process by FindOut(Action) until first action is inferred.

R	IF	THEN
1	Distance > 5 miles	Means is "drive"
2	Distance > 1 mile, time < 15 minutes	Means is "drive"
3	Distance > 1 mile, time > 15 minutes	Means is "walk"
4	Means is "drive", location is "downtown"	Action is "take a cab"
5	Means is "drive", location is not "downtown"	Action is "drive your car"
6	Means is "walk", weather is "bad"	Action is "take a coat and walk"
7	Means is "walk", weather is "good"	Action is "walk"

Request facts:
Distance is about 6 miles;
Weather is "bad";
Location is downtown;
Time is about 20 minutes

Procedure MONITOR (RULE)
Test first proposition of condition part of RULE
repeat
  If parameters known then
    if proposition true then
      proposition ← next proposition
    else RULE not executable
  else FINDOUT(PARAMETERS)
Until (no more propositions to test) or (RULE not executable)
If (no more propositions to test) then
execute action part of RULE

kayaknya ini slidenya salah

R	IF	THEN
1	Distance > 5 miles	Means is "walk"
2	Distance > 1 mile, time < 15 minutes	Means is "drive"
3	Distance > 1 mile, time > 15 minutes	Means is "walk"
4	Means is "drive", location is "downtown"	Action is "take a cab"
5	Means is "drive", location is not "downtown"	Action is "drive your car"
6	Means is "walk", weather is "bad"	Action is "take a coat and walk"
7	Means is "walk", weather is "good"	Action is "walk"

```
FindOut(Action)                 {R4,R5,R6,R7}
    Monitor(R4)                 {means is drive ? F,
        FindOut(means)          {R1,R2,R3}
            Monitor(R1)         {distance > 5 miles ?
                FindOut(distance)    request(distance): 6 miles
                Execute(R1)          + means is walk
            Delete(R1)
        R4 not executable
    Delete(R4)
    Monitor(R5)
Stop.
```

```
Procedure FINDOUT (GOAL)
If (GOAL can be inferred)
then
    set RULE-LIST = list all rules whose action part fulfills GOAL
    until (RULE-LIST = empty) or (GOAL inferred) do
        MONITOR(first or next rule from RULE-LIST)
        delete this rule from RULE-LIST
else (request GOAL)
```

```
Procedure MONITOR (RULE)
Test first proposition of condition part of RULE
repeat
    If parameters known then
        If proposition true then
            proposition ← next proposition
        else RULE not executable
    else FINDOUT(PARAMETERS)
Until (no more propositions to test) or (RULE not executable)
If (no more propositions to test) then
    execute action part of RULE
```

kan means is drive, kok rule yg masuk ga R2 aja?
krn yg masuk itu parameternya => means
drive ini cuman value doang.
maksanya semua yg masuk itu semua yg mengandung.
means => maksanya R1, R2, R3 masuk.

R	IF	THEN
1	Distance > 5 miles	Means is "drive"
2	Distance > 1 mile, time < 15 minutes	Means is "drive"
3	Distance > 1 mile, time > 15 minutes	Means is "walk"
4	Means is "drive", location is "downtown"	Action is "take a cab"
5	Means is "drive", location is not "downtown"	Action is "drive your car"
6	Means is "walk", weather is "bad"	Action is "take a coat and walk"
7	Means is "walk", weather is "good"	Action is "walk"

Inference using Backward Chaining to decide what action to take to get to a theatre. Start the process by FindOut(Action) until first action is inferred, and working memory is empty.

Request facts:
Distance is about 6 miles;
Weather is "bad";
Location is downtown;
Time is about 20 minutes

```
findout(action)                          {R4,R5,R6,R7}
Monitor(R4)                              {means=drive,
                                         location=downtown}
  findout(means)                         {R1, R2, R3}
    monitor(R1)                          {distance > 5 miles}
      FindOut(distance)                  request(distance): 6 miles
      execute(R1)                        + means=drive
    delete(R1)
    stop
  findout(location)                      request(location): downtown
  execute(R4)                            + action=take a cab
delete(R4)
stop
```

R7 gpp gausah di cek soalnya goalnya sudah berhasil di inference. (jd ga kosong rule-list isoke).

Answer: action=take a cab

berhasil menginferensi action.

mau ga mau utk soal ini semua rules harus di cek.

Latihan

Basis pengetahuan dari sistem yang menentukan *resort* bagi *skier*:

R1: if Rating = beginner, Purpose = fun then Resort = St.Sartre
R2: if Rating = beginner, Purpose = serious then Resort = Schloss Heidegger
R3: if Rating = advanced, Purpose = serious then Resort = Chateau Derrida
R4: if Rating = advanced, Purpose = fun then Resort = Wittgenstein Gladbach
R5: if Lessons < 30 hours then Rating = beginner
R6: if Lessons >= 30 hours, Fitness = poor then Rating = beginner
R7: if Lessons >= 30 hours, Fitness = good then Rating = advanced
R8: if Pressups < 10 then Fitness = poor
R9: if Pressups >= 10 then Fitness = good

BC: WM kosong, jawaban saat request: purpose = fun, lesson = 178, pressups = 15

↳ goal

Backward Chaining

R1: If Rating = beginner, Purpose = fun then Resort = St.Sorry
R2: If Rating = beginner, Purpose = serious then Resort = Schloss Hohenwerfen
R3: If Rating = advanced, Purpose = serious then Resort = Chateau Dornida
R4: If Rating = advanced, Purpose = fun then Resort = Wittgenstein Gladbach
R5: If Lessons < 30 hours then Rating = beginner
R6: If Lessons >= 30 hours, Fitness = poor then Rating = beginner
R7: If Lessons >= 30 hours, Fitness = good then Rating = advanced
R8: If Pressups < 10 then Fitness = poor
R9: If Pressups >= 10 then Fitness = good

FindOut(Resort)	{R1,R2,R3,R4}
Monitor(R1)	Rating = beginner, Purpose = fun
FindOut(rating)	{R5,R6,R7}
Monitor(R5)	Lessons < 30 hours
FindOut(lessons)	request(lesson): 178
R5 not executable	
Delete R5	
Monitor(R6)	Lessons >= 30 hours, Fitness = poor
FindOut(fitness)	{R8,R9}
Monitor(R8)	Pressups < 10
FindOut(pressups)	request(pressups): 15
R8 not executable	
Delete R8	
Monitor(R9)	Pressups >= 10
Execute R9	+ Fitness=good
Delete R9	
R6 not executable	
Delete R6	
Monitor(R7)	Lessons >= 30 hours, Fitness = good
Execute R7	+ Rating=advanced
Delete R7	
R1 not executable	
Delete R1	
Monitor(R2)	Rating = beginner, Purpose = serious
R2 not executable	
Delete R2	
Monitor (R3)	Rating = advanced, Purpose = serious
FindOut(purpose)	request(purpose): fun
R3 not executable	
Delete R3	
Monitor(R4)	Rating = advanced, Purpose = fun
Execute R4	+ Resort=Wittgenstein Gladbach
Delete R4	
Terminate	

Fakta pada WM
purpose = fun,
lesson = 178,
pressups = 15

Rule-based System Features

Modularity

- Each rule defines a small, relatively independent piece of knowledge

Incrementability

- New rules can be added to the knowledge base relatively independently of other rules

Modifiability

- Old rules can be changed relatively independently of other rules

Support systems transparency

Summary

Forward vs Backward Chaining

Goal-driven reasoning; Match RHS

FindOut(Goal) & Monitor(Rule)

Referensi

1. Frank Puppe, Systematic Introduction to Expert Systems: Knowledge Representations and Problem-Solving Methods, Springer, 1st ed. 1993
2. Peter Jackson, Introduction To Expert Systems, Addison-Wesley 3rd Edition, 1999,


INSTITUT TEKNOLOGI BANDUNG
1920

RBS

- R1: If temperature<37 then no fever
- R2: If temperature>37 and temperature<38 then low fever
- R3: If temperature>38 then high fever
- R4: If light nasal breathing then nasal discharge
- R5: If heavy nasal breathing then sinus membranes swelling
- R6: If low fever and headache and nasal discharge and cough then cold
- R7: If cold and not(soar throat) then don't treat
- R8: If cold and soar throat then treat
- R9: If don't treat then action: don't give medication
- R10: If treat then give medication
- R11: If give medication and antibiotics allergy then action: give Tylenol
- R12: If give medication and not(antibiotics allergy) then action: give antibiotics

Aksi apa yang harus dilakukan?

WM kosong, ketika request berikut fakta yang dimasukkan:

Fakta: headache; light nasal breathing; temperature=37,5; cough;
not(antibiotics allergy); soar throat

findout(Action) R9, R11, R12

→ lihat yg RHSnya don't treat R7, R8.

→ lihat yg RHSnya cold R6

→ RHSnya fever R1, R2, R3

monitor(R1) false delete R1

monitor(R2) cocok, masuk ke R2.

→ R6 udah true yg fever, lanjut cek headache

headache ga ada, lanjut check nasal R4

R6 → true.

R7 → salah karena not(sore throat) salah

jadi lanjut cek R8.