

PERTEMUAN 14

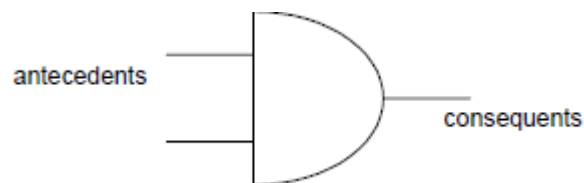
KNOWLEDGE-BASED EXPERT SYSTEMS

Knowledge-Based Expert Systems (KBS) pengertiannya serupa dengan *Expert Systems* (ES), yaitu program pemberi advis/nasehat yang terkomputerisasi yang ditujukan untuk menirukan atau menggantikan proses *reasoning* dan pengetahuan (*knowledge*) dari para pakar dalam menyelesaikan permasalahan masalah yang spesifik.

Adapun karakteristik KBS, antara lain:

1. Dapat belajar dari pengalaman.
2. Mentransfer knowledge dari satu domain ke domain yang lain.
3. Dapat memberikan proses reasoning dalam pelbagai level.
4. Menggunakan tool-tool: heuristics (rule of thumb), mathematical models, simulations.

9.1 Rules dan Rule Chaining



IF (antecedents) THEN (consequents)

Contoh: Mengidentifikasi binatang.

Sebuah robot dapat mempersepsikan fitur-fitur dasar: color, size, has hair, gives milk.

Tetapi, kemampuannya mengidentifikasi obyek berdasarkan fitur-fitur tersebut diatas adalah

terbatas. Dia dapat membedakan binatang-binatang dari obyek yang lain, tetapi ia tak dapat menggunakan fakta bahwa sebagian binatang yang memiliki leher panjang disebut dengan jerapah.

Rule-nya:

1. IF ?x has hair

THEN ?x is a mammal

2. IF ?x gives milk

THEN ?x is a mammal

3. IF ?x has feathers

THEN ?x is a bird

4. IF ?x flies

?x lays eggs

THEN ?x is a bird

5. IF ?x is a mammal

?x eats meat

THEN ?x is a carnivore

6. IF ?x is a mammal

?x has pointed teeth

?x has claws

?x has forward-pointing eyes

THEN ?x is a carnivore

7. IF ?x is a mammal

?x has hoofs

THEN ?x is an ungulate

8. *IF ?x is a mammal*

?x chews end

THEN ?x is an ungulate

9. *IF ?x is a carnivore*

?x has tawny color

?x has dark spots

THEN ?x is a cheetah

10. *IF ?x is a carnivore*

?x has tawny color

?x has black strips

THEN ?x is a tiger

11. *IF ?x is a ungulate*

?x has long neck

?x has long legs

?x has tawny color

?x has dark spots

THEN ?x is a giraffe

12. *IF ?x is a ungulate*

?x has white color

?x has black strips

THEN ?x is a zebra

13. *IF ?x is a bird*

?x does not fly

?x has long legs

?x has long neck

?x has black and white

THEN ?x is a ostrich

14. IF ?x is a bird

?x does not fly

?x swims

?x has black and white

THEN ?x is a penguin

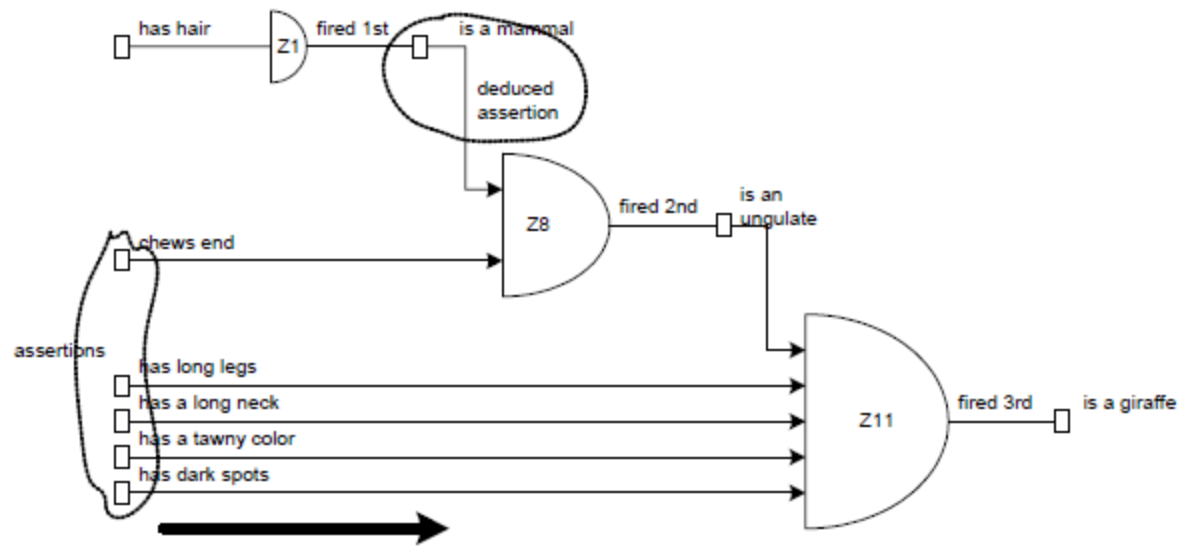
15. IF ?x is a bird

?x is a good flyer

THEN ?x is an albatross

9.2 Forward dan Backward chaining

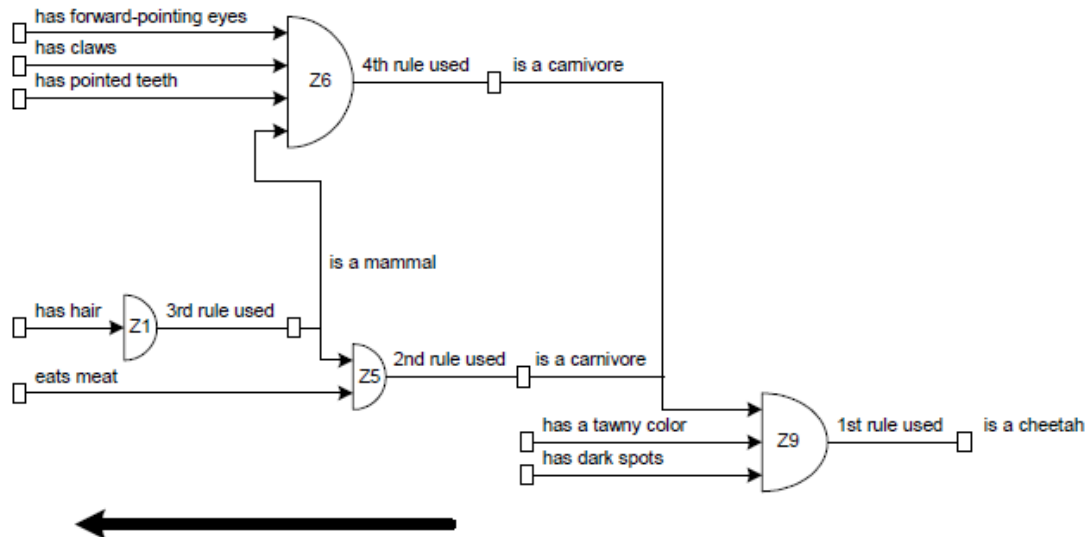
9.2.1 Forward chaining



Identify an animal

- *Until no rule produces a new assertion or the animal is identified*
- *For each rule:*
 - ✓ *Try to support each of the rule's antecedents by matching it to known facts.*
 - ✓ *If all the rule's antecedents are supported, assert the consequent rules there is an identical assertion already.*
 - ✓ *Report for all matching and instantiation alternatives*

9.2.2 Backward chaining



➤ Repeat until all hypothesis have been tried and none have been supported or until the animal is identified.

- ✓ For each hypothesis,
- ✓ For each rule whose consequent matches the current hypothesis, Try to support each of the rule's antecedents by matching it to assertions.
- ✓ If all the rule's antecedents are supported, announce success And conclude that the hypothesis is true.

Note: The chaining ends unsuccessfully if any required antecedent assertions cannot be supported.

Bagaimanakah hubungan antara rule dengan fakta-faktanya, sehingga didapatkan konklusinya.

1. Jika masalah yang dihadapi lebih dekat ke fan out (sekumpulan fakta yang bisa menuju banyak konklusi), maka pilihlah *backward chaining*.

2. Jika masalah yang dihadapi lebih dekat ke fan in (sekumpulan hipotesis yang bisa menuju ke banyak pertanyaan), maka pilihlah *forward chaining*.
3. Banyak cara untuk mendapatkan sedikit konklusi, maka digunakanlah *forward chaining*.
4. Sedikit cara untuk mendapatkan banyak konklusi, maka digunakanlah *backward chaining*.
5. Jika kita belum mendapatkan berbagai fakta, dan kita tertarik hanya pada satu konklusi yang mungkin, maka digunakanlah *backward chaining*.
6. Jika kita benar-benar sudah mendapatkan pelbagai fakta, dan kita ingin untuk mendapatkan konklusi dari fakta-fakta itu, maka digunakanlah *forward chaining*.

Tipe ataupun karakteristik sistem yang dapat menggunakan metode *forward chaining* atau yang dapat dicari dengan *forward chaining*, antara lain:

1. Sistem yang dipresentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian IF.
3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian THEN. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

Tipe ataupun karakteristik sistem yang dapat menggunakan metode *backward chaining* atau yang dapat dicari dengan *backward chaining*, antara lain:

1. Sistem yang dipresentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap konklusi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan konklusi pada bagian THEN.

3. Setiap konklusi dihasilkan dari kondisi-kondisi yang terdapat pada bagian IF. Selanjutnya kondisi-kondisi tersebut menjadi konklusi baru yang dimasukkan ke stack di atas konklusi yang sudah ada.
4. Setiap konklusi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu konklusi, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berhenti.