

TEKNIK INFERENSI

Sistem Informasi

Definisi Inferensi

- ▶ Inferensi adalah : Proses yang digunakan dalam Sistem Pakar untuk menghasilkan informasi baru dari informasi yang telah diketahui
- ▶ Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut *Inference Engine* (Mesin inferensi)
- ▶ Ketika representasi pengetahuan (RP) pada bagian *knowledge base* telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan.
- ▶ Inference engine merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning*.



REASONING

- ▶ **Definisi** : Proses bekerja dengan pengetahuan, fakta dan strategi pemecahan masalah, untuk mengambil suatu kesimpulan. (Berpikir dan mengambil kesimpulan)
- ▶ **Metode Reasoning**
 - ▶ Deductive Reasoning
 - ▶ Inductive Reasoning
 - ▶ Abductive Reasoning
 - ▶ Analogical Reasoning
 - ▶ Common Sense Reasoning



Deductive Reasoning

- ▶ Kita menggunakan *reasoning* deduktif untuk mendeduksi informasi baru dari hubungan logika pada informasi yang telah diketahui.
 - ▶ **Contoh:**
 - ▶ Implikasi : Saya akan basah kuyup jika berdiri ditengah-tengah hujan deras
 - ▶ Aksioma : Saya berdiri ditengah-tengah hujan deras
 - ▶ Konklusi : Saya akan basah kuyup
- IF A is True AND IF A IMPLIES B is True, Then B is True***



Inductive Reasoning

- ▶ Kita menggunakan *reasoning* induktif untuk mengambil kesimpulan umum dari sejumlah fakta khusus tertentu.
- ▶ **Contoh:**
 - ▶ Premis : Monyet di Kebun Binatang Ragunan makan pisang
 - ▶ Premis : Monyet di Kebun Raya Bogor makan pisang
 - ▶ Konklusi : Semua monyet makan pisang



Abductive Reasoning

- ▶ Merupakan bentuk dari proses deduksi yang memungkinkan inferensi *plausible*.
 - ▶ *Plausible* berarti bahwa konklusi mungkin bisa mengikuti informasi yang tersedia, tetapi juga bisa salah.
 - ▶ Contoh:
 - ▶ Implikasi : Tanah menjadi basah jika terjadi hujan
 - ▶ Aksioma : Tanah menjadi basah
 - ▶ Konklusi : Apakah terjadi hujan?
- IF B is True AND A implies B is true, Then A is True?*



Analogical Reasoning

- ▶ Kita menggunakan pemodelan analogi untuk membantu kita memahami situasi baru atau objek baru.
- ▶ Kita menggambar analogi antara 2 objek/situasi, kemudian melihat persamaan dan perbedaan untuk memandu proses *reasoning*.



Common Sense Reasoning

- ▶ Melalui pengalaman, manusia belajar untk memecahkan masalahnya secara efisien.
- ▶ Dengan menggunakan *common sense* untuk secara cepat memperoleh suatu solusi.
- ▶ Dalam sistem pakar, dapat dikategorikan sebagai *Heuristic*.
- ▶ Proses ***heuristic search*** atau ***best first search*** digunakan pada aplikasi yang membutuhkan solusi yang cepat



Reasoning dengan Logika

▶ **Modus Ponens**

- ▶ Definisi: Rule dari logika yang menyatakan bahwa jika kita tahu A adalah benar dan A implies B adalah juga benar, maka kita dapat mengasumsikan bahwa B benar.

$[A \text{ AND } (A \rightarrow B)] \rightarrow B$

IF A is True

AND $A \rightarrow B$ is True

THEN B is True



Contoh:

- ▶ A = Udara Cerah
- ▶ B = Kita akan pergi ke pantai
- ▶ $A \rightarrow B$ = Jika udara cerah, maka kita pergi ke pantai
- ▶ Dengan menggunakan Modus Ponens, kita bisa menarik kesimpulan bahwa
“Kita akan pergi ke Pantai”



Resolusi

- ▶ Definisi: Strategi inferensi yang digunakan pada sistem logika untuk menentukan kebenaran dari suatu assertion (penegasan)
 - ▶ Metoda Resolusi mencoba untuk membuktikan bahwa beberapa teorema atau ekspresi sebagai proposisi P adalah TRUE, dengan memberikan sejumlah aksioma dari masalah tersebut.
 - ▶ *Proof by Refutation*, suatu teknik yang ingin membuktikan bahwa $\neg P$ tidak dapat menjadi TRUE.
 - ▶ *Resolvent* : ekspresi baru yang muncul dari metode resolusi yang merupakan gabungan (*union*) dari aksioma yang ada dengan teorema negasi.
-



Misalnya:

- ▶ Ada 2 aksioma:
 - ▶ $A \vee B$ (A is True OR B is True) dan $\neg B \vee C$ (B is Not True OR C is True).
 - ▶ $(A \vee B) \wedge (\neg B \vee C) = A \vee C$
- ▶ *Resolvent* tersebut kemudian ditambahkan pada list dari aksioma dan akan menghasilkan resolvent baru. Proses ini berulang sampai menghasilkan kontradiksi.



Nonresolusi

- ▶ Pada resolusi, tidak ada perbedaan antara *goals* (tujuan), *premises* maupun *rules*. Semua dianggap sebagai aksioma dan diproses dengan rule resolusi untuk inferensi.
- ▶ Cara tersebut dapat menyebabkan kebingungan karena menjadi tidak jelas apa yang ingin dibuktikan
- ▶ Teknik **Nonresolusi** atau ***natural-deduction*** mencoba mengatasi hal tersebut dengan menyediakan beberapa statement sebagai *goal*-nya
 - ▶ Untuk membuktikan $[H \wedge (A \rightarrow B) \rightarrow C]$:
 - ▶ If $(B \rightarrow C)$, then membuktikan $(H \rightarrow A)$



Perhatikan contoh kasus berikut:

- ▶ Untuk menjelaskan pendekatan ini, *Misalkan kita ingin membuktikan apakah Jack suka tim sepakbola Arema-Malang. Asumsinya adalah bahwa semua orang yang tinggal di Malang menyukai Arema. Karenanya, jika kita bisa mengetahui bahwa Jack tinggal di Malang, maka kita bisa membuktikan tujuan kita.*
- ▶ Kasus tadi dapat direpresentasi menjadi:
- ▶ **Antecedents:**
 $[\text{Lives-Malang}(\text{Jack}) \wedge (\text{Lives-Malang}(X) \rightarrow \text{Likes-Arema}(X))$
- ▶ **Goals:**
 $\rightarrow \text{Likes-Arema}(\text{Jack})$]:
- ▶ Pada contoh tersebut, kita bisa membuktikan *goal*-nya jika **$(\text{Lives-Malang}(X) \rightarrow \text{Likes-Arema}(X))$** , dan **$(\text{Lives-Malang}(\text{Jack}))$** adalah BENAR.

Beberapa hukum dalam Inferensi dengan Logika

- Hukum *Detachment*:
$$\begin{array}{l} p \rightarrow q \\ p \\ \text{maka, } q \end{array}$$
- Hukum Kontrapositif:
$$\begin{array}{l} p \rightarrow q \\ \text{maka, } \sim q \rightarrow \sim p \end{array}$$
- Chain rule (Hukum *Syllogism*)
$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \text{maka, } p \rightarrow r \end{array}$$
- Hukum Inferensi *Disjunctive*:
$$\begin{array}{ll} p \vee q & p \vee q \\ \sim p & \sim q \\ \text{maka, } q & p \end{array}$$
- Hukum Dobel Negasi:
$$\begin{array}{l} \sim(\sim p) \\ \text{maka, } p \end{array}$$



Beberapa hukum dalam Inferensi dengan Logika

Hukum DeMorgan:

$$\begin{array}{l} \sim(p \wedge q) \qquad \sim(p \vee q) \\ \text{maka, } \sim p \vee \sim q \qquad \sim p \wedge \sim q \end{array}$$

Hukum Simplifikasi:

$$\begin{array}{l} p \wedge q \qquad \sim(p \vee q) \\ \text{maka, } p \qquad q \end{array}$$

Hukum Konjungsi:

$$\begin{array}{l} p \\ q \\ \text{maka, } p \wedge q \end{array}$$

Hukum *Disjunctive Addition*:

$$\begin{array}{l} p \\ \text{maka, } p \vee q \end{array}$$

Hukum *Conjunctive Argument*:

$$\begin{array}{l} \sim(p \wedge q) \qquad \sim(p \wedge q) \\ p \qquad q \\ \text{maka, } \sim q \qquad \sim p \end{array}$$



INFERENCE DENGAN RULES: FORWARD dan BACKWARD CHAINING

- ▶ Inferensi dengan rules merupakan implementasi dari modus ponens, yang direfleksikan dalam mekanisme *search* (pencarian).
- ▶ **Firing a rule:** Bilamana semua hipotesis pada *rules* (bagian “IF”) memberikan pernyataan BENAR
- ▶ Dapat mengecek semua *rule* pada *knowledge base* dalam arah *forward* maupun *backward*
- ▶ Proses pencarian berlanjut sampai tidak ada rule yang dapat digunakan (*fire*), atau sampai sebuah tujuan (*goal*) tercapai.
- ▶ Ada dua metode *inferencing* dengan rules, yaitu **Forward Chaining** atau *Data-Driven* dan **Backward Chaining** atau *Goal-Driven*.



BACKWARD CHAINING

- ▶ Pendekatan *goal-driven*, dimulai dari ekspektasi apa yang diinginkan terjadi (hipotesis), kemudian mengecek pada sebab-sebab yang mendukung (ataupun kontradiktif) dari ekspektasi tersebut.
- ▶ Jika suatu aplikasi menghasilkan *tree* yang sempit dan cukup dalam, maka gunakan *backward chaining*.
- ▶ Beberapa sifat dari *backward chaining*:
 - ▶ *Good for Diagnosis.*
 - ▶ *Looks from present to past.*
 - ▶ *Works from consequent to antecedent.*
 - ▶ *Is goal-driven, top-down reasoning.*
 - ▶ *Works backward to find facts that support the hypothesis.*
 - ▶ *It facilitates a depth-first search.*
 - ▶ *The consequents determine the search.*
 - ▶ *It does facilitate explanation.*



BACKWARD CHAINING

- ▶ Pada komputer, program dimulai dengan tujuan (*goal*) yang diverifikasi apakah bernilai TRUE atau FALSE
- ▶ Kemudian melihat pada suatu rule yang mempunyai GOAL tersebut pada bagian konklusinya.
- ▶ Mengecek pada premis dari rule tersebut untuk menguji apakah rule tersebut terpenuhi (bernilai TRUE)
- ▶ Pertama dicek apakah ada assertion-nya
 - ▶ Jika pencarian disitu gagal, maka ES akan mencari rule lain yang memiliki konklusi yang sama dengan rule pertama tadi
 - ▶ Tujuannya adalah membuat rule kedua terpenuhi (*satisfy*)
- ▶ Proses tersebut berlanjut sampai semua kemungkinan yang ada telah diperiksa atau sampai rule inisial yang diperiksa (dg GOAL) telah terpenuhi
- ▶ Jika GOAL terbukti FALSE, maka GOAL berikut yang dicoba.



Forward chaining

- ▶ Merupakan grup dari multipel inferensi yang melakukan pencarian dari suatu masalah kepada solusinya.
- ▶ Jika klausa premis sesuai dengan situasi (bernilai TRUE), maka proses akan meng-assert konklusi
- ▶ Forward Chaining adalah *data driven* karena inferensi dimulai dengan informasi yg tersedia dan baru konklusi diperoleh
- ▶ Beberapa Sifat *forward chaining*:
 - ▶ *Good for monitoring, planning, and control*
 - ▶ *Looks from present to future.*
 - ▶ *Works from antecedent to consequent.*
 - ▶ *Is data-driven, bottom-up reasoning.*
 - ▶ *Works forward to find what solutions follow from the facts.*
 - ▶ *It facilitates a breadth-first search.*
 - ▶ *The antecedents determine the search.*
 - ▶ *It does not facilitate explanation.*
- ▶ Jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam, maka gunakan *forward chaining*.



Contoh Kasus

- ▶ **Sistem Pakar:** Penasihat Keuangan
- ▶ **Kasus :** Seorang *user* ingin berkonsultasi apakah tepat jika dia berinvestasi pada *stock* IBM?
- ▶ **Variabel-variabel yang digunakan:**
 - ▶ A = memiliki uang \$10.000 untuk investasi
 - ▶ B = berusia < 30 tahun
 - ▶ C = tingkat pendidikan pada level college
 - ▶ D = pendapatan minimum pertahun \$40.000
 - ▶ E = investasi pada bidang Sekuritas (Asuransi)
 - ▶ F = investasi pada saham pertumbuhan (*growth stock*)
 - ▶ G = investasi pada saham IBM
- ▶ Setiap variabel dapat bernilai TRUE atau FALSE



FAKTA YANG ADA:

- ▶ Diasumsikan si *user* (investor) memiliki data:
 - ▶ Memiliki uang \$10.000 (A TRUE)
 - ▶ Berusia 25 tahun (B TRUE)
 - ▶ Dia ingin meminta nasihat apakah tepat jika berinvestasi pada IBM stock?
 - ▶ **RULES**
 - ▶ R1 : IF seseorang memiliki uang \$10.000 untuk berinvestasi AND dia berpendidikan pada level college THEN dia harus berinvestasi pada bidang sekuritas
 - ▶ R2 : IF seseorang memiliki pendapatan pertahun min \$40.000 AND dia berpendidikan pada level college THEN dia harus berinvestasi pada saham pertumbuhan (*growth stocks*)
-



FAKTA YANG ADA:

- ▶ R3 : IF seseorang berusia < 30 tahun AND dia berinvestasi pada bidang sekuritas THEN dia sebaiknya berinvestasi pada saham pertumbuhan
- ▶ R4 : IF seseorang berusia < 30 tahun dan > 22 tahun THEN dia berpendidikan college
- ▶ R5 : IF seseorang ingin berinvestasi pada saham pertumbuhan THEN saham yang dipilih adalah saham IBM.

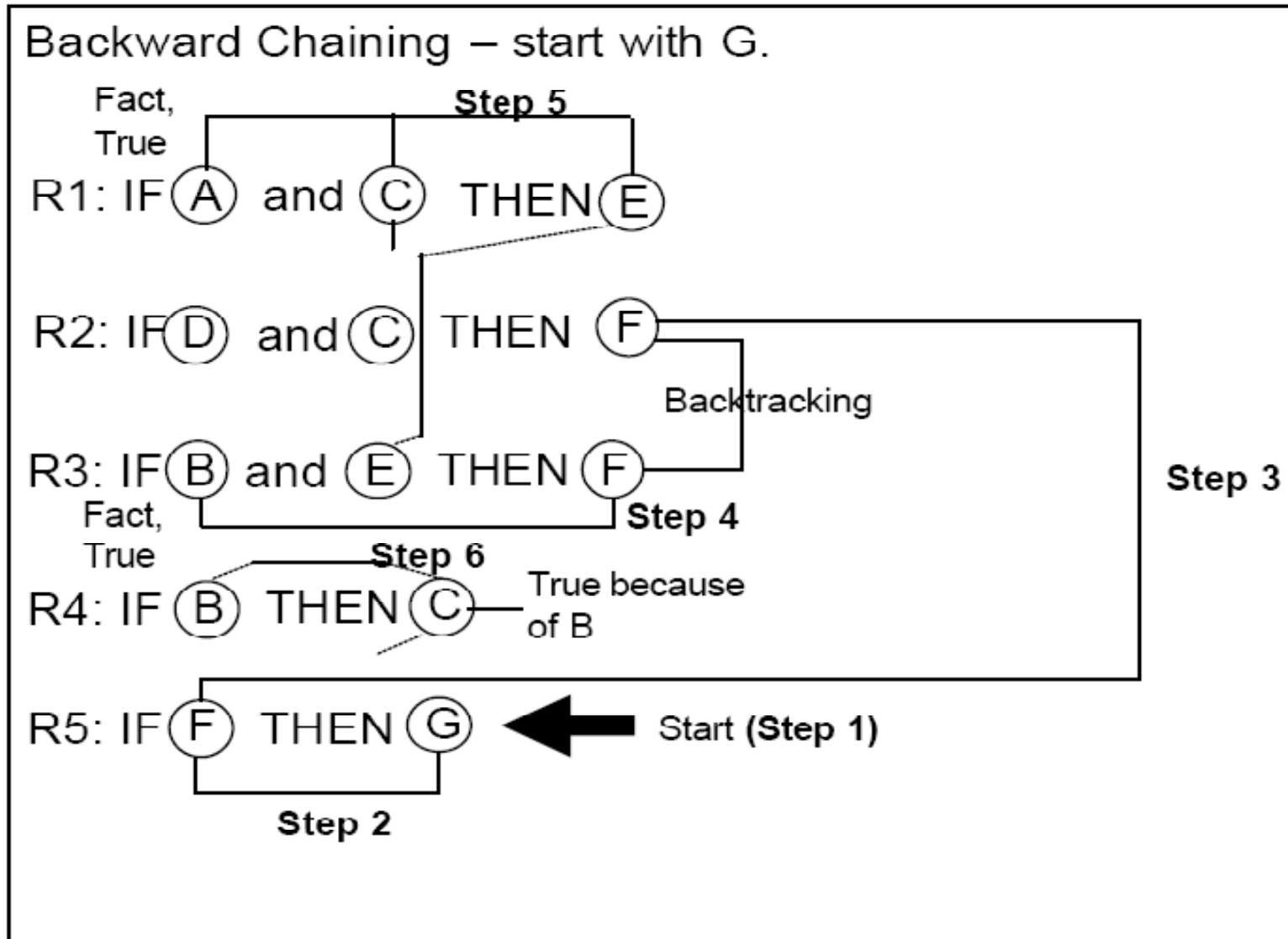


Rule simplification:

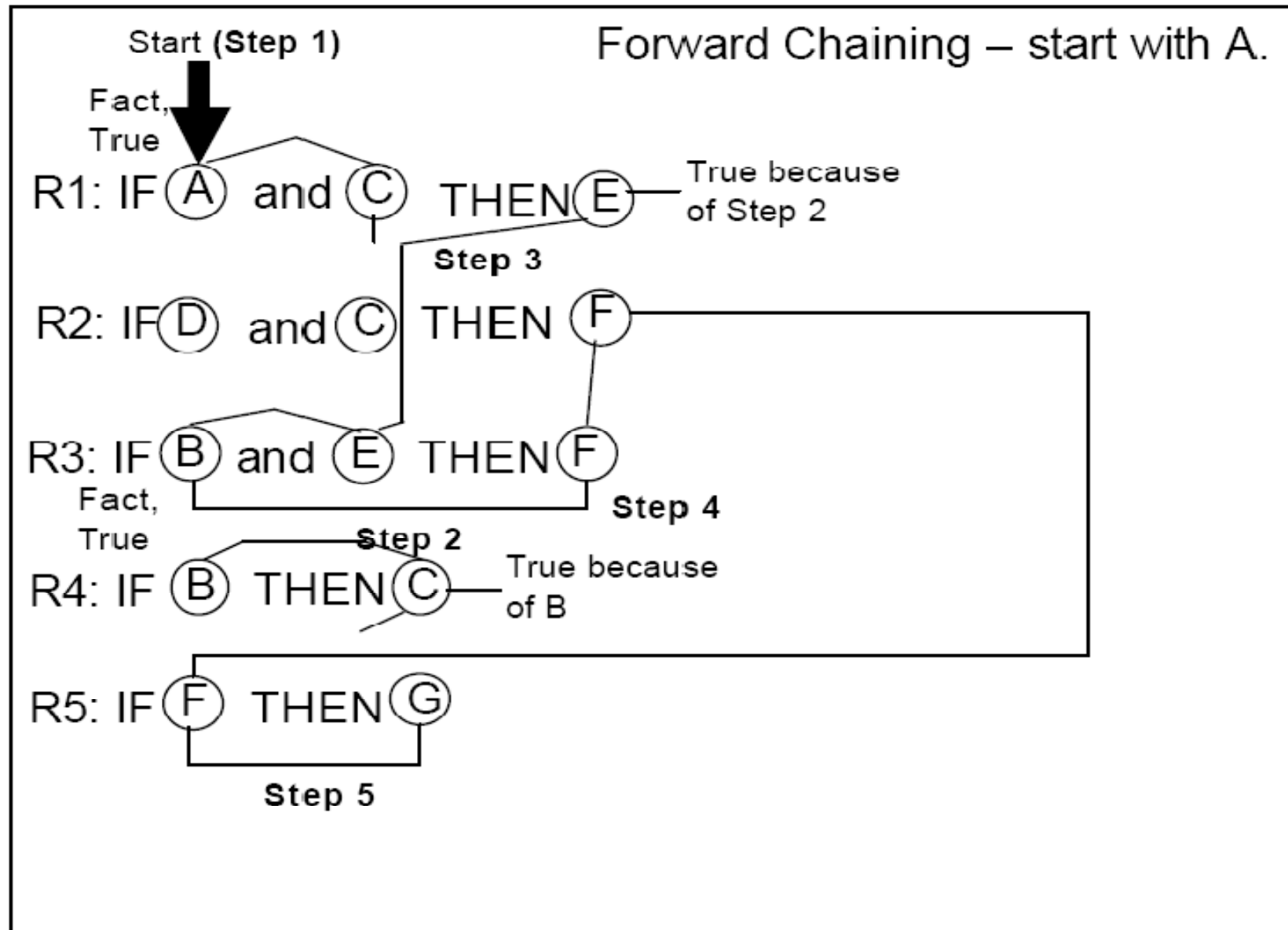
- ▶ – R1: IF A and C, THEN E
- ▶ – R2: IF D and C, THEN F
- ▶ – R3: IF B and E, THEN F
- ▶ – R4: IF B, THEN C
- ▶ – R5: IF F, THEN G



Solusi dengan Backward Chaining



Solusi dengan Forward Chaining



-
- ▶ Inferensi dengan rules (sebagaimana juga dengan Logika) dapat sangat efektif, tapi terdapat beberapa keterbatasan pada teknik-teknik tersebut.
 - ▶ Misalnya, perhatikan contoh berikut:
 - ▶ Proposisi 1 : Semua burung dapat terbang
 - ▶ Proposisi 2 : Burung Unta (Kasuari) adalah burung
 - ▶ Konklusi : Burung Unta dapat terbang
 - ▶ Konklusi tersebut adalah valid, tetapi dalam kenyataannya adalah salah, karena burung unta tidak dapat terbang. Untuk kasus seperti ini maka terkadang kita harus menggunakan teknik inferensi yang lain.
-



Fungsi dari Inference Engine

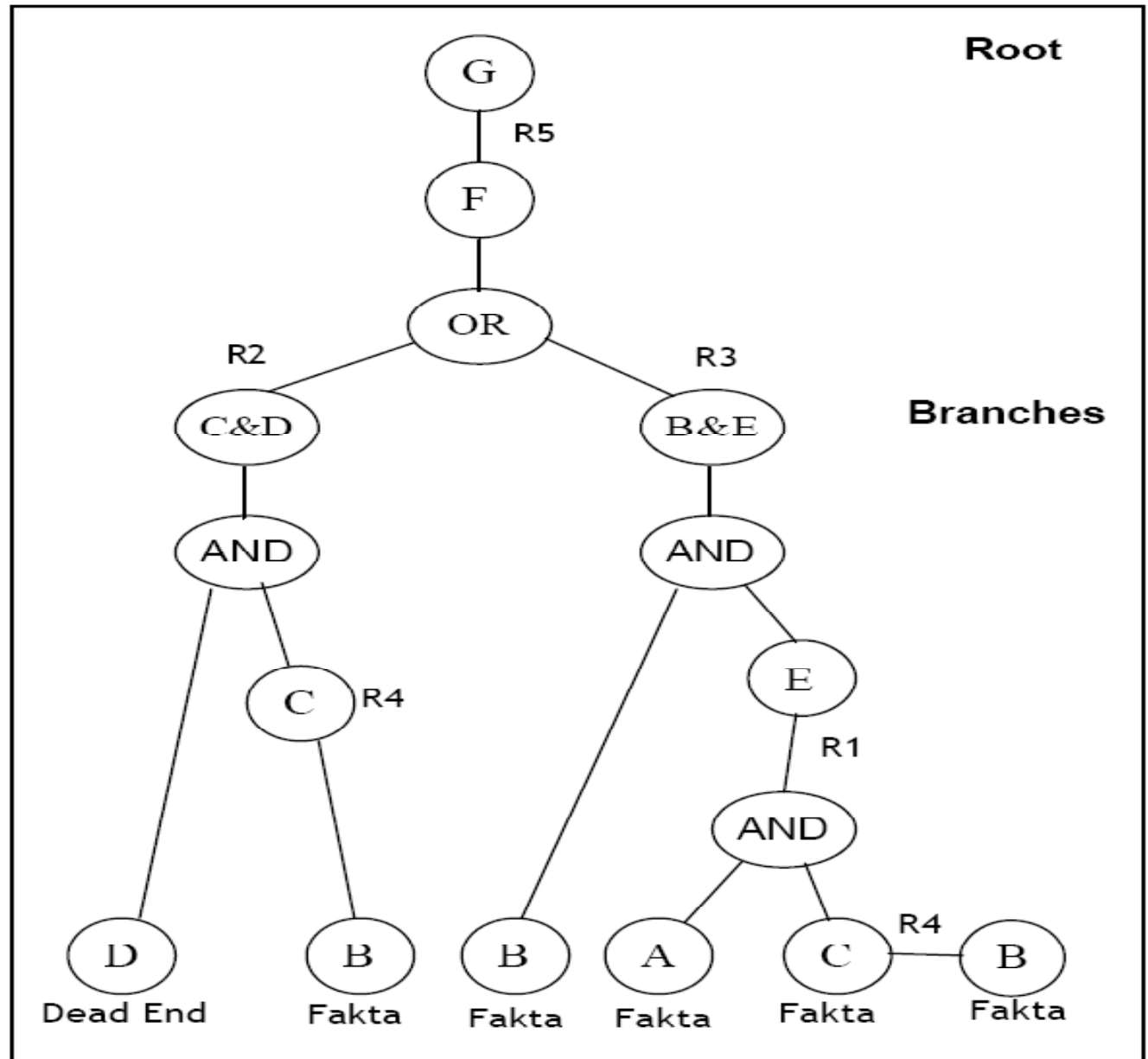
1. *Fire the rules*
 2. Memberikan pertanyaan pada user
 3. Menambahkan jawaban pada *Working Memory (Blackboard)*
 4. Mengambil fakta baru dari suatu rule (dari hasil inferensi)
 5. Menambahkan fakta baru tersebut pada *working memory*
 6. Mencocokkan fakta pada *working memory* dengan rules
 7. Jika ada yang cocok (*matches*), maka *fire rules* tersebut
 8. Jika ada dua rule yang cocok, cek dan pilih rule mana yang menghasilkan *goal* yang diinginkan
 9. *Fire the lowest-numbered unfired rule*
-



INFERENCE TREE (POHON INFERENSI)

- ▶ Penggambaran secara skematik dari proses inferensi
- ▶ Sama dengan *decision tree*
- ▶ Inferencing: *tree traversal*
- ▶ Advantage: Panduan untuk *Explanations* **Why** dan **How**





Gambar Inference Tree dari proses backward chaining

METODE INFERENSI YANG LAIN

- ▶ Inferensi dengan Frame
- ▶ *Model Based Reasoning*
- ▶ *Case Based Reasoning*



EXPLANATION

- ▶ *Human experts* memberikan justifikasi dan penjelasan (*explain*) dari apa yang mereka lakukan
- ▶ ES harus dapat melakukan hal yang sama
- ▶ **Explanation**: disediakan oleh ES untuk mengklarifikasi proses *reasoning*, rekomendasi, dan tindakan lainnya (mis: *asking a question*)
- ▶ *Explanation facility (justifier)*
- ▶ **Tujuan Explanation**
 - ▶ Membuat sistem menjadi lebih *intelligible*
 - ▶ Menjelaskan situasi yang *unanticipated* (tidak terantisipasi)
 - ▶ Memuaskan psikologis user dan/atau *social needs*
 - ▶ Klarifikasi dari asumsi yang diperoleh pada saat sistem beroperasi
 - ▶ Mengatur sensitivitas analisis

Dua Explanation Dasar

- ▶ **Why** - *Why* (mengapa) sebuah fakta dibutuhkan?
 - ▶ Pertanyaan ‘Why’ dapat diajukan oleh user pada saat diberikan sebuah pertanyaan untuk dijawab, untuk mengetahui alasan mengapa informasi itu diperlukan.
 - ▶ Misalnya, kita lihat pada kasus sebelumnya. Bisa saja muncul dialog sbb:
COMPUTER: *Berapa pendapatan Anda per tahun?*
CLIENT: *Why?*



Dua Explanation Dasar

- ▶ Komputer akan merespon dengan menunjukkan rule R2. Tetapi secara ideal jawaban komputer harus bisa sbb:
- ▶ **COMPUTER:**
 - ▶ *Untuk mengecek R2, Saya ingin tahu apakah pendapatan Anda per tahun diatas \$40000. Jika itu benar, saya akan membuat kesimpulan: karena anda berpendidikan pada level college, anda sebaiknya berinvestasi pada saham pertumbuhan*



Dua Explanation Dasar (lanj..)

- ▶ **How** – Menjelaskan bagaimana suatu konklusi atau rekomendasi diperoleh
 - ▶ Sistem sederhana – hanya pada konklusi final (terakhir)
 - ▶ Sistem yang kompleks menyediakan *chain of rules* yang digunakan untuk mencapai konklusi.



Misalnya, dari kasus sebelumnya, komputer memberikan konklusi:

- ▶ **COMPUTER:** *Inverstasi pada saham IBM*
- ▶ **CLIENT:** *How?*
- ▶ **COMPUTER:** (Akan menunjukkan urutan *rules* yang digunakan/*fires* : R5-R3-R1-R4)
- ▶ **IDEAL COMPUTER:** *Anda memiliki uang \$10000 untuk investasi dan anda berusia kurang dari 30 tahun, kemudian mengacu pada R4 bahwa anda berpendidikan pada level college. Untuk investor muda seperti anda jika ingin berinvestasi pada bidang saham pertumbuhan, maka saham IBM adalah pilihan terbaik yang bisa anda coba.*



INFERENSI DENGAN UNCERTAINTY

- ▶ **Representasi Uncertainty**
- ▶ Numerik
 - ▶ Menggunakan penskalaan numerik.
 - ▶ 0 : untuk *complete uncertainty* (FALSE)
 - ▶ 1 atau 100 : *complete certainty* (TRUE)
- ▶ Grafik : Horizontal bar
- ▶ Simbolik : *Likert Scale: Ranking, Ordinal, Cardinal*



INFERENSI DENGAN UNCERTAINTY

▶ **KOMBINASI CERTAINTY FACTOR**

- ▶ Certainty Factors (CF) mengekspresikan tingkat kepercayaan terhadap suatu event (atau fakta atau hipotesis) berdasarkan *evidence* (atau *the expert's assessment*)

▶ **Kombinasi beberapa CF dalam satu rule**

▶ **Operator AND:**

- ▶ IF *inflation is high*, CF = 50%, (A), AND
- ▶ IF *unemployment rate is above 7 percent*, CF = 70%, (B), AND
- ▶ IF *bond prices decline*, CF = 100%, (C)
- ▶ THEN *stock prices decline*



INFERENSI DENGAN UNCERTAINTY

- ▶ CF dari konklusi adalah nilai CF minimum pada bagian IF
- ▶ **CF(A, B, and C) = minimum(CF(A),CF(B),CF(C))**
- ▶ CF(A) adalah yang minimum → 50%, sehingga CF untuk “*stock prices decline*” adalah **50%**



INFERENSI DENGAN UNCERTAINTY

Operator OR:

- ▶ IF *inflation is low*, CF = 70%, (A), AND
- ▶ IF *bond prices are high*, CF = 85%, (B)
- ▶ THEN *stock prices will be high*
- ▶ CF dari konklusi adalah nilai CF maksimum pada bagian IF

CF(A or B) = maksimum (CF(A),CF(B))

- ▶ CF(B) adalah yang maksimum → 85%, sehingga CF untuk “*stock prices will be high*” adalah **85%**



INFERENSI DENGAN UNCERTAINTY

Kombinasi dua atau lebih Rules

▶ Misalnya:

- ▶ **R1:** IF *the inflation rate is less than 5 percent*, THEN *stock market prices go up* (CF = 0.7)
- ▶ **R2:** IF *unemployment level is less than 7 percent*, THEN *stock market prices go up* (CF = 0.6)
- ▶ Inflation rate = 4 percent and the unemployment level = 6.5 percent



Efek Kombinasi:

- ▶ **$CF(R1,R2) = CF(R1) + CF(R2)[1 - CF(R1)]$; or**
- ▶ **$CF(R1,R2) = CF(R1) + CF(R2) - CF(R1) \times CF(R2)$**
- ▶ Jika $CF(R1) = 0.7$ AND $CF(R2) = 0.6$, maka:
 $CF(R1,R2) = 0.7 + 0.6(1 - 0.7) = 0.7 + 0.6(0.3) = 0.88$
- ▶ Jadi peluang *stock prices will increase* adalah sebesar 88 persen

- ▶ **Jika ada rule ketiga yang ditambahkan, maka:**
- ▶ **$CF(R1,R2,R3) = CF(R1,R2) + CF(R3) [1 - CF(R1,R2)]$,**
Misalnya:
 - ▶ R3: IF *bond prices increases*, THEN *stock market prices go up* (CF = 0.85)
- ▶ Jika diasumsikan semua rule pada bagian IF bernilai TRUE maka, peluang *stock prices will increase* adalah:
 **$CF(R1,R2,R3) = 0.88 + 0.85 (1 - 0.88) = 0.88 + 0.85 (.12)$
 $= 0.982$**
- ▶ **catatan:**
 - ▶ *bond* = surat obligasi
 - ▶ *stock* = saham
 - ▶ *unemployment* = PHK

