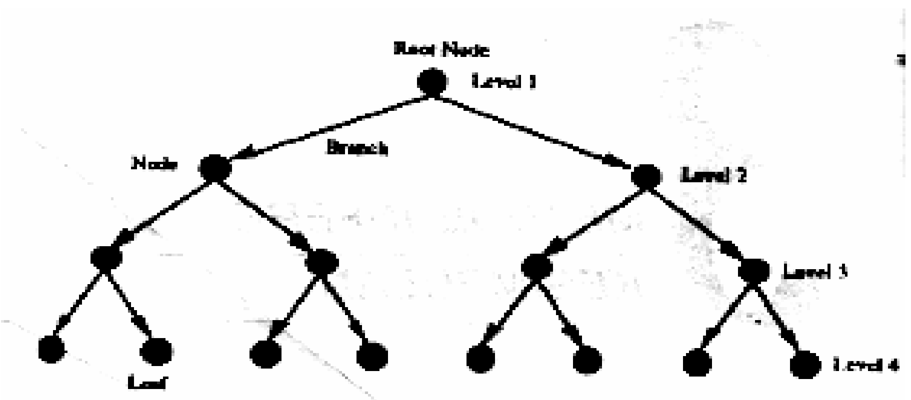
METODE INFERENSI/KESIMPULAN

TREES,LATTICES DAN GRAF

Tree :struktur data hirarki yg berisi node/vertices/objek yg menyimpan informasi/pengetahuan dan link/edges/cabang yg menghubungkan node

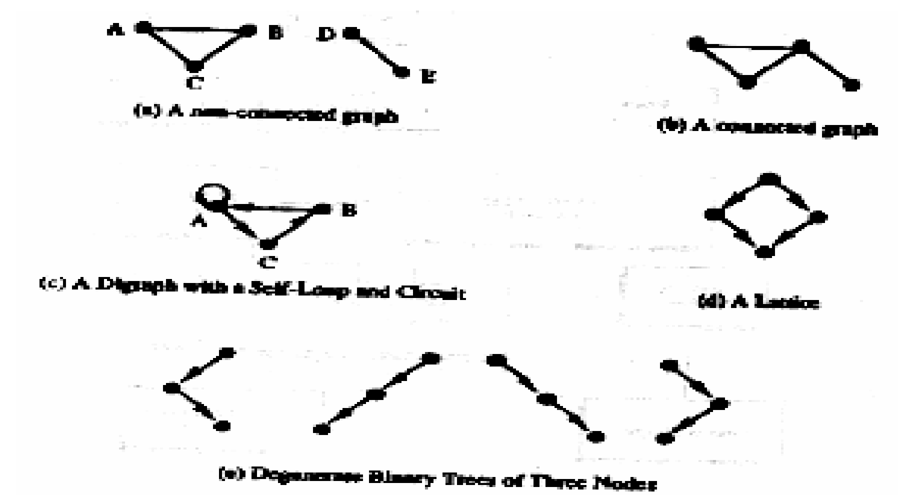
- ✓ Disebut juga dg tipe **jaringan semantik khusus**
- ✓ Merupakan kasus khusus yg disebut **graf**
- ✓ Suatu graf dapat mempunyai nol atau lebih link, dan tidak ada perbedaan antara root dan child
- ✓ Root : node tertinggi, leaves : terendah



- ✓ **Stuktur keputusan** : skema representasi pengetahuan dan metode pemberian alasan tentang pengetahuannya.
- ✓ Jika suatu keputusan adalah binary, maka tree keputusan binary mudah dibuat dan sangat efisien.
- ✓ Setiap pertanyaan, turun satu tingkat dalam tree.Jika seluruh leaves adalah jawaban dan seluruh node yg turun adalah pertanyaan, maka ada max 2^n untuk jawaban dan n pertanyaan

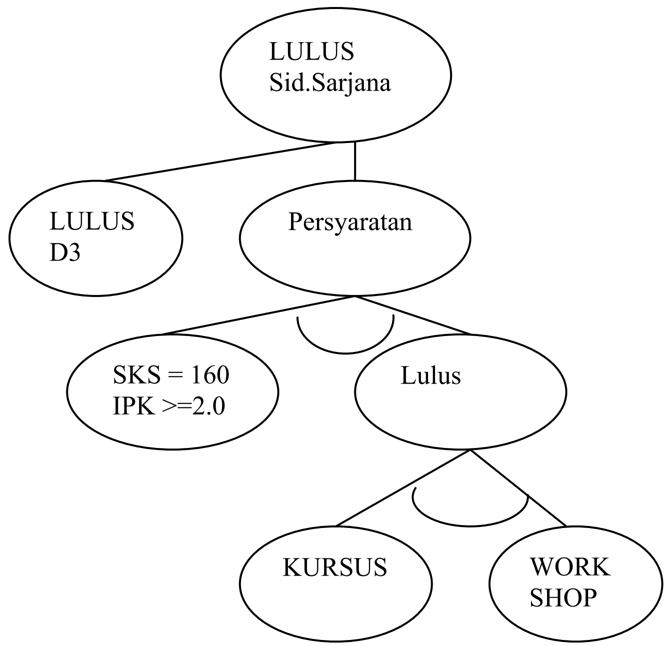
STATE SPACE

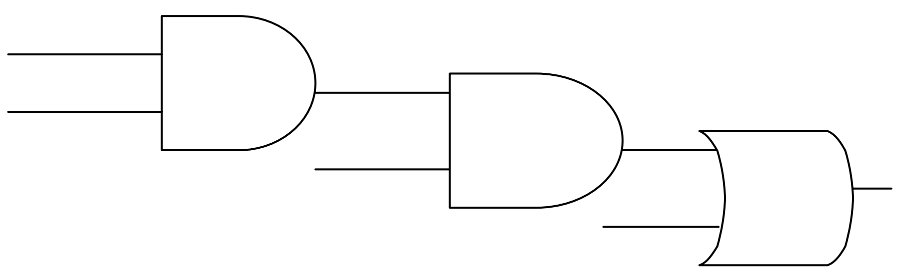
- ✓ State adalah kumpulan karakteristik yg dapat digunakan untuk menentukan status.
- ✓ State Space adalah rangkaian pernyataan yg menunjukkan transisi antara state dimana objek diekspreimen



POHON AND-OR

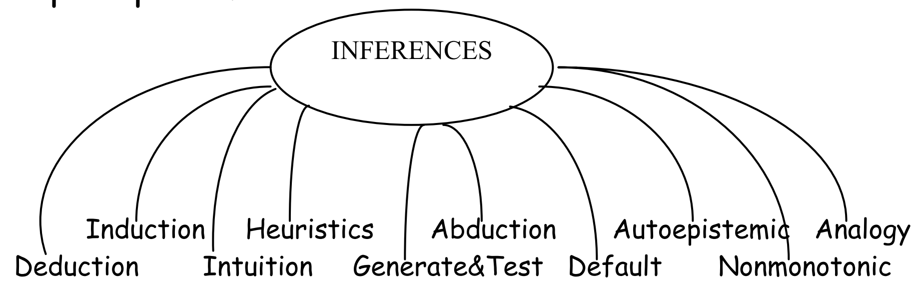
- ✓ Dalam SP, untuk menemukan solusi problem dapat menggunakan rangkaian backward yaitu dengan tree AND-OR dan AND-OR-NOT





LOGIKA DEDUKTIF DAN SILOGISME

Tipe-tipe Inferensi



Deduction

- ➢ Pemberian alasan logikal dimana kesimpulan harus mengikuti premis

Induction

- ➢ Inferensi dari khusus ke umum

Intuition

- ➢ Tidak ada teori yg menjamin. Jawabannya hanya muncul, mungkin dengan penentuan pola yg ada secara tidak disadari.

Heuristic

- ➢ Aturan yg didasarkan pada pengalaman

Generate & Test

- ➢ Trial dan error. Digunakan dgn perencanaan.

Abduction

- ➢ Pemberian alasan kembali dari kesimpulan yg benar ke premis .

Default
- Diasumsikan pengetahuan umum sebagai default

Autoepistemic
- Self-knowledge

Nonmonotonic
- Pengetahuan yg sebelumnya mungkin tdk benar jika bukti baru didapatkan

Analogy
- Kesimpulan yg berdasarkan pada persamaan untuk situasi yg lainnya.

Yang paling sering dipakai : deductive logic, unruk menentukan validitas "argument".

Silogisme merupakan satu type argumen logika.

Contoh :

Premise : Anyone who can program is intelligent

Premise : John can program

Conclusion : Therefore, John is intelligent

Premise
- ✓ Digunakan sebagai bukti untuk mendukung sutu kesimpulan.
- ✓ Disebut juga **antecedent**

Kesimpulan/Conclusion
- ✓ Disebut juga **consequent**

Karakteristik logika deduktif adalah *kesimpulan benar harus mengikuti dari premis yg benar*

Anyone who can program is intelligent

John can program

∴John is intelligent

Dalam bentuk IF-THEN

IF Anyone who can program is intelligent And
John can program

THEN John is intelligent

Silogisme klasik disebut **categorical syllogism.**

Premis dan kesimpulan ditentukan sebagai statement categorical dari 4 bentuk berikut :

FORM	SCHEMA
A	All S is P
E	No S is P
I	Some S is P
O	Some S is not P

S : Subjek kesimpulan disebut minor term

P : Predikat kesimpulan disebut major term

Major premise : All M is P

Minor premise : All S is M

Concluusion : All S is P

Silogisme diatas disebut **standard form** dimana major dan minor premis diidentifikasi.

Categorical Silogisme

- **A dan I** disebut "affirmative in quality" , subjek dimasukkan kedalam jenis predikat
- **E dan O** disebut "negative in quality", subjek tidak masuk dalam jenis predikat
- **IS = capula** = menghubungkan, menunjukkan bentuk tense dari kata kerja "tobe"
- **Middle term (M)**
- **All** dan **No** : universal quantifier, **Some** :particular quantifier
- **Mood** silogisme ditentukan dengan 3 huruf yg memberikan bentuk premis pokok, minor premis, dan kesimpulan.

	Figure 1	**Figure 2**	**Figure 3**	**Figure 4**
Major Premise	M P	P M	M P	P M
Minor Premise	S M	S M	M S	M S

Contoh :

All M is P
All S is M
∴ All S is P
type AAA-1

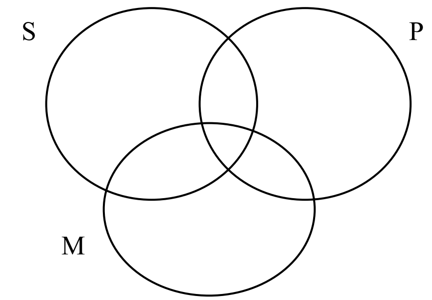
All M is P
No S is M
∴ No S is P
type ????

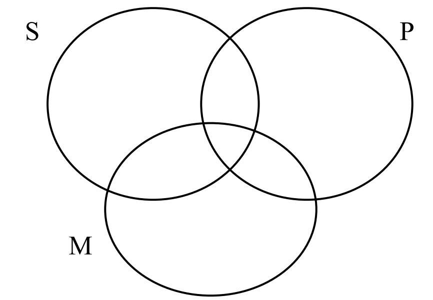
Some P are M
All M are S
∴Some S are P
type ????

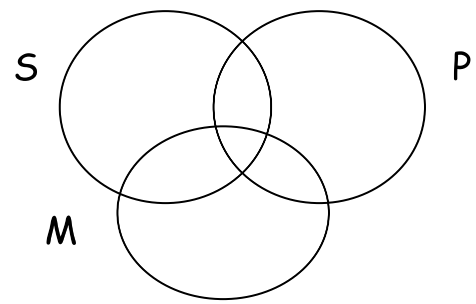
Untuk membuktikan validitas argumen silogisme, ada metode yang dinamakan "decision prosedure" yaitu dengan menggunakan diagram venn.

Contoh :

All M is P
All S is M
∴ All S is P
type AAA-1







BARIS INFERENCE (RULES OF INFERENCE)

Yaitu modus ponens dan modus tollens

Diagram venn tidak sesuai untuk argumen yg lebih kompleks karena menjadi sulit untuk dibaca pada decision tree untuk silogisme

Pada logika proposisional,
If there is power, the computer will work
<u>There is power</u>
∴ The computer will work

Maka dapat ditulis

$$\begin{array}{l} A \to B \\ \underline{A \qquad} \\ \therefore B \end{array} \quad \equiv \quad \begin{array}{l} p \to q \\ \underline{p \qquad} \\ \therefore q \end{array} \quad \equiv \quad p,\ p \to q;\ \therefore q$$

→ disebut "direct reasoning,modus ponenes, law of detachment dan assuming the antecedent"
p,q disebut variabel logika
A,B disebut konstanta proposisional

Bagaimana dengen skema untuk argumen dari tipe ini :

1. $\begin{array}{l} p \to q \\ \underline{q \qquad} \\ \therefore p \end{array}$
2. $\begin{array}{l} p \to q \\ \underline{\sim q \qquad} \\ \therefore \sim p \end{array}$

1. Disebut dg **fallacy of converse**
2. Disebut dg **indirect reasoning, modus tollens, law of contrapositive**

Hukum Inferensi	*Skema*	
1. Hukum Data Skema	$\begin{array}{l} p \to q \\ \underline{p} \\ \therefore q \end{array}$	
2. Hukum kontra positif	$\begin{array}{l} \underline{p \to q} \\ \therefore \neg q \to \neg p \end{array}$	
3. Hukum Modus Tollen	$\begin{array}{l} p \to q \\ \underline{\neg q} \\ \therefore \neg q \end{array}$	
4. Aturan Rangkaian (Hukum Silogisme)	$\begin{array}{l} p \to q \\ \underline{q \to r} \\ \therefore p \to r \end{array}$	
5. Hukum Inferensi Disjunktif	$\begin{array}{l} \underline{p \vee q} \\ \neg p \\ \therefore q \end{array}$	$\begin{array}{l} \underline{p \vee q} \\ \neg q \\ \therefore p \end{array}$
6. Hukum Negasi Dobel	$\begin{array}{l} \underline{\neg(\neg p)} \\ \therefore p \end{array}$	
7. Hukum De Morgan	$\begin{array}{l} \underline{\neg(p \wedge q)} \\ \therefore \neg p \vee \neg q \end{array}$	$\begin{array}{l} \underline{\neg(p \vee q)} \\ \therefore \neg p \wedge \neg q \end{array}$
8. Hukum Penyederhanaan	$\begin{array}{l} \underline{p \wedge q} \\ \therefore p \end{array}$	$\begin{array}{l} \underline{p \vee q} \\ \therefore q \end{array}$
9. Hukum Konjungsi	$\begin{array}{l} p \\ \underline{q} \\ \therefore p \wedge q \end{array}$	
10. Hukum Konjungsi	$\begin{array}{l} \underline{p} \\ \therefore p \vee q \end{array}$	
11. Hukum De Morgan	$\begin{array}{l} \neg(p \wedge q) \\ \underline{p} \\ \therefore \neg q \end{array}$	$\begin{array}{l} \neg(p \wedge q) \\ \underline{q} \\ \therefore \neg p \end{array}$

Tabel Kondisional dan variantnya

Kondisional	$p \rightarrow q$
Konversi	$q \rightarrow p$
Invensi	$\sim p \rightarrow \sim q$
Kontrapositif	$\sim q \rightarrow \sim p$

Contoh argumen dengan lebih dari 2 promise:
Chip prices rise only if the yen rises
The yen rises only if the dollar falls and
If the dollar falls then the yen rises.
Since chip proses have risen,
the dollar must have fallen

Proposisinya C = chip prices rise
Y = yen rises
D = dollar falls

$$\begin{array}{l} C \rightarrow Y \\ (Y \rightarrow D) \wedge (D \rightarrow Y) \\ \underline{C \qquad\qquad\qquad} \\ \therefore D \end{array}$$

Buktikan !.....

Solusi :

1. Ingat $p \rightarrow q$ dan $q \rightarrow p$ benar maka p dan q ekuivalen
2. Jika $(p \rightarrow q) \wedge (q \rightarrow p)$ maka ekuivalen dg $p \leftrightarrow q$ dg kata lain $p \equiv q$

Maka argumennya menjadi

$$\begin{array}{l} C \rightarrow Y \\ Y \equiv D \\ \underline{C \qquad} \\ \therefore D \end{array}$$

3. Karena Y sama dengan D maka substitusi D kedalam Y

Maka argumennya menjadi :

$$\begin{array}{l} C \rightarrow D \\ \underline{C \qquad} \\ \therefore D \end{array}$$ (TERBUKTI valid bahwa ini adalah modus ponens)

SOAL :

All men are mortal	(p)
Socrates is a man	(q)
Therefore, Socrates is mortal	$\therefore r$

Buktikan valid atau tidak ?....

FIRST ORDER PREDICATE LOGIC

Kategori silogisme dengan menggunakan predikat logik

TIPE	SKEMA	REPRESENTASI PREDIKAT
A	All S is P	$(\forall x)\ (S(x) \rightarrow P(x))$
E	No S is P	$(\forall x)\ (S(x) \rightarrow \sim P(x))$
I	Some S is P	$(\exists x)\ (S(x) \wedge P(x))$
O	Some S is not P	$(\exists x)\ (S(x) \wedge \sim P(x))$

Rule Hukum Universal Instantion menunjukkan individual yg mungkin digantikan dg universal yaitu simbol ϕ yg berarti fungsi proposisional

$\underline{(\forall x)\ \phi(x)}$ x= variabel yg mengatur seluruh individual

$\therefore \phi(a)$ a= individual khusus

Contoh : Socrates is human

$\underline{(\forall x)\ H\ (x)}$

$\therefore$H (Socrates)

dimana H(x) : fungsi proposissional dg x adalah human

Contoh lain

All men are mortal

<u>Socrates is a man</u>

$\therefore$Socrates is mortal

dimana H=man, M=mortal, s=socrates

Solusi :

1. $(\forall x)\ (H(x) \rightarrow M(x))$
2. $H(s)$
3. $\therefore M(s)$
4. $H(s) \rightarrow M(s)$
5. $M(s)$

LOGIC SYSTEMS = WFFS = WFF

- ✓ Koleksi objek seperti baris, aksioma, pernyataan dsb
- ✓ Tujuan :
  1. Menentukan bentuk argumen (WFFS=Well Formed Formulas)
     Contoh All S is P
  2. Menunjukkan baris inference yg valid
  3. Mengembangkan sendiri dg menemukan baris baru dari inference shg memperluas rentangan argumen yg dapat dibuktikan
- ✓ **Aksioma :**fakta sederhana atau assertion yg tidak dapat dibuktikan dari dalam sistem
- ✓ **System formal yang diperlukan :**
  1. Alfabet simbol
  2. String finite dari simbol tertentu, wffs
  3. Aksioma, definisi system
  4. Baris inference, yang memungkinkan wff, A untuk dikurangi sebagai kesimpulan dari set finite Γ wff lain dimana Γ = {A1,A2,…An}. Wffs harus berupa aksioma atau teori lain dari sistem logis

RESOLUSI

- ✓ Diperkenalkan oleh Robinson (1965)
- ✓ Merupakan baris inference yg utama dalam prolog
- ✓ Prolog menggunakan notasi "quantifier-free"
- ✓ Prolog didasarkan pada logika predikat first-order
- ✓ Sebelum resolusi diterapkan, wff harus berada dalam keadaan normal (bentuk standar) yaitu hanya menggunakan V , $\wedge$, ~

Mis wff (A V B) $\wedge$ (~B V C)   disebut bentuk normal konjungtif

A V B dan ~B V C

Ekspresi clausal umumnyya dituliskan dalam bentuk khusus yg disebut kowalski :

$A1, A2, \ldots\ldots\ldots A_N \rightarrow B1, B2, \ldots.B_M$

Dalam notasi predikat standar :

$A1 \wedge A2 \wedge \ldots\ldots\ldots A_N \rightarrow B1 \vee B2 \vee, \ldots.B_M$

Bentuk disjungsinya menggunakan

$(p \rightarrow q) \equiv \sim p \vee q$

menjadi :

$A1 \wedge A2 \wedge \vee \ldots\ldots\ldots A_N \rightarrow B1 \vee B2 \vee, \ldots.B_M$

$\equiv \sim(A1 \wedge A2 \wedge \ldots\ldots\ldots A_N) \vee (B1 \vee B2 \vee, \ldots.B_M)$

$\equiv \sim A1 \vee \sim A2 \vee \ldots\ldots\ldots \sim A_N \vee B1 \vee B2 \vee, \ldots. B_M$ INGAT De Morgan $\sim(p \wedge q) \equiv \sim p \vee \sim q$

Dengan klausa Horn menjadi :

$A1, A2, \ldots\ldots\ldots A_N \rightarrow B$

Dalam prolog :

$B :- A1, A2, \ldots A_N$

Untuk membuktikan teori benar dengan metode klasik "reductio ad absurdum" metode kontradiksi.

Tujuan resolusi adalah meng-infer klause baru **"revolvent"** dari 2 clause yang disebut parent clauses

Contoh

$$\begin{array}{l} A \vee B \\ \underline{A \vee \sim B} \\ \therefore \forall A \end{array}$$

dapat ditulis sbb

(A V B) $\wedge$ (A V ~B)

ingat distribusi :

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

sehingga

$(A \vee B) \wedge (A \vee \sim B) \equiv A \vee (B \wedge \sim B) \equiv A$ (resolvent)

ingat $(B \wedge \sim B) \equiv$ nil/null

Clause sekarang	*Resolven*	*Arti*
$p \rightarrow q, p$ or $\neg p \vee q, p$	q	*Ponen Modus*
$p \rightarrow q, q \rightarrow r$ or $\neg p \vee q, \neg q \vee r$	$p \rightarrow r$ or $\neg p \vee r$	Rangkaian atau silogisme Hipotetis
$\neg p \vee q\ p \vee$	q	Penggabungan
$\neg p \vee \neg q, p \vee q$	$\neg p \vee p$ or $\neg q \vee q$	Benar (Ulangan)
$\neg p, p$	nil	Salah (Kontradiksi)

SISTEM RESOLUSI DAN DEDUKSI

- **Refutation** adalah salah satu type pembuktian yang salah
- **Contoh** $A \rightarrow B$

  $B \rightarrow C$

  $C \rightarrow D$

  $\therefore A \rightarrow D$

  $A \rightarrow B, B \rightarrow C, C \rightarrow D \vdash A \rightarrow B$

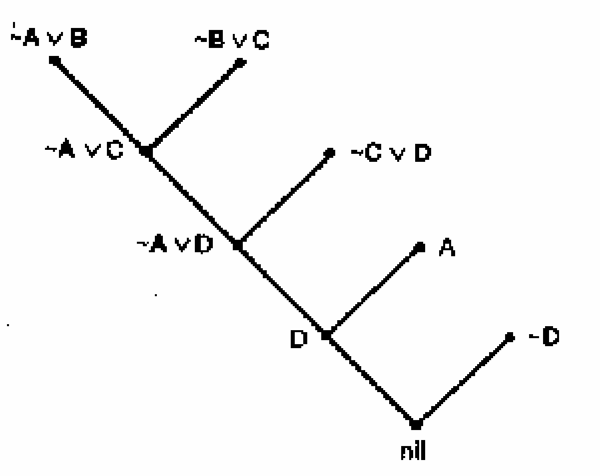
  Buktikan **bahwa kesimpulan adalah teori resolusi refulasi**

  Solusi :

  Gunakan $(p \rightarrow q) \equiv \sim p \vee q$ untuk semua premise dan kesimpulan, kemudian negasikan untuk kesimpulannya, sehingga menjadi

  $(\sim A \vee B) \wedge (\sim B \vee C) \wedge (\sim C \vee D) \wedge A \wedge \sim D$

  Pohon resolusi refutation



Terbukti bahwa $A \rightarrow D$ adalah teori

Latihan :

$B \rightarrow E$

$E \wedge E$

$E \wedge S \rightarrow F$

$F \wedge G \rightarrow R$

$R \wedge T \rightarrow C$

$B \wedge S \wedge G \wedge T \rightarrow C$

RESOLUSI DAN LOGIKA PREDIKAT FIRT ORDER

Sebelum resolusi dapat diterapkan, wff harus diletakkan dalam bentuk casual

Contoh :

Some programmers hate all failures

No programmer hates any success

$\therefore$ No failure is a success

P(x) = x is a progammer

F(x) = x is a failure

S(x) = x is a success

H(x,y) = x hates y

Premise dan kesimpulannya

(1) $(\exists x) [P(x) \wedge (\forall y) (F(y) \rightarrow H(x,y))]$

(2) $(\forall x) (P(x) \rightarrow (\forall y) (S(y) \rightarrow \sim H(x,y))]$

(3) $\sim(\forall y) (F(y) \rightarrow \sim S(,y))$

Konversi ke bentuk clausal

1. Hilangkan kondisional, $(p \rightarrow q) \equiv \sim p \vee q$
2. Geser negasi ke dalam (reduksi skope ~).
   Negasi digeser hanya berlaku untuk atomik formula
3. Hilangkan quantifier eksistensial
   - Jika $\exists$ tidak ada dalam skope $\forall$, ganti variabel dengan suatu konstanta baru
     $(\exists x)$ $P(x)$ diganti $P(a)$
   - Jika $\exists$ berada dalam skope $\forall$, ganti variabel dengan suatu fungsi yang memiliki argumen semua variabel dari $\forall$ tersebut
     $\forall x, \forall y, \exists z$ $P(x,y,z)$ diganti menjadi
     $\forall x, \forall y, P(x,y,F(x,y))$
4. Standarisasi variabel (jika perlu) sehingga tiap quantifier memiliki variabel yang berbeda
5. Geser semua $\forall$ ke kiri (karena semua quantifier punya nama yang berbeda, pergeseran tidak mempengaruhi hasil)
   Bentuk ini disebut prenex normal form terdiri atas prefix quantifier yang diikuti matriks
6. Hilangkan $\forall$. $\forall$ tidak perlu ditulis, diasumsikan semua variabel terkuantifikasi universal
7. Geser disjungsi (V) kedalam, sehingga terbentuk conjungsi normal form
8. Buang konjungsi dan uraikan menjadi klausa-klausa
9. Standarisasi variabel (jika perlu) sehingga tidak ada variabel yang muncul pada lebih dari 1 klausa.

Contoh : Ubah ke bentuk klausal !!!!!!

$\forall x\ (Balok(x) \rightarrow (\exists y\ (Diatas(x,y) \wedge \sim Piramid(y))$
$\quad \wedge \sim\exists y\ (Diatas(x,y) \wedge Diatas(y,x))$
$\quad \wedge \forall y\ (\sim Balok(y) \rightarrow \sim Sama(x,y))))$

Solusi :

1. $\forall x\ (\boldsymbol{\sim} Balok(x)\ \mathbf{V}\ (\exists y\ (Diatas(x,y) \wedge \sim Piramid(y))$
   $\wedge \sim\exists y\ (Diatas(x,y) \wedge Diatas(y,x))$
   $\wedge \forall y\ (\sim Balok(y)\ \mathbf{V} \sim Sama(x,y))))$

2. $\forall x\ (\sim Balok(x)\ V\ (\exists y\ (Diatas(x,y) \wedge \sim Piramid(y))$
   $\wedge \forall \mathbf{y}\ (\boldsymbol{\sim} Diatas(x,y)\ \mathbf{V\sim}\ Diatas(y,x))$
   $\wedge \forall y\ (\sim Balok(y)\ V \sim Sama(x,y))))$

3. $\forall x\ (\sim Balok(x)\ V\ (Diatas(x,\mathbf{f(x)}) \wedge \sim Piramid(\mathbf{f(x)}))$
   $\wedge \forall y\ (\sim Diatas(x,y)\ V\sim Diatas(y,x))$
   $\wedge \forall y\ (\sim Balok(y)\ V \sim Sama(x,y))))$

4. $\forall x\ (\sim Balok(x)\ V\ (Diatas(x,f(x)) \wedge \sim Piramid(f(x)\mathbf{)})$
   $\wedge \forall y\ (\sim Diatas(x,y)\ V\sim Diatas(y,x))$
   $\wedge \forall \mathbf{z}\ (\sim Balok(\mathbf{z})\ V \sim Sama(x,\mathbf{z}))))$

5. $\forall x \forall \mathbf{y} \forall \mathbf{z}\ (\sim Balok(x)\ V\ (Diatas(x,f(x)) \wedge \sim Piramid(f(x)\mathbf{)})$
   $\wedge\ (\sim Diatas(x,y)\ V\sim Diatas(y,x))$
   $\wedge\ (\sim Balok(z)\ V \sim Sama(x,z))))$

6. (~Balok (x) V ((Diatas(x,f(x)) ∧ ~Piramid(f(x)))
∧ (~Diatas(x,y) V~ Diatas(y,x))
∧ (~Balok(z) V ~Sama(x,z))))

7. (~Balok (x) V Diatas(x,f(x))
∧ (~Balok (x) V ~Piramid(f(x)))
∧ (~Balok (x) V ~Diatas(x,y) V~ Diatas(y,x))
∧ (~Balok (x) V ~Balok(z) V ~Sama(x,z))))

8. 1. ~Balok (x) V Diatas(x,f(x))
2. ~Balok (x) V ~Piramid(f(x))
3. ~Balok (x) V ~Diatas(x,y) V~ Diatas(y,x)
4. ~Balok (x) V ~Balok(z) V ~Sama(x,z)

9. 1. ~Balok (x) V Diatas(x,f(x))
2.~Balok (k) V ~Piramid(f(k))
3.~Balok (m) V ~Diatas(m,y) V~ Diatas(y,m)
4. ~Balok (n) V ~Balok(z) V ~Sama(n,z)

RANGKAIAN BACKWARD DAN FORWARD

Forward : bottom-up reasoning, breadth first
Backward : top-down reasoning, depth-first

Rangkaian forward	Rangkaian Backward
-Planning, monitoring,control -Saat sekarang ke masa depan -Antecedent ke consequent -Data driven, bottom-up -Kerja mundur untuk menemukan pemecahan yg mengikuti fakta -Breadth-first search -Antecedent menentukan pencarian -Fasilitas bukan penjelasan	-Diagnosis -Sekarang ke masa lalu -Consequent ke antecedent -Goal driven, top-down -Kerja mundur untuk menemukan fakta yg mendukung hipotesa -Depth-first search -Consequent menentukan pencarian -Fasilitas penjelasan

METODE LAIN DARI INFERENCE/KESIMPULAN

ANALOGI

- Mencoba dan menghubungkan situasi lama sebagai penuntun ke situasi baru.
- Contoh : diagnosis medical
- Pemberian alasan analogis berhubungan dgn induksi

GENERATE AND TEST

- Pembuatan solusi kemudian pengetesan untuk melihat apakah solusi yg diajukan memenuhi semua persyaratan. Jika solusi memenuhi maka berhenti yg lain membuat sollusi yg baru kemudian test lagi dst
- Contoh : Dendral, prog AM(artificial Mathematician),Mycin

ABDUCTION/PENGAMBILAN

- Metodenya sama dg modus ponens
- Abduction / Modus ponens

Abduction	Modus ponens
$p \rightarrow q$	$p \rightarrow q$
q	p
$\therefore p$	$\therefore q$

- Bukan argument deduksi yg valid
- Berguna untuk baris/rules heuristik inference
- Analogi,generate and test, abduction adalah metode bukan deduksi. Dari premise yg benar, metode ini tidak dapat membuktikan kesimpulan yg benar

Perbedaan :

Inference	Start	Tujuan
FORWARD	Fakta	Kesimpulan yg harus mengikuti
BACKWARD	Kesimpulan tdk pasti	Fakta pendukung kesimpulan
ABDUCTION	Kesimpulan benar	Fakta yg dpt mengikuti

NONMONOTONIC REASONING

- *Tambahan aksioma* yg baru pada sistem logika berarti bahwa banyak teori yg dapat dibuktikan jika ada banyak aksioma dari teori yg didapat, disebut ***monotonik sistem***

METAKNOWLEDGE

- Program meta-DENDRAL menggunakan induksi untuk menyimpulkan baris baru dari struktur kimia.
- Contoh : TEIRESIAS yg menambah pengetahuan secara interaktif dari expert