

CSCM603130: Sistem Cerdas

First Order Logic Inference

Fariz Darari, Aruni Yasmin Azizah

Fakultas Ilmu Komputer
Universitas Indonesia

2019/2020 • Semester Ganjil



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Logika untuk knowledge representation & reasoning (KRR)

- **Declarative**: knowledge base terpisah dari mekanisme penalaran (*inference*).
- **Unambiguous**: thd. suatu interpretasi, arti sebuah *sentence* jelas.
- Seberapa **expressive**-kah propositional logic?
 - Wumpus World: “Hembusan angin dirasakan pada kamar-kamar yang bertetangga dengan lubang jebakan”. Berapa banyak kalimat proposisi yang diperlukan?
 - Bahasa alami mengenal **objek**, **relasi** antar objek, dan penalaran tentang relasi tersebut. Bagaimana hal ini dilakukan dalam logika proposisi?
- Dibutuhkan logika lain sebagai bahasa representasi pengetahuan yang lebih **expressive**.



First Order Logic

- Menggunakan propositional logic sebagai fondasi untuk membangun logika yang lebih expressive:
 - **Objects**, mis: orang, bangunan, buku, meja, ...
 - **Relations**: menyatakan sifat objects (merah, bulat, cantik, ...) atau hubungan beberapa objects (lebih besar dari, di atas, ...)
 - **Functions**: memetakan object ke tepat satu object lain, seperti ayah dari, satu lebihnya dari, ...



Outline

1 FOL: Syntax & Semantics

- Model, Simbol, & Interpretasi
- Term & Kalimat FOL

2 KBA dengan FOL

- Substitution
- Contoh KBA dengan FOL
- Knowledge Engineering

3 Inference untuk FOL

- Inference Rules untuk Quantifiers
- Reduksi FOL Inference ke PL Inference
- Unification
- Forward chaining & Backward Chaining
- Resolution



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



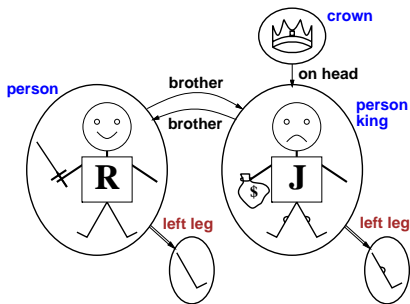
Model, Simbol, & Interpretasi (1)

- Dalam semantics propositional logic (PL), *model* mengaitkan propositional symbol dengan suatu nilai kebenaran.
- Dalam semantics FOL, *model* dibangun oleh:
 - **Domain**: himpunan object di “dunia” yang dibicarakan (**domain elements**)
 - **Interpretasi**: menentukan object, relasi, dan fungsi mana (di “dunia” yang dibicarakan) yang dirujuk masing-masing oleh **simbol constant**, **simbol predicate**, dan **simbol fungsi**.



Model, Simbol, & Interpretasi (2)

Model dengan 5 objects (Richard the Lionheart, King John, kaki kiri Richard dan John, dan mahkota), 2 binary relation (“brother” dan “on head”), 3 unary relations (“crown”, “person”, “king”), dan 1 unary function (“left leg”).



Model, Simbol, & Interpretasi (3)

- Dalam sebuah model, sebuah kalimat dapat di-*interpretasi*-kan dengan banyak cara.

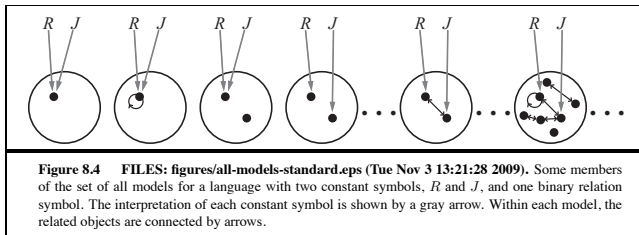
Contoh intepretasi

- $Richard \rightarrow Richard\ the\ Lionheart,$
 $John \rightarrow King\ John$
- $Brother \rightarrow \text{hubungan persaudaraan} =$
 $\{\langle Richard\ the\ Lionheart, King\ John \rangle, \langle King\ John, Richard\ the\ Lionheart \rangle\};$
 $OnHead \rightarrow \{\langle mahkota, King\ John \rangle\};$
 $Person \rightarrow \{Richard\ the\ Lionheart, King\ John\};$
 $Crown \rightarrow \{mahkota\};$
 $King \rightarrow \{King\ John\}$
- $LeftLeg \rightarrow \text{fungsi "left leg"}:$
 $Richard\ the\ Lionheart \mapsto \text{kaki kiri Richard}$
 $King\ John \mapsto \text{kaki kiri John}$



Model, Simbol, & Interpretasi (4)

Seperti halnya dalam PL, *entailment*, *validity*, *satisfiability*, dll. pada FOL ditentukan dengan memperhatikan **semua kemungkinan model**.



- Menentukan *entailment* dengan enumerasi model mustahil!
- Biasanya ada satu interpretasi yang “dimaksudkan” → **intended interpretation** (lihat contoh sebelumnya).



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Pembentukan Kalimat FOL

Sentence \rightarrow *AtomicSentence* | *ComplexSentence*

AtomicSentence \rightarrow *Predicate* | *Predicate*(*Term*, ...) | *Term* = *Term*

ComplexSentence \rightarrow (*Sentence*) | [*Sentence*]

| \neg *Sentence*

| *Sentence* \wedge *Sentence*

| *Sentence* \vee *Sentence*

| *Sentence* \Rightarrow *Sentence*

| *Sentence* \Leftrightarrow *Sentence*

| *Quantifier* *Variable*, ... *Sentence*

Term \rightarrow *Function*(*Term*, ...)

| *Constant*

| *Variable*

Quantifier \rightarrow \forall | \exists

Constant \rightarrow *A* | *X*₁ | *John* | ...

Variable \rightarrow *a* | *x* | *s* | ...

Predicate \rightarrow *True* | *False* | *After* | *Loves* | *Raining* | ...

Function \rightarrow *Mother* | *LeftLeg* | ...

OPERATOR PRECEDENCE : $\neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow$



Term dan Kalimat atomik

Syntax *term*

variable atau

constant atau

function($term_1, \dots, term_n$), di mana $n \geq 1$

Syntax *atomic sentence*

predicate($term_1, \dots, term_m$), di mana $m \geq 0$

atau $term_1 = term_2$

Contoh:

- *Brother*(*KingJohn*, *RichardTheLionheart*)
- $>$ (*Length*(*LeftLegOf*(*Richard*)), *Length*(*LeftLegOf*(*KingJohn*)))



Arti kalimat atomik FOL

Arti dari sebuah kalimat atomik FOL:

Kalimat atomik $\textit{predicate}(term_1, \dots, term_n)$ dikatakan bernilai *true* dalam model m jhj *relasi* yang dirujuk oleh simbol *predicate* berlaku benar untuk *objek-objek* yang dirujuk oleh $term_i$, $1 \leq i \leq n$, dalam model m .



Kalimat kompleks dengan logical connectives

Syntax *complex sentence* dengan logical connectives

$\neg S$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$

Contoh:

- $Brother(John, Richard) \Rightarrow Brother(Richard, John)$
- $>(1, 2) \vee \leq(1, 2)$
- $>(1, 2) \wedge \neg >(1, 2)$



Kalimat kompleks dengan universal quantification

Syntax:

Jika S kalimat, \forall *variables* S adalah kalimat kompleks

Contoh: "Semua mahasiswa Fasilkom UI adalah pintar"

$\forall x (Mahasiswa(x, FasilkomUI) \Rightarrow Pintar(x))$

Semantics:

$\forall x$ S bernilai *true* dalam model m jh S bernilai *true* untuk **semua** kemungkinan *extended interpretation* untuk variabel x dalam model m .
Extended interpretation (dibentuk dari intended interpretation)
 menspesifikasikan domain element yang dirujuk oleh x .

$\forall x$ $S \equiv$ **conjunction** dari S untuk instantiation var. x dengan domain element:

$x \rightarrow Ani$: Jika Ani mahasiswa Fasilkom UI, maka Ani pintar.

$x \rightarrow Anto$: Jika Anto mahasiswa Fasilkom UI, maka Anto pintar.

⋮

$x \rightarrow Zakky$: Jika Zakky mahasiswa Fasilkom UI, maka Zakky pintar.



Perhatian!

- Biasanya, \Rightarrow adalah *operator/connective* yang digunakan dengan \forall .
- Masalah yang sering terjadi: menggunakan \wedge sebagai *connective* untuk \forall :
 $\forall x (mahasiswa(x, FasilkomUI) \wedge pintar(x))$
- Kalimat ini berarti “Semua orang adalah mahasiswa Fasilkom UI dan pintar”.



Kalimat kompleks dengan existential quantification

Syntax:

Jika S kalimat, \exists *variable* S adalah kalimat kompleks

Contoh: “Ada mahasiswa UI yang pintar”

$\exists x (Mahasiswa(x, UI) \wedge Pintar(x))$

Semantics:

$\exists x S$ bernilai *true* dalam model m jh S bernilai *true* untuk *setidaknya satu extended interpretation* dalam model m untuk variabel x .

$\exists x S \equiv$ *disjunction* dari S untuk instantiation var. x dengan domain element:

$x \rightarrow Ani$: Ani adalah mahasiswa UI \wedge Ani pintar;

$x \rightarrow Anto$: Anto adalah mahasiswa UI \wedge Anto pintar;

\vdots

$x \rightarrow Zakky$: Zakky adalah mahasiswa UI \wedge Zakky pintar;



Perhatian!

- Biasanya, \wedge adalah *operator/connective* yang digunakan dengan \exists .
- Masalah yang sering terjadi: menggunakan \Rightarrow sebagai *connective* untuk \exists :
 $\exists x \text{ (mahasiswa}(x, UI) \Rightarrow \text{pintar}(x))$
- Kalimat ini *true* jika ada setidaknya 1 orang (object) yang bukan mahasiswa UI!



Beberapa sifat \forall dan \exists

- $\forall x \forall y S$ sama dengan $\forall y \forall x S$, biasa ditulis $\forall x, y S$
- $\exists x \exists y S$ sama dengan $\exists y \exists x S$, biasa ditulis $\exists x, y S$
- $\forall x \exists y S$ TIDAK sama dengan $\exists y \forall x S$!
 - $\forall x \exists y \text{ Mencintai}(x, y)$
 “Setiap orang mencintai (sekurang-kurangnya) seseorang.”
 - $\exists y \forall x \text{ Mencintai}(x, y)$
 “Ada (sekurang-kurangnya) seseorang yang dicintai oleh semua orang.”
- Quantifier bisa dinyatakan dengan yang lain:

$\forall x \text{ Doyan}(x, \text{Bakso})$	sama dengan	$\neg \exists x \neg \text{Doyan}(x, \text{Bakso})$
$\exists x \text{ Doyan}(x, \text{Dodol})$	sama dengan	$\neg \forall x \neg \text{Doyan}(x, \text{Dodol})$



Equality

- Kalimat $term_1 = term_2$ bernilai *true* dalam model m jhj $term_1$ and $term_2$ merujuk ke *object* yang sama dalam model m .
- Contoh:
 - $Ayah(Anto) = Budi$ adalah *satisfiable*
 - $Anto = Budi$ juga *satisfiable*!
 - $Anto = Anto$ adalah *valid*.
- Bisa digunakan dengan negasi untuk membedakan dua *term*:
 $\exists x, y \ (Mencintai(Anto, x) \wedge Mencintai(Anto, y))$



Equality

- Kalimat $term_1 = term_2$ bernilai *true* dalam model m jhj $term_1$ and $term_2$ merujuk ke *object* yang sama dalam model m .
- Contoh:
 - $Ayah(Anto) = Budi$ adalah *satisfiable*
 - $Anto = Budi$ juga *satisfiable*!
 - $Anto = Anto$ adalah *valid*.
- Bisa digunakan dengan negasi untuk membedakan dua *term*:
 - $\exists x, y \ (Mencintai(Anto, x) \wedge Mencintai(Anto, y))$
 - $\exists x, y \ (Mencintai(Anto, x) \wedge Mencintai(Anto, y) \wedge \neg(x = y))$
 - (Anto mendua atau lebih!)



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Knowledge-based Agent dengan FOL

- Dalam KBA, terdapat dua operasi: TELL dan ASK.
- **TELL**: memberikan informasi dengan menambahkan suatu sentence ke dalam KB (**assertion**).
 - $\text{TELL}(\text{KB}, \text{King}(\text{John}))$
 - $\text{TELL}(\text{KB}, \forall x (\text{King}(x) \Rightarrow \text{Person}(x)))$
- **ASK**: memberikan pertanyaan kepada KB (**query**).
 - $\text{ASK}(\text{KB}, \text{King}(\text{John}))$ jawabannya *true*.
 - $\text{ASK}(\text{KB}, \text{Person}(\text{John}))$ jawabannya *true*.
 - $\text{ASK}(\text{KB}, \exists x \text{ Person}(x))$ jawabannya *true*.
- Satu fungsi lain untuk menghasilkan jawaban yang lebih informatif: **ASKVARS**



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Substitution

- Terkadang jawaban "Yes-or-No" tidak memberikan informasi yang bermanfaat.

TELL(*KB*, *King(John)*)

TELL(*KB*, *Person(Richard)*)

TELL(*KB*, $\forall x \ (King(x) \Rightarrow Person(x))$)

- **ASK**VARs(*KB*, *Person(x)*), jawabannya:



Substitution

- Terkadang jawaban "Yes-or-No" tidak memberikan informasi yang bermanfaat.

TELL(*KB*, *King(John)*)

TELL(*KB*, *Person(Richard)*)

TELL(*KB*, $\forall x \ (King(x) \Rightarrow Person(x))$)

- **ASKVARS(*KB*, *Person(x)*)**, jawabannya:
 - $\{x/John\}$



Substitution

- Terkadang jawaban "Yes-or-No" tidak memberikan informasi yang bermanfaat.

TELL(*KB*, *King(John)*)

TELL(*KB*, *Person(Richard)*)

TELL(*KB*, $\forall x \ (King(x) \Rightarrow Person(x))$)

- ASK*VARs*(*KB*, *Person(x)*), jawabannya:
 - $\{x/John\}$
 - $\{x/Richard\}$



Substitution

- Terkadang jawaban "Yes-or-No" tidak memberikan informasi yang bermanfaat.

TELL(*KB*, *King(John)*)

TELL(*KB*, *Person(Richard)*)

TELL(*KB*, $\forall x \ (King(x) \Rightarrow Person(x))$)

- **ASKVARS(*KB*, *Person(x)*)**, jawabannya:
 - $\{x/John\}$
 - $\{x/Richard\}$
- ASKVARS khususnya digunakan untuk *KB* yang hanya terdiri dari Horn clauses, karena memungkinkan mengikat variabel dengan nilai yang spesifik (**substitution**).



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term (binding list)* θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term (binding list)* θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.

- $S = \text{LebihPintar}(x, y), \theta = \{x/\text{Ani}, y/\text{Anto}\}$



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term (binding list)* θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.

- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, y/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Ani}, \text{Anto})$



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term* (*binding list*) θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.

- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, y/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Ani}, \text{Anto})$
- $S = \text{LebihPintar}(x, y)$, $\theta = \{y/\text{Ani}, x/\text{Anto}\}$



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term (binding list)* θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.

- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, y/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Ani}, \text{Anto})$
- $S = \text{LebihPintar}(x, y)$, $\theta = \{y/\text{Ani}, x/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Anto}, \text{Ani})$



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term (binding list)* θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.

- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, y/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Ani}, \text{Anto})$
- $S = \text{LebihPintar}(x, y)$, $\theta = \{y/\text{Ani}, x/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Anto}, \text{Ani})$
- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, z/\text{Anto}\}$



Substitution

Substitution/ Binding List

Mengaplikasikan suatu himpunan pasangan *variable/term (binding list)* θ pada suatu kalimat S yang menghasilkan kalimat baru $S\theta$.

- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, y/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Ani}, \text{Anto})$
- $S = \text{LebihPintar}(x, y)$, $\theta = \{y/\text{Ani}, x/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Anto}, \text{Ani})$
- $S = \text{LebihPintar}(x, y)$, $\theta = \{x/\text{Ani}, z/\text{Anto}\}$
menghasilkan $S\theta = \text{LebihPintar}(\text{Ani}, y)$



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”

$$\forall x, y \ (Ibu(x) = y \Leftrightarrow Orangtua(y, x) \wedge Perempuan(y))$$



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”
 $\forall x, y \ (Ibu(x) = y \Leftrightarrow Orangtua(y, x) \wedge Perempuan(y))$
- “Laki-laki dan perempuan adalah dua kelompok berbeda”



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”
 $\forall x, y \ (Ibu(x) = y \Leftrightarrow Orangtua(y, x) \wedge Perempuan(y))$
- “Laki-laki dan perempuan adalah dua kelompok berbeda”
 $\forall x \ (Lelaki(x) \Leftrightarrow \neg Wanita(x))$
- “Hubungan orang tua dan anak berkebalikan”



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”
 $\forall x, y \ (Ibu(x) = y \Leftrightarrow Orangtua(y, x) \wedge Perempuan(y))$
- “Laki-laki dan perempuan adalah dua kelompok berbeda”
 $\forall x \ (Lelaki(x) \Leftrightarrow \neg Wanita(x))$
- “Hubungan orang tua dan anak berkebalikan”
 $\forall x, y \ (Orangtua(x, y) \Leftrightarrow Anak(y, x))$



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”
 $\forall x, y \ (Ibu(x) = y \Leftrightarrow Orangtua(y, x) \wedge Perempuan(y))$
- “Laki-laki dan perempuan adalah dua kelompok berbeda”
 $\forall x \ (Lelaki(x) \Leftrightarrow \neg Wanita(x))$
- “Hubungan orang tua dan anak berkebalikan”
 $\forall x, y \ (Orangtua(x, y) \Leftrightarrow Anak(y, x))$
- “Kakek-nenek (grandparents) adalah orang tua dari orang tua seseorang”



KBA dengan FOL: Hubungan Keluarga

- “Ibu adalah orangtua berjenis kelamin perempuan”
 $\forall x, y \ (Ibu(x) = y \Leftrightarrow Orangtua(y, x) \wedge Perempuan(y))$
- “Laki-laki dan perempuan adalah dua kelompok berbeda”
 $\forall x \ (Lelaki(x) \Leftrightarrow \neg Wanita(x))$
- “Hubungan orang tua dan anak berkebalikan”
 $\forall x, y \ (Orangtua(x, y) \Leftrightarrow Anak(y, x))$
- “Kakek-nenek (grandparents) adalah orang tua dari orang tua seseorang”
 $\forall x, y \ (Grandparents(x, y) \Leftrightarrow \exists z \ (Orangtua(x, z) \wedge Orangtua(z, y)))$



KBA dengan FOL: Persepsi dan Aksi pada Wumpus World

- Representasi hasil *percept* dari sensor:

$Percept([sBau, sAngin, sKilau, sBentur, sTeriak], t)$.

- $TELL(KB, Percept([Bau, Angin, Kilau, Kosong, Kosong], 5))$

- Untuk menentukan tindakan terbaik yang diambil:

$ASKVARS(KB, TindakanTerbaik(a, 5))$ memberikan $\{a/Ambil\}$

- Data “mentah” dari sensor perlu diolah (proses *perception*):

- $\forall sa, sk, sr, st, t \ (Percept([Bau, sa, sk, sr, st], t) \Rightarrow MenciumBau(t))$

- $\forall sb, sk, sr, st, t \ (Percept([sb, Angin, sk, sr, st], t) \Rightarrow MerasaHembus(t))$

- $\forall sb, sa, sr, st, t \ (Percept([sb, sa, Kilau, sr, st], t) \Rightarrow MelihatKilauan(t))$

- Tindakan “rational reflex” bisa dinyatakan sebuah kalimat, mis:

$\forall t \ (MelihatKilauan(t) \Rightarrow TindakanTerbaik(Ambil, t))$



- Keberadaan wumpus: $\forall t \text{ Di}(Wumpus, [2, 2], t)$
- Kondisi kamar berdasarkan proses perception:
 - $\forall k, t \text{ (Di}(\text{Agent}, k, t) \wedge \text{MerasaHembus}(t) \Rightarrow \text{KmrAngin}(k))$
- “Terdapat lubang jebakan pada kamar yang bersebelahan dengan kamar berangin”.
 - $\forall y \text{ (KmrAngin}(y) \Leftrightarrow [\exists x \text{ (Jebakan}(x) \wedge \text{Sebelahan}(x, y))])$
 di mana:
 $\forall x, y, a, b \text{ (Sebelahan}([x, y], [a, b]) \Leftrightarrow$
 $(x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1)))$



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Knowledge Engineering

- Proses merancang kalimat-kalimat KRL yang dengan tepat “merepresentasikan” dunia/masalah (d.k.l. membangun *KB*): **knowledge engineering**.
- Proses knowledge engineering:
 - *Identifikasi masalah*: serupa dengan PEAS pada perancangan agent;
 - *Knowledge acquisition*: mengumpulkan knowledge yang relevan;
 - *Pemilihan vocabulary*: memilih simbol predicate, function, constants;
 - *Encode pengetahuan umum tentang domain*: menghasilkan *axioms* dalam kalimat KRL;
 - *Encode deskripsi spesifik dari problem*: menggunakan kalimat KRL untuk merepresentasikan instance dari problem;
 - *Query KB*: mendapatkan jawaban dari inference procedure;
 - *Debug KB*: menyempurnakan KB, jika diperoleh jawaban yang salah.



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Instantiation

- **Ground term**: sebuah *term* tanpa variable, mis: *Ani*, *Ayah(Anto)*
- **Instantiation**: kalimat di mana variable diganti dengan ground term (diperoleh dengan mengaplikasikan *substitution*)
- $\text{SUBST}(\theta, \alpha)$: mengaplikasikan substitution θ pada kalimat α

Contoh

$$\alpha = \text{mahasiswa}(x, UI) \Rightarrow \text{pintar}(x)$$

$$\beta = \text{mahasiswa}(x, UI) \wedge \text{pintar}(x)$$

$$\theta = \{x / \text{Anto}\}$$

$\text{SUBST}(\theta, \alpha)$ menghasilkan *instantiation*:



Instantiation

- **Ground term**: sebuah *term* tanpa variable, mis: *Ani*, *Ayah(Anto)*
- **Instantiation**: kalimat di mana variable diganti dengan ground term (diperoleh dengan mengaplikasikan *substitution*)
- $\text{SUBST}(\theta, \alpha)$: mengaplikasikan substitution θ pada kalimat α

Contoh

$$\alpha = \text{mahasiswa}(x, UI) \Rightarrow \text{pintar}(x)$$

$$\beta = \text{mahasiswa}(x, UI) \wedge \text{pintar}(x)$$

$$\theta = \{x/ \text{Anto}\}$$

$\text{SUBST}(\theta, \alpha)$ menghasilkan *instantiation*:

$$\text{mahasiswa}(\text{Anto}, UI) \Rightarrow \text{pintar}(\text{Anto})$$

$\text{SUBST}(\theta, \beta)$ menghasilkan *instantiation*:



Instantiation

- **Ground term**: sebuah *term* tanpa variable, mis: *Ani*, *Ayah(Anto)*
- **Instantiation**: kalimat di mana variable diganti dengan ground term (diperoleh dengan mengaplikasikan *substitution*)
- $\text{SUBST}(\theta, \alpha)$: mengaplikasikan substitution θ pada kalimat α

Contoh

$\alpha = \text{mahasiswa}(x, UI) \Rightarrow \text{pintar}(x)$

$\beta = \text{mahasiswa}(x, UI) \wedge \text{pintar}(x)$

$\theta = \{x/ Anto\}$

$\text{SUBST}(\theta, \alpha)$ menghasilkan *instantiation*:

$\text{mahasiswa}(Anto, UI) \Rightarrow \text{pintar}(Anto)$

$\text{SUBST}(\theta, \beta)$ menghasilkan *instantiation*:

$\text{mahasiswa}(Anto, UI) \wedge \text{pintar}(Anto)$



Universal Instantiation

Sebuah kalimat dengan universal quantifier (\forall) meng-*entail* **semua** *instantiation*-nya:

$$\frac{\forall v \ \alpha}{SUBST(\{v/g\}, \alpha)}$$

untuk sembarang variable v dan ground term g .

Contoh

$\forall x \ (King(x) \wedge Greedy(x) \Rightarrow Evil(x))$ meng-entail:

$$King(John) \wedge Greedy(John) \Rightarrow Evil(John)$$

$$King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$$

$$King(Father(John)) \wedge Greedy(Father(John)) \Rightarrow Evil(Father(John))$$

\vdots



Existential Instantiation

Untuk sembarang variable v , kalimat α dan constant k yang **tidak muncul di knowledge base**:

$$\frac{\exists v \ \alpha}{SUBST(\{v/k\}, \alpha)}$$

Contoh

$\exists x \ (Crown(x) \wedge OnHead(x, John))$ meng-entail:

$Crown(C_1) \wedge OnHead(C_1, John)$

dengan syarat C_1 adalah *constant symbol* yang **baru**, disebut *Skolem constant*



Penggunaan Universal vs. Existential Instantiation

Penggunaan \forall -Instantiation

- Universal instantiation bisa digunakan **berkali-kali** untuk menambahkan kalimat baru.
- KB yang baru *logically equivalent* dengan yang lama.

Penggunaan \exists -Instantiation

- Existential instantiation cukup digunakan **sekali** untuk **menggantikan** kalimat existential.
- KB yang baru **tidak** *logically equivalent* dengan yang lama, tetapi *satisfiable* jhj KB yang lama juga *satisfiable* \rightarrow **inferentially equivalent**



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Reduksi KB FOL ke KB PL: Contoh

Andaikan *KB* berisi kalimat-kalimat berikut:

$\forall x \ (King(x) \wedge Greedy(x) \Rightarrow Evil(x))$

$\forall y \ Greedy(y)$

King(John)

Brother(Richard, John)

Jika kita terapkan \forall -instantiation menggunakan semua kemungkinan ground-term substitution dari vocabulary KB, didapatkan KB sbb.:

$King(John) \wedge Greedy(John) \Rightarrow Evil(John)$

$King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$

Greedy(John)

Greedy(Richard)

King(John)

Brother(Richard, John)

KB yang baru dikatakan **propositionalized** dengan proposition symbol-nya:

King(John), *Greedy(John)*, *Evil(John)*, *King(Richard)*, etc.



Inference FOL menggunakan inference PL

- Ide dasar: ubah ($KB + query$) dari FOL menjadi PL, lalu gunakan PL inference procedure (mis: *resolution*).



Inference FOL menggunakan inference PL

- Ide dasar: ubah ($KB + query$) dari FOL menjadi PL, lalu gunakan PL inference procedure (mis: *resolution*).

Masalah: dengan adanya function, jumlah ground term menjadi infinite

Greedy(Father(John))

Greedy(Father(Father(John)))

Greedy(Father(Father(Father(John)))), dst.



Inference FOL menggunakan inference PL

- Ide dasar: ubah $(KB + query)$ dari FOL menjadi PL, lalu gunakan PL inference procedure (mis: *resolution*).

Masalah: dengan adanya function, jumlah ground term menjadi infinite

Greedy(Father(John))

Greedy(Father(Father(John)))

Greedy(Father(Father(Father(John)))), dst.

- Teorema Herbrand (1930): jika $FOL\ KB \models \alpha$, ada sebuah **finite subset** $PL\ KB \models \alpha$.
- Ide dasar: For $n = 0$ to ∞
 Buat propositional KB_n dengan depth- n ground term
 Periksa apakah $KB_n \models \alpha$



Inference FOL menggunakan inference PL

- Ide dasar: ubah $(KB + query)$ dari FOL menjadi PL, lalu gunakan PL inference procedure (mis: *resolution*).

Masalah: dengan adanya function, jumlah ground term menjadi infinite

$Greedy(Father(John))$

$Greedy(Father(Father(John)))$

$Greedy(Father(Father(Father(John))))$, dst.

- Teorema Herbrand (1930): jika $FOL\ KB \models \alpha$, ada sebuah **finite subset** $PL\ KB \models \alpha$.
- Ide dasar: For $n = 0$ to ∞
 Buat propositional KB_n dengan depth- n ground term
 Periksa apakah $KB_n \models \alpha$

Masalah (lagi!)

Kalau α dapat di-*entail*, inferensi FOL **complete**.

Kalau α tidak dapat di-*entail*, tidak dapat dibuktikan (terjadi infinite loop).



Masalah lain dengan *propositionalization*

- **Propositionalization** menghasilkan banyak kalimat yang tidak relevan.
- Contohnya, dari KB berikut:

$\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$

$\forall y \text{ Greedy}(y)$

$\text{King}(\text{John})$

$\text{Brother}(\text{Richard}, \text{John})$

cukup jelas bahwa $\text{Evil}(\text{John})$ berlaku, namun propositionalization menghasilkan juga:

$\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$

$\text{Greedy}(\text{Richard})$

yang tidak relevan.

- Dengan p buah predicate k -ary dan n constant, ada $p \times n^k$ instantiation!



Generalized Modus Ponens

Inference rule GMP

$$\frac{p_1', p_2', \dots, p_n', (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{q\theta}$$

di mana $p_i'\theta = p_i\theta$ untuk semua i

$$\begin{array}{ll} p_1' = King(John) & p_1 = King(x) \\ p_2' = Greedy(y) & p_2 = Greedy(x) \\ \theta = \{x/John, y/John\} & q = Evil(x) \\ q\theta = Evil(John) & \end{array}$$

- GMP dengan KB yang berisi **definite clauses** (seperti Horn clause pada PL): $p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q$
- Semua variable diasumsikan **universally quantified**
- GMP adalah hasil **lifting** MP: “mengangkat” inference rule PL ke FOL.
 - Hanya melakukan *substitution* yang dibutuhkan oleh *inference*.



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - **Unification**
 - Forward chaining & Backward Chaining
 - Resolution



Unification

Isi KB

$$\forall x \ (King(x) \wedge Greedy(x) \Rightarrow Evil(x))$$
$$\forall y \ Greedy(y)$$
$$King(John)$$
$$Brother(Richard, John)$$

Inference bahwa $KB \models Evil(John)$ bisa langsung disimpulkan jika kita bisa mencari *substitution* θ sehingga $King(x)$ dan $Greedy(x)$ bisa di-unifikasi dengan $King(John)$ dan $Greedy(y)$.

Contoh: $\theta = \{x/John, y/John\}$

Definisi unification

$$UNIFY(\alpha, \beta) = \theta \text{ jika } SUBST(\alpha, \theta) = SUBST(\beta, \theta)$$


Contoh unification

α	β	θ
$Sayang(Anto, x)$	$Sayang(Anto, Ani)$	



Contoh unification

α	β	θ
$Sayang(Anto, x)$	$Sayang(Anto, Ani)$	$\{x/Ani\}$
$Sayang(Anto, x)$	$Sayang(y, Ani)$	



Contoh unification

α	β	θ
$Sayang(Anto, x)$	$Sayang(Anto, Ani)$	$\{x/Ani\}$
$Sayang(Anto, x)$	$Sayang(y, Ani)$	$\{x/Ani, y/Anto\}$
$Sayang(Anto, x)$	$Sayang(y, Ibu(y))$	



Contoh unification

α	β	θ
$Sayang(Anto, x)$	$Sayang(Anto, Ani)$	$\{x/Ani\}$
$Sayang(Anto, x)$	$Sayang(y, Ani)$	$\{x/Ani, y/Anto\}$
$Sayang(Anto, x)$	$Sayang(y, Ibu(y))$	$\{y/Anto, x/Ibu(Anto)\}$
$Sayang(Anto, x)$	$Sayang(x, Ani)$	



Contoh unification

α	β	θ
$Sayang(Anto, x)$	$Sayang(Anto, Ani)$	$\{x/Ani\}$
$Sayang(Anto, x)$	$Sayang(y, Ani)$	$\{x/Ani, y/Anto\}$
$Sayang(Anto, x)$	$Sayang(y, Ibu(y))$	$\{y/Anto, x/Ibu(Anto)\}$
$Sayang(Anto, x)$	$Sayang(x, Ani)$	<i>fail</i>

- **Standardized apart** variable menghilangkan *overlap*, mis:
 $Sayang(x_{101}, Ani)$
- $UNIFY(Sayang(Anto, x), Sayang(y, z)) = \{y/Anto, x/z\}$
 - Bagaimana dengan $\{y/Anto, x/Ani, z/Ani\}$?
 - $\{y/Anto, x/z\}$ disebut **most general unifier**. (MGU)



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution



Contoh knowledge base

Versi bahasa Inggris

“The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.”

Buktikan bahwa Col. West adalah criminal!



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:
 $Missile(x) \Rightarrow Weapon(x)$



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:
 $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as "hostile":



- 

Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:
 $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as "hostile":
 $Enemy(x, America) \Rightarrow Hostile(x)$
- West, who is American ...



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:
 $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as "hostile":
 $Enemy(x, America) \Rightarrow Hostile(x)$
- West, who is American ...
 $American(West)$



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
(secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:
 $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as "hostile":
 $Enemy(x, America) \Rightarrow Hostile(x)$
- West, who is American ...
 $American(West)$
- The country Nono, an enemy of America ...



Representasi ke FOL

- ... it is a crime for an American to sell weapons to hostile nations:
 $American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
 (secara implisit, variabel pada definite clause dikuantifikasi universal)
- Nono ... has some missiles:
 $\exists x Owns(Nono, x) \wedge Missile(x)$
 $Owns(Nono, M_1)$ dan $Missile(M_1)$ (Skolemization)
- ... all of its missiles were sold to it by Colonel West
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- Missiles are weapons:
 $Missile(x) \Rightarrow Weapon(x)$
- An enemy of America counts as "hostile":
 $Enemy(x, America) \Rightarrow Hostile(x)$
- West, who is American ...
 $American(West)$
- The country Nono, an enemy of America ...
 $Enemy(Nono, America)$

Perhatikan: semua kalimat KB ini berbentuk **definite clause**.



Forward chaining pada FOL dengan GMP

- Mirip dengan forward chaining pada PL
- Mulai dari fakta yang diketahui (clause tanpa premise), mis:
 $Owns(Nono, M_1), Missile(M_1)$
- “Aktifkan” (*trigger*) rule yang premise-nya terpenuhi →
tambahkan kesimpulan rule ke KB, mis:
 $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
 - “Cocokkan” premise setiap rule dengan fakta-fakta yang diketahui → **pattern matching**
- Ulangi sampai *query* terbukti, atau tidak ada fakta baru yang bisa ditambahkan ke KB.



Contoh Forward Chaining FOL

American(West)

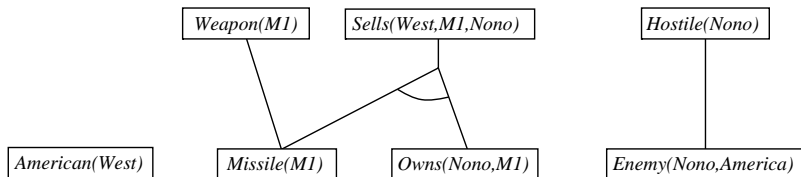
Missile(M1)

Owns(Nono,M1)

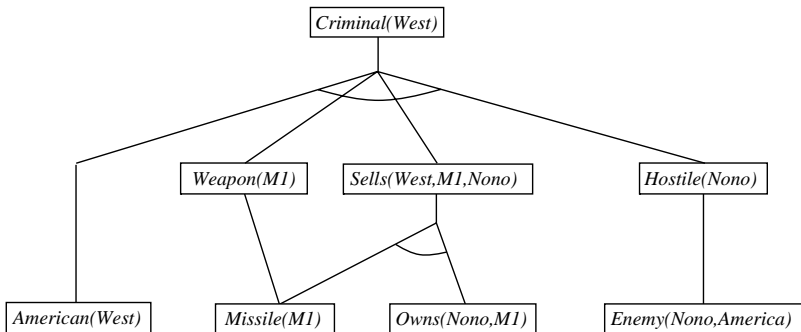
Enemy(Nono,America)



Contoh Forward Chaining FOL



Contoh Forward Chaining FOL



Backward chaining pada FOL dengan GMP

- Berangkat dari gol, melalui rule-rule yang ada, cari fakta-fakta yang dapat digunakan untuk mendukung pembuktian.
- Efek berantai muncul dari upaya untuk pemenuhan rule-rule dari gol tersebut.

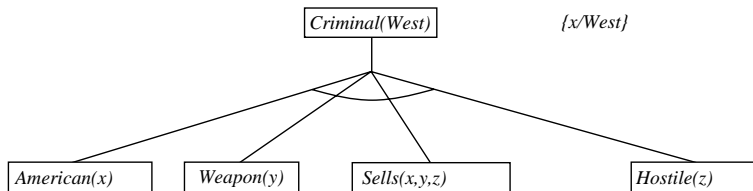


Contoh Backward Chaining FOL

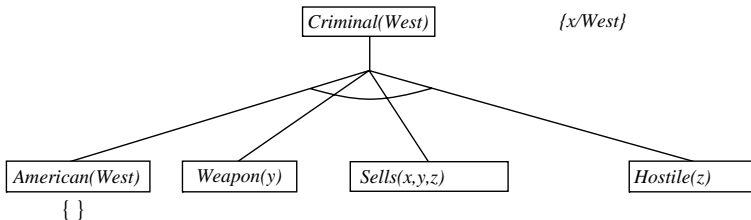
Criminal(West)



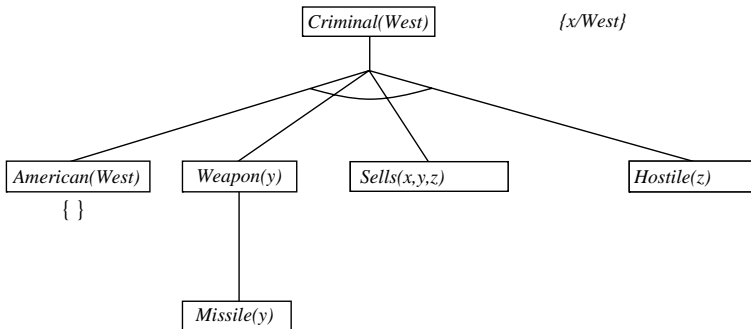
Contoh Backward Chaining FOL



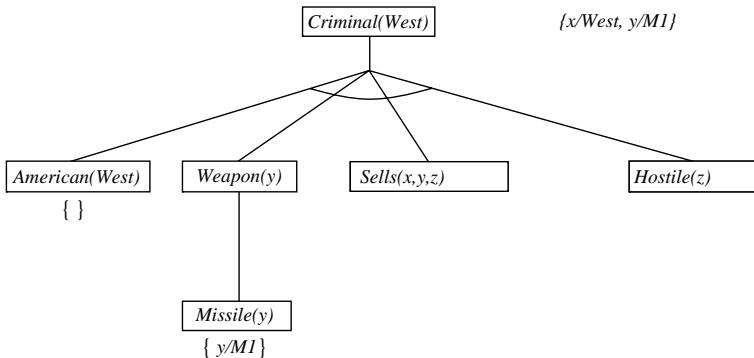
Contoh Backward Chaining FOL



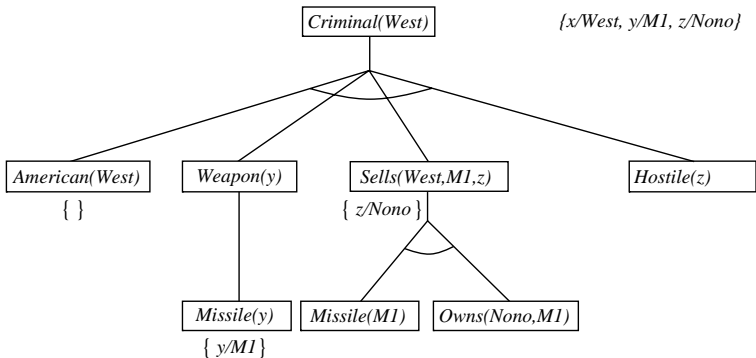
Contoh Backward Chaining FOL



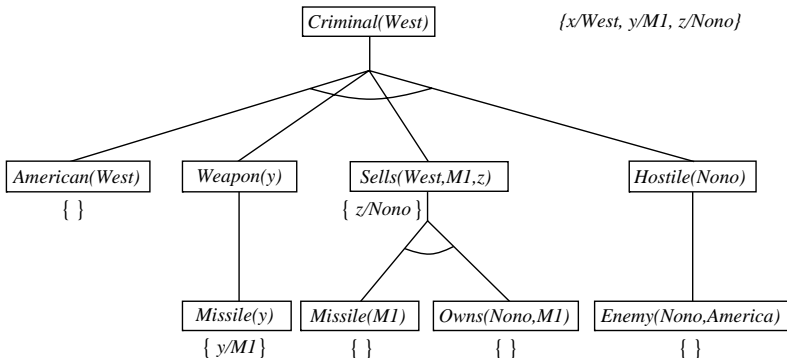
Contoh Backward Chaining FOL



Contoh Backward Chaining FOL



Contoh Backward Chaining FOL



Logic Programming

Algorithm = Logic + Control

- **Prolog**: bahasa Logic Programming
- Dasar: backward chaining pada Horn clause
- Program berupa himpunan *clause*:

$$\text{positive_literal} \text{ :- literal}_1, \dots \text{literal}_n.$$

Contoh

$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
 $\text{criminal}(X) \text{ :- american}(X), \text{weapon}(Y), \text{sells}(X, Y, Z), \text{hostile}(Z).$

- Aspek “non-logic” dari Prolog:
 - BC berjalan atas-ke-bawah (rule) dan kiri-ke-kanan (clause)
 - Ada predikat untuk aritmetika: $X \text{ is } Y * Z + 3$
 - Ada clause dengan “efek samping”: I/O



Outline

- 1 FOL: Syntax & Semantics
 - Model, Simbol, & Interpretasi
 - Term & Kalimat FOL
- 2 KBA dengan FOL
 - Substitution
 - Contoh KBA dengan FOL
 - Knowledge Engineering
- 3 Inference untuk FOL**
 - Inference Rules untuk Quantifiers
 - Reduksi FOL Inference ke PL Inference
 - Unification
 - Forward chaining & Backward Chaining
 - Resolution**



Resolution pada FOL (lifting resolution PL)

■ Binary resolution rule:

$$\frac{\ell_1 \vee \dots \vee \ell_k, \quad m_1 \vee \dots \vee m_n}{(\ell_1 \vee \dots \vee \ell_{i-1} \vee \ell_{i+1} \vee \dots \vee \ell_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)\theta}$$

di mana $\text{UNIFY}(\ell_i, \neg m_j) = \theta$.

Contoh:

$$\frac{\neg Kaya(x) \vee Sedih(x) \quad Kaya(Anto)}{Sedih(Anto)}$$

di mana $\theta = \{x/Anto\}$

- **Factoring**: reduksi dua literal yang unifiable pada sebuah clause menjadi satu literal + terapkan MGU pada literal hasil factoring dan literal lainnya.

Contoh: Factoring dari $P(b, x) \vee P(x, x) \vee Q(x, c)$ adalah $P(b, b) \vee Q(b, c)$ dengan MGU $\theta = \{x/b\}$

Untuk menunjukkan $KB \models \alpha$: terapkan resolution pada CNF dari $KB \wedge \neg \alpha$.

- Resolution adalah **refutation-complete**: jika sekumpulan kalimat FOL *unsatisfiable*, maka penerapan resolution akan menghasilkan *contradiction* dari sekumpulan kalimat FOL tersebut.



Mengubah Kalimat FOL ke CNF

“Everyone who loves all animals is loved by someone.”

$$\forall x [\forall y \text{ Animal}(y) \implies \text{Loves}(x, y)] \implies [\exists y \text{ Loves}(y, x)]$$

- 1 Eliminasi implikasi dan biimplikasi

$$\forall x [\neg \forall y \neg \text{Animal}(y) \vee \text{Loves}(x, y)] \vee [\exists y \text{ Loves}(y, x)]$$

- 2 Pindahkan \neg ke “dalam”: $\neg \forall x p \equiv \exists x \neg p$, $\neg \exists x p \equiv \forall x \neg p$:

$$\forall x [\exists y \neg (\neg \text{Animal}(y) \vee \text{Loves}(x, y))] \vee [\exists y \text{ Loves}(y, x)]$$

$$\forall x [\exists y \neg \neg \text{Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists y \text{ Loves}(y, x)]$$

$$\forall x [\exists y \text{ Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists y \text{ Loves}(y, x)]$$

- 3 **Standardize** variables: setiap quantifier variable-nya beda

$$\forall x [\exists y \text{ Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists z \text{ Loves}(z, x)]$$

- 4 **Skolemize**: eliminasi existential quantifier. $\exists x$ dieliminasi dengan menggunakan **Skolem function** dari universally quantified variables sebelumnya: $\forall x [\text{Animal}(F(x)) \wedge \neg \text{Loves}(x, F(x))] \vee \text{Loves}(G(x), x)$

- 5 Buang universal quantifiers:

$$[\text{Animal}(F(x)) \wedge \neg \text{Loves}(x, F(x))] \vee \text{Loves}(G(x), x)$$

- 6 Distribusi \wedge over \vee :

$$[\text{Animal}(F(x)) \vee \text{Loves}(G(x), x)] \wedge [\neg \text{Loves}(x, F(x)) \vee \text{Loves}(G(x), x)]$$



Contoh pembuktian dengan resolution

