Artificial Intelligence: Search Methods for Problem Solving

Deduction as Search
First Order Logic

A First Course in Artificial Intelligence: Chapter 11

Deepak Khemani
Department of Computer Science & Engineering
IIT Madras

What do you know?

An autonomous agent

operating in a dynamic world

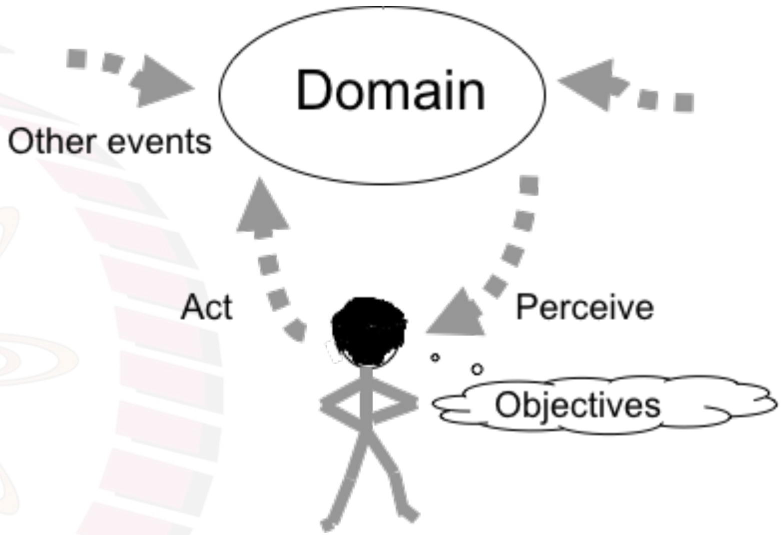
creates a representation of the world

senses the world around it

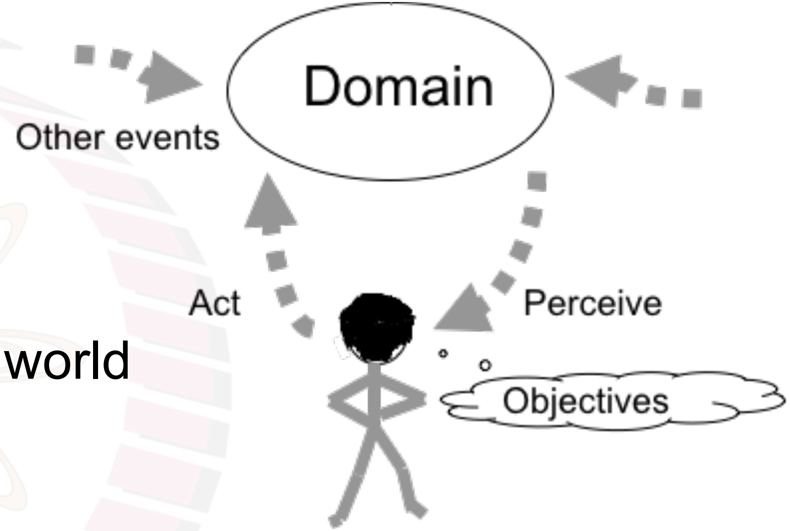
makes plans to achieve its goals

monitors the execution of those plans

and constantly updates what it knows



What else do you know?



An autonomous agent

operating in a dynamic world

...needs to make inferences about its world

This week,

Representation and Reasoning in Logic

Another string in the bow of a problem solving agent

Knowledge and Reasoning – necessary for intelligence

What does the agent know

and

what else does the agent know as a consequence of what it knows?

A preview of the course
AI: Knowledge Representation & Reasoning

The Greek Syllogism

Given

All men are mortal

and

Socrates is a man

conclude

Socrates is mortal

The Socratic Argument

What *else* do you know?

True / Given

Snigdha's mother is Sneha
Snigdha's father is Salil
Sameer's mother is Sneha
Sameer's father is Salil
Salil's sister is Arushi
Arnav's mother is Arushi
Arnav tested positive for Covid

+

Knowledge of gender
and relationships

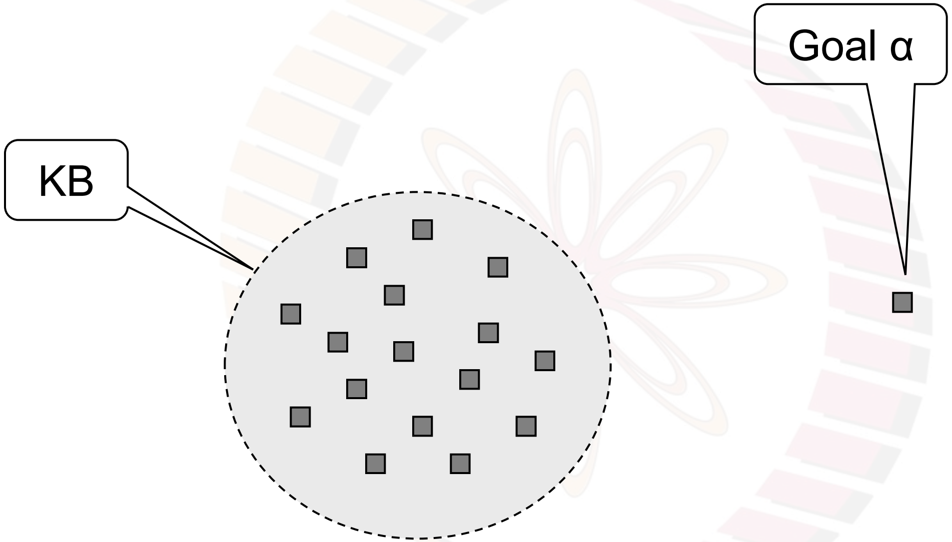
Necessarily true

Snigdha is Sneha's daughter
Snigdha is Sameer's sister
Snigdha is Arushi's niece
•
•
Salil is not Arushi's sister
Salil is not Sneha's sister
Arushi is not Sneha's daughter
•
•
There exists a woman who is Snigdha and Arnav's grandma
•
•

Possibly true

Sneha loves Snigdha
Snigdha loves Sneha
Snigdha loves idli-sambhar
Snigdha goes to school
Arushi is older than Salil
Snigdha is a Arnav's friend
Arnav has Covid
•
•

Deduction: The Goal



Given a knowledge base (KB)
which is set of sentences assumed to be true,

is a given (query) sentence **α** *necessarily* true?

Representation

Semiotics: A symbol is something that stands for something else

Examples:

- The "number" seven can be represented in many different ways.
- Road signs – curves, pedestrians, schools, U-turns, eating places…

All languages are semiotic systems

Biosemiotics: How complex behaviour
emerges when simple systems
interact with each other through signs

Reasoning

The manipulation of symbols in a *meaningful* manner.

- we can *only* manipulate *symbols*
- the *meaning* or *truth value* is in only our minds

Maths is replete with *algorithms* we use –

- Addition and multiplication of multi-digit numbers
- Long division
- Solving systems of linear equations
- Fourier transforms, convolution…

The Syllogism

The Greek syllogism embodies the notion of **formal logic**

An argument is valid if it conforms to a **valid form**

All men are mortal
Socrates is a man
Socrates is mortal

All cities are congested
Chennai is a city
Chennai is congested

All politicians are honest
Sambit is a politician
Sambit is honest

All Xs are Y
C is X
C is Y

If premises are true
Then conclusion is true

Formal Logic

Logic is a formal system

Logical reasoning is only concerned ONLY with the FORM of the argument, and not with CONTENT.

If the form is valid AND If the antecedents are true
THEN the conclusion is true.

Thus, the conclusion holds only if the antecedents are true.

Logic does not concern itself with the truth of antecedents
OR

what the sentences are talking about (content).

Logics

Epistemic Logics

Temporal Logics

Modal Logics

Classical two valued logics

Second Order Logic

First Order Logic

Propositional Logic

Event Calculus / Situation Calculus

Default Logics

Probabilistic reasoning

Horn Clauses

Description Logics

Fuzzy logics

Constraint logic programming

Rough sets

Qualitative reasoning

Classical Logics

- Mathematical Logic
- Once a statement is true it remains true
  - for example the Pythagoras Theorem
  - no notion of time, events, or statements becoming false
- The RHS of a rule has only ADD actions

- Non-monotonic logics
  - Fluents: sentences which can flip their truth value
  - Knowledge and Belief
  - Default Reasoning
  - Inheritance

AI: Knowledge Representation and Reasoning

Rules of Inference

Rules of inference in logic are similar to the rules in production systems

In classical logic the RHS of a rule has only one action
- add a new sentence
  - to the knowledge base (KB)

In logic the rules are viewed as
Premises → Conclusion
or Antecedents → Consequent

Rules do have names
but the names are not used directly

p rule
pattern 1
pattern 2
.
.
→
actions...

Some common rules of inference

From $\alpha \supset \beta$
and $\underline{\alpha}$
Infer β
Modus Ponens (MP)

From $\alpha \supset \beta$
and $\underline{\sim\beta}$
Infer $\sim\alpha$
Modus Tollens (MT)

From α
and $\underline{\beta}$
Infer $\alpha \wedge \beta$
Conjunction (C)

From $(\alpha \supset \beta) \wedge (\gamma \supset \delta)$
and $\underline{\alpha \vee \gamma}$
Infer $\beta \vee \delta$
Constructive Dilemma (CD)

From $\underline{\alpha}$
Infer $\alpha \vee \beta$
Addition (A)

From $\underline{\alpha \wedge \beta}$
Infer α
Simplification (S)

From $(\alpha \supset \beta) \wedge (\gamma \supset \delta)$
and $\underline{\sim\beta \vee \sim\delta}$
Infer $\sim\alpha \vee \sim\gamma$
Destructive Dilemma (DD)

From $\alpha \supset \beta$
and $\underline{\beta \supset \gamma}$
Infer $\alpha \supset \gamma$
Hypothetical Syllogism (HS)

From $\alpha \vee \beta$
and $\underline{\sim\alpha}$
Infer β
Disjuncntive Syllogism (DS)

MoveGen in Logical Reasoning

KB: A set of sentences in some language

Pattern: A subset of the sentences

Augmenting the KB in a piecewise fashion

Move: Pattern → Inference

Premises → Consequent

Rules of Substitution

A rule of substitution allows one to replace one sentence with another. This is possible when one sentence is logically equivalent to another. For example

$$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$$

If the above equivalence is a tautology, then the sentence $(\alpha \supset \beta)$ will always take the same truth value as the sentence $(\neg\alpha \vee \beta)$. We can verify that the equivalence is a tautology by constructing a truth table.

α	β	$(\alpha \supset \beta)$	$\neg\alpha$	$(\neg\alpha \vee \beta)$	$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$
true	true	true	false	true	true
false	true	true	true	true	true
true	false	false	false	false	true
false	false	true	true	true	true

Common rules of substitution

$\alpha \equiv (\alpha \vee \alpha)$	idempotence of $\vee$
$\alpha \equiv (\alpha \wedge \alpha)$	idempotence of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	DeMorgan's Law
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	DeMorgan's Law
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$
$(\alpha \supset \beta) \equiv (\neg\beta \supset \neg\alpha)$	contrapositive
$(\alpha \supset \beta) \equiv (\neg\alpha \vee \beta)$	implication
$(\alpha \equiv \beta) \equiv ((\alpha \supset \beta) \wedge (\beta \supset \alpha))$	equivalence
$((\alpha \wedge \beta) \supset \gamma) \equiv (\alpha \supset (\beta \supset \gamma))$	exportation
$((\alpha \supset \beta) \wedge (\alpha \supset \neg\beta)) \equiv \neg\alpha$	absurdity

$(\alpha \vee \text{true}) \equiv \text{true}$

$(\alpha \vee \text{false}) \equiv \alpha$

$(\alpha \wedge \text{true}) \equiv \alpha$

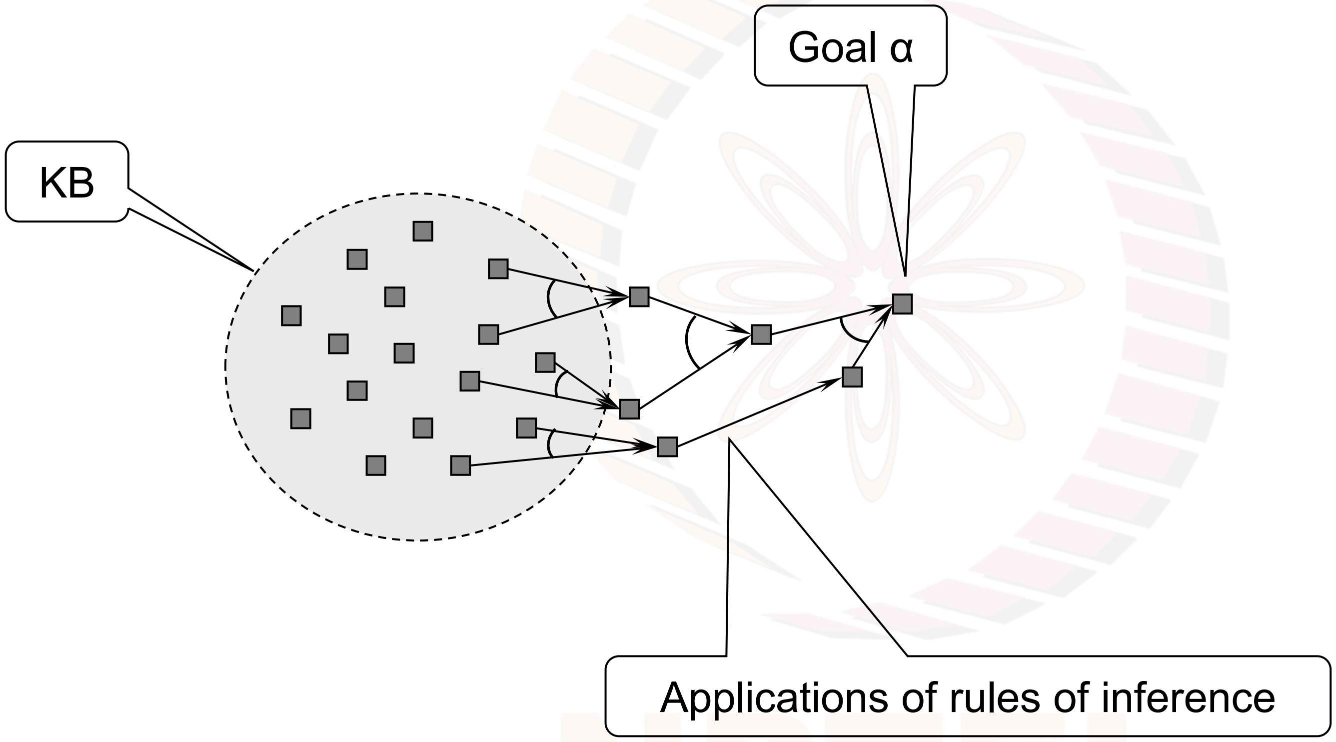
$(\alpha \wedge \text{false}) \equiv \text{false}$

$(\alpha \wedge \neg\alpha) \equiv \text{false}$

$(\alpha \vee \neg\alpha) \equiv \text{true}$

$\alpha \equiv \neg(\neg\alpha)$

Deduction: Proof



Provability: KB ⊢ α

Finding Proofs

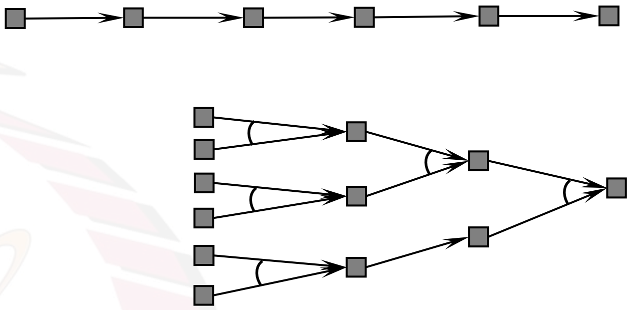
The proof is the end product,
that is a justification of the sentence α being true

It represents a chain of inferences linking
the given facts
to the desired goal

Proofs are found by a process of search
Remember the 4-colour theorem?

The irrelevant inferences are discarded
and only the final proof remains

Forward Reasoning



A data driven chain of inferences

From facts to goals

Forward Chaining

Pick a rule with matching facts

Add the consequent to the KB

Till the goal is added to the KB

The key question is

which rule and what facts?

Logic: Semantics

Denotation: What does a sentence stand for?

Truth Functional: Is the sentence *true*?

Axioms / Premises (KB): Assumed to be *true*.

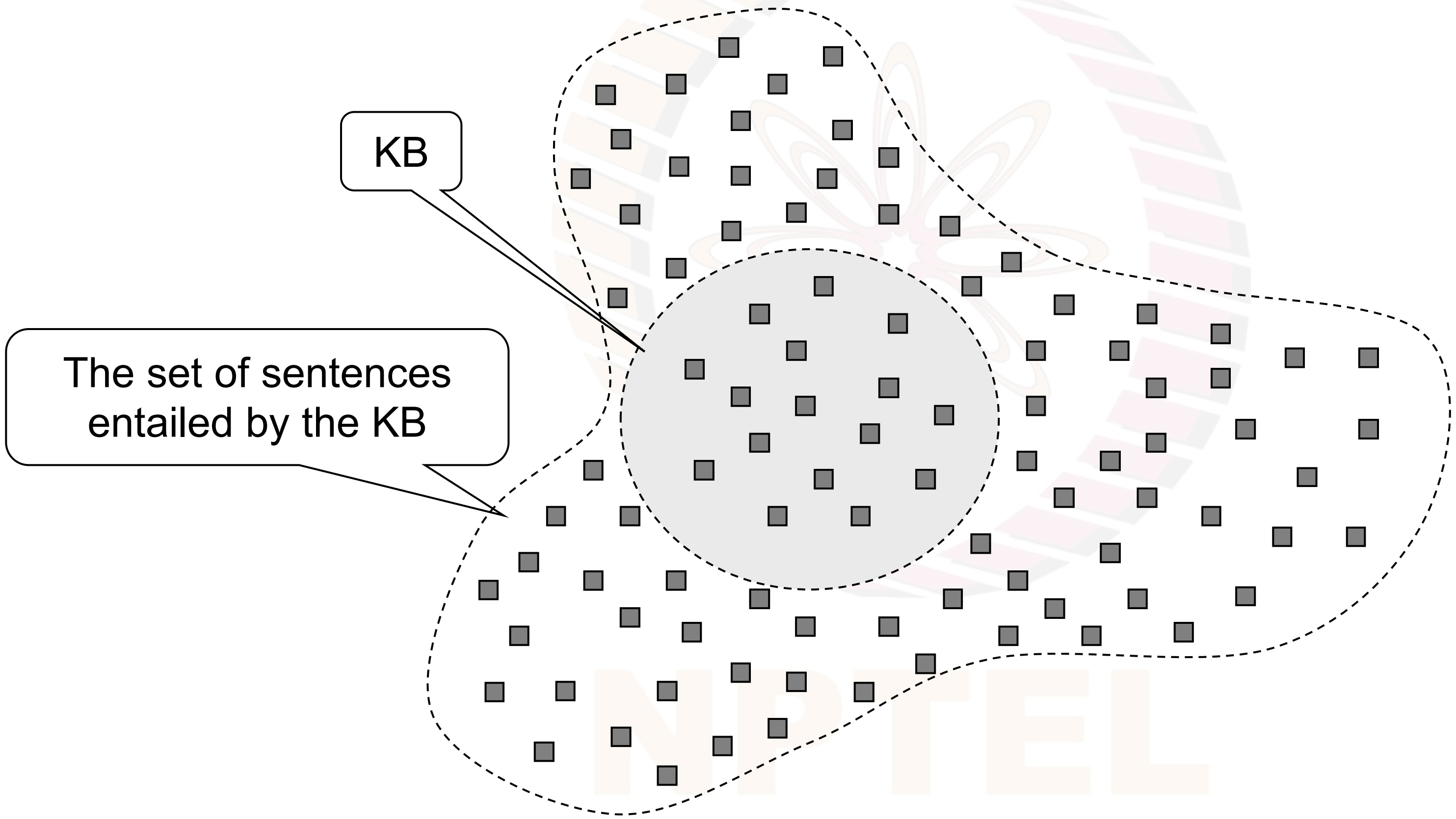
KB is *true* iff every sentence in the KB is *true*.

Entailment: A sentence α is said to be entailed by a set of sentences *S*/KB if the sentence is necessarily *true* whenever S/KB is *true*

$$\text{KB} \models \alpha$$

We also say that α is *true* (given the KB)

The set of true statements



Soundness and Completeness

Given a knowledge base and a reasoning algorithm –

Entailment: which other sentences in the language are necessarily true?

Proof: which other sentences in the language can one produce by the reasoning algorithm?

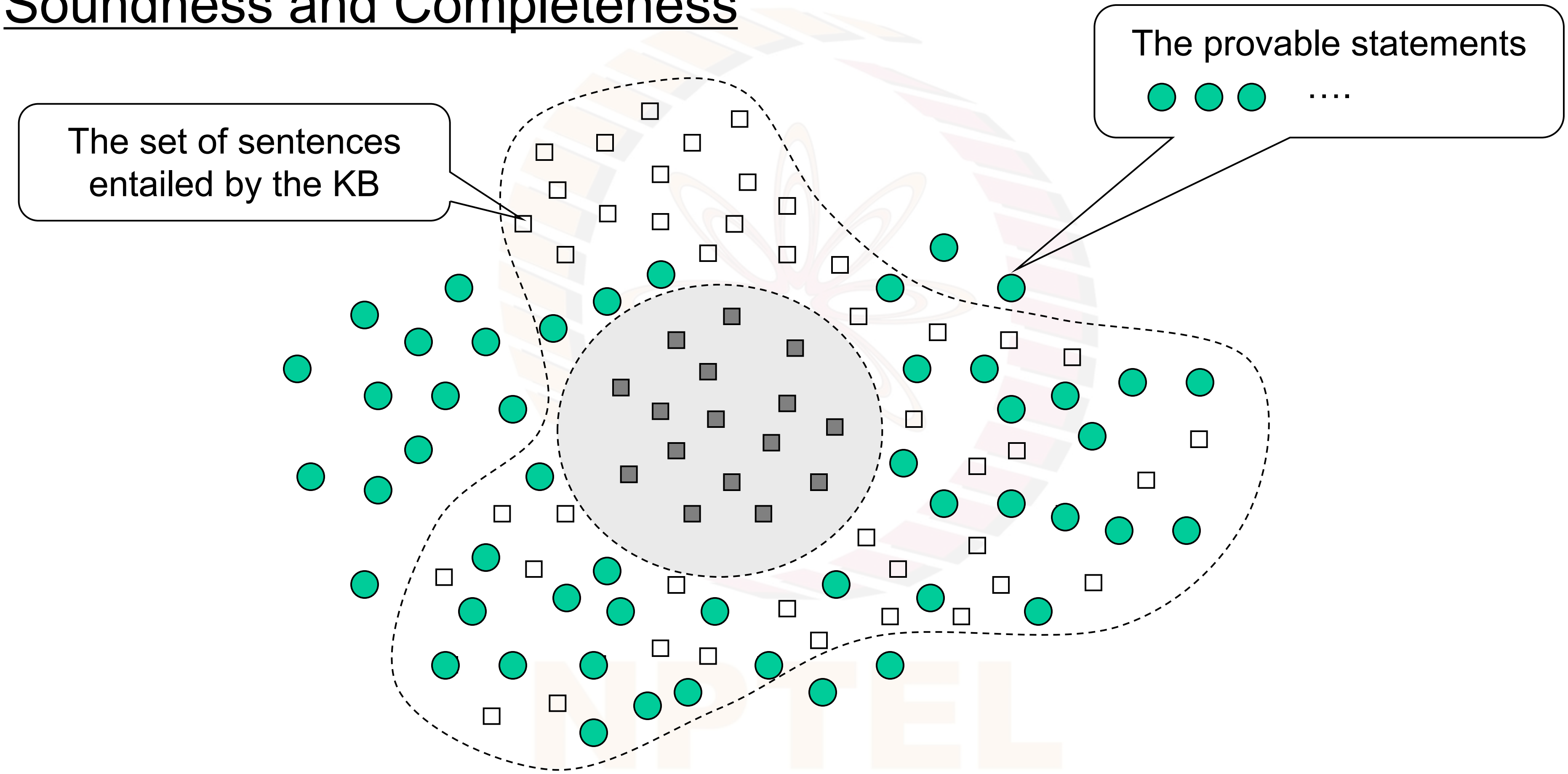
Soundness (of the reasoning algorithm):

A logic is sound
if **only** true statements in the language can be proved

Completeness (of the reasoning algorithm):

A logic is complete
if **all** true statements in the language can be proved

Soundness and Completeness



First Order Logic (FOL): Syntax

The *logical* part of the vocabulary

- Symbols that stand for connectives or operators
  - "∧", "∨", "~", and "⊃"…
- Brackets "(", ")", "{", "}"…
- The constant symbols "⊥" and "⊤".
- A set of variable symbols $V = \{v_1, v_2, v_3, \dots\}$
  - commonly used $\{x, y, z, x_1, y_1, z_1, \dots\}$
- Quantifiers: "∀" read as "for all", and "∃" read as "there exists".

  The former is the *universal quantifier*

  and the latter the *existential quantifier*.
- The symbol "=" read as "equals".

FOL Syntax (contd)

The non-logical part of *FOL vocabulary* constitutes of three sets.

- A set of predicate symbols P = {P_1, P_2, P_3, …}. We also use the symbols {*P*, *Q*, R, …}. More commonly we use words like “Man”, “Mortal”, “GreaterThan”. Each symbol has an arity associated with it.
- A set of function symbols F = {f_1, f_2, f_3, …}. We commonly used the symbols {f, g, h…} or words like “Successor” and “Sum”. Each function symbol has an arity that denotes the number of argument it takes.
- A set of constant symbols C = {c_1, c_2, c_3, …}. We often used symbols like “0”, or “Socrates”, or “Darjeeling” that are meaningful to us.

The three sets define a specific language L(*P*,*F,C*).

Atomic Formulas of L(P,F,C)

The set of formulas is defined using terms and predicate symbols. By default the logical symbols "$\bot$" and "$\top$" are also formulas. The set of well formed formulas *F* of L(*P*,*F*,*C*) is defined as follows.

Atomic formulas $\mathcal{A}$

$\bot \in \mathcal{A}$

$\top \in \mathcal{A}$

If $t_1, t_2 \in \mathfrak{I}$ then $(t_1=t_2) \in \mathcal{A}$

If $t_1, t_2, \ldots, t_n \in \mathfrak{I}$ and
$P \in \mathrm{P}$ is an n-place predicate symbol
then $P(t_1,t_2,\ldots,t_n) \in \mathcal{A}$

Formulas of L(P,F,C)

The set of formulas of L(P,F,C) $\mathcal{F}$ is defined as follows

If $\alpha \in \mathcal{A}$ then $\alpha \in \mathcal{F}$

If $\alpha \in \mathcal{F}$ then $\sim\alpha \in \mathcal{F}$

If $\alpha, \beta \in \mathcal{F}$ then $(\alpha \wedge \beta) \in \mathcal{F}$

If $\alpha, \beta \in \mathcal{F}$ then $(\alpha \vee \beta) \in \mathcal{F}$

If $\alpha, \beta \in \mathcal{F}$ then $(\alpha \supset \beta) \in \mathcal{F}$

Universal and Existential Quantifiers

If $\alpha \in \mathcal{F}$ and $x \in V$ then $\forall x\ (\alpha) \in \mathcal{F}$

$\forall x\ (\alpha)$ is read as "for all x (α)"

If $\alpha \in \mathcal{F}$ and $x \in V$ then $\exists x\ (\alpha) \in \mathcal{F}$

$\exists x\ (\alpha)$ is read as "there exists x (α)"

We will also use the notation (forall (x) (α)) and (exists (x) (α)) as given in the book Artificial Intelligence by Eugene Charniak and Drew McDermott.

Makes representation for use in programs simpler.

List notation

Standard mathematical notation

1. $\forall x\ (Man(x) \supset Human(x))$: all men are human beings
2. $Happy(suresh) \vee Rich(suresh)$: Suresh is rich or happy
3. $\forall x\ (CitrusFruit(x) \supset \neg Human(x))$: all citrus fruits are non-human
4. $\exists x\ (Man(x) \wedge Bright(x))$: some men are bright

List notation (a la Charniak & McDermott, "Artificial Intelligence")

1.(forall (x) (if (man x) (human x)))

2.(or (happy suresh) (rich suresh))

3.(forall (x) (if (citrusFruit x) (not (human x))))

4.(exists (x) (and (man x) (bright x)))

FOL: Rules of Inference

The propositional logic rules we saw earlier are valid in *FOL* as well. In addition we need new rules to handle quantified statements. The two commonly used rules of inference are,

$$\frac{\forall x\ P(x)}{P(a)} \quad \text{where } a \in C \qquad \text{Universal Instantiation (UI)}$$

$$\frac{P(a)}{\exists x\ P(x)} \quad \text{where } a \in C \qquad \text{Generalization}$$

Examples:

$$\frac{\forall x\ (Man(x) \supset Mortal(x))}{(Man(Socrates) \supset Mortal(Socrates))}$$

$$\frac{(Man(Socrates) \supset Mortal(Socrates))}{\exists x\ (Man(x) \supset Mortal(x))}$$

Semantics (Propositional Logic)

Atomic sentences in Propositional Logic can stand for **anything**. Consider,

Alice likes mathematics and she likes stories. If she likes mathematics she likes algebra. If she likes algebra and likes physics she will go to college. She does not like stories or she likes physics. She does not like chemistry and history.

Encoding: P = Alice likes mathematics. Q = Alice likes stories. R = Alice likes algebra. S = Alice likes physics. T = Alice will go to college. U = Alice likes chemistry. V = Alice likes history.

Then the given facts are,

$(P \wedge Q)$

$(P \supset R)$

$((R \wedge S) \supset T)$

$(\sim Q \vee S)$

$(\sim U \wedge \sim V)$

If the above sentences are true is it necessarily true that "Alice will go to college"?

That is " Is T *true*?"

Proofs in Propositional Logic

1.	$(P \wedge Q)$	premise	*Alice likes mathematics and she likes stories.*
2.	$(P \supset R)$	premise	*If she likes mathematics she likes algebra.*
3.	$((R \wedge S) \supset T)$	premise	•
4.	$(\neg Q \vee S)$	premise	•
5.	P	1, simplification	
6.	Q	1, simplification	
7.	R	2, 5, modus ponens	
8.	S	4, 6, disjunctive syllogism	
9.	$(R \wedge S)$	7, 8, conjunction	
10.	T	3, 9, modus ponens	

The First Order version

Let us rephrase our example (Alice) problem in first order terminology.

- Alice likes mathematics and she likes stories.
- If someone likes mathematics she likes algebra[1].
- If someone likes algebra and likes physics she will go to college.
- Alice does not like stories or she likes physics.
- Alice does not like chemistry and history.”

We can formalize the statements in *FOL* as follows.

1. likes(Alice, Math) $\wedge$ likes(Alice, stories)
2. $\forall$x(likes(x, Math) $\supset$ likes(x, Algebra))
3. $\forall$x((likes(x, Algebra) $\wedge$ likes(x, Physics)) $\supset$ goesTo(x, College))
4. $\neg$likes(Alice, stories) $\vee$ likes(Alice, Physics)
5. $\neg$likes(Alice, Chemistry) $\wedge$ $\neg$likes(Alice, History)

[1] Here we must emphasize that *she* stands for both *she* and *he*.

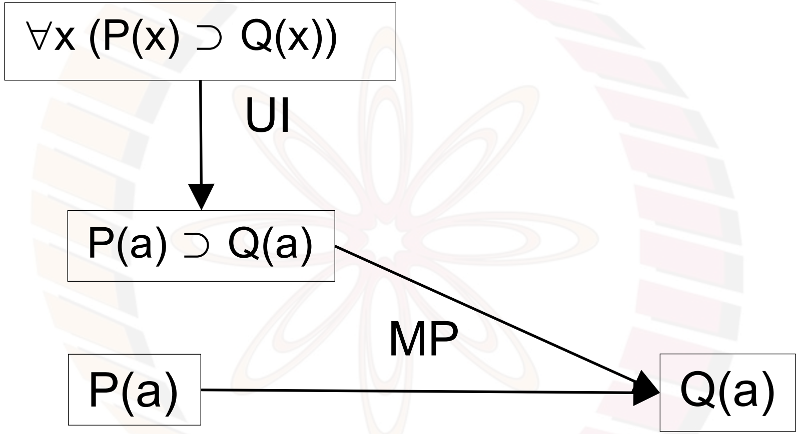
The FOL Proof

1. likes(Alice, Math) $\wedge$ likes(Alice, stories)
2. $\forall$x(likes(x, Math) $\supset$ likes(x, Algebra))
3. $\forall$x((likes(x, Algebra) $\wedge$ likes(x, Physics)) $\supset$ goesTo(x, College))
4. $\neg$likes(Alice, stories) $\vee$ likes(Alice, Physics)
5. $\neg$likes(Alice, Chemistry) $\wedge$ $\neg$likes(Alice, History)

We can now generate a proof that is analogous to the proof in propositional logic.

6.	likes(Alice, Math)	1, simplification
7.	likes(Alice, stories)	1, simplification
8.	(likes(Alice, Math) $\supset$ likes(Alice, Algebra))	2, UI
9.	likes(Alice, Algebra))	6, 8, modus ponens
10.	likes(Alice, Physics)	4, 7, disjunctive syllogism
11.	((likes(Alice, Algebra) $\wedge$ likes(Alice,Physics))	9, 10, conjunction
12.	((likes(Alice, Algebra) $\wedge$ likes(Alice,Physics)) $\supset$ goesTo(Alice,College))	3, UI
13.	goesTo(Alice, College)	12, 11, modus ponens

Forward Chaining in FOL



Forward chaining in FOL is a two step process.

The use of Implicit Quantifier Notation
collapses this two step inference into one.

Implicit Quantifier notation

Prefix universally quantified variables with a "?". Replace existentially quantified variables not in the scope of a universal quantified with a *Skolem constant* (named after the mathematician Thoralf Skolem)

1. Man(?x) $\supset$ Human(?x) : all men are human beings
2. Happy(suresh) $\vee$ Rich(suresh) : Suresh is rich or happy
3. CitrusFruit(?x) $\supset$ $\neg$Human(?x) : all citrus fruits are non-human
4. Man(sk-11) $\wedge$ Bright(sk-11)) : some men are bright

List notation

1. (if (man ?x) (human ?x))
2. (or (happy suresh) (rich suresh))
3. (if (citrusFruit ?x) (not (human ?x)))
4. (and (man sk-11) (bright sk-11))

Unifier: Substitution

A *substitution* θ is a set of <variable value> pairs each denoting the value to be substituted for the variable.

A *unifier* for two formulas α and β is a substitution that makes the two formulas identical. We say that α *unifies* with β. A unifier θ unifies a set of formulas $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ if,

$$\alpha_1\theta = \alpha_2\theta = \ldots = \alpha_N\theta = \varphi$$

Modified Modus Ponens (MMP)

MPP: From ($\alpha \supset \gamma$) and β infer $\gamma\theta$ where θ is a unifier* for α and β and $\gamma\theta$ is the formula obtained by applying the substitution θ to γ.

For example,

MMP

$\boxed{P(?x) \supset Q(?x)}$

$\theta = \{\langle ?x, a \rangle\}$

$\boxed{P(a)} \longrightarrow \boxed{Q(a)}$

*A substitution θ is a *unifier* for two (or more) formulas α and β if when applied it makes the two formulas identical. That is, $\alpha\theta = \beta\theta$

MPP: an example

Thus if

α = (Sport(tennis) ∧ Likes(Alice, tennis))

$\beta \supset \delta$ = (Sport(?y) ∧ Likes(?x, ?y)) ⊃ Watches(?x, ?y)

then α unifies with β

with the substitution θ = {<?x, Alice>, <?y, tennis>}

and one can infer

$\delta\theta$ = Watches(?x, ?y)θ = Watches(Alice, tennis)

Modified Modus Ponens (MMP)

From	α
and	$\underline{\beta\quad}$
Infer	$\alpha \wedge \beta$
Conjunction (C)	

$(P(?x,?y) \wedge R(?y)) \supset Q(?x)$

$P(a,?z) \wedge R(?z)$

$Q(a)$

$\theta = \{<?x, a>, <?y,?z>\}$

Bypassing Conjunction

$P(a,?z)$ $R(?z)$

$P(?x,?y) \wedge R(?y)$

$(P(?x,?y) \wedge R(?y)) \supset Q(?x)$

$P(a,?z)$

$R(?z)$

$Q(a)$

$\theta = \{<?x, a>, <?y,?z>\}$

A shorter proof with Modified Modus Ponens

1. likes(Alice, Math) $\wedge$ likes(Alice, stories)
2. likes(?x, Math) $\supset$ likes(?x, Algebra)
3. (likes(?x, Algebra) $\wedge$ likes(?x, Physics)) $\supset$ goesTo(?x, College)
4. $\neg$likes(Alice, stories) $\vee$ likes(Alice, Physics)
5. $\neg$likes(Alice, Chemistry) $\wedge$ $\neg$likes(Alice, History)

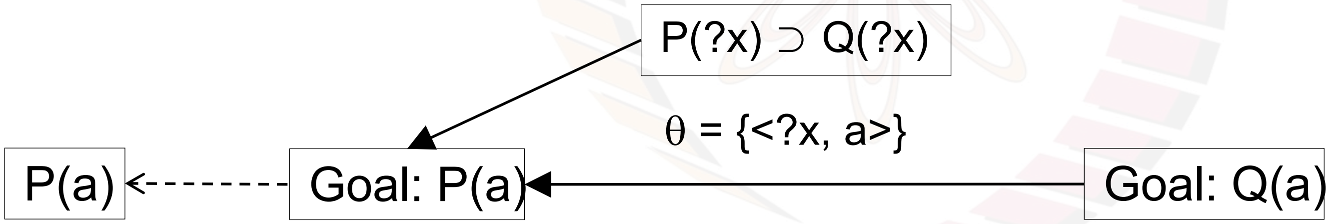
6. likes(Alice, Math)	1, simplification
7. likes(Alice, stories)	1, simplification
8. likes(Alice, Algebra)	6, 2, MPP
9. likes(Alice, Physics)	4, 7, disjunctive syllogism
10. goesTo(Alice, College)	3, 8, 9, MPP

Backward Chaining

We move from the goal to be proved towards facts.

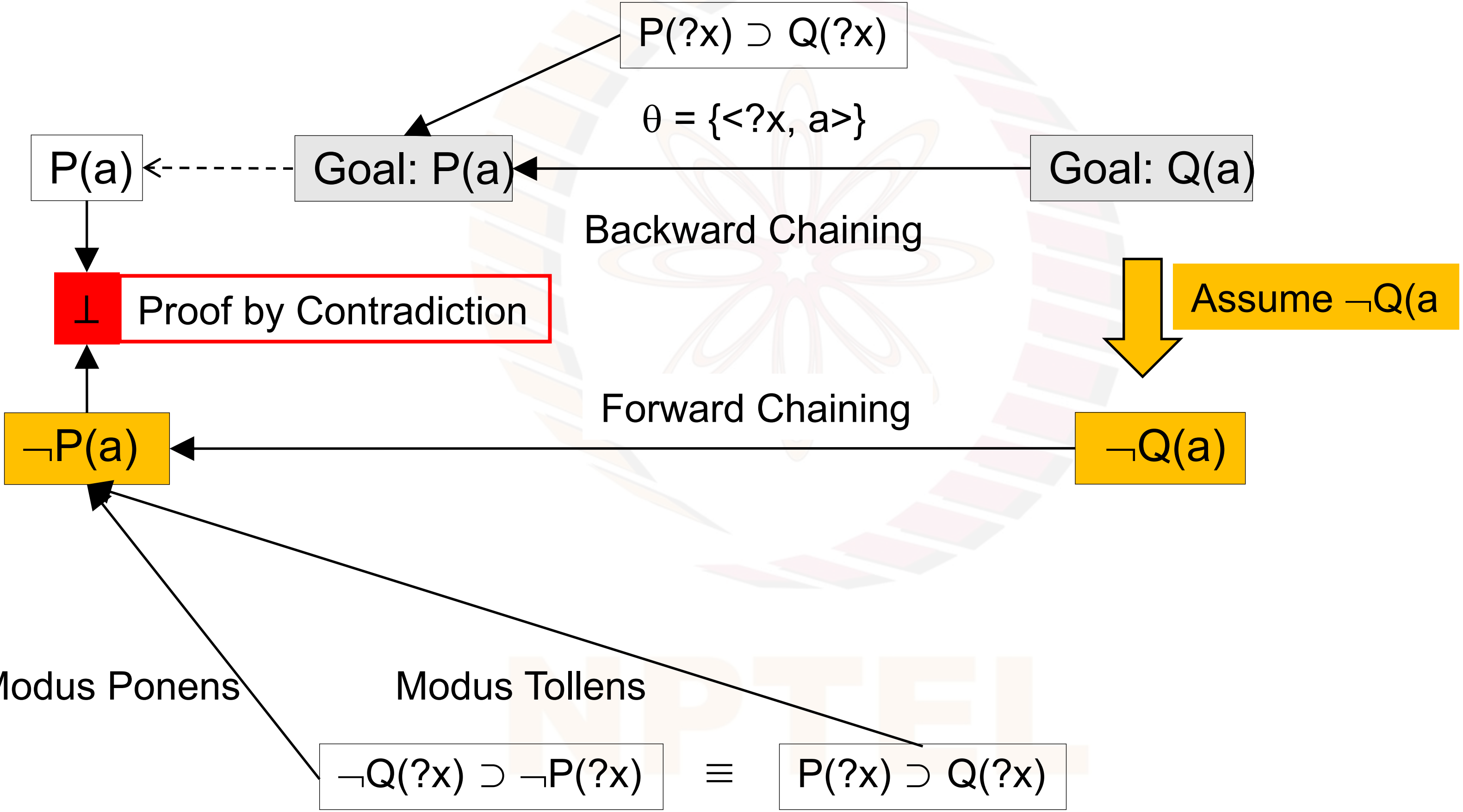
From $(\alpha \supset \gamma)$ and *Goal:*β infer *Goal:*$\alpha\theta$ where θ is a unifier for γ and β and $\alpha\theta$ is the formula obtained by applying the substitution θ to α.

For example,



A *goal* is said to be *solved* if it matches a fact in the KB.

Backward Chaining and Modus Tollens



Backward Reasoning

- Backward reasoning is goal directed
- We only look for rules for which the consequent matches the goal.
- This results in low branching factor in the search tree
  – which rule to apply from the matching set of rules?
- Foundations of Logic Programming
  – the programming language Prolog

Deductive Retrieval

The goal need not be a *specific proposition*

It can be have variables as well

Formulas with variables can match facts.

For example

Goal: Mortal(?z)

can be interpreted as an existential statement

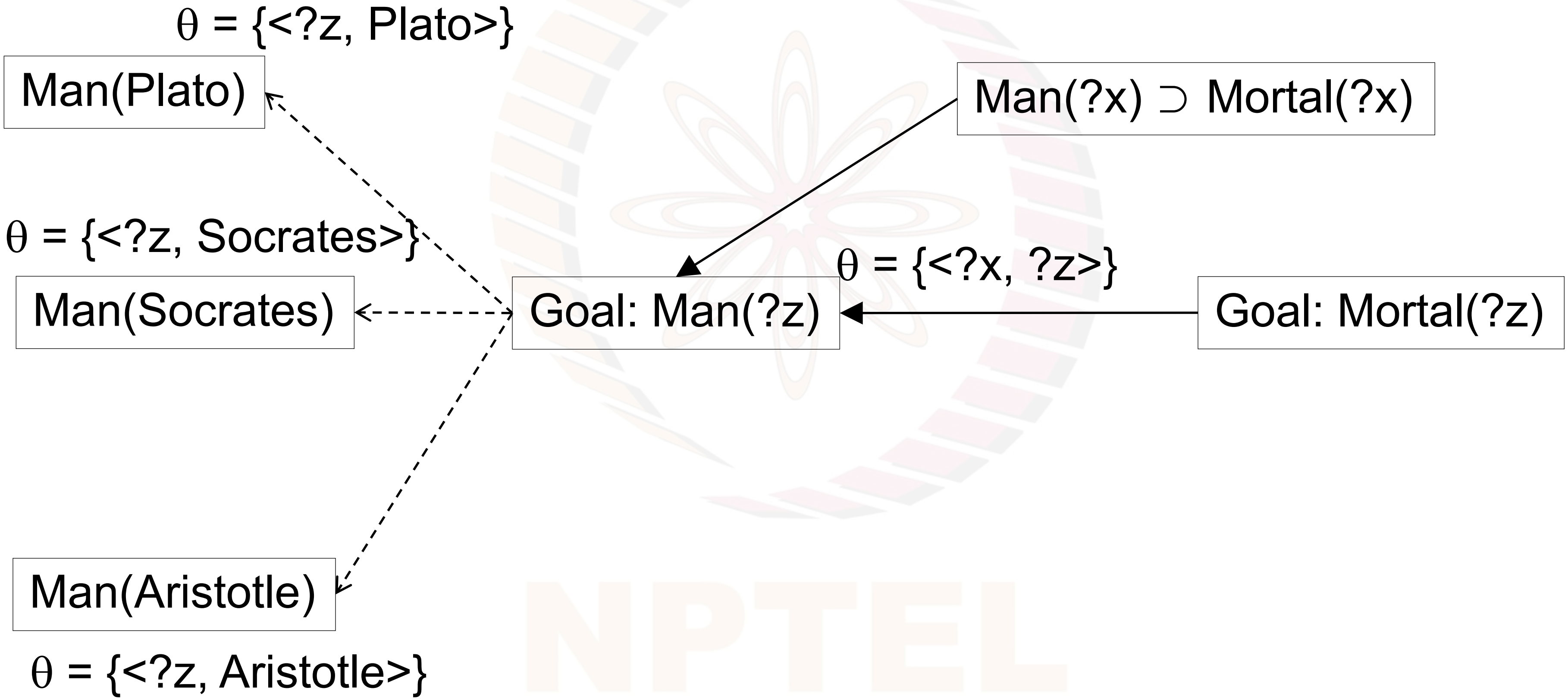
Is (there a z such that) $\exists z\, Mortal(z)$ true?

The answer, in addition to *yes* or *no*,

can also return a *value* for the *variable*

for which it is true.

Deductive Retrieval: possible answers



Backward Chaining (Propositional Logic)

Alice likes mathematics (P) and she likes stories (Q). If she likes mathematics (P) she likes algebra (R). If she likes algebra (R) and likes physics (S) she will go to college (T). She does not like stories (Q) or she likes physics (S). She does not like chemistry (U) and history (V).

Then the given facts are, $(P \land Q)$, $(P \supset R)$, $((R \land S) \supset T)$, $(\sim Q \lor S)$, $(\sim U \land \sim V)$

Equivalently

1. P
2. Q
3. $(P \supset R)$
4. $((R \land S) \supset T)$
5. $(Q \supset S)$
6. $\sim U$
7. $\sim V$

Goal Set

{T}	Given goal
{R, S}	from 4
{P, S}	from 3
{S}	matches 1
{Q}	from 5
{ }	matches 2, success

"Is T *true*?"

We answer this by backward chaining.

Backward Chaining with Conjunctive Antecedents

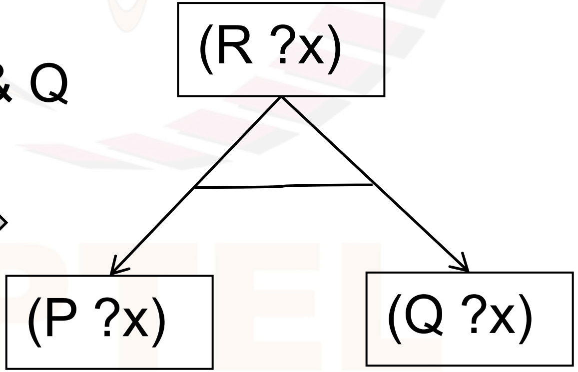
A goal (R ?x) and a rule (if (and (P ?x) (Q ?x)) (R ?x))

A *goal which matches the consequent* of a rule reduces to the antecedents in the rule.

{(R ?x)}
↓
{(P ?x) (Q ?x)}

To solve R
solve both P & Q

An AND node

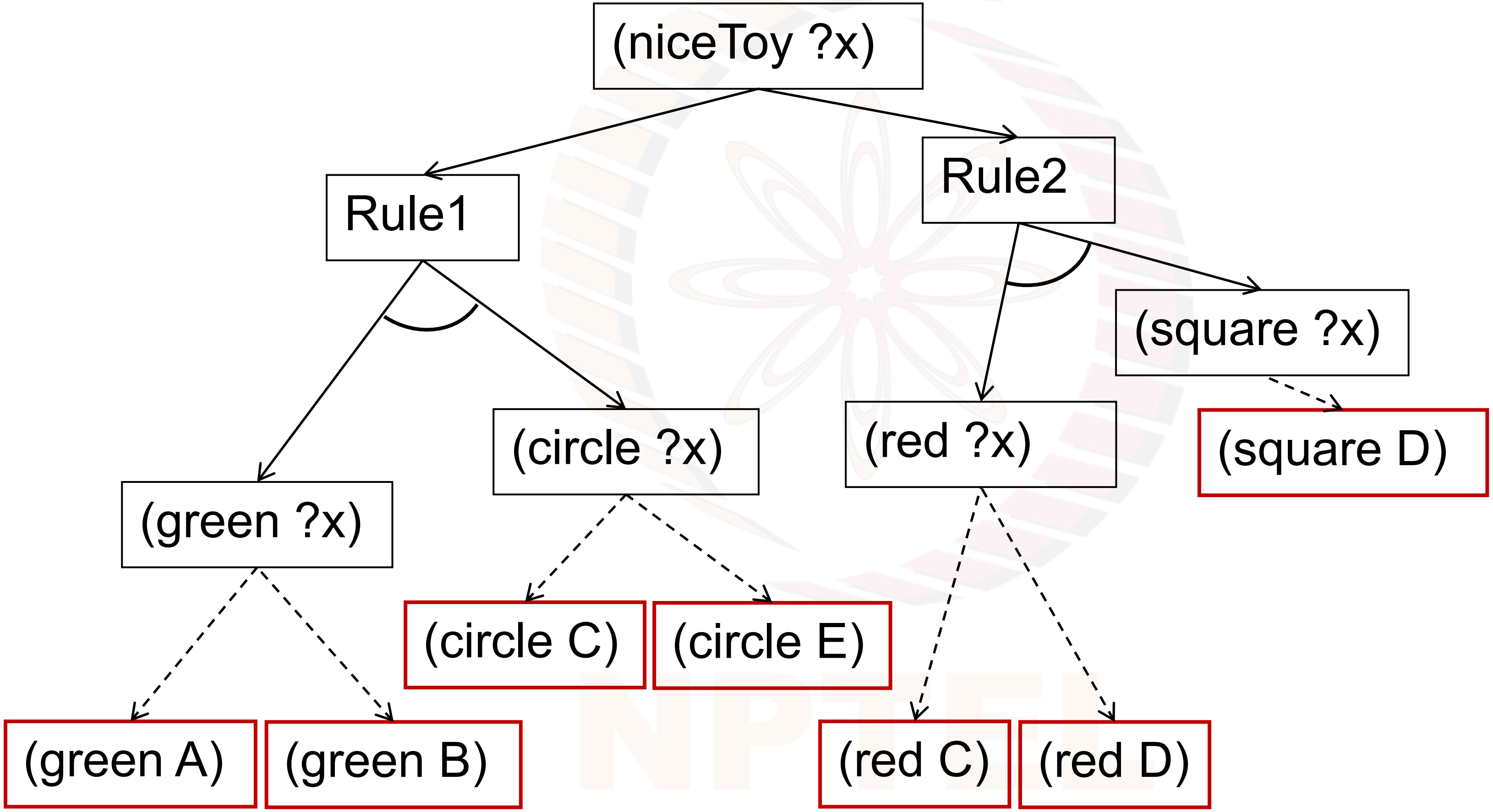


Goal Trees

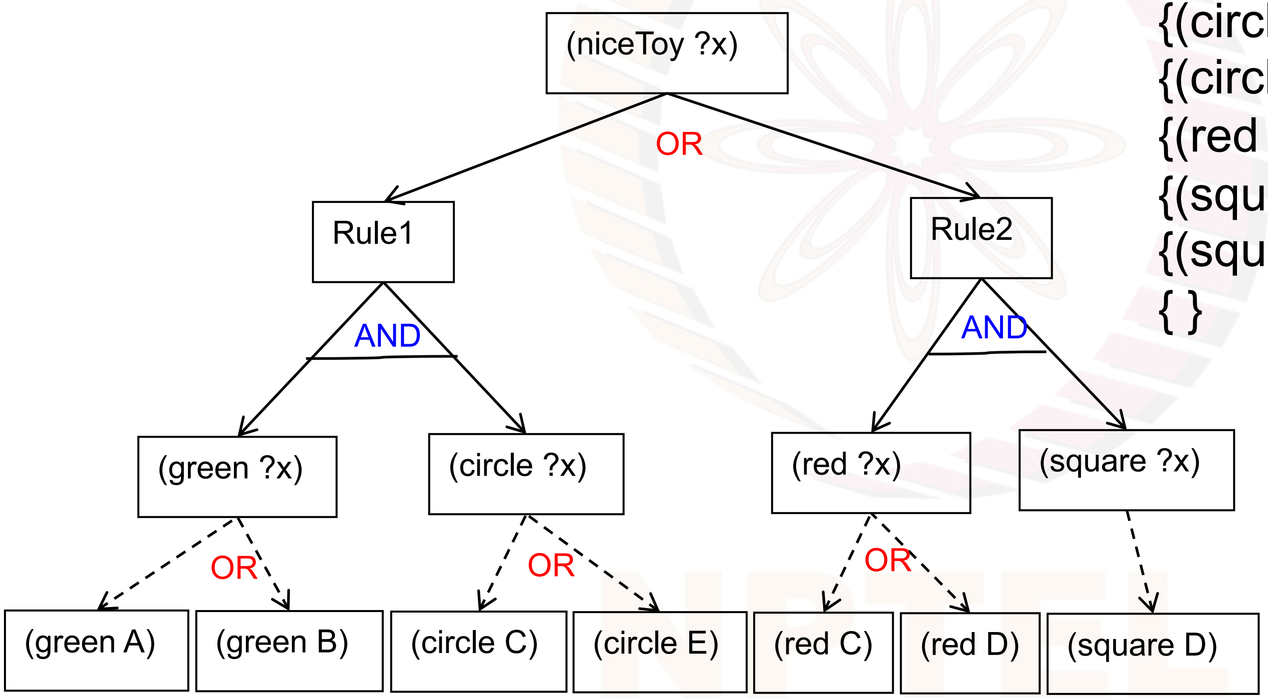
Consider the following KB in skolemized list notation, and the goal (niceToy ?z)

Rule1: (if (and (green ?x) (circle ?x)) (niceToy ?x))
Rule2: (if (and (red ?x) (square ?x)) (niceToy ?x))
(green A)
(green B)
(circle C)
(red C)
(red D)
(square D)
(circle E)

Goal tree = AND/OR tree



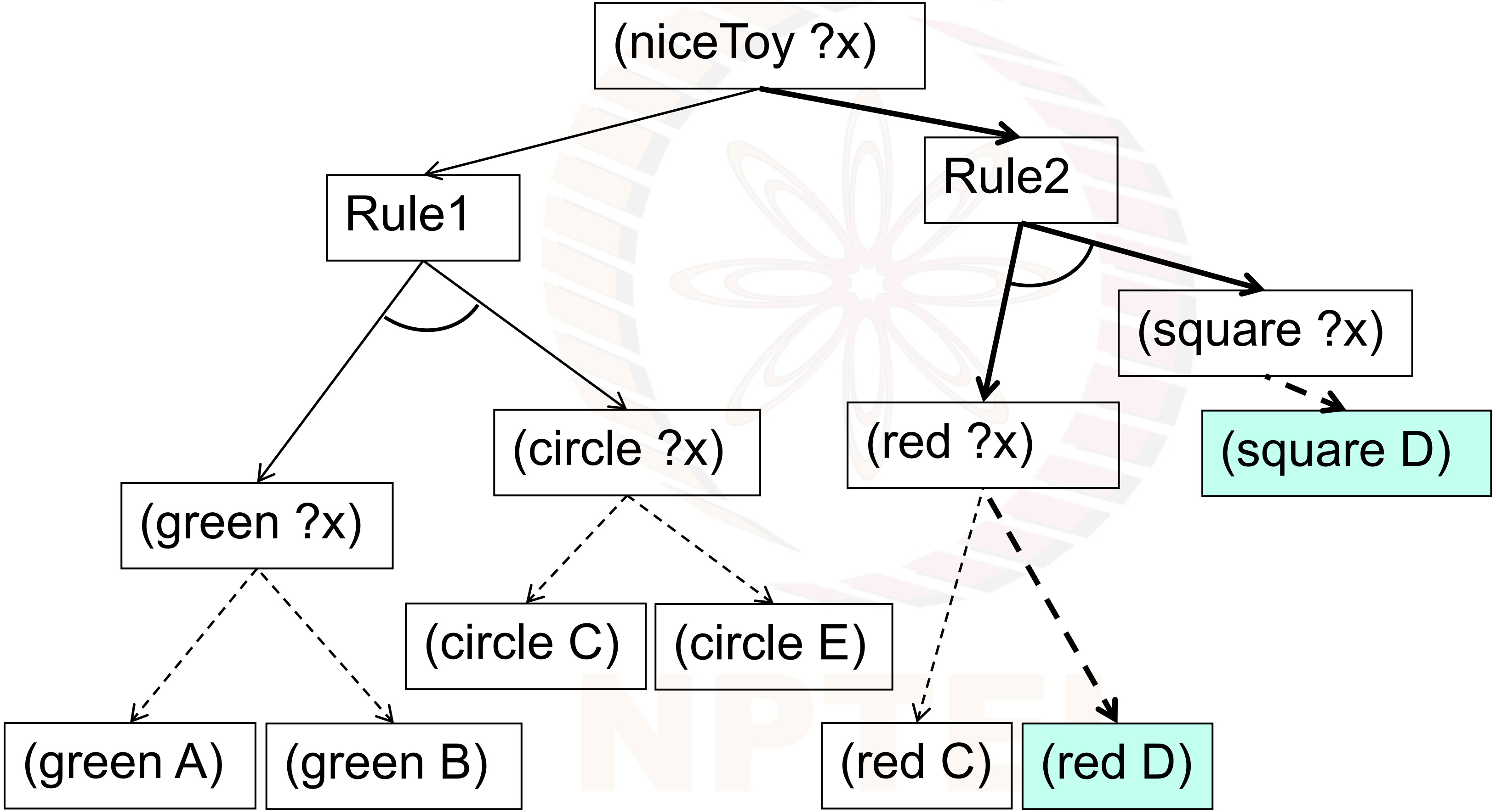
Depth First Search



Goal Set

{(niceToy ?x)}		
{(green ?x), (circle ?x)}		Rule1
{(circle A)	?x=A	FAIL
{(circle B)}	?x=B	FAIL
{(red ?x), (square ?x)}		Rule2
{(square C)}	?x=C	FAIL
{(square D)}	?x=D	
{ }	Success	

AND/OR tree: Solution = subtree

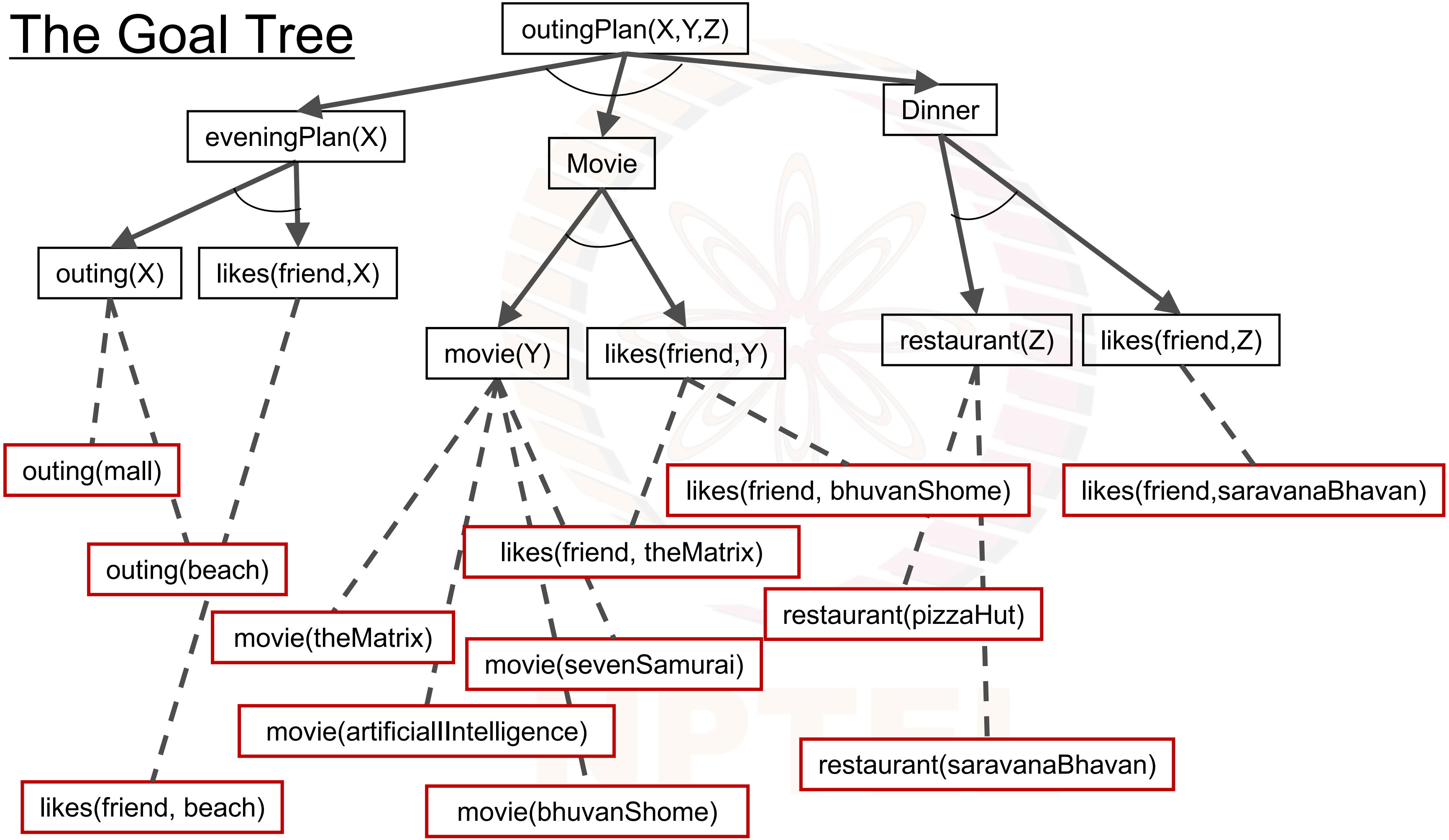


A Prolog KB (program)

```
outingPlan(X,Y,Z) :- eveningPlan(X), moviePlan(Y), dinnerPlan(Z).
eveningPlan(X) :- outing(X), likes(friend, X).
moviePlan(X) :- movie(X), likes(friend,X).
dinnerPlan(X) :- restaurant(X), likes(friend,X).
outing(mall).
outing(beach).
movie(theMatrix).
movie(artificialIntelligence).
movie(bhuvanShome).
movie(sevenSamurai).
restaurant(pizzaHut).
restaurant(saravanaBhavan).
likes(friend, beach).
likes(friend, theMatrix).
likes(friend, bhuvanShome).
likes(friend, sarvanaBhavan).
```

```
(if (and (restaurant ?x) (likes friend ?x)) (dinnerPlan ?x))
```

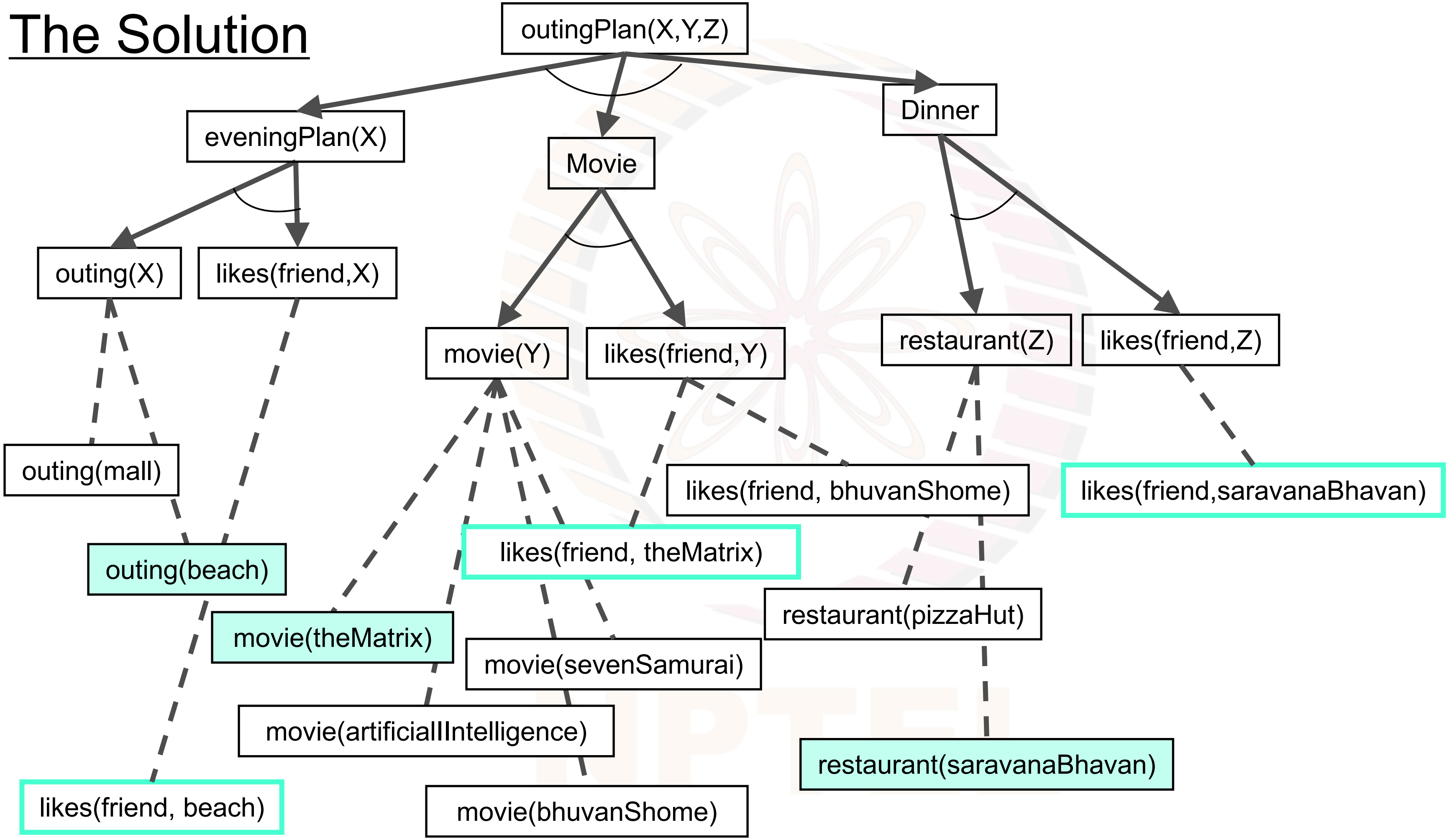
The Goal Tree



Backward Chaining: Depth First Search

{outingPlan(X,Y,Z)}	theta = { }
{eveningPlan(X), moviePlan(Y), dinnerPlan(Z)}	theta = { }
{outing(X), likes(friend, X), moviePlan(Y), dinnerPlan(Z)}	theta = { }
{likes(friend, mall), moviePlan(Y), dinnerPlan(Z)}	theta = {X=mall}
{“fail”, moviePlan(Y), dinnerPlan(Z)}	theta = {X=mall}
{outing(X), likes(friend, X), moviePlan(Y), dinnerPlan(Z)}	theta = { }backtrack
{likes(friend, beach), moviePlan(Y), dinnerPlan(Z)}	theta = {X=beach}
{moviePlan(Y), dinnerPlan(Z)}	theta = {X=beach}
{movie(Y), likes(friend,Y), dinnerPlan(Z)}	theta = {X=beach}
{likes(friend, theMatrix), dinnerPlan(Z)}	theta = {X=beach, Y=theMatrix}
{dinnerPlan(Z)}	theta = {X=beach, Y=theMatrix}
{restaurant(Z), likes(friend,Z)}	theta = {X=beach, Y=theMatrix}
{likes(friend,pizzaHut)}	theta = {X=beach, Y=theMatrix, Z=pizzaHut}
{“fail”}	theta = {X=beach, Y=theMatrix, Z=pizzaHut}
{restaurant(Z), likes(friend,Z)}	theta = {X=beach, Y=theMatrix} backtrack
{likes(friend, saravanaBhavan)}	theta = {X=beach, Y=theMatrix, Z= saravanaBhavan }
{ }	theta = {X=beach, Y=theMatrix, Z= saravanaBhavan }

The Solution



A not so easy problem

Given the following knowledge base (in list notation)

{(O A B), (O B C), (not (M A)), (M C)}

What is the KB talking about?
What is the semantics?

Remember, logic is formal

Depends upon the interpretation $\vartheta = \langle D, I \rangle$!

Two sample interpretations….

Interpretation 1

{(O A B), (O B C), (not (M A)), (M C)}

Domain: Blocks World

Predicate symbols

O → On

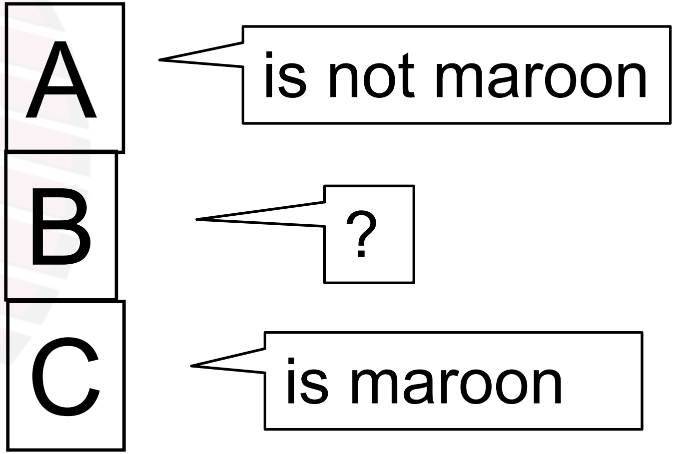
M → Maroon

Constant Symbols

A, B, C → blocks

A is on B

B is on C



Interpretation 2

{(O A B), (O B C), (not (M A)), (M C)}

Domain: People

Predicate symbols
O → LookingAt
M → Married

Constant Symbols
A → Jack
B → Anne
C → John

Anne is looking at John

Jack is looking at Anne

John

Anne

Jack

is married

is not married

?

The Goal

{(O A B), (O B C), (not (M A)), (M C)}

Given the KB and the goal

(exists (x y) (and (O x y) (not (M x)) (M y)))

or equivalently (and (O ?x ?y) (not (M ?x)) (M ?y))

…is *clearly* (?) entailed

Blocks World:

Is there a not-maroon block on a maroon block?

People:

Is a not-married person looking at a married one?

Incompleteness of Backward and Forward Chaining

Given the KB,

{(O A B), (O B C), (not (M A)), (M C)}

And the Goal,

(and (O ?x ?y) (not (M ?x)) (M ?y))

Neither Forward Chaining nor Backward Chaining is able to generate a proof.

Both are Incomplete!

A complete procedure…

in AI: Knowledge Representation & Reasoning

End : Deduction as Search