Set 7:
Predicate logic
Chapter 8 R&N

ICS 271 Fall 2018

Outline

- New ontology
  - objects, relations, properties, functions
- New Syntax
  - Constants, predicates, properties, functions
- New semantics
  - meaning of new syntax
- Inference rules for Predicate Logic (FOL)
  - Unification
  - Resolution
  - Forward-chaining, Backward-chaining
- Readings: Russel and Norvig Chapter 8 & 9

Pros and cons of propositional logic

- 😊 Propositional logic is *declarative*: pieces of syntax correspond to facts
- 😊 Propositional logic allows partial/disjunctive/negated information (unlike most data structures and databases)
- 😊 Propositional logic is *compositional*: meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- 😊 Meaning in propositional logic is *context-independent* (unlike natural language, where meaning depends on context)
- ☹ Propositional logic has very limited expressive power (unlike natural language)
  E.g., cannot say "pits cause breezes in adjacent squares"
  except by writing one sentence for each square

Propositional logic is not expressive

- Needs to refer to objects in the world,
- Needs to express general rules
  - On(x,y) $\rightarrow$ $\neg$ clear(y)
  - All men are mortal; Socrates is a man, therefore mortal
  - Everyone who passed the age of 21 can drink
  - One student in this class got perfect score
  - Etc….
- First order logic, also called Predicate calculus allows more expressiveness

Propositional logic is not expressive, cont.

- Combinatorial explosion when trying to express general rules :
  - Exactly one student in the class got perfect score
    - Propositional logic
      - $P_1 \vee P_2 \vee \ldots \vee P_n$
      - For all i,j : $\neg P_i \vee \neg P_j$
    - First order logic
      - $\exists x[P(x) \wedge \neg \exists y[x \neq y \wedge P(y)]]$
    - Q : exactly two students have perfect score?

Logics in general

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

First-order logic

Whereas propositional logic assumes world contains *facts*,
first-order logic (like natural language) assumes the world contains

- Objects: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries ...
- Relations: red, round, bogus, prime, multistoried ..., brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, ...
- Functions: father of, best friend, third inning of, one more than, beginning of ...

Syntax of FOL: Basic elements

Constants	$KingJohn,\ 2,\ UCB, \ldots$
Predicates	$Brother,\ >, \ldots$
Functions	$Sqrt,\ LeftLegOf, \ldots$
Variables	$x,\ y,\ a,\ b, \ldots$
Connectives	$\wedge\ \vee\ \neg\ \Rightarrow\ \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall\ \exists$

Atomic sentences

$$\text{Atomic sentence} = predicate(term_1, \ldots, term_n)$$
$$\text{or } term_1 = term_2$$

$$\text{Term} = function(term_1, \ldots, term_n)$$
$$\text{or } constant \text{ or } variable$$

E.g., $Brother(KingJohn, RichardTheLionheart)$
$> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Complex sentences

Complex sentences are made from atomic sentences using connectives

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

E.g. $Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$
$>(1,2) \vee \leq(1,2)$
$>(1,2) \wedge \neg>(1,2)$

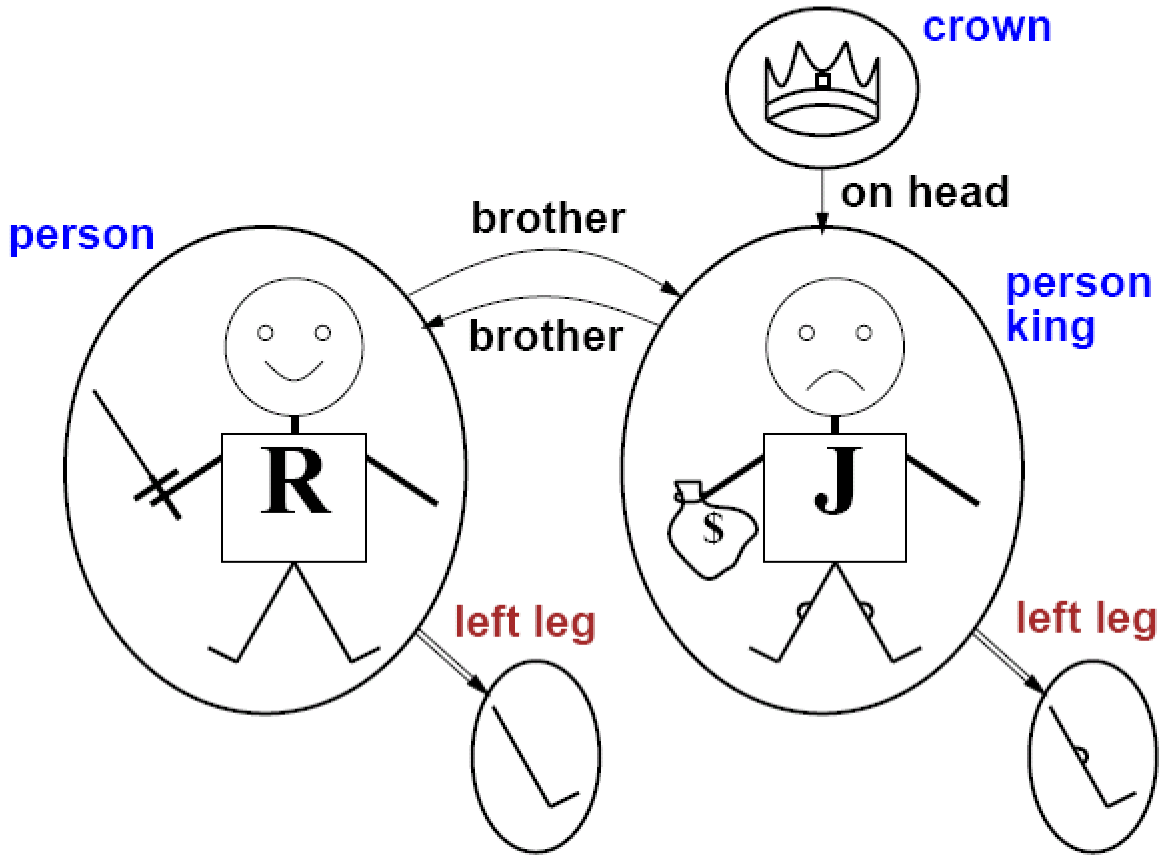
FOL : syntax

1. Terms – refer to objects
   - Constants : a, b, c, …
   - Variables : x, y, …
     - Can be free or bound
   - Functions (over terms) : f, g, …
   - Ground term : constants + fully instantiated functions (no variables) : f(a)
2. Predicates
   - E.g. P(a), Q(x), …
   - Unary = property, arity>1 = relation between objects
   - Atomic sentences
   - Evaluate to true/false
   - Special relation '='
3. Logical connectives : $\neg \wedge \vee \rightarrow$
4. Quantifiers : $\exists$ $\forall$
   - Typically want sentences wo free variables (fully quantified)
5. Function vs Predicate
   - FatherOf(John) vs Father(X,Y) [Father(FatherOf(John),John)]
   - Q : BrotherOf(John) vs Brothers(X,Y)?

Semantics: Worlds

- **The** world **consists of** objects **that have** properties**.**
  - **There are** relations **and** functions **between these objects**
  - **Objects in the world, individuals:** people, houses, numbers, colors, baseball games, wars, centuries
    - Clock A, John, 7, the-house in the corner, Los Angeles, …
  - **Functions** on individuals:
    - father-of, best friend, third inning of, one more than
  - **Relations**:
    - brother-of, bigger than, inside, part-of, has color, occurred after
  - **Properties (a relation of arity 1)**:
    - red, round, bogus, prime, multistoried, beautiful
  - Note : FOL possible world has no variables! Just objects/functions/relations.

Models for FOL: Example



Truth in first-order logic

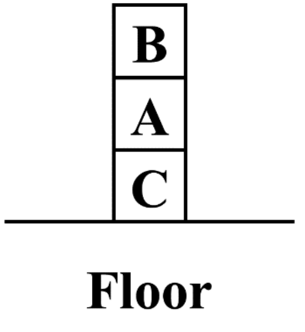
- World contains objects (domain elements) and relations/functions among them
- Interpretation specifies referents for

constant symbols	→	objects
predicate symbols	→	relations
function symbols	→	functions

- Sentences are true with respect to a world and an interpretation
- An atomic sentence *predicate*$(term_1,...,term_n)$ is true
  iff the objects referred to by $term_1,...,term_n$
  are in the relation referred to by *predicate*

Semantics: Interpretation

- An interpretation of a sentence (wff) is defined wrt a world that has a set of constants, functions, relations
- An interpretation of a sentence (wff) is a structure that maps
  - Constant symbols of the language to constants in the worlds,
  - n-ary function symbols of the language to n-ary functions in the world,
  - n-ary predicate symbols of the language to n-ary relations in the world
- Given an interpretation, an atom has the value “true” if it denotes a relation that holds for those individuals denoted in the terms. Otherwise it has the value “false”
  - Example: Block world:
    - A, B, C, Floor, On, Clear
  - World:
    - On(A,B) is false, Clear(B) is true, On(C,F) is true…

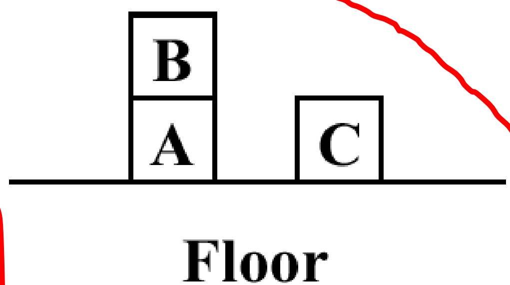


Example of Models (Blocks World)

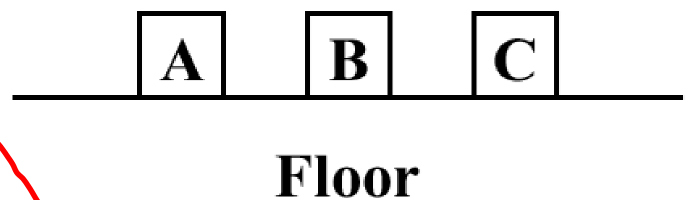
- **The formulas:**
  - On(A,F) → Clear(B)
  - Clear(B) and Clear(C) → On(A,F)
  - Clear(B) or Clear(A)
  - Clear(B)
  - Clear(C)

Possible interpretations where the KB is true:

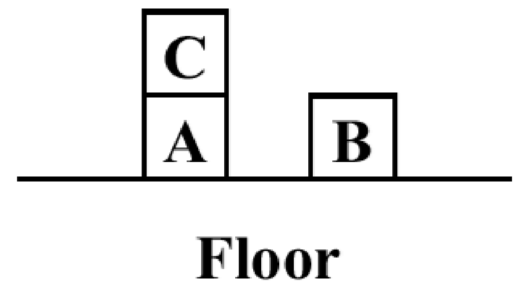
- **Checking truth value of Clear(B)**
  - Map B (sentence) to B' (interpretation)
  - Map Clear (sentence) to Clear' (interpretation)
  - Clear(B) is true iff B' is in Clear'



On = {<B,A>,<A,F>,<C,F>}
Clear = {<C>,<B>}

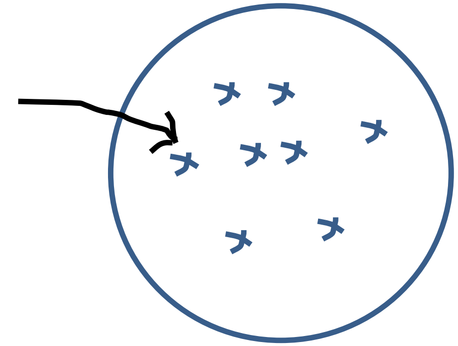
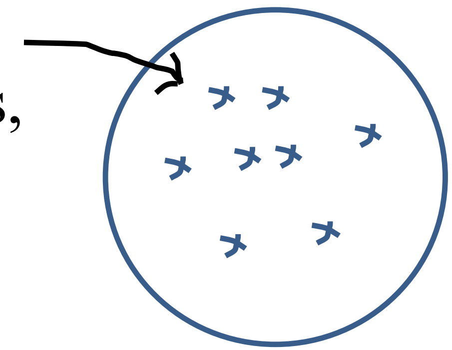


On = {<A,F>, <B,F>,<C,F>}
Clear = {<A>,<B>,<C>}



On = {<C,A>,<A,F>,<B,F>}
Clear = {<C>,<B>}

Semantics : PL vs FOL

Language	Possible worlds (interpretations)
KB : CNF over prop symbols	Semantics: an interpretation maps prop symbols to {true,false} 
KB : CNF over predicates over terms (fn + var + const) Note : const, fn, pred symbols	Semantics: an interpretation has obj's and maps : const symbols to const's, fn symbols to fn's, pred symbols to pred's  Note : const's, fn's, pred's Note : var's not mapped!

Semantics: Models

- **An interpretation** satisfies **a sentence if the sentence has the value "true" under the interpretation.**
- **Model: An interpretation that satisfies a sentence is a** model **of that sentence**
- **Validity: Any sentence that has the value "true" under all interpretations is** valid
- **Any sentence that does not have a model is inconsistent or unsatisfiable**
- **If a sentence w has a value true under all the models of a set of sentences KB then KB logically entails w**

- **Note :**
  - **In FOL a set of possible worlds is infinite**
  - **Cannot use model checking!!!**

Quantification

- ***Universal*** and ***existential*** quantifiers allow expressing general rules with variables
- *Universal quantification*
  - Syntax: if **w** is a sentence (wff) then ∀**x w** is a wff.
  - All cats are mammals $\forall x\, Cat(x) \rightarrow Mammal(x)$
  - It is equivalent to the conjunction of all the sentences obtained by substitution the name of an object for the variable x.

$$Cat(Spot) \rightarrow Mammal(Spot) \wedge$$
$$Cat(Rebbeka) \rightarrow Mammal(Rebbeka) \wedge$$
$$Cat(Felix) \rightarrow Mammal(Felix) \wedge$$
$$,,,,$$

Universal quantification

$\forall \langle variables \rangle \; \langle sentence \rangle$

Everyone at Berkeley is smart:
$\forall x \; At(x, Berkeley) \Rightarrow Smart(x)$

$\forall x \; P$ is true in a model m iff P with x holding for each possible object in the model

Roughly speaking, equivalent to the conjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn, Berkeley) \Rightarrow Smart(KingJohn) \\
\wedge \; & At(Richard, Berkeley) \Rightarrow Smart(Richard) \\
\wedge \; & At(Berkeley, Berkeley) \Rightarrow Smart(Berkeley) \\
\wedge \; & \ldots
\end{aligned}
$$

A common mistake to avoid

Typically, $\Rightarrow$ is the main connective with $\forall$

Common mistake: using $\land$ as the main connective with $\forall$:

$$\forall x \;\; At(x, Berkeley) \land Smart(x)$$

means "Everyone is at Berkeley and everyone is smart"

Quantification: Existential

- **Existential quantification : $\exists$ an existentially quantified sentence is true if it is true for some object**

$$\exists x Sister(x, Spot) \wedge Cat(x)$$

- **Equivalent to disjunction:**

$$Sister(Spot, Spot) \wedge Cat(Spot) \vee$$
$$Sister(Rebecca, Spot) \wedge Cat(Rebecca) \vee$$
$$Sister(Felix, Spot) \wedge Cat(Felix) \vee$$
$$Sister(Richard, Spot) \wedge Cat(Richard) \ldots$$

- **We can mix existential and universal quantification.**

Existential quantification

$\exists \langle variables \rangle \ \langle sentence \rangle$

Someone at Stanford is smart:
$\exists x \ \ At(x, Stanford) \wedge Smart(x)$

$\exists x \ \ P$ is is true in a model m iff P with x holding for some possible object in the model

Roughly speaking, equivalent to the disjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn, Stanford) \wedge Smart(KingJohn) \\
\vee \ & At(Richard, Stanford) \wedge Smart(Richard) \\
\vee \ & At(Stanford, Stanford) \wedge Smart(Stanford) \\
\vee \ & \ldots
\end{aligned}
$$

Another common mistake to avoid

Typically, $\wedge$ is the main connective with $\exists$

Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists\, x \;\; At(x, Stanford) \Rightarrow Smart(x)$$

is true if there is anyone who is not at Stanford!

Properties of quantifiers

- ∀x ∀y is the same as ∀y ∀x
- ∃x ∃y is the same as ∃y ∃x
- ∃x ∀y is not the same as ∀y ∃x
  - ∃x ∀y Loves(x,y)
    - "There is a person who loves everyone in the world"
  - ∀y ∃x Loves(x,y)
    - "Everyone in the world is loved by at least one person"
- ¬∀x Likes(x,IceCream) ∃x ¬ Likes(x,IceCream)
  - "not true that P(X) holds for all X" ≡ "exists X for which P(X) is false"
- ¬∃x Likes(x, Broccoli) ∀x ¬ Likes(x, Broccoli)
- Quantifier duality : each can be expressed using the other
- ∀x Likes(x,IceCream) ¬∃x ¬ Likes(x,IceCream)
- ∃x Likes(x,Broccoli) ¬∀x ¬ Likes(x,Broccoli)

Fun with sentences

Brothers are siblings

Fun with sentences

Brothers are siblings

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$

"Sibling" is symmetric

Fun with sentences

Brothers are siblings

$\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$

"Sibling" is symmetric

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$

One's mother is one's female parent

Fun with sentences

Brothers are siblings

$\forall x, y \;\; Brother(x, y) \; \Rightarrow \; Sibling(x, y).$

"Sibling" is symmetric

$\forall x, y \;\; Sibling(x, y) \; \Leftrightarrow \; Sibling(y, x).$

One's mother is one's female parent

$\forall x, y \;\; Mother(x, y) \; \Leftrightarrow \; (Female(x) \wedge Parent(x, y)).$

A first cousin is a child of a parent's sibling

Fun with sentences

Brothers are siblings

$\forall x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$

"Sibling" is symmetric

$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$

One's mother is one's female parent

$\forall x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \land Parent(x, y)).$

A first cousin is a child of a parent's sibling

$\forall x, y \;\; FirstCousin(x, y) \;\Leftrightarrow\; \exists\, p, ps \;\; Parent(p, x) \land Sibling(ps, p) \land Parent(ps, y)$

Equality

- $term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object

- E.g., definition of *Sibling* in terms of *Parent*:

$\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(x = y) \land \exists m,f\ \neg(m = f) \land Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$

Using FOL

- **The kinship domain:**
  - Objects are people
  - Properties include gender and they are related by relations such as parenthood, brotherhood, marriage
  - predicates: Male, Female (unary) Parent, Sibling, Daughter, Son...
  - Function: Mother Father

- Brothers are siblings

  $\forall$x,y *Brother(x,y)* $\Rightarrow$ *Sibling(x,y)*

- One's mother is one's female parent

  $\forall$m,c *Mother(c)* = m $\Leftrightarrow$ *(Female(m)* $\wedge$ *Parent(m,c))*

- “Sibling” is symmetric

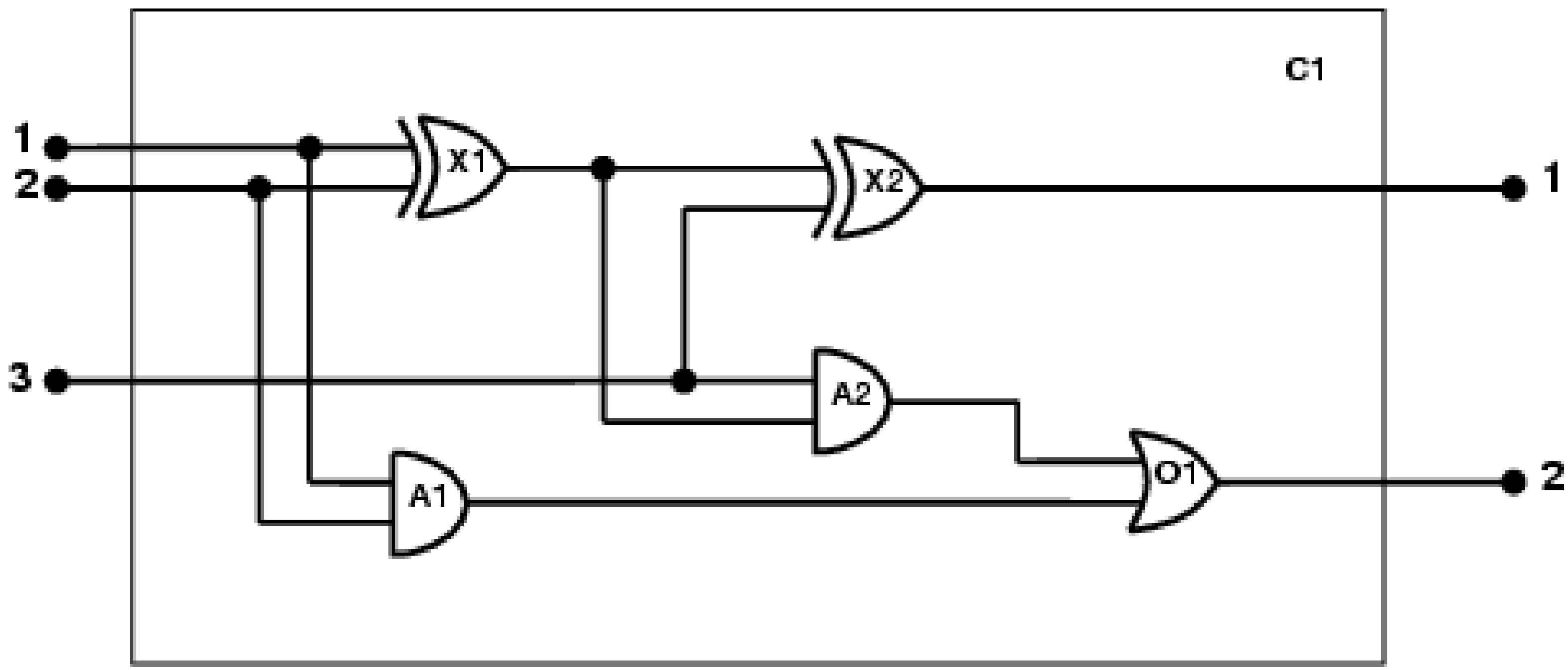
  $\forall$x,y *Sibling(x,y)* $\Leftrightarrow$ *Sibling(y,x)*

Knowledge engineering in FOL

1. **Identify the task**
2. **Assemble the relevant knowledge; identify important concepts**
3. **Decide on a vocabulary of predicates, functions, and constants**
4. **Encode general knowledge about the domain**
5. **Encode a description of the specific problem instance**
6. **Pose queries to the inference procedure and get answers**
7. **Debug the knowledge base**

The electronic circuits domain

One-bit full adder



The electronic circuits domain

1. **Identify the task**
   - **Does the circuit actually add properly? (circuit verification)**
2. **Assemble the relevant knowledge**
   - **Composed of I/O terminals, connections and gates; Types of gates (AND, OR, XOR, NOT)**
   - **Irrelevant: size, shape, color, cost of gates**
3. **Decide on a vocabulary**
   - **Alternatives :**

     Type(X_1) = XOR
     Type(X_1, XOR)
     XOR(X_1)

The electronic circuits domain

4. Encode general knowledge of the domain

- $\forall t_1, t_2$ Connected(t_1, t_2) $\Rightarrow$ Signal(t_1) = Signal(t_2)
- $\forall t$ Signal(t) = 1 $\vee$ Signal(t) = 0
- 1 ≠ 0
- $\forall t_1, t_2$ Connected(t_1, t_2) $\Rightarrow$ Connected(t_2, t_1)
- $\forall g$ Type(g) = OR $\Rightarrow$ Signal(Out(1,g)) = 1 $\Leftrightarrow$ $\exists n$ Signal(In(n,g)) = 1
- $\forall g$ Type(g) = AND $\Rightarrow$ Signal(Out(1,g)) = 0 $\Leftrightarrow$ $\exists n$ Signal(In(n,g)) = 0
- $\forall g$ Type(g) = XOR $\Rightarrow$ Signal(Out(1,g)) = 1 $\Leftrightarrow$ Signal(In(1,g)) ≠ Signal(In(2,g))
- $\forall g$ Type(g) = NOT $\Rightarrow$ Signal(Out(1,g)) ≠ Signal(In(1,g))

The electronic circuits domain

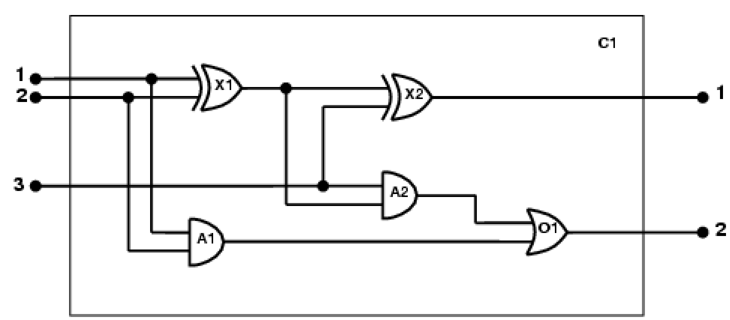
5. Encode the specific problem instance

Type(X_1) = XOR
Type(A_1) = AND
Type(O_1) = OR

Type(X_2) = XOR
Type(A_2) = AND

Connected(Out(1,X_1),In(1,X_2))
Connected(Out(1,X_1),In(2,A_2))
Connected(Out(1,A_2),In(1,O_1))
Connected(Out(1,A_1),In(2,O_1))
Connected(Out(1,X_2),Out(1,C_1))
Connected(Out(1,O_1),Out(2,C_1))

Connected(In(1,C_1),In(1,X_1))
Connected(In(1,C_1),In(1,A_1))
Connected(In(2,C_1),In(2,X_1))
Connected(In(2,C_1),In(2,A_1))
Connected(In(3,C_1),In(2,X_2))
Connected(In(3,C_1),In(1,A_2))



The electronic circuits domain

6. Pose queries to the inference procedure

What are the possible sets of values of all the terminals for the adder circuit?

$\exists i_1,i_2,i_3,o_1,o_2$ Signal(In(1,C_1)) = $i_1 \wedge$ Signal(In(2,C_1)) = $i_2 \wedge$ Signal(In(3,C_1)) = $i_3 \wedge$ Signal(Out(1,C_1)) = $o_1 \wedge$ Signal(Out(2,C_1)) = o_2

7. Debug the knowledge base

May have omitted assertions like $1 \neq 0$

Interacting with FOL KBs

Suppose a wumpus-world agent is using an FOL KB
and perceives a smell and a breeze (but no glitter) at $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$
$Ask(KB, \exists a \; Action(a, 5))$

I.e., does the KB entail any particular actions at $t = 5$?

Answer: Yes, $\{a/Shoot\}$ $\leftarrow$ substitution (binding list)

$Ask(KB, S)$ returns some/all σ such that $KB \models S\sigma$

Given a sentence S and a substitution σ,
$S\sigma$ denotes the result of plugging σ into S; e.g.,
$S = Smarter(x, y)$
$\sigma = \{x/Hillary, y/Bill\}$
$S\sigma = Smarter(Hillary, Bill)$

Knowledge base for the wumpus world

"Perception"

$\forall\, b, g, t \;\; Percept([Smell, b, g], t) \;\Rightarrow\; Smelt(t)$

$\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

Reflex: $\forall\, t \;\; AtGold(t) \;\Rightarrow\; Action(Grab, t)$

Reflex with internal state: do we have the gold already?

$\forall\, t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$

$Holding(Gold, t)$ cannot be observed

$\Rightarrow$ keeping track of change is essential

Deducing hidden properties

Properties of locations:
$\forall x, t \;\; At(Agent, x, t) \land Smelt(t) \;\Rightarrow\; Smelly(x)$
$\forall x, t \;\; At(Agent, x, t) \land Breeze(t) \;\Rightarrow\; Breezy(x)$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall y \;\; Breezy(y) \;\Rightarrow\; \exists x \;\; Pit(x) \land Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall x, y \;\; Pit(x) \land Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

Definition for the $Breezy$ predicate:

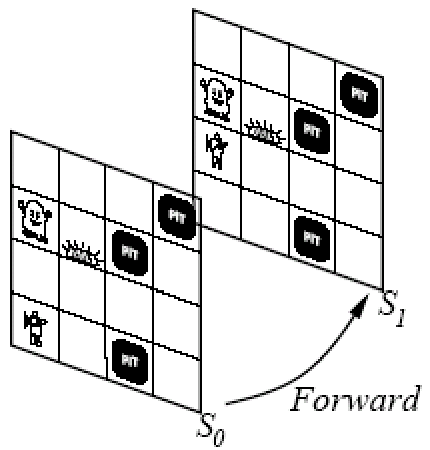
$$\forall y \;\; Breezy(y) \;\Leftrightarrow\; [\exists x \;\; Pit(x) \land Adjacent(x, y)]$$

Keeping track of change

Facts hold in situations, rather than eternally
E.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$

Situation calculus is one way to represent change in FOL:
Adds a situation argument to each non-eternal predicate
E.g., Now in $Holding(Gold, Now)$ denotes a situation

Situations are connected by the $Result$ function
$Result(a, s)$ is the situation that results from doing a in s



Describing actions I

"Effect" axiom—describe changes due to action
$\forall s \;\; AtGold(s) \;\Rightarrow\; Holding(Gold, Result(Grab, s))$

"Frame" axiom—describe non-changes due to action
$\forall s \;\; HaveArrow(s) \;\Rightarrow\; HaveArrow(Result(Grab, s))$

Frame problem: find an elegant way to handle non-change
(a) representation—avoid frame axioms
(b) inference—avoid repeated "copy-overs" to keep track of state

Qualification problem: true descriptions of real actions require endless caveats—what if gold is slippery or nailed down or ...

Ramification problem: real actions have many secondary consequences—what about the dust on the gold, wear and tear on gloves, ...

Yale Shooting Problem

- Fred, Gun
  - alive(0)
  - not loaded(0)
- Load
  - loaded(1)
- Shoot
  - loaded(2) → not alive(3)
- Cannot show
  - Fred not alive at (3) since "loaded(2)" not entailed
  - alive(1), since in "not alive(1)" has a model

Describing actions II

Successor-state axioms solve the representational frame problem

Each axiom is "about" a predicate (not an action per se):

P true afterwards ⇔ [an action made P true
∨ P true already and no action made P false]

For holding the gold:

$$\forall\, a, s\;\; Holding(Gold, Result(a,s)) \;\Leftrightarrow$$
$$[(a = Grab \wedge AtGold(s))$$
$$\vee\; (Holding(Gold, s) \wedge a \neq Release)]$$

Summary

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power: sufficient to define wumpus world