Using FOL

The set domain:

- The only sets are the empty set and the set to which something is added:
  $\forall s\ Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x,s_2\ Set(s_2) \wedge s = \{x|s_2\})$
- No element is added to the empty set:
  $\neg\exists x,s\ \{x|s\} = \{\}$
- Adding an in already existing element to a set has no effect:
  $\forall x,s\ x \in s \Leftrightarrow s = \{x|s\}$
- The only elements of a set are those which have been added to it (x is element of s if s is equal to s_2 which has an added y where y=x or x element of s_2):
  $\forall x,s\ x \in s \Leftrightarrow [\exists y,s_2\} (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$
- A set is a subset of a second set if all elements of the first are elements of the second:
  $\forall s_1,s_2\ s_1 \subseteq s_2 \Leftrightarrow (\forall x\ x \in s_1 \Rightarrow x \in s_2)$
- Two sets are equal if they are mutual subsets:
  $\forall s_1,s_2\ (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$
- An element is in the intersection of two sets if it is in both sets:
  $\forall x,s_1,s_2\ x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$
- An element is in the union of two sets if it is in one of them:
  $\forall x,s_1,s_2\ x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$

Knowledge base for the wumpus world

- **Perception and action:**
  - Agents in Wumpus world receive a 5-element vector as input.
  - Sentence must include information about time-step.
  - A typical perception sentence (here binary predicate with list of input costants):
    Percept([Stench, Breeze, Glitter, None, None],5)
  - Actions can be represented as:
    Turn(Right), Turn(Left), Forward, Shoot, Grab, Release, Climb)
  - Finding the right action at a time:
    $\exists a\ BestAction(a,5)$
  - Perception implicates facts about the current state:
    $\forall t,s,b\ Percept([s,b,Glitter],t) \Rightarrow Glitter(t)$
  - Reflex behavior for agent:
    $\forall t\ Glitter(t) \Rightarrow BestAction(Grab,t)$

Deducing hidden properties

Environment:

- Using list terms with indices to denote the different squares:
  - [1,2] instead of $\text{Square}_{1,2}$
  - $\forall$x,y,a,b *Adjacent*([x,y],[a,b]) $\Leftrightarrow$ [a,b] $\in$ {[x+1,y], [x-1,y],[x,y+1],[x,y-1]}
  - Wumpus
  - Home(X)
  - Pit(X)
- Position of Agent at time:
  - At(Agent,s,t)
- Properties of squares:
  - $\forall$s,t *At*(Agent,s,t) $\wedge$ Breeze(t) $\Rightarrow$ Breezy(s)

Squares are breezy near a pit:

- **Diagnostic** rule---infer cause from effect
  - $\forall$s Breezy(s) $\Rightarrow$ $\exists$r Adjacent(r,s) $\wedge$ Pit(r)
- **Causal** rule---infer effect from cause
  - $\forall$r Pit(r) $\Rightarrow$ [$\forall$s Adjacent(r,s) $\Rightarrow$ Breezy(s)]

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 BestAction(a,5))

- I.e., does the KB entail some best action at *t=5*?
- Answer: *Yes*, {*a/Shoot*} $\leftarrow$ **substitution** (binding list)
- Given a sentence *S* and a substitution σ,
- *S*σ denotes the result of plugging σ into *S*; e.g.,

  S = Smarter(x,y)
  σ = {x/Hillary,y/Bill}
  *S*σ = Smarter(Hillary,Bill)

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

Knowledge engineering in FOL

1. Identify the task
2. Assemble the relevant knowledge
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

correct errors

correct errors

Using FOL

The X domain:

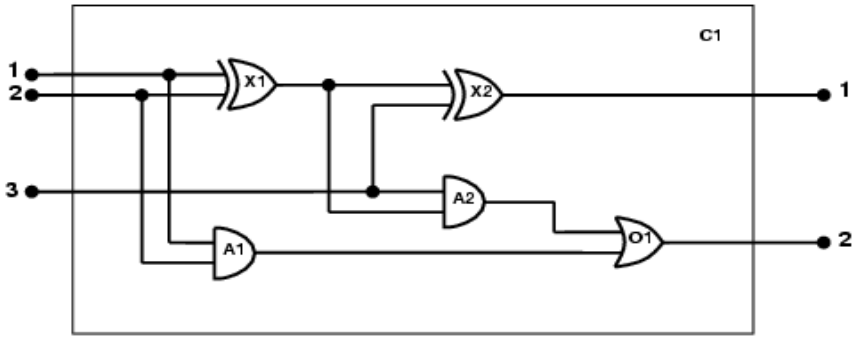
- What is this knowledge base's domain? What "*are*" the relations?

$\forall x\ Z(x) \Leftrightarrow \neg Y1(x)$
$\forall x,y\ W34(x,y) \Leftrightarrow CZ23(y,x)$
$\forall x,y\ ZS2(x,y) \Leftrightarrow ZS2(y,x)$
$\forall x,y\ X1(y) = x \Leftrightarrow (Y1(x) \wedge W34(x,y))$
$\forall x,y\ H21(x,y) \Leftrightarrow (Z1(y) \wedge S45\ (y,x))$
$\forall x,y\ W123(x,y) \Leftrightarrow \exists z\ W34\ (x,z) \wedge W34\ (z,y)$
$\forall x,y\ ZS2(x,y) \Leftrightarrow x \neq y \wedge \exists\ z\ W34(z,x) \wedge W34(z,y)$

- Compare: $\forall m,c\ Mother(c) = m \Leftrightarrow (Female(m) \wedge Parent(m,c))$
- Does this look more familiar?
- ➢ **The kinship domain**
- Using human readable (and understandable) names helps to maintain knowledge base (but might lead to misinterpretation)

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 wires 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

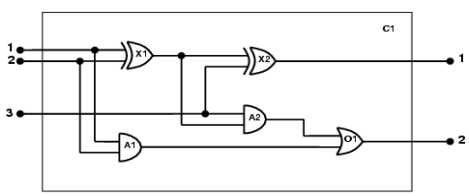
4. Encode general knowledge of the domain

1. Two connected in-/outputs have same signal:
   $\forall t_1,t_2\ Connected(t_1, t_2) \Rightarrow Signal(t_1) = Signal(t_2)$
2. Signal at in-/output has 0 oder 1 (never both):
   $\forall t\ Signal(t) = 1 \vee Signal(t) = 0$
   $1 \neq 0$
3. Connected is commutative:
   $\forall t_1,t_2\ Connected(t_1, t_2) \Rightarrow Connected(t_2, t_1)$
4. Output of OR-gate is 1 iff an input is 1
   $\forall g\ Type(g) = OR \Rightarrow Signal(Out(1,g)) = 1 \Leftrightarrow \exists n\ Signal(In(n,g)) = 1$
5. Output of AND-gate is 0 iff an input is 0:
   $\forall g\ Type(g) = AND \Rightarrow Signal(Out(1,g)) = 0 \Leftrightarrow \exists n\ Signal(In(n,g)) = 0$
6. Output of XOR-gate is 1 iff its inputs are unequal:
   $\forall g\ Type(g) = XOR \Rightarrow Signal(Out(1,g)) = 1 \Leftrightarrow Signal(In(1,g)) \neq Signal(In(2,g))$
7. Output of NOT-gate differs from its input:
   $\forall g\ Type(g) = NOT \Rightarrow Signal(Out(1,g)) \neq Signal(In(1,g))$

The electronic circuits domain

5. Encode the specific problem instance

$Type(X_1) = XOR$ $\quad$ $Type(X_2) = XOR$
$Type(A_1) = AND$ $\quad$ $Type(A_2) = AND$
$Type(O_1) = OR$



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

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

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 $1 \neq 0$

Summary

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