Artificial Intelligence

Chapter 8 (& extra Material)

First Order Logic
Syntax and Semantics

Dr. Mustafa Jarrar

Sina Institute, University of Birzeit

mjarrar@birzeit.edu
www.jarrar.info

Watch this lecture and download the slides from

http://jarrar-courses.blogspot.com/2011/11/artificial-intelligence-fall-2011.html

Reading

This lecture is based on chapter 8 + other material.

Some slides are borrowed Enrico Franconi
http://www.inf.unibz.it/~franconi/dl/course/

(But notice that I introduced some modifications.)

Outline

First Order Logic

- Motivation (why FOL)
- Syntax
- Semantics

Lecture Keywords:

Logic, First Order Logic, FOL, Entailment, Interpretation, Semantics, Formal Semantics, First Order Interpretation, Logical Implication, satisfiable, Unsatisfiable, Falsifiable, Valid, Tautology

المنطق، المنطق الشكلي، المنطق الصوري، المنطق أولي الدرجة، التفسير المنطقي، التفسير الشكلي، تفسير الجمل المنطقية، تحليل القضايا، صحة الجمل المنطقية، الحدود، التناقض،

Motivation

- We can already do a lot with propositional logic.
- But it is unpleasant that we cannot access the *structure of atomic sentences.*
- Atomic formulas of propositional logic are *too atomic . they are just statement.*
- which my be <u>true</u> or <u>false</u> but which have no internal structure.
- In *First Order Logic (FOL) the atomic formulas are interpreted as statements* about ***relationships between objects.***

Predicates and Constants

Let's consider the statements:

- *Mary is female*
- *John is male*
- *Mary and John are siblings*

In propositional logic the above statements are atomic propositions:

- Mary-is-female
- John-is-male
- Mary-and-John-are-siblings

In FOL atomic statements use predicates, with constants as argument:

- Female(mary)
- Male(john)
- Siblings(mary, john)

Variables and Quantifiers

Let's consider the statements:

- *Everybody is male or female*
- *A male is not a female*

In FOL predicates may have variables as arguments, whose value is bounded by quantifiers:

- $\forall x.\ Male(x) \vee Female(x)$
- $\forall x.\ Male(x) \rightarrow \neg Female(x)$

Deduction (why?):

- *Mary is not male*
- $\neg$ Male(Mary)

Functions

Let's consider the statement:

- *The father of a person is male*

In FOL objects of the domain may be denoted by functions applied to (other) objects:

- $\forall x.\ Male(father(x))$

Outline

First Order Logic

- Motivation (why FOL)
- Syntax
- Semantics

Syntax of FOL: atomic sentences

Countably infinite **supply of symbols (*signature*):**

individual constants: $a, b, c, \ldots$

variable symbols: $x, y, z, \ldots$

n-are predicate symbols: $P, Q, R, \ldots$

n-ary function symbols: $f, g, h, \ldots$

Terms: $t \rightarrow x$ variable

$\mid a$ constant

$\mid f(t_1, \ldots, t_n)$ function application

Ground terms: terms that do not contain variables

Formulas: $\phi \rightarrow P(t_1, \ldots, t_n)$ atomic formulas

E.g., Brother(KingJohn; RichardTheLionheart)

>(length(leftLegOf(Richard)), length(leftLegOf(KingJohn)))

Syntax of Propositional Logic

Formulas: $\phi, \psi \rightarrow$		$P(t_1,\ldots,t_n)$	Atomic Formulas
	$\mid$	$\perp$	False
	$\mid$	T	True
	$\mid$	$\neg\phi$	Negation
	$\mid$	$\phi \wedge \psi$	Conjunction
	$\mid$	$\phi \vee \psi$	Disjunction
	$\mid$	$\phi \rightarrow \psi$	Implication
	$\mid$	$\phi \leftrightarrow \psi$	Equivalence

(Ground) **atoms** and (ground) **literals.**

E.g. Sibling(kingJohn, Richard) $\rightarrow$ Sibling(Richard, KingJohn)

$>(1, 2) \vee \leq(1, 2)$

$>(1, 2) \wedge \neg >(1, 2)$

Syntax of First Order Logic

Formulas: $\phi, \psi \rightarrow$	$P(t_1,\ldots,t_n)$	Atomic Formulas
|	$\perp$	False
|	T	True
|	$\neg\phi$	Negation
|	$\phi \wedge \psi$	Conjunction
|	$\phi \vee \psi$	Disjunction
|	$\phi \rightarrow \psi$	Implication
|	$\phi \leftrightarrow \psi$	Equivalence
|	$\forall x.\phi$	Universal quantification
|	$\exists x.\phi$	*Existential* quantification

E.g. Everyone in Italy is smart: $\forall$x. In(x, Italy) $\rightarrow$ Smart(x)

Someone in France is smart: $\exists$x. In(x, France) $\wedge$ Smart(x)

Summary of Syntax of FOL

Terms

- Variables
- Constants
- Functions

Literals

- Atomic Formula
  - Relation (Predicate)
- Negation

Well formed formulas

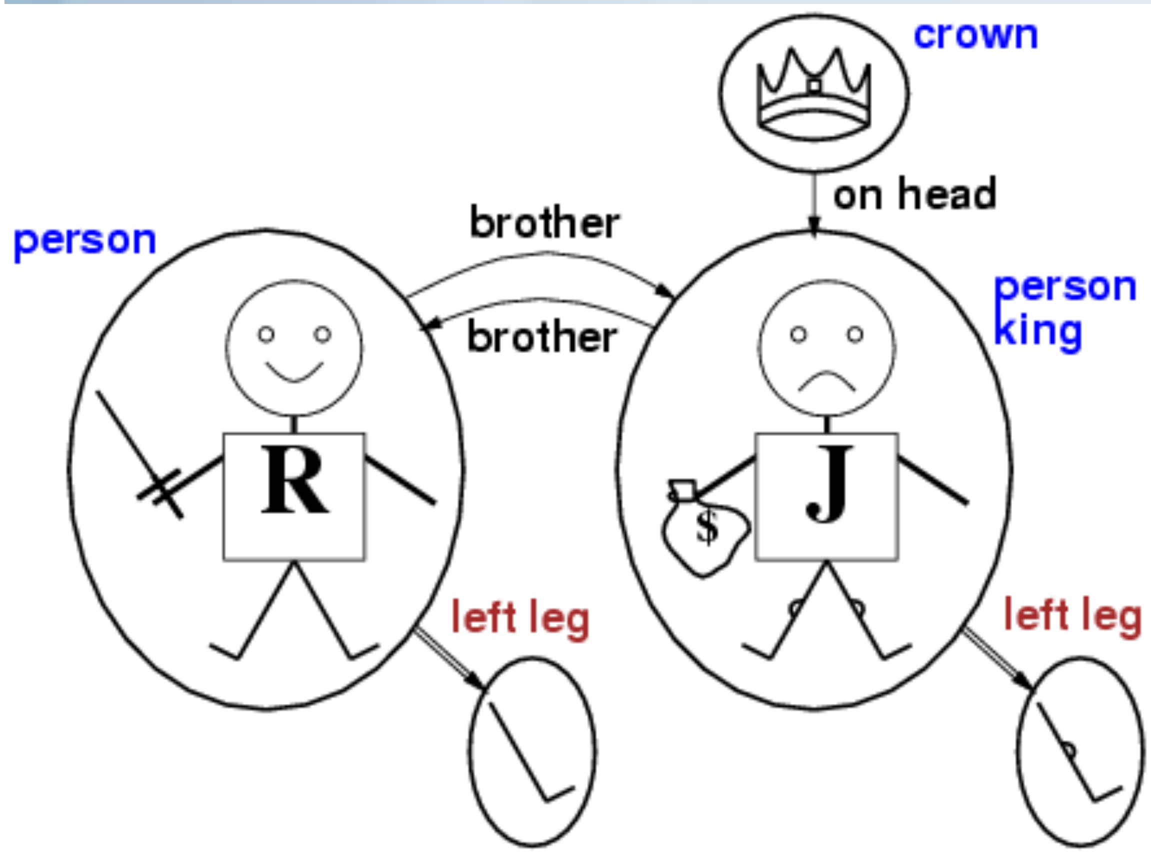
- Truth-functional connectives
- Existential and universal quantifiers

Outline

- **First Order Logic**
  - Motivation (why FOL)
  - Syntax
  - ➡ Semantics (=how to interpret FOL statements)

What is a domain Δ

Δ = Set of objects, relations, and functions



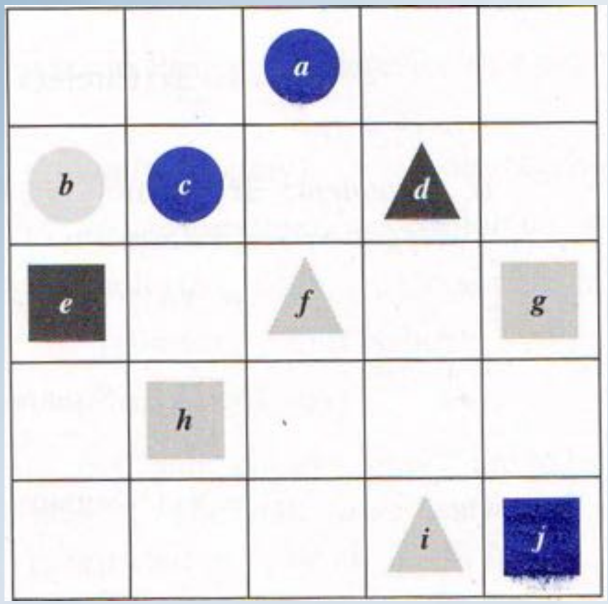
Objects

Relations

Functional relations

Example: Tarski's World

Domain Δ



Δ = objects + relations + functions

How do you interpret these statements?

$\forall x\ \mathrm{Circle}(x) \rightarrow\ \mathrm{Above}(x, f)$

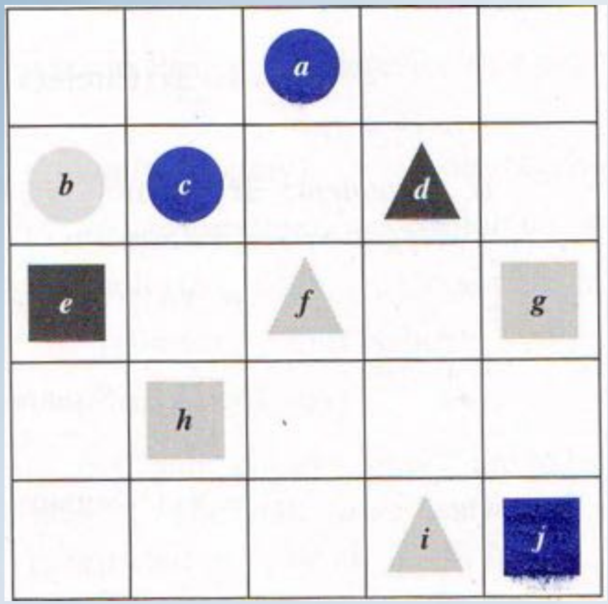
$\exists x\ \mathrm{Square}(x) \wedge \mathrm{Black}(x, f)$

$\forall x\ (\mathrm{Circle}(x) \rightarrow\ \exists x\ (\mathrm{Square}(y) \wedge \mathrm{SameColor}(x, y)))$

$\exists x\ (\mathrm{Square}(x) \wedge \forall y\ (\mathrm{Triangle}(y) \rightarrow \mathrm{RightOf}(x, y)))$

Motivation Example: Tarski's World

Domain Δ



Conceptualization of Domain Δ

Block = {a,b,c,d,e,f,g,h,i,j}
Circle = {a,b,c}
Square = {e,h,g,j}
Triangle = {d,f,i}
Blue = {a,c,j}
Black = {e,d}
….
SameColor = {<a,c>, <a,j>, <c,j>, <b,f>, <b,g>,<b,h>,<b,i>,<f,g><f,h>, <f,i>, <g,h>, <e,d>,}
RightOf = {<a,b>,<a,c>,…,<j,i>}
LiftOf = {<b,a>,<c,a>,…,<I,j>}
Above = {<a,b>,<a,c>,<a,d>,<b,e>,<b,j>…}
….

How do you interpret these statements?

$\forall x\ \text{Circle}(x) \rightarrow\ \text{Above}(x, f)$ ✓

$\exists x\ \text{Square}(x) \wedge \text{Black}(x, f)$ ✓

$\forall x\ (\text{Circle}(x) \rightarrow\ \exists x\ (\text{Square}(y) \wedge \text{SameColor}(x, y)))$ ✓

$\exists x\ (\text{Square}(x) \wedge \forall y\ (\text{Triangle}(y) \rightarrow \text{RightOf}(x, y)))$ ✓

Semantics of FOL: Intuition

- Just like in propositional logic, a (complex) FOL formula may be true (or false) with respect to a given interpretation.

- An interpretation specifies referents for

  constant symbols $\rightarrow$ ***objects***

  predicate symbols $\rightarrow$ ***relations***

  function symbols $\rightarrow$ ***functional relations***

- An atomic sentence $P(t_1,\ldots, t_n)$ is true in a given interpretation iff the *objects referred to by* $t_1,\ldots, t_n$ are in the *relation referred to by the predicate* P.

- An interpretation in which a formula is true is called a ***model*** *for the formula.*

Semantic of FOL statements
(First-Order Interpretations)

Interpretation: $I = \langle \Delta, .^I \rangle$ where Δ is an arbitrary non-empty set and $.^I$ is a function that maps:

- Individual constants to elements of Δ :

  $a^I \in \Delta$

- n-ary predicate symbols to relation over Δ :

  $P^I \subseteq \Delta^n$

- n-ary function symbols to functions over Δ :

  $f^I \in [\Delta^n \rightarrow \Delta]$

Semantic of FOL: Satisfaction

Interpretation of ground terms:

$$(f(t_1,\ldots,t_n))^I \quad = \quad f^I(t^I{}_1,\ldots,t^I{}_n) \ \ (\in \Delta)$$

$$\text{SameColor}(a,j)^I = \text{SameColor}^I(a^I,j^I) \in \Delta$$

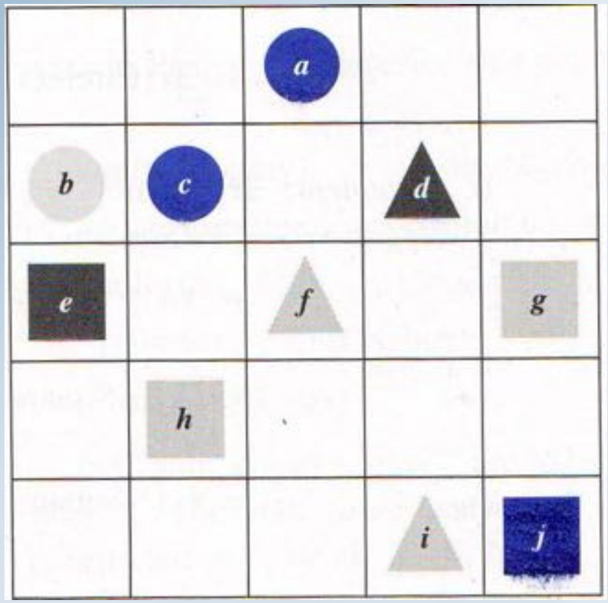
Satisfaction of ground atoms $P(t_1,\ldots,t_n)$:

$$I \models P(t_1,\ldots,t_n) \quad \text{iff} \quad \langle t^I{}_1,\ldots,t^I{}_n \rangle \in P^I$$

$$I \models \text{SameColor}(a,j) \quad \textit{iff} \ \ \langle a^I,j^I \rangle \in \text{SameColor}^I$$

Interpretation Example: Tarski's World

Domain Δ



Conceptualization of Domain Δ

Block = {a,b,c,d,e,f,g,h,i,j}
Circle = {a,b,c}
Square = {e,h,g,j}
Triangle = {d,f,i}
Blue = {a,c,j}
Black = {e,d}
….
SameColor = {<a,c>, <a,j>, <c,j>, <b,f>, <b,g>,<b,h>,<b,i>,<f,g><f,h>, <f,i>, <g,h>, <e,d>,}
RightOf = {<a,b>,<a,c>,…,<j,i>}
LiftOf = {<b,a>,<c,a>,…,<I,j>}
Above = {<a,b>,<a,c>,<a,d>,<b,e>,<b,j>…}
….

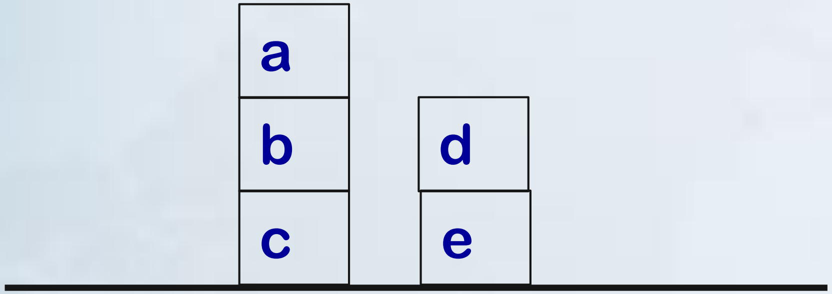
$I \models \text{Circle}(a)$

$I \not\models \text{Circle}(h)$

$I \models \text{SameColor}(g, h)$

$I \not\models \text{Abov}(e, b)$

Interpretation (Example)



Block^{I} = {<a>,<b>,<c>,<d>,<e>}

Above^{I} = {<a,b>,<b,c>,<d,e>}

Clear^{I} = {<a>,<d>}

Table^{I} = {<c>,<e>}

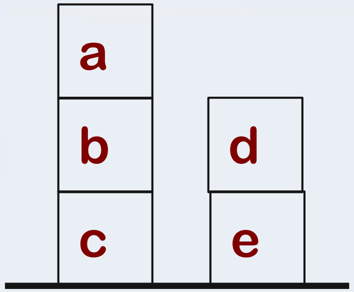
I |= Block(a)

I |≠ Block(f)

I |≠ Above(b,e)

I |= Above(b,c)

Semantics of FOL: Variable Assignments



V set of all variables. Function α: $V \rightarrow \Delta$.

Notation: $\alpha[x/d]$ means assign d to x

Interpretation of terms

$$x^{I,\alpha} = \alpha(x)$$

$$a^{I,\alpha} = a^I$$

$$(f(t_1,..,t_n))^{I,\alpha} = f^I(t_1^{I,\alpha},\ldots,t_n^{I,\alpha})$$

$Above(a,b)^{I,\alpha} = Above^I(b^{I,\alpha}, c^{I,\alpha}$

Satisfiability of atomic formulas:

$$I,\alpha \models P(t_1,\ldots,t_n) \quad \text{iff} \quad \langle t_1^{I,\alpha},\ldots,t_n^{I,\alpha}\rangle \in P^I$$

Variable Assignment example

$$\alpha = \{(x \rightarrow d_1), (y \rightarrow d_2)\}$$

$$I, \alpha \models \text{Red}(x)$$

$$I, \alpha[y/d_1] \models \text{Block}(y)$$

Semantics of FOL: Satisfiability of formulas

A formula ϕ is satisfied by (*is true in)* an interpretation I under a variable assignment α.

$I,\alpha \models \phi$:

$I,\alpha \models P(t_1,\ldots,t_n)$	iff	$\langle t_1^{I,\alpha},\ldots,t_n^{I,\alpha}\rangle \in P^I$
$I,\alpha \models \neg\phi$	iff	$I,\alpha \not\models \phi$
$I,\alpha \models \phi \wedge \psi$	iff	$I,\alpha \models \phi$ and $I\ \alpha \models \psi$
$I,\alpha \models \phi \vee \psi$	iff	$I,\alpha \models \phi$ or $I\ \alpha \models \psi$
$I,\alpha \models \forall x.\phi$	iff	for all $d \in \Delta$: $I, \alpha[x/d] \models \phi$
$I,\alpha \models \exists x.\phi$	iff	there exits a $d \in \Delta$: $I, \alpha[x/d] \models \phi$

Satisfiability and Validity

An interpretation I is a **model** of ϕ under α, if

$$I, \alpha \models \phi$$

Similarly as in propositional logic, a formula ϕ can be **satisfiable, unsatisfiable, falsifiable** or **valid** -the definition is in terms of the pair (I,α).

A formula ϕ is

- **Satisfiable**, if there is some (I, α) that satisfies ϕ,
- **Unsatisfiable**, if ϕ is not satisfiable,
- **Falsifiable**, if there is some (I, α) that does not satisfy ϕ,
- **Valid** (i.e., a **Tautology**), if every (I, α) is a model of ϕ.

Equivalence

Analogously, two formulas are **logically** equivalent ($\phi \equiv \psi$), if for all I; α we have:

$$I, \alpha \models \phi \quad \text{iff} \quad I, \alpha \models \psi$$

Entailment

Entailment is defined similarly as in propositional logic.

The formula ϕ is logically implied by a formula ψ, if ϕ is true in all models of ψ

(symbolically, $\psi \models \phi$):

$$\psi \models \phi \text{ iff } I \models \text{ for all models } I \text{ of } \psi$$

Properties of quantifiers

$(\forall x\,.\forall y.\ \phi)$ is the same as $(\forall y\,.\forall x.\ \phi)$

$(\exists x\ .\ \exists y.\ \phi)$ is the same as $(\exists y\ .\ \exists x.\ \phi)$

$(\exists\, x\,.\forall y.\ \phi)$ is **not** the same as $(\forall y\ .\ \exists x.\ \phi)$

$\exists x\,.\forall y.\ Loves(x,y)$ “There is a person who loves everyone in the world”

$\forall y.\ \exists x.\ Loves(x,y)$ “Everyone in the world is loved by at least one person”

Quantifier duality: each can be expressed using the other

$\forall x.\ Likes(x,Falafel)$ $\neg\exists x.\neg Likes(x,Falafel)$

$\exists x.Likes(x,Salad)$ $\neg\forall x\ \neg Likes(x,Salad)$

Equivalences

$$
\begin{aligned}
(\forall x.\, \phi) \land \psi &\equiv \forall x.\, (\phi \land \psi) \\
(\forall x.\, \phi) \lor \psi &\equiv \forall x.\, (\phi \lor \psi) \\
(\exists x.\, \phi) \land \psi &\equiv \exists x.\, (\phi \land \psi) \\
(\exists x.\, \phi) \lor \psi &\equiv \exists x.\, (\phi \lor \psi) \\
\forall x.\, \phi \land \forall x.\, \psi &\equiv \forall x.\, (\phi \land \psi) \\
\exists x.\, \phi \lor \exists x.\, \psi &\equiv \exists x.\, (\phi \lor \psi)
\end{aligned}
$$

$$
\begin{aligned}
\neg \forall x.\, \phi &\equiv \exists x.\, \neg \phi \\
\neg \exists x.\, \phi &\equiv \forall x.\, \neg \phi
\end{aligned}
$$

& propositional equivalences

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

A simple genealogy KB (Another Example)

- **Build a small genealogy knowledge base by FOL that**
  - contains facts of immediate family relations (spouses, parents, etc.)
  - contains definitions of more complex relations (ancestors, relatives)
  - is able to answer queries about relationships between people

- **Predicates:**
  - parent(x, y), child (x, y), father(x, y), daughter(x, y), etc.
  - spouse(x, y), husband(x, y), wife(x,y)
  - ancestor(x, y), descendent(x, y)
  - relative(x, y)

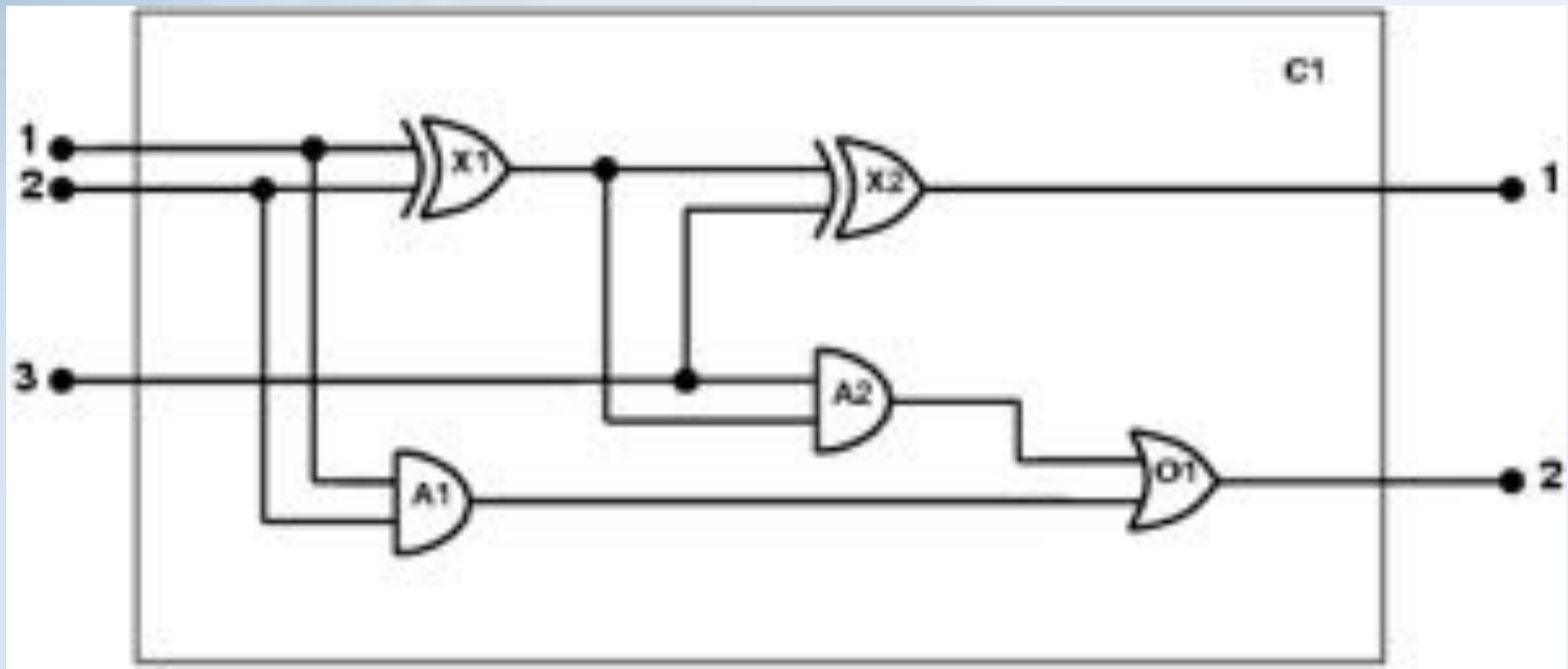
- **Facts:**
  - husband(Joe, Mary), son(Fred, Joe)
  - spouse(John, Nancy), male(John), son(Mark, Nancy)
  - father(Jack, Nancy), daughter(Linda, Jack)
  - daughter(Liz, Linda)
  - etc.

A simple genealogy KB (Another Example)

- **Rules for genealogical relations**
  - (∀x,y) parent(x, y) <=> child (y, x)
    (∀x,y) father(x, y) <=> parent(x, y) ^ male(x) (similarly for mother(x, y))
    (∀x,y) daughter(x, y) <=> child(x, y) ^ female(x) (similarly for son(x, y))
  - (∀x,y) husband(x, y) <=> spouse(x, y) ^ male(x) (similarly for wife(x, y))
    (∀x,y) spouse(x, y) <=> spouse(y, x) (**spouse relation is symmetric**)
  - (∀x,y) parent(x, y) => ancestor(x, y)
    (∀x,y)(∃z) parent(x, z) ^ ancestor(z, y) => ancestor(x, y)
  - (∀x,y) descendent(x, y) <=> ancestor(y, x)
  - (∀x,y)(∃z) ancestor(z, x) ^ ancestor(z, y) => relative(x, y)
    (related by common ancestry)
    (∀x,y) spouse(x, y) => relative(x, y) (related by marriage)
    (∀x,y)(∃z) relative(z, x) ^ relative(z, y) => relative(x, y) (**transitive**)
    (∀x,y) relative(x, y) => relative(y, x) (**symmetric**)
- **Queries**
  - ancestor(Jack, Fred) /* the answer is yes */
  - relative(Liz, Joe) /* the answer is yes */
  - relative(Nancy, Mathews)
    /* no answer in general, no if under closed world assumption */

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
5.
5. $\forall t_1,t_2$ Connected(t_1, t_2) $\Rightarrow$ Signal(t_1) = Signal(t_2)
   - $\forall$t Signal(t) = 1 $\vee$ Signal(t) = 0
   - 1 $\neq$ 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)) $\neq$ Signal(In(2,g))
   - $\forall$g Type(g) = NOT $\Rightarrow$ Signal(Out(1,g)) $\neq$ Signal(In(1,g))

The electronic circuits domain

5. Encode the specific problem instance

$Type(X_1) = XOR$

$Type(X_2) = XOR$

$Type(A_1) = AND$

$Type(A_2) = AND$

$Type(O_1) = OR$

$Connected(Out(1,X_1),In(1,X_2))$

$Connected(Out(1,X_1),In(2,A_2))$

$Connected(Out(1,A_2),In(1,O_1))$

$Connected(Out(1,A_1),In(2,O_1))$

$Connected(Out(1,X_2),Out(1,C_1))$

$Connected(Out(1,O_1),Out(2,C_1))$

$Connected(In(1,C_1),In(1,X_1))$

$Connected(In(1,C_1),In(1,A_1))$

$Connected(In(2,C_1),In(2,X_1))$

$Connected(In(2,C_1),In(2,A_1))$

$Connected(In(3,C_1),In(2,X_2))$

$Connected(In(3,C_1),In(1,A_2))$

The electronic circuits domain

6. Pose queries to the inference procedure
7.
7. What are the possible sets of values of all the terminals for the adder circuit?
8.
8. $\exists i_1,i_2,i_3,o_1,o_2$ 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 $1 \neq 0$