AUTOMATA FORMAL LANGUAGES AND LOGIC

First Order Logic – The Electronic Circuits Domain

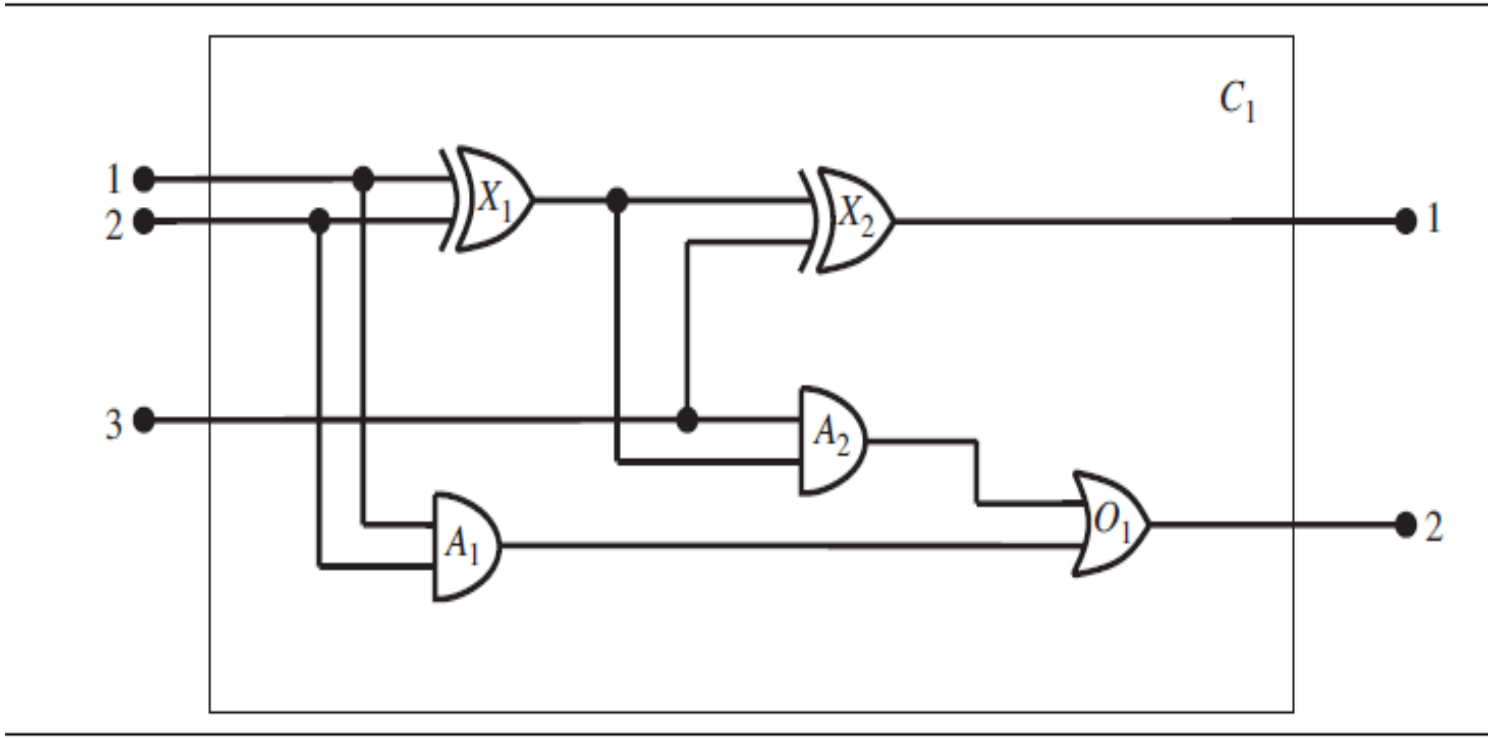
Dr Pooja Agarwal
Department of Computer Science & Engineering

Electronic Circuits Domain

Seven-step process for Knowledge Engineering

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on a vocabulary
4. Encode general knowledge of the domain
5. Encode the specific problem instance
6. Pose queries to the inference procedure
7. Debug the knowledge base

Digital circuits



Decide on a vocabulary

- **Next we consider terminals, which are identified by the predicate Terminal (x)**.
- A gate or circuit can have one or more input terminals and one or more output terminals.
- We use the function ***In(1,X1)*** to denote the first input terminal for gate X1.
- A similar function Out is used for output terminals.
- The function ***Arity(c, i, j)*** says that circuit c has i input and j output terminals.
- The connectivity between gates can be represented by a predicate, Connected,which takes two terminals as arguments, as in ***Connected(Out(1,X1), In(1,X2))***.

Decide on a vocabulary

- Finally, we need to know whether a ***signal*** is on or off.
- One possibility is to use a unary predicate, ***On(t)***, which is true when the signal at a terminal is on.
- This makes it a little difficult, however, to pose questions such as **"What are all the possible values of the signals at the output terminals of circuit C1 ?"**
- We therefore introduce as objects two signal values, 1 and 0, and a function ***Signal (t)*** that denotes the signal value for the terminal t.

Encode general knowledge of the domain

- One sign that we have a good ontology is that we require only a few general rules, which can be stated clearly and concisely.
- **These are all the axioms we will need:**
- ***1. If two terminals are connected, then they have the same signal:***

  $\forall t1, t2\ Terminal(t1) \wedge Terminal(t2) \wedge Connected(t1, t2) \Rightarrow Signal(t1)=Signal(t2)$.
- ***2. The signal at every terminal is either 1 or 0:***

  $\forall t\ Terminal(t) \Rightarrow Signal(t)=1 \vee Signal(t)=0$.
- ***3. Connected is commutative:***

  $\forall t1, t2\ Connected(t1, t2) \Leftrightarrow Connected(t2, t1)$.

Encode general knowledge of the domain

4. There are four types of gates:

$\forall g\ Gate(g) \wedge k = Type(g) \Rightarrow k = AND \vee k = OR \vee k = XOR \vee k = NOT$.

5. An AND gate's output is 0 if and only if any of its inputs is 0:

$\forall g\ Gate(g) \wedge Type(g)=AND \Rightarrow Signal(Out(1, g))= 0 \Leftrightarrow \exists n\ Signal(In(n, g))=0$.

6. An OR gate's output is 1 if and only if any of its inputs is 1:

$\forall g\ Gate(g) \wedge Type(g)=OR \Rightarrow Signal(Out(1, g))= 1 \Leftrightarrow \exists n\ Signal(In(n, g))=1$.

Encode general knowledge of the domain

8. A NOT gate's output is different from its input:

$\forall g\ Gate(g) \wedge (Type(g)=NOT) \Rightarrow Signal(Out(1, g)) = Signal(In(1, g))$.

9. The gates (except for NOT) have two inputs and one output.

$\forall g\ Gate(g) \wedge Type(g) = NOT \Rightarrow Arity(g, 1, 1)$.

$\forall g\ Gate(g) \wedge k = Type(g) \wedge (k = AND \vee k = OR \vee k = XOR) \Rightarrow Arity(g, 2, 1)$

Encode general knowledge of the domain

10. A circuit has terminals, up to its input and output arity, and nothing beyond its arity:

$\forall$ c, i, j Circuit(c) $\wedge$ Arity(c, i, j) $\Rightarrow \forall$ n (n $\leq$ i $\Rightarrow$ Terminal (In(c, n))) $\wedge$ (n > i $\Rightarrow$ In(c, n) = Nothing) $\wedge$

$\forall$ n (n $\leq$ j $\Rightarrow$ Terminal (Out(c, n))) $\wedge$ (n > j $\Rightarrow$ Out(c, n) =Nothing)

11. Gates, terminals, signals, gate types, and Nothing are all distinct.

$\forall$ g, t Gate(g) $\wedge$ Terminal (t) $\Rightarrow$g = t = 1 = 0 = OR = AND = XOR = NOT = Nothing .

12. Gates are circuits. $\forall$ g Gate(g) $\Rightarrow$ Circuit(g)

Encode the specific problem instance

The circuit is encoded as circuit C1 with the following description.

First, we categorize the circuit and its component gates:

$$Circuit(C1) \land Arity(C1, 3, 2)$$
$$Gate(X1) \land Type(X1)=XOR$$
$$Gate(X2) \land Type(X2)=XOR$$
$$Gate(A1) \land Type(A1)=AND$$
$$Gate(A2) \land Type(A2)=AND$$
$$Gate(O1) \land Type(O1)=OR .$$

Encode the specific problem instance

Then, we show the connections between them:

Connected(Out(1,X1), In(1,X2)) Connected(In(1,C1), In(1,X1))
Connected(Out(1,X1), In(2,A2)) Connected(In(1,C1), In(1,A1))
Connected(Out(1,A2), In(1,O1)) Connected(In(2,C1), In(2,X1))
Connected(Out(1,A1), In(2,O1)) Connected(In(2,C1), In(2,A1))
Connected(Out(1,X2), Out(1,C1)) Connected(In(3,C1), In(2,X2))
Connected(Out(1,O1), Out(2,C1)) Connected(In(3,C1), In(1,A2))

Pose queries to the inference procedure

What combinations of inputs would cause the first output of C1 (the sum bit) to be 0 and the second output of C1 (the carry bit) to be 1?

$\exists$ i1, i2, i3 Signal (In(1, C1))=i1 $\wedge$ Signal (In(2, C1))=i2 $\wedge$ Signal (In(3, C1))=i3$\wedge$ Signal (Out(1, C1))=0 $\wedge$ Signal (Out(2, C1))=1 .

The answers are substitutions for the variables i1, i2, and i3 such that the resulting sentence is entailed by the knowledge base.

ASKVARS will give us three such substitutions:

{i1/1, i2/1, i3/0} {i1/1, i2/0, i3/1} {i1/0, i2/1, i3/1} .

Pose queries to the inference procedure

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

$\exists$ i1, i2, i3, o1, o2 Signal (In(1, C1))=i1 $\wedge$ Signal (In(2, C1))=i2 $\wedge$ Signal (In(3, C1))=i3 $\wedge$ Signal (Out(1, C1))=o1 $\wedge$ Signal (Out(2, C1))=o2 .

Debug the knowledge base

- We can perturb the knowledge base in various ways to see what kinds of erroneous behaviors emerge.
- For example, we can ask ∃ i1, i2, o Signal (In(1,C1))=i1 ∧ Signal (In(2,C1))=i2 ∧ Signal (Out(1,X1)), which reveals that no outputs are known at X1 for the input cases 10 and 01.
- Then, we look at the axiom for XOR gates, as applied to X1:

Signal (Out(1,X1))=1 ⇔ Signal (In(1,X1)) = Signal (In(2,X1)) .

- If the inputs are known to be, say, 1 and 0, then this reduces to Signal (Out(1,X1))=1 ⇔ 1 = 0 .
- Now the problem is apparent: the system is unable to infer that Signal (Out(1,X1))=1, so we need to tell it that 1 = 0.

Reference

- Some of the definitions/diagrams/examples are from the book “Artificial Intelligence – A Modern Approach”, Stuart Russell and Peter Norvig, Pearson, 3rd Edition (Paperback),2016

THANK YOU

Pooja Agarwal
Department of Computer Science & Engineering

poojaagarwal@pes.edu