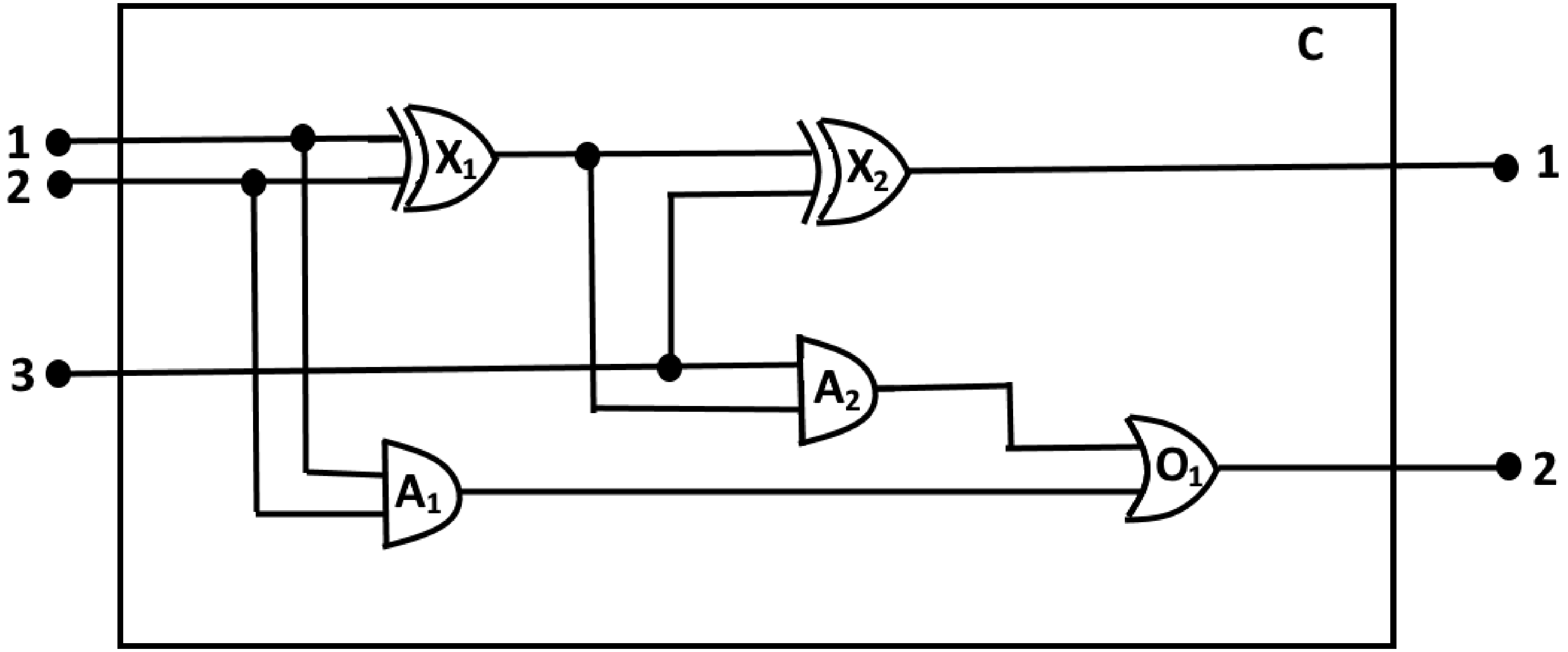
Model problems using FOL

Andrea Galassi

Languages and Algorithms for Artificial Intelligence

Case study from Russel-Norvig's AIMA 3rd edition, Chapter 8.4.2

Model a digital circuit using FOL



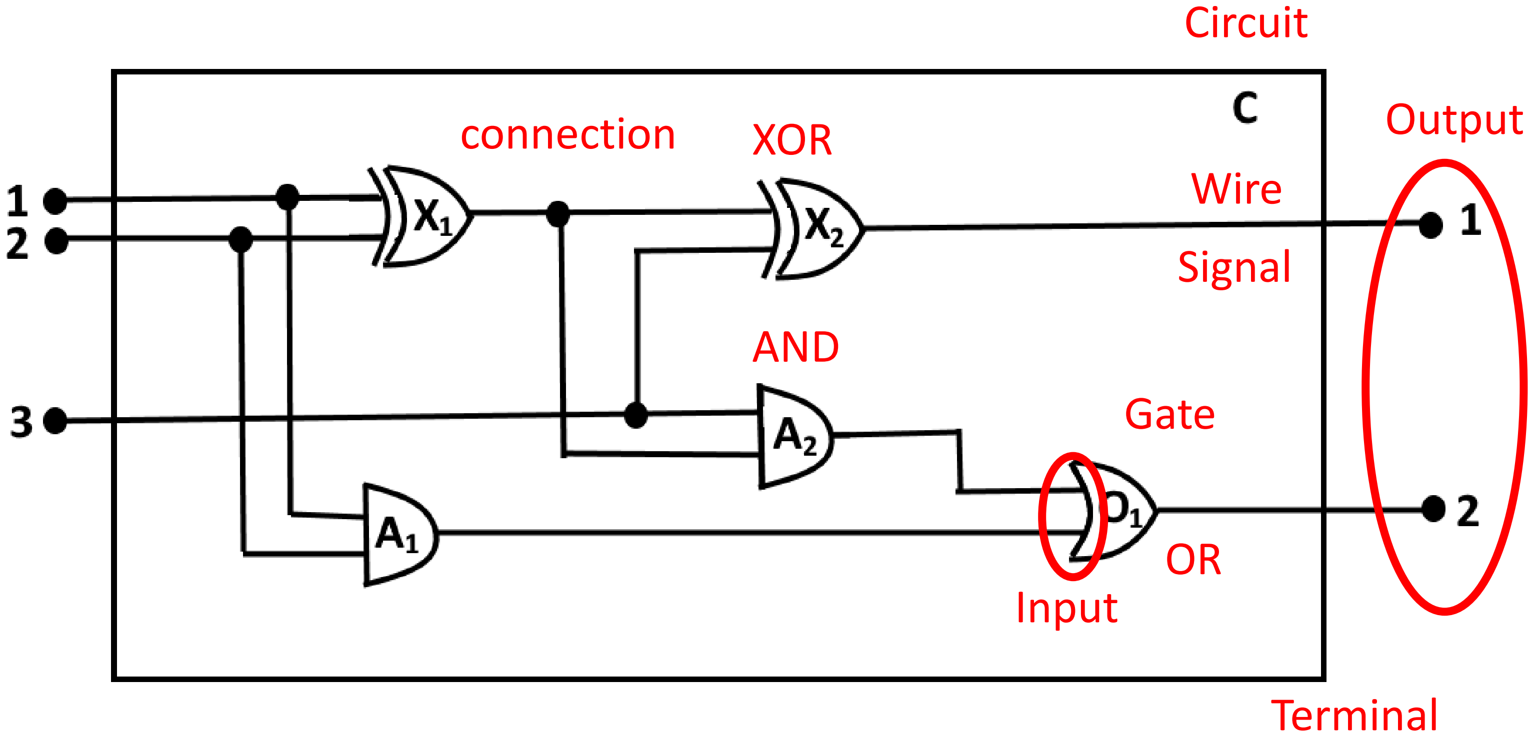
How to proceed?

1) Identify the relevant knowledge
2) Decide a vocabulary
3) Encode general knowledge of the domain
4) Encode the specific instance of the problem

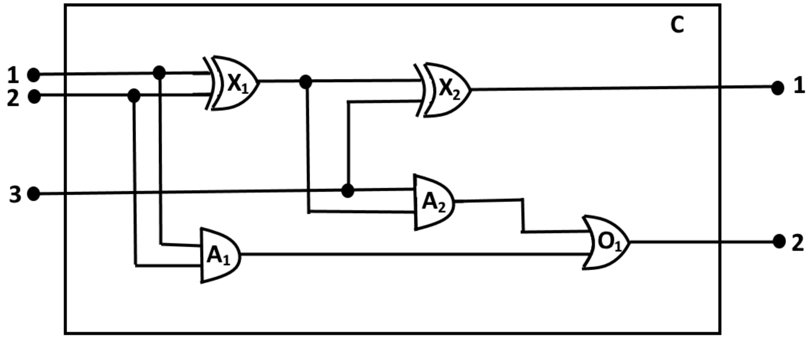
1. Identify the relevant knowledge

What do we need to know about digital circuits?

What “entities” are involved in a digital circuit?



2. Decide the vocabulary



Entities:

- Circuit
- Output, Input, Terminal
- Signal (0, 1)
- Gate (OR, AND, XOR)
- Connection

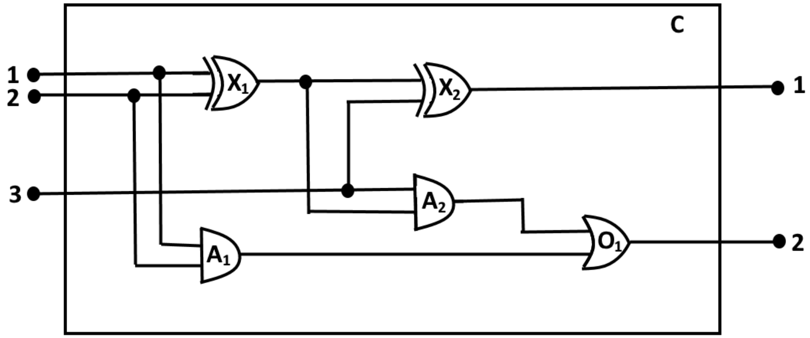
Predicates:

Functions:

Constants:

C, X1, X2, A1, A2, O2,

2. Decide the vocabulary



Entities:

- Circuit
- Output, Input, Terminal
- Signal (0, 1)
- Gate (OR, AND, XOR)
- Connection

Constants:

C, X1, X2, A1, A2, O2,

Predicates:

Circuit(C)

Gate(X1)

Connected(Out(X1),In(X2))

Connected(t1, t2)

Terminal(t)

Model(c, i, o)

Circuit c has i inputs, o outputs

Functions:

Type(X1)

Type of gate X1

In(1, C)

Input #1 of C

Out(1, C)

Output #1 of C

Signal(t)

Value of the signal of terminal t

3. Knowledge of the domain (1/4)

Which terms are distinct?

- Terms that are not distinct:

- Terms that are distinct:

3. Knowledge of the domain (1/4)

Which terms are distinct?

- Terms that are not distinct: Gates are Circuits

- Terms that are distinct: Gates, Terminals, Signals, Gate Types, nothing

3. Knowledge of the domain (1/4)

Which terms are distinct?

- Terms that are not distinct: Gates are Circuits

$\forall g\ Gate(g) \Rightarrow Circuit(g)$

- Terms that are distinct: Gates, Terminals, Signals, Gate Types, nothing

$\forall g,t\ Gate(g) \wedge Terminal(t) \Rightarrow$

$g \neq t \neq 1 \neq 0 \neq OR \neq AND \neq XOR \neq NOT \neq Nothing$

3. Encode Knowledge of the domain (2/4)

Signals and connections

- Value of the signals

- If two terminals are connected,

- Property of connected:

3. Encode Knowledge of the domain (2/4)

Signals and connections

- Value of the signals is 1 or 0

- If two terminals are connected, they have the same signal

- Property of connected: commutative property

3. Encode Knowledge of the domain (2/4)

Signals and connections

- Value of the signals is 1 or 0

$\forall t\ Terminal(t) \Rightarrow Signal(t) = 1 \vee Signal(t) = 0$

- If two terminals are connected, they have the same signal

$\forall t1,t2\ Terminal(t1) \wedge Terminal(t2) \wedge Connected(t1,t2) \Rightarrow$
$Signal(t1) = Signal(t2)$

- Property of connected: commutative property

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

3. Encode Knowledge of the domain (3/4)

- Types of gates
- AND gate:
- OR gate:
- XOR gate:

3. Encode Knowledge of the domain (3/4)

- Types of gates

- AND gate: output 0 when one input is 0

- OR gate: output 1 when one input is 1

- XOR gate: output 1 when both inputs are different

3. Encode Knowledge of the domain (3/4)

- Types of gates

$\forall g\ Gate(g) \land Type(g)=k \Rightarrow k=AND \lor k=OR \lor k=XOR$

- AND gate: output 0 when one input is 0

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

- OR gate: output 1 when one input is 1

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

- XOR gate: output 1 when both inputs are different

$\forall g\ Gate(g) \land Type(g)=XOR \Rightarrow Signal(Out(1,g))=1 \Leftrightarrow Signal(In(1,g)) \neq Signal(In(2,g))$

3. Encode Knowledge of the domain (4/4)

Use model to establish the number of I/O terminal of circuits

- Model of the gates:

- How does Model limit the number of terminals?

3. Encode Knowledge of the domain (4/4)

Use model to establish the number of I/O terminal of circuits

- Model of the gates: gates have 2 inputs and one output

- How does Model limit the number of terminals?

A circuit has terminals, up to its input/output model and nothing beyond that

$\forall$c,i,o Circuit(c) $\wedge$ Model(c,i,o) $\Rightarrow$

3. Encode Knowledge of the domain (4/4)

Use model to establish the number of I/O terminal of circuits

- Model of the gates: gates have 2 inputs and one output

$\forall g\ Gate(g) \Rightarrow Model(g, 2,1)$

- How does Model limit the number of terminals?

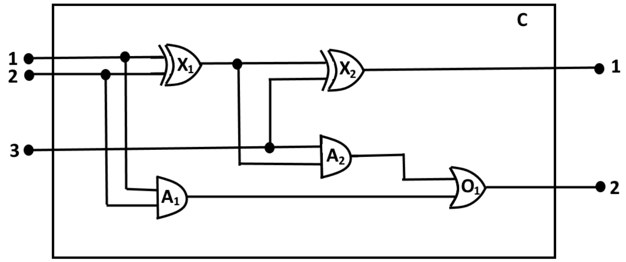
A circuit has terminals, up to its input/output model and nothing beyond that

$\forall c,i,o\ Circuit(c) \wedge Model(c,i,o) \Rightarrow$

$\quad \forall n\ (i \geq n \Rightarrow Terminal(In(c,n))) \wedge (\forall n > i \Rightarrow In(c,n) = Nothing) \wedge$

$\quad \forall n\ (o \geq n \Rightarrow Terminal(Out(c,n))) \wedge (\forall n > o \Rightarrow Out(c,n) = Nothing)$

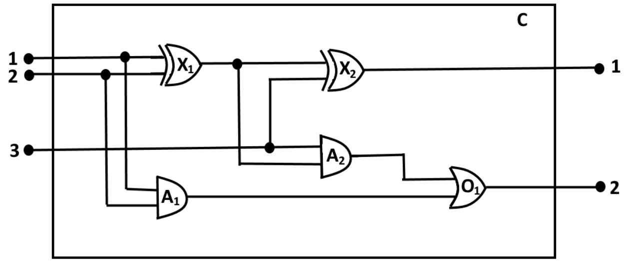
4. Encode Problem Instance (1/2)



What are the predicate and the functions that do apply to our constants?

- C
- X1
- X2
- A1
- A2
- O1

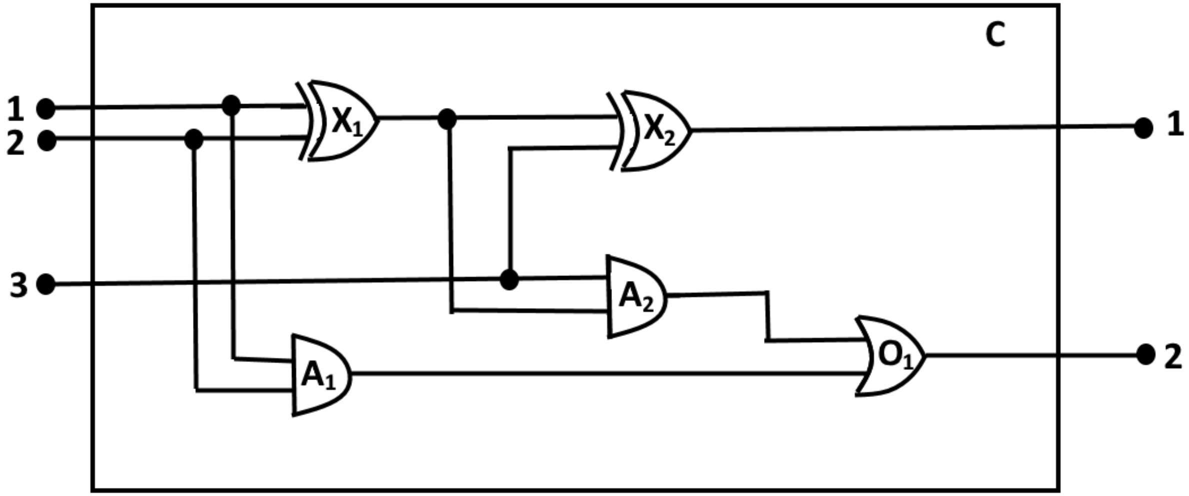
4. Encode Problem Instance (1/2)



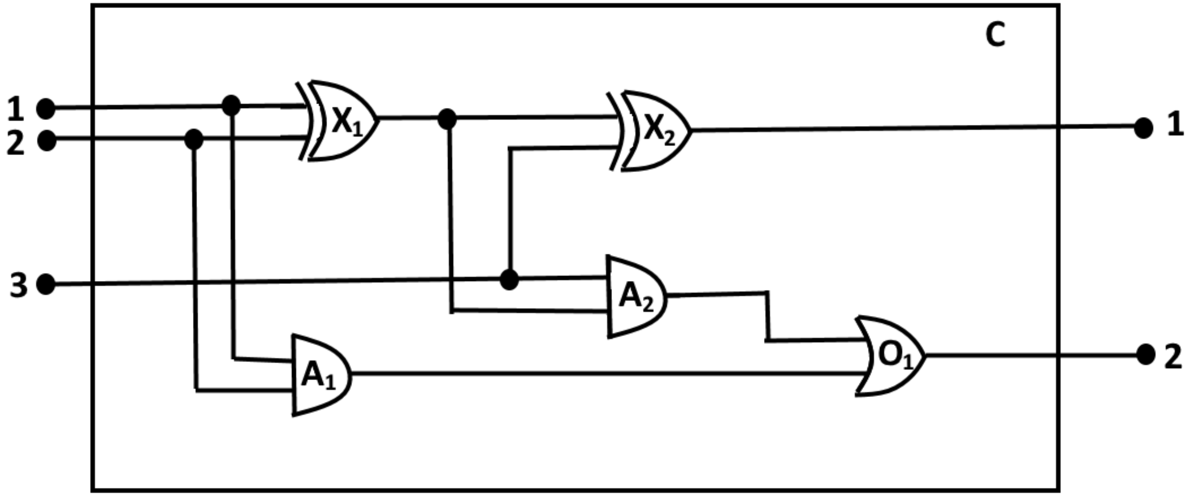
What are the predicate and the functions that do apply to our constants?

- C   Circuit(C) $\wedge$ Model(C,3,2)
- X1   Gate(X1) $\wedge$ Type(X1) = XOR
- X2   Gate(X2) $\wedge$ Type(X2) = XOR
- A1   Gate(A1) $\wedge$ Type(A1) = AND
- A2   Gate(A2) $\wedge$ Type(A2) = AND
- O1   Gate(O1) $\wedge$ Type(O1) = OR

4. Encode Problem Instance (2/2)



4. Encode Problem Instance (2/2)



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

Connected(Out(1, X2), Out(1, C1))
Connected(Out(1, O1), Out(1, C1))

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