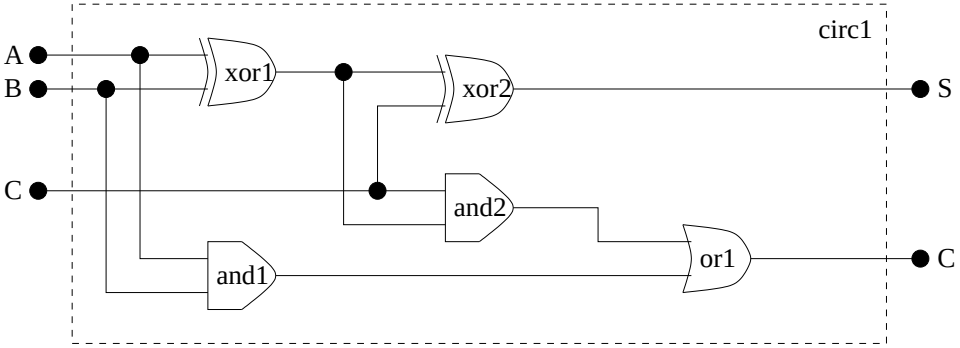
Knowledge Engineering Case Study

1 The Domain

We're going to build a knowledge base for a very simple domain: combinatorial circuits. Here's a circuit comprising two XOR-gates, two AND-gates and one OR-gate which, it is claimed, acts as a full-adder. In other words, it takes in three bits (the two to be added and a previous carry bit) and it outputs two bits (the sum and the carry).



This is a simple domain for a number of reasons: it's easy to know, and hence encode, everything about it; there is no uncertainty or indefiniteness (or, at least, none that we're going to worry about); and it's made up entirely of physical objects.

We want to use this knowledge base to verify circuits, i.e. our queries will ask about signals at various points in the circuits, so that we can verify whether the circuit computes whatever its designer claims that it computes.

2 Ontological Engineering

2.1 Decide what to talk about

Question. What are the relevant concepts?

The fact that we have a particular task here (circuit verification) simplifies our task. It means that, during ontological engineering, we now know that various concepts are not relevant. If our task were something other than circuit verification, other concepts might be relevant. For example, if we wanted a knowledge base for debugging circuits, then wires would be relevant, since we would have to reason about whether a wire was faulty; time might also be relevant as we would need to handle timing faults. If our task was circuit design, then costs and speeds might be relevant.

2.2 Decide on a vocabulary

Question. What predicate symbols, function symbols and constant symbols should we use?

3 Axiomatisation

3.1 Encode general rules

The axioms we need will depend on the ontological engineering: different decisions there will have ramifications here.

Question. What axioms do we need?

3.2 Encode specific knowledge

Now we can axiomatise specific circuits about which we wish to ask questions.

Question. How will you axiomatise the full adder from earlier?

Once this has been done, we can `tell` our knowledge base all the axioms we have come up with, and then we can `ask` it questions. For example, we could ask the following question: what combination of inputs to the circuit would cause the first output of the circuit (the sum) to be off and the second output (the carry) to be on.

Question. Convert the question into logic.

4 Concluding Remarks

Encoding a small, simple domain may seem hard enough, but it is easy compared with encoding more general knowledge, especially what we might call commonsense knowledge.

Consider trying to represent even the knowledge that every child knows: a liquid poured into a cup will take on the shape of that cup; if the cup is turned over, the liquid runs out; people don't exist until they are born; people typically sleep at night; people seek shelter in heavy rain; an object can't be in two places at once. And so on, and so on. There is a myriad of knowledge about objects, parts of objects, classes of objects, measurements, events, states, processes, times, space, mental objects, mental events, mental states and mental processes.

If we are to build an agent that can exist in a rich environment, especially if we are to build a robot that can exist in the physical world, it needs this commonsense knowledge. And yet AI has not made very great inroads into representing this knowledge. Somehow it seems harder to represent this commonsense knowledge than it is to represent more specialised, scientific knowledge. The naïve physics that gets us through our daily lives is hard to elicit and represent; the abstruse physics theories of physics professors somehow is more easily captured.

Question. Why do you think it is so difficult to represent commonsense knowledge?

In the future, artificially intelligent agents might learn more of this knowledge themselves (through experimentation in the world, through conversing, and through consuming sources of information such as books, TV programmes, films and web sites). But in the short- to medium-term, we still rely on knowledge engineering. But capturing this knowledge is a daunting task. Notable is the Cyc project — a huge knowledge engineering effort. But this has had limited success. (Follow the links on the web site.)