6.825 Techniques in Artificial Intelligence

Logic Miscellanea

- Completeness and Incompleteness
- Equality
- Paramodulation

Logic is a huge subject. It includes esoteric mathematical and philosophical arguments as well as hard-core engineering of knowledge representations and efficient inference algorithms. I’m going to use this lecture to cover a set of random topics in logic that, even if you don’t understand them exactly, it’s good to know a little bit about.

The paramodulation rule will also be important to homework 2.

Completeness and Decidability

Complete: If KB $\models \alpha$ then KB $\vdash \alpha$

- If it's entailed, there is a proof

Semi-decidable:

- If there's a proof, we'll halt with it
- If not, maybe halt, maybe not

We found that resolution for propositional logic was sound and complete. Recall that a proof system is complete if whenever a conclusion is entailed by some premises, it can be proved from those premises.

Completeness and Decidability

Complete: If KB $\models \alpha$ then KB $\vdash \alpha$

- If it's entailed, there is a proof

Semi-decidable:

- If there's a proof, we'll halt with it
- If not, maybe halt, maybe not

Gödel's Completeness Theorem: *There exists a complete proof system for FOL*

Godel proved a completeness theorem for first-order logic: There exists a complete proof system for FOL. But, living up to his nature as a very abstract logician, he didn't come up with such a proof system; he just proved one existed.

Completeness and Decidability

Complete: If KB $\models \alpha$ then KB $\vdash \alpha$

- If it's entailed, there is a proof

Semi-decidable:

- If there's a proof, we'll halt with it
- If not, maybe halt, maybe not

Gödel's Completeness Theorem: *There exists a complete proof system for FOL*

Robinson's Completeness Theorem: *Resolution refutation is a complete proof system for FOL*

Then, Robinson came along and showed that resolution refutation is sound and complete for FOL.

Completeness and Decidability

Complete: If KB $\models \alpha$ then KB $\vdash \alpha$

- If it's entailed, there is a proof

Semi-decidable:

- If there's a proof, we'll halt with it
- If not, maybe halt, maybe not

Gödel's Completeness Theorem: *There exists a complete proof system for FOL*

Robinson's Completeness Theorem: *Resolution refutation is a complete proof system for FOL*

FOL is semi-decidable: if the desired conclusion follows from the premises then eventually resolution refutation will find a contradiction.

This makes first-order logic semi-decidable. If the answer is "yes", that is, if there is a proof, then the theorem prover will eventually halt and say so. But if there isn't a proof, it might run forever! (unlike in the propositional case, in which it is guaranteed that if there isn't a proof, resolution will eventually stop).

Adding Arithmetic

So, things are relatively good with regular first-order logic. And they're still fine if you add addition to the language. But if you add addition and multiplication, it starts to get weird!

Adding Arithmetic

Gödel's Incompleteness Theorem: *There is no consistent, complete proof system for FOL + Arithmetic.*

Either there are sentences that are true, but not provable or there are sentences that are provable, but not true.

Godel also had an **in**completeness theorem, which says that there is no consistent, complete proof system for FOL plus arithmetic. (Consistent is the same as sound). Either there are sentences that are true, but not provable, or there are sentences that are provable, but not true. It's not so good either way.

Adding Arithmetic

Gödel's Incompleteness Theorem: *There is no consistent, complete proof system for FOL + Arithmetic.*

Either there are sentences that are true, but not provable or there are sentences that are provable, but not true.

Arithmetic gives you the ability to construct code-names for sentences within the logic.

P = "P is not provable."

Here's the roughest cartoon of how the proof goes. Arithmetic gives you the ability to construct code names for sentences within the logic, and therefore to construct sentences that are self-referential. This sentence, P, is sometimes called the Godel-sentence. P is "P is not provable".

Adding Arithmetic

Gödel's Incompleteness Theorem: *There is no consistent, complete proof system for FOL + Arithmetic.*

Either there are sentences that are true, but not provable or there are sentences that are provable, but not true.

Arithmetic gives you the ability to construct code-names for sentences within the logic.

P = "P is not provable."

- If P is true: it's not provable (incomplete)
- If P is false: it's provable (inconsistent)

If P is true, then P is not provable (so the system is incomplete). If P is false, then P is provable (so the system is inconsistent). Too bad!

Equality

We need to talk a bit about equality. So far we've been doing proofs in a language with predicates and functions, but no equality statements. When we talked about the definition of first-order logic we put equality in our language, so the question is what do you do in resolution, if you have equality, if you want to talk about things being equal to each other.

Equality

$\forall$ x. x = x

$\forall$ xy. x=y $\rightarrow$ y = x

$\forall$ xyz. x=y Æ y = z $\rightarrow$ x = z

One solution is to just treat equality like any other relation and give axioms that specify how it has to work. So, for instance, equality is an equivalence relation, which means it's symmetric, reflexive, and transitive. We can say that in logic like this.

Equality

$\forall$ x. x = x

$\forall$ xy. x=y $\rightarrow$ y = x

$\forall$ xyz. x=y Æ y = z $\rightarrow$ x = z

$\forall$ xy. x=y $\rightarrow$ (P(x) $\leftrightarrow$ P(y))

etc., etc., etc., …

But then, very sadly, you need more than that. You need to say, for every predicate P, for all X and Y if X = Y then (P(X) if and only if P(Y)). The idea is that you should be able to substitute equals for equals into any context. That doesn't follow from our equality axioms, and so we'd have add a new axiom for every predicate in our language that says it's okay to substitute equals into it. That could get to be pretty tedious.

Paramodulation

Need one more rule to deal with resolution and equality.

So what typically happens is that people build special inference procedures for equality into the proof procedure, so in resolution there's a rule called paramodulation. I'll show it to you by example. You'll get the idea. The book talks about demodulation. This is a slightly more complicated version, but it's not too much more complicated.

Paramodulation

Need one more rule to deal with resolution and equality.

$$\frac{\begin{array}{l} F(x) = B \\ Q(y) \vee W(y, F(y)) \end{array}}{Q(y) \vee W(y, B)}$$

We'll start by looking at a simple example. What if you know that F(x) = B, and that either Q(y) or W(y, F(y)). It seems like you should be able to substitute B in for F(y), because F(x) unifies with F(y). And you can! You can conclude Q(y) or W(y,B).

Paramodulation

Need one more rule to deal with resolution and equality.

$$\frac{\begin{array}{l}\alpha \vee (s = t) \\ \beta \vee \gamma[r]\end{array}}{(\alpha \vee \beta \vee \gamma[t])\theta}$$

$\gamma[r]$ is a literal containing term r

$\theta = \text{unify}(s, r)$

$$\frac{\begin{array}{l}F(x) = B \\ Q(y) \vee W(y, F(y))\end{array}}{Q(y) \vee W(y, B)}$$

Here's the general paramodulation rule. Like resolution, it lets you take two clauses and make a third with elements from the first two. The first clause has to have an equality (we'll call it s = t) in it. The second clause has to have some literal, which we'll call gamma that contains a term r, such that r unifies with s. Then, the basic idea is that, since r matches with s, and since s equals t, then we should be able to substitute r in for t. And we're allowed to do that, but with some additional work. First of all, it might have required application of a substitution theta to make r unify with s, and we'll have to apply that substitution to the whole resulting clause. Secondly, we might not have known, unconditionally, that s = t. If s = t actually occurs in a clause with other literals, then we need to disjoin those literals into our resulting sentence.

Paramodulation

Need one more rule to deal with resolution and equality.

$$\frac{\alpha \vee (s = t) \qquad \beta \vee \gamma[r]}{(\alpha \vee \beta \vee \gamma[t])\theta}$$

$\gamma[r]$ is a literal containing term r

$\theta = \text{unify}(s, r)$

$$\frac{F(x) = B \qquad Q(y) \vee W(y, F(y))}{Q(y) \vee W(y, B)}$$

where

$s = F(x)$

$t = B$

$\gamma[\cdot] = W(y, \cdot)$

$r = F(y)$

$\theta = \{x / y\}$

So, to see how we can do our simple example with paramodulation, here is the mapping from the abstract variables in the rule to the actual components of the example.

Paramodulation

Need one more rule to deal with resolution and equality.

$$\frac{\begin{array}{l}\alpha \vee (s = t) \\ \beta \vee \gamma[r]\end{array}}{(\alpha \vee \beta \vee \gamma[t])\theta}$$

$\gamma[r]$ is a literal containing term r

$\theta = \text{unify}(s, r)$

$$\frac{\begin{array}{l}F(x) = B \\ Q(y) \vee W(y, F(y))\end{array}}{Q(y) \vee W(y, B)}$$

$$\frac{\begin{array}{l}P(x) \vee F(x) = B \\ Q(y) \vee W(y, F(y))\end{array}}{P(y) \vee Q(y) \vee W(y, B)}$$

where

$s = F(x)$

$t = B$

$\gamma[\cdot] = W(y, \cdot)$

$r = F(y)$

$\theta = \{x / y\}$

Now we can make our example a bit more complicated, adding an additional literal to the first sentence. Now we're required to include it in the result, and to apply the substitution theta to it, as well.

Recitation Problems

Formalize each group of sentences (using the given function and predicate symbols), then prove the last from the others using resolution and paramodulation.

Jane's lover drives a red car.
Fred is the only person who drives a red car.
Therefore, Fred is Jane's lover.
(L(x) = the lover of x; D(x) = x drives a red car)

Mrs. Abbot only teaches good students.
John and Mary have the same teacher.
Mrs. Abbot is Mary's teacher.
Therefore, John is a good student.
(T(x) = the teacher of x; G(x) = x is a good student)

Please do these recitation problems before the next recitation.

More Recitation Problems

- Every part is either made by FooCorp or BarCorp.
- All fragile parts are stored in the warehouse of their manufacturer.
- BarCorp can't manufacture titanium parts.
- The part I need is fragile and made of titanium.
- Therefore, the part I need is the FooCorp's warehouse.

(M(x) = the manufacturer of part x; W(x,y) = part x is stored in the warehouse of company y; T(x) = part x is made of titanium; F(x) part x is fragile; use a constant for "the part I need".)