L09: Propositional Logic Proof

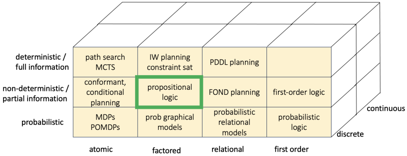
AIMA4e: Chapter 7.5

What you should know after this lecture

- Propositional resolution theorem proving

Reasoning about partially-specified world states

Factored states
Boolean-valued factors



Implication and entailment

What is the difference between $\alpha \Rightarrow \beta$ and $\alpha \models \beta$?

- $\alpha \Rightarrow \beta$ is a sentence in propositional logic.
  - It can be manipulated by a theorem prover.
  - We (mathematicians) can't say whether it's true or false.
  - We can say whether it holds in some model m
- $\alpha \models \beta$ is a mathematical claim.
  - It can't be manipulated by a theorem prover (unless we are trying to encode math in logic (Russell and Whitehead tried this with first-order logic and ran aground.))
  - We (mathematicians) can say whether it's true or false.

Here are some entailments:

- $A \wedge B \models B$
- $A \models A \vee B$
- $A \not\models B$
- **False** $\models A$
- **False** $\models$ **True**
- The only sentence that **True** entails is **True**
- The only sentence that entails **False** is **False**

Implication and entailment

You can prove (using simple set theory on sets of models):

Theorem
If $\textbf{\textit{True}} \models (\alpha \Rightarrow \beta)$ *then* $\alpha \models \beta$.

Theorem
If $\alpha \models \beta$ *then* $\textbf{\textit{True}} \models (\alpha \Rightarrow \beta)$.

Syntactic proof

Recall, a proof procedure takes two sentences, α and β, and tells you whether it can prove β from α:

$$\alpha \vdash \beta$$

Proof procedure is

- sound iff for all α, β, if $\alpha \vdash \beta$ then $\alpha \models \beta$
- complete iff for all α, β, if $\alpha \models \beta$ then $\alpha \vdash \beta$

We have looked at proof procedures that operate via enumerating models. But that is inefficient in many cases.
So, we will look at purely syntactic proof, that operates entirely on logical sentences.

One proof strategy: refutation

To prove $\alpha \models \beta$:

- Write α as one or more premises
- Inference rules tell you what you can add to your proof given what you already have. Logic is monotonic.
- When the rules have allowed you to write down β, then you're done.

Proof by refutation:

- To prove $\alpha \models \beta$
- Instead show that $\alpha \wedge \neg\beta \models$ **False**

Inference rules:

- Lots of interesting proof systems (sets of inference rules)
- We would like one that is sound and complete: $(\alpha \vdash \beta) \equiv (\alpha \models \beta)$
- Refutation using the resolution inference rule is sound and complete!!

Inference rules

General inference rule form: If you have α and β written down in your proof, you can now write γ.

$$\frac{\alpha \quad \beta}{\gamma}$$

Some "natural deduction" inference rules (don't learn these!):

- Modus Ponens
$$\frac{\alpha \Rightarrow \beta \quad \alpha}{\beta}$$
- Modus Tollens
$$\frac{\alpha \Rightarrow \beta \quad \neg\beta}{\neg\alpha}$$
- And introduction
$$\frac{\alpha \quad \beta}{\alpha \wedge \beta}$$
- Or introduction
$$\frac{\alpha}{\alpha \vee \beta}$$
- And elimination
$$\frac{\alpha \wedge \beta}{\alpha}$$

Resolution: One rule to prove them all!

Propositional Resolution (where Q_i and R_j are literals):

$$\frac{(P \vee Q_1 \vee \ldots, \vee Q_n) \quad (\neg P \vee R_1 \vee \ldots \vee R_m)}{(Q_1 \vee \ldots \vee Q_n \vee R_1 \vee \ldots \vee R_m)}$$

Theorem: Resolution is refutation complete.

If $\phi \models$ **False** then applying the propositional resolution rule, starting with the clauses in ϕ, until it cannot be applied any further will allow you to derive **False** (the empty clause).

Resolution refutation example

We know

- I'll go by bus or by train.
- If I go by train, I will be late.
- If I go by bus, I will be late.

In propositional logic

- $B \vee T$
- $T \Rightarrow L$
- $B \Rightarrow L$

Can I infer that I will be late (L)?
Negate conclusion conjoin with assumptions, convert to CNF

$$(B \vee T) \wedge (\neg T \vee L) \wedge (\neg B \vee L) \wedge \neg L$$

Resolution refutation example, continued

Does this formula entail **False**?

$$(\mathsf{B} \vee \mathsf{T}) \wedge (\neg \mathsf{T} \vee \mathsf{L}) \wedge (\neg \mathsf{B} \vee \mathsf{L}) \wedge \neg \mathsf{L}$$

Proof:

1. $\mathsf{B} \vee \mathsf{T}$ // assumption
2. $\neg \mathsf{T} \vee \mathsf{L}$ // assumption
3. $\neg \mathsf{B} \vee \mathsf{L}$ // assumption
4. $\neg \mathsf{L}$ // assumption
5. $\neg \mathsf{T}$ // 2, 4
6. $\neg \mathsf{B}$ // 3, 4
7. B // 1, 5
8. **False** // 6, 7

Proof strategies

Automated proof systems perform a kind of search. Search guidance is important.

- **Unit preference**: Prefer to do a resolution step involving a unit clause (clause with one literal.)

  Produces a shorter clause, which tends to be helpful, because we are trying to produce an empty clause.

- **Set of support**: Prefer to do a resolution step involving the negated goal or any clause derived from the negated goal.

  We are trying to produce a contradiction that follows from the negated goal, so these clauses are relevant.

If a contradiction exists, it can always be reached using the set-of-support strategy.

The power of False

Can we make formal sense of the idea that you can derive any conclusion from a contradiction?

$$(P \wedge \neg P) \models Z$$

Does this formula entail **False**? (Is it unsatisfiable?)

$$P \wedge \neg P \wedge \neg Z$$

Proof:

1. P // assumption
2. $\neg P$ // assumption
3. $\neg Z$ // assumption
4. **False** // 1, 2

Yes!

Practice example

Prove that these sentences

- $(P \to Q) \to Q$
- $(P \to P) \to R$
- $(R \to S) \to \neg(S \to Q)$

entail R

Horn clauses can have more efficient inference

A Horn clause is a clause (disjunction of literals) with exactly one positive literal. Here are some:

$$
\begin{aligned}
A \wedge B \wedge C &\Rightarrow D \\
E \wedge F &\Rightarrow A \\
&B
\end{aligned}
$$

Prolog: Depth-first backward chaining from a goal conjunction. Basis of logic programming which then adds extra tricks for handling negation, equality, and even side-effects.

More kinds of logic

- First order: adds to propositional logic
  - variables ranging over objects
  - quantifiers $\exists$ and $\forall$
  - Resolution can be generalized to do FOL proofs
- Non-boolean valued: probability, fuzzy, trinary
- Modal:
  - Temporal: always, until, eventually, ...
  - Alethic: necessary, possible
  - Deontic: obligatory, permitted
  - Epistemic: $\mathsf{K}(a, \phi)$ (agent a knows that ϕ)
- Special purpose (usually with efficient inference procedures)
  - Description logic (basically, taxonomies)
  - Reasoning about regular expressions

Next time

- Probability!