L08 – Propositional Logic

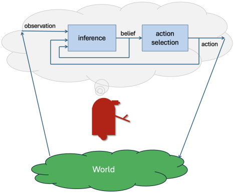
AIMA4e: 7.3–7.5 (rest of chapter 7 good for context!)

What you should know after this lecture

- Definition of logic: syntax and semantics
- What is logic good for?
- Propositional logic syntax and semantics
- Inference strategies
  - Model-checking: enumerative and efficient
  - Theorem proving (will do in detail next time)

Logic: structured representation and proof

- Observations are facts about the world
- Belief is a set of states described in very compact logical language, as the conjunction of observed facts
- Important question:
  - Does my current belief b entail some conclusion ϕ? That is:
  - Is ϕ guaranteed to be true, if b is?



What is propositional logic and what is it good for?

- Assume a very large (for now, finite) set of possible states
- Representation is factored into of a set of Boolean state variables, called propositions
- Language for specifying huge sets of states with short descriptions (which ones depends on how we do the formalization)

It is raining $\wedge$ Nick is at the beach

- Inference procedures for determining the truth of some statement given the truth of others: semantics-preserving syntactic manipulation. Domain independent!

Logic, in general

- possible worlds: set of all possible ways the world could be (states of the environment)
- syntax: set of sentences that you can write down on paper; compositionally defined
- semantics: relationship between syntactic sentences and sets of possible worlds; also compositionally defined
- inference: ways of generating new syntactic expressions from given ones, which
  - preserve semantics,
  - no matter what the semantics are!

Propositional logic syntax

<u>propositional symbols</u>: uppercase letters, **True**, **False**

- propositional symbols are sentences // Called "atoms"
- if α is a sentence, then $\neg\alpha$ is a sentence // negation
- if α and β are sentences, then
    - $\alpha \vee \beta$ is a sentence // or
    - $\alpha \wedge \beta$ is a sentence // and
    - $\alpha \Rightarrow \beta$ is a sentence // implies
    - $\alpha \Leftrightarrow \beta$ is a sentence // iff

<u>literal</u>: an atomic sentence or a negated atomic sentence

Propositional logic models

Can think of this in two steps:

1. Imagine a domain (set of possible worlds (environment states)) you'd want to describe (e.g. classrooms of students, or hiking trips, or cars)
2. Assign a meaning of each propositional symbol to a subset of that domain that is interesting or important to your problem: e.g.,
   - P: there were more than 10 students
   - Q: there were fewer than 20 students
   - R: the lecturer was witty

For any given possible world and interpretation of the symbols, we end up with

<u>model</u>: propositional symbols $\rightarrow$ truth value in $\{true, false\}$

Propositional logic semantics

Model $\mathfrak{m}$ satisfies sentence α if and only if one of the following holds:

- α is **True**
- α is a propositional symbol: $\mathfrak{m}(\alpha) = \textit{true}$
- $\alpha = \neg\beta$: $\mathfrak{m}$ **does not** satisfy β
- $\alpha = (\beta \vee \gamma)$: $\mathfrak{m}$ satisfies β **or** $\mathfrak{m}$ satisfies γ
- $\alpha = (\beta \wedge \gamma)$: $\mathfrak{m}$ satisfies β **and** $\mathfrak{m}$ satisfies γ
- $\alpha = (\beta \Rightarrow \gamma)$: $\mathfrak{m}$ satisfies $\neg\beta$ **or** $\mathfrak{m}$ satisfies γ
- $\alpha = (\beta \Leftrightarrow \gamma)$: $\mathfrak{m}$ satisfies $\beta \Rightarrow \gamma$ **and** $\mathfrak{m}$ satisfies $\gamma \Rightarrow \beta$

Logical terminology

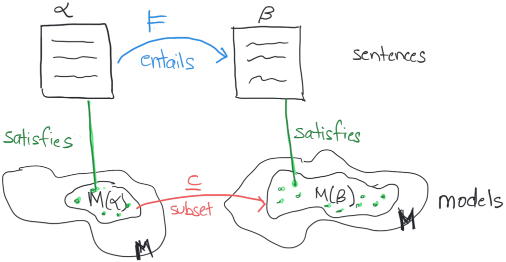
- model: a mapping between objects in the syntax and objects in the semantics; also called an interpretation
- satisfies: a model m satisfies a sentence α if α is true in m
  - Sometimes (but not in our book) written $m \models \alpha$
  - Sometimes we say m is a model of α
  - Sometimes we say α holds in m
  - $M(\alpha)$: set of all models of α
- entails: a sentence α entails sentence β, $\alpha \models \beta$, if and only if $M(\alpha) \subseteq M(\beta)$
- valid: a sentence is valid if it is satisfied in all models
- unsatisfiable: a sentence is unsatisfiable if it not satisfied in any model
- satisfiable: a sentence is satisfiable if there is at least one model in which it is satisfied

Entailment

A sentence α entails sentence β, $\alpha \models \beta$, if and only if

$$M(\alpha) \subseteq M(\beta)$$

That is, no matter whether you're thinking about hiking trips or classrooms or llamas, and what you think your symbols stand for, any model that satisfies α will also satisfy β.



Formalization practice

- W: lecturer is witty
- T: more than 10 students in class
- Z: students are asleep
- R: it's raining

Statements:

1. If the lecturer is witty, there will be more than 10 students in class.
2. If the lecturer is not witty, the students will be asleep.
3. More than 10 students will come to class only if it's not raining.

More formalization practice

AA: Alice admits; BA: Barbara admits; AP: Alice prison; BP Barbara prison

1. If both Alice and Barbara admit to having hacked into government computers, then neither of them will receive a prison sentence.
2. But if either of them admits to having hacked into a computer while the other doesn't, she will be sentenced to imprisonment while the other won't.
3. So unless both don't admit the deed, it cannot happen that both receive a prison sentence.

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 **True*** $\models (\alpha \Rightarrow \beta)$ *then* $\alpha \models \beta$.

Theorem
If $\alpha \models \beta$ *then **True*** $\models (\alpha \Rightarrow \beta)$.

Inference

- Given some information (observations) (α) what can I conclude <u>must</u> be true about the world (β)?
- Does α entail β??

Note that we can always take several observed sentences $\alpha_1, ..., \alpha_k$ and make them into a single sentence

$$\alpha_1 \wedge \ldots \wedge \alpha_n$$

Proof

Generally, 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 $\alpha \vdash \beta$

Proof is completely in syntax-land!

Stupidest possible propositional inference procedure

Recall that a model is an assignment of truth values to propositional symbols; we know the set of symbols for any given domain.

STUPID-ENTAILMENT(α, β)

> **for** each possible model m:
>
> > **if** SATISFIES(m, α) and not SATISFIES(m, β):
> >
> > > **return False**
>
> **return True**

How many possible models are there?
When would this be especially painful?

Reduction of proof to satisfiability testing

Recall that:

- A sentence is unsatisfiable if it is not true in any model
- If $\alpha \wedge \neg\beta$ is unsatisfiable then $\alpha \models \beta$.

Why??

Sometimes it's easier to think up algorithms for testing satisfiability (SAT). Two strategies:

- Backtracking (DPLL)
- Local search (e.g. simulated annealing, WalkSat, etc.)

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.

Proof: Inference rules

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.

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

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

One proof strategy: refutation

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!!

Clausal form (conjunctive normal form (CNF))

Many provers first convert all of their input to clausal form, which makes subsequent operations easier.

1. Turn all instances of $\alpha \Leftrightarrow \beta$ into $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
2. Turn all instances of $\alpha \Rightarrow \beta$ into $(\neg\alpha \vee \beta)$
3. Push negations all the way "in" using deMorgan's laws: $\neg(\alpha \wedge \beta) = (\neg\alpha \vee \neg\beta)$ and $\neg(\alpha \vee \beta) = (\neg\alpha \wedge \neg\beta)$
4. Distribute $\vee$ over $\wedge$: convert $\alpha \vee (\beta \wedge \gamma)$ to $(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

You end up with a formula of the form

$$(\alpha \vee \beta \vee \ldots) \wedge (\gamma \vee \delta \vee \ldots) \wedge \ldots \wedge (\epsilon \vee \zeta \vee \ldots)$$

where all the components are literals (negated or non-negated atoms). Elements of the form $(\alpha \vee \beta \vee \ldots)$ are called clauses.

Check yourself: what do each of these sets of clauses mean?

- $\{\}$: The set of no clauses
- $\{\{\}\}$: The set containing the empty clause

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 <u>refutation complete</u>.

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**?

$$(B \lor T) \land (\neg T \lor L) \land (\neg B \lor L) \land \neg L$$

Proof:

1. $B \lor T$ // assumption
2. $\neg T \lor L$ // assumption
3. $\neg B \lor L$ // assumption
4. $\neg L$ // assumption
5. $\neg T$ // 2, 4
6. $\neg B$ // 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:

$$A \wedge B \wedge C \Rightarrow D$$
$$E \wedge F \Rightarrow A$$
$$B$$

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: $K(a, \phi)$ (agent a knows that ϕ)
- Special purpose (usually with efficient inference procedures)
  - Description logic (basically, taxonomies)
  - Reasoning about regular expressions