Propositional Logic: Bottom-Up Proofs

CPSC 322 – Logic 3

Textbook §5.2

Lecture Overview

Quiz

- **8 points** Without using the words “model” or “interpretation”, explain what it means to say that $KB \models g$.

- Grading scheme:
  - approximately correct: 8 points
  - some good ideas: 4 points
  - incorrect: 0 points

Lecture Overview

Propositional Definite Clauses: Semantics

Semantics allows you to relate the symbols in the logic to the domain you're trying to model.

Definition (interpretation)

An interpretation I assigns a truth value to each atom.

We can use the interpretation to determine the truth value of clauses and knowledge bases:

Definition (truth values of statements)

- A body $b_1 \wedge b_2$ is true in I if and only if b_1 is true in I and b_2 is true in I.
- A rule $h \leftarrow b$ is false in I if and only if b is true in I and h is false in I.
- A knowledge base KB is true in I if and only if every clause in KB is true in I.

Models and Logical Consequence

Definition (model)

A model of a set of clauses is an interpretation in which all the clauses are *true*.

Definition (logical consequence)

If KB is a set of clauses and g is a conjunction of atoms, g is a logical consequence of KB, written $KB \models g$, if g is *true* in every model of KB.

- we also say that g logically follows from KB, or that KB entails g.
- In other words, $KB \models g$ if there is no interpretation in which KB is *true* and g is *false*.

Lecture Overview

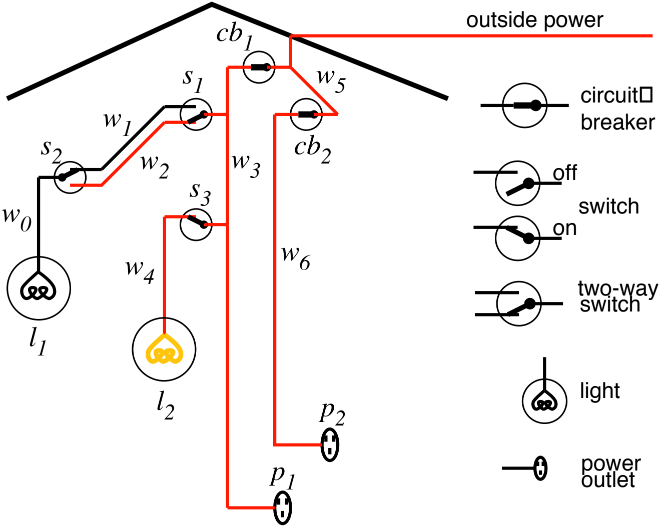
User's view of Semantics

1. Choose a task domain: intended interpretation.
2. Associate an atom with each proposition you want to represent.
3. Tell the system clauses that are true in the intended interpretation: axiomatizing the domain.
4. Ask questions about the intended interpretation.
5. If $KB \models g$, then g must be true in the intended interpretation.
6. The user can interpret the answer using their intended interpretation of the symbols.

Computer's view of semantics

- The computer doesn't have access to the intended interpretation.
    - All it knows is the knowledge base.
- The computer can determine if a formula is a logical consequence of KB.
    - If $KB \models g$ then g must be true in the intended interpretation.
    - If $KB \not\models g$ then there is a model of KB in which g is false. This could be the intended interpretation.

Electrical Environment



Representing the Electrical Environment

$light\_l_1.$

$light\_l_2.$

$down\_s_1.$

$up\_s_2.$

$up\_s_3.$

$ok\_l_1.$

$ok\_l_2.$

$ok\_cb_1.$

$ok\_cb_2.$

$live\_outside.$

$live\_l_1 \leftarrow live\_w_0$

$live\_w_0 \leftarrow live\_w_1 \wedge up\_s_2.$

$live\_w_0 \leftarrow live\_w_2 \wedge down\_s_2.$

$live\_w_1, \leftarrow live\_w_3 \wedge up\_s_1.$

$live\_w_2 \leftarrow live\_w_3 \wedge down\_s_1.$

$live\_l_2 \leftarrow live\_w_4.$

$live\_w_4 \leftarrow live\_w_3 \wedge up\_s_3.$

$live\_p_1 \leftarrow live\_w_3.$

$live\_w_3 \leftarrow live\_w_5 \wedge ok\_cb_1.$

$live\_p_2 \leftarrow live\_w_6.$

$live\_w_6 \leftarrow live\_w_5 \wedge ok\_cb_2.$

$live\_w_5 \leftarrow live\_outside.$

Role of semantics

In user's mind:

- $l2\_broken$: light $l2$ is broken
- $sw3\_up$: switch is up
- $power$: there is power in the building
- $unlit\_l2$: light $l2$ isn't lit
- lit_l1: light $l1$ is lit

In Computer:

$$l2\_broken \leftarrow sw3\_up \wedge power \wedge unlit\_l2.$$

$$sw3\_up.$$

$$power \leftarrow lit\_l1.$$

$$unlit\_l2.$$

$$lit\_l1.$$

Conclusion: $l2\_broken$

- The computer doesn't know the meaning of the symbols
- The user can interpret the symbols using their meaning

Lecture Overview

Proofs

- A proof is a mechanically derivable demonstration that a formula logically follows from a knowledge base.
- Given a proof procedure, $KB \vdash g$ means g can be derived from knowledge base KB.
- Recall $KB \models g$ means g is true in all models of KB.

Definition (soundness)

A proof procedure is sound if $KB \vdash g$ implies $KB \models g$.

Definition (completeness)

A proof procedure is complete if $KB \models g$ implies $KB \vdash g$.

Lecture Overview

Bottom-up Ground Proof Procedure

One rule of derivation, a generalized form of *modus ponens*:

> *If "$h \leftarrow b_1 \wedge \ldots \wedge b_m$" is a clause in the knowledge base, and each b_i has been derived, then h can be derived.*

You are forward chaining on this clause.
(This rule also covers the case when $m = 0$.)

Bottom-up proof procedure

$KB \vdash g$ if $g \subseteq C$ at the end of this procedure:

$C := \{\};$
repeat
  select clause "$h \leftarrow b_1 \wedge \ldots \wedge b_m$" in KB such that
    $b_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected.