Propositional Logic: Bottom-Up Proofs

CPSC 322 – Logic 3

Textbook §5.2

Lecture Overview

1. Recap
2. Bottom-Up Proofs
3. Soundness of Bottom-Up Proofs
4. Completeness of Bottom-Up Proofs
5. Resolution Proofs

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*.

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.

Example

$$a \leftarrow b \wedge c.$$

$$a \leftarrow e \wedge f.$$

$$b \leftarrow f \wedge k.$$

$$c \leftarrow e.$$

$$d \leftarrow k.$$

$$e.$$

$$f \leftarrow j \wedge e.$$

$$f \leftarrow c.$$

$$j \leftarrow c.$$

Example

$$a \leftarrow b \wedge c.$$
$$a \leftarrow e \wedge f.$$
$$b \leftarrow f \wedge k.$$
$$c \leftarrow e.$$
$$d \leftarrow k.$$
$$e.$$
$$f \leftarrow j \wedge e.$$
$$f \leftarrow c.$$
$$j \leftarrow c.$$

$$\{\}$$

Example

$a \leftarrow b \wedge c.$

$a \leftarrow e \wedge f.$

$b \leftarrow f \wedge k.$

$c \leftarrow e.$

$d \leftarrow k.$

$e.$

$f \leftarrow j \wedge e.$

$f \leftarrow c.$

$j \leftarrow c.$

$\{\}$

$\{e\}$

Example

$$a \leftarrow b \wedge c.$$
$$a \leftarrow e \wedge f.$$
$$b \leftarrow f \wedge k.$$
$$c \leftarrow e.$$
$$d \leftarrow k.$$
$$e.$$
$$f \leftarrow j \wedge e.$$
$$f \leftarrow c.$$
$$j \leftarrow c.$$

$$\{\}$$
$$\{e\}$$
$$\{c, e\}$$

Example

$a \leftarrow b \wedge c.$

$a \leftarrow e \wedge f.$

$b \leftarrow f \wedge k.$

$c \leftarrow e.$

$d \leftarrow k.$

$e.$

$f \leftarrow j \wedge e.$

$f \leftarrow c.$

$j \leftarrow c.$

$\{\}$

$\{e\}$

$\{c, e\}$

$\{c, e, f\}$

Example

$$a \leftarrow b \wedge c.$$
$$a \leftarrow e \wedge f.$$
$$b \leftarrow f \wedge k.$$
$$c \leftarrow e.$$
$$d \leftarrow k.$$
$$e.$$
$$f \leftarrow j \wedge e.$$
$$f \leftarrow c.$$
$$j \leftarrow c.$$

$$\{\}$$
$$\{e\}$$
$$\{c, e\}$$
$$\{c, e, f\}$$
$$\{c, e, f, j\}$$

Example

$a \leftarrow b \wedge c.$

$a \leftarrow e \wedge f.$

$b \leftarrow f \wedge k.$

$c \leftarrow e.$

$d \leftarrow k.$

$e.$

$f \leftarrow j \wedge e.$

$f \leftarrow c.$

$j \leftarrow c.$

$\{\}$

$\{e\}$

$\{c, e\}$

$\{c, e, f\}$

$\{c, e, f, j\}$

$\{a, c, e, f, j\}$

Lecture Overview

Soundness of bottom-up proof procedure

If $KB \vdash g$ then $KB \models g$.

- Suppose there is a g such that $KB \vdash g$ and $KB \not\models g$.
- Let h be the first atom added to C that's not true in every model of KB.
  - In particular, suppose h isn't *true* in model I of KB.
- There must be a clause in KB of form

$$h \leftarrow b_1 \wedge \ldots \wedge b_m$$

  Each b_i is true in I. h is false in I. So this clause is false in I.
- Therefore I isn't a model of KB. Contradiction: thus no such g exists.

Lecture Overview

Minimal Model

We can use proof procedure to find a model of KB.

- First, observe that the C generated at the end of the bottom-up algorithm is a fixed point.
  - further applications of our rule of derivation will not change C.

Minimal Model

We can use proof procedure to find a model of KB.

- First, observe that the C generated at the end of the bottom-up algorithm is a fixed point.
  - further applications of our rule of derivation will not change C.

Definition (minimal model)

Let the minimal model I be the interpretation in which every element of the fixed point C is true and every other atom is false.

Minimal Model

We can use proof procedure to find a model of KB.

- First, observe that the C generated at the end of the bottom-up algorithm is a fixed point.
  - further applications of our rule of derivation will not change C.

Definition (minimal model)

Let the minimal model I be the interpretation in which every element of the fixed point C is true and every other atom is false.

Claim: I is a model of KB. Proof:

- Assume that I is not a model of KB. Then there must exist some clause $h \leftarrow b_1 \wedge \ldots \wedge b_m$ in KB (having zero or more b_i's) which is false in I.
- This can only occur when h is false and each b_i is true in I.
- If each b_i belonged to C, we would have added h to C as well.
- Since C is a fixed point, no such I can exist.

Completeness

If $KB \models g$ then $KB \vdash g$.

- Suppose $KB \models g$. Then g is true in all models of KB.
- Thus g is true in the minimal model.
- Thus g is generated by the bottom up algorithm.
- Thus $KB \vdash g$.

Lecture Overview

Top-down Ground Proof Procedure

Idea: search backward from a query to determine if it is a logical consequence of KB.

An answer clause is of the form:

$$yes \leftarrow a_1 \wedge a_2 \wedge \ldots \wedge a_m$$

The SLD Resolution of this answer clause on atom a_i with the clause:

$$a_i \leftarrow b_1 \wedge \ldots \wedge b_p$$

is the answer clause

$$yes \leftarrow a_1 \wedge \cdots \wedge a_{i-1} \wedge b_1 \wedge \cdots \wedge b_p \wedge a_{i+1} \wedge \cdots \wedge a_m.$$

Derivations

- An answer is an answer clause with $m = 0$. That is, it is the answer clause $yes \leftarrow$.
- A derivation of query "$?q_1 \wedge \ldots \wedge q_k$" from KB is a sequence of answer clauses $\gamma_0, \gamma_1, \ldots, \gamma_n$ such that
  - γ_0 is the answer clause $yes \leftarrow q_1 \wedge \ldots \wedge q_k$,
  - γ_i is obtained by resolving γ_{i-1} with a clause in KB, and
  - γ_n is an answer.

Top-down definite clause interpreter

To solve the query $?q_1 \wedge \ldots \wedge q_k$:

$ac :=$ "$yes \leftarrow q_1 \wedge \ldots \wedge q_k$"
repeat
  select atom a_i from the body of ac;
  choose clause C from KB with a_i as head;
  replace a_i in the body of ac by the body of C
until ac is an answer.

Recall:

- Don't-care nondeterminism If one selection doesn't lead to a solution, there is no point trying other alternatives. select
- Don't-know nondeterminism If one choice doesn't lead to a solution, other choices may. choose

Example: successful derivation

$a \leftarrow b \wedge c.$	$a \leftarrow e \wedge f.$	$b \leftarrow f \wedge k.$
$c \leftarrow e.$	$d \leftarrow k.$	$e.$
$f \leftarrow j \wedge e.$	$f \leftarrow c.$	$j \leftarrow c.$

Query: $?a$

$\gamma_0 : \quad yes \leftarrow a$	$\gamma_4 : \quad yes \leftarrow e$
$\gamma_1 : \quad yes \leftarrow e \wedge f$	$\gamma_5 : \quad yes \leftarrow$
$\gamma_2 : \quad yes \leftarrow f$	
$\gamma_3 : \quad yes \leftarrow c$	

Example: failing derivation

$$
\begin{array}{lll}
a \leftarrow b \wedge c. & a \leftarrow e \wedge f. & b \leftarrow f \wedge k. \\
c \leftarrow e. & d \leftarrow k. & e. \\
f \leftarrow j \wedge e. & f \leftarrow c. & j \leftarrow c.
\end{array}
$$

Query: $?a$

$$
\begin{array}{ll}
\gamma_0 : & yes \leftarrow a \\
\gamma_1 : & yes \leftarrow b \wedge c \\
\gamma_2 : & yes \leftarrow f \wedge k \wedge c \\
\gamma_3 : & yes \leftarrow c \wedge k \wedge c \\
\gamma_4 : & yes \leftarrow e \wedge k \wedge c \\
\gamma_5 : & yes \leftarrow k \wedge c
\end{array}
$$

Search Graph

$a \leftarrow b \wedge c.$
$a \leftarrow h.$
$b \leftarrow k.$
$d \leftarrow p.$
$f \leftarrow p.$
$g \leftarrow f.$
$h \leftarrow m.$
$?a \wedge d$

$a \leftarrow g.$
$b \leftarrow j.$
$d \leftarrow m.$
$f \leftarrow m.$
$g \leftarrow m.$
$k \leftarrow m.$
$p.$