Propositional Logic: Bottom-Up Proofs

CPSC 322 – Logic 3

Textbook §5.2
Lecture Overview

1 Recap
2 Proofs
3 Bottom-Up Proofs
4 Soundness of Bottom-Up Proofs
5 Completeness of Bottom-Up 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 $b_1 \land b_2$ 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

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

1. Recap
2. Proofs
3. Bottom-Up Proofs
4. Soundness of Bottom-Up Proofs
5. Completeness of Bottom-Up Proofs
One rule of derivation, a generalized form of *modus ponens*:

If “$h ← b_1 ∧ \ldots ∧ b_m$” 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 $m = 0$.)

Propositional Logic: Bottom-Up Proofs

CPSC 322 – Logic 3, Slide 8
Bottom-up proof procedure

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

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

\[
\begin{align*}
    a & \leftarrow b \land c. \\
    a & \leftarrow e \land f. \\
    b & \leftarrow f \land k. \\
    c & \leftarrow e. \\
    d & \leftarrow k. \\
    e. \\
    f & \leftarrow j \land e. \\
    f & \leftarrow c. \\
    j & \leftarrow c. \\
\end{align*}
\]
Example

\[ a \leftarrow b \land c. \]
\[ a \leftarrow e \land f. \]
\[ b \leftarrow f \land k. \]
\[ c \leftarrow e. \]
\[ d \leftarrow k. \]
\[ e. \]
\[ f \leftarrow j \land e. \]
\[ f \leftarrow c. \]
\[ j \leftarrow c. \]

\[ \{ \} \]
Example

\[
\begin{align*}
  a & \leftarrow b \land c. \\
  a & \leftarrow e \land f. \\
  b & \leftarrow f \land k. \\
  c & \leftarrow e. \\
  d & \leftarrow k. \\
  e & . \\
  f & \leftarrow j \land e. \\
  f & \leftarrow c. \\
  j & \leftarrow c.
\end{align*}
\]
Example

\[ a \leftarrow b \land c. \]
\[ a \leftarrow e \land f. \]
\[ b \leftarrow f \land k. \]
\[ c \leftarrow e. \]
\[ d \leftarrow k. \]
\[ e. \]
\[ f \leftarrow j \land e. \]
\[ f \leftarrow c. \]
\[ j \leftarrow c. \]
\[ \{\} \]
\[ \{e\} \]
\[ \{c, e\} \]
\[ \{c, e, f\} \]
\[ \{c, e, f, j\} \]
\[ \{a, c, e, f, j\} \]
Example

\[ a \leftarrow b \land c. \]
\[ a \leftarrow e \land f. \]
\[ b \leftarrow f \land k. \]
\[ c \leftarrow e. \]
\[ d \leftarrow k. \]
\[ e. \]
\[ f \leftarrow j \land e. \]
\[ f \leftarrow c. \]
\[ j \leftarrow c. \]
Example

\[
a \leftarrow b \land c.
\]
\[
a \leftarrow e \land f.
\]
\[
b \leftarrow f \land k.
\]
\[
c \leftarrow e.
\]
\[
d \leftarrow k.
\]
\[
e.
\]
\[
f \leftarrow j \land e.
\]
\[
f \leftarrow c.
\]
\[
j \leftarrow c.
\]
Example

\[
\begin{align*}
a & \leftarrow b \land c. \\
a & \leftarrow e \land f. \\
b & \leftarrow f \land k. \\
c & \leftarrow e. \\
d & \leftarrow k. \\
e. \\
f & \leftarrow j \land e. \\
f & \leftarrow c. \\
j & \leftarrow c.
\end{align*}
\]
Lecture Overview

1 Recap
2 Proofs
3 Bottom-Up Proofs
4 Soundness of Bottom-Up Proofs
5 Completeness of Bottom-Up Proofs
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 $KB$.
- Suppose $h$ isn't true in model $I$ of $KB$.
- There must be a clause in $KB$ of form

$$h \leftarrow b_1 \land \ldots \land b_m$$

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

1. Recap
2. Proofs
3. Bottom-Up Proofs
4. Soundness of Bottom-Up Proofs
5. Completeness of Bottom-Up Proofs
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 $h \leftarrow b_1 \land \ldots \land b_m$ 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.
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 $KB \vdash g$. 