Propositional Definite Clause Logic: Syntax, Semantics and Bottom-up Proofs

Computer Science cpsc322, Lecture 20

(Textbook Chpt 5.1.2 – 5.2.2)

June, 6, 2017
Lecture Overview

- Recap: Logic intro
- Propositional Definite Clause Logic: Semantics
- PDCL: Bottom-up Proof
Logics as a R&R system

Represent

• formalize a domain

\[
on_{-}l_1 \iff \text{live}_{-}w_1 \quad \text{AND} \quad \text{live}_{-}w_2 \iff \text{on}_{-}sw_1 \land \text{live}_{-}w_3
\]

• reason about it

If the agent knows \text{on}_{-}sw_1 and \text{live}_{-}w_3 it should be able to infer \text{on}_{-}l_1
Logics in AI: Similar slide to the one for planning

- Propositional Definite Clause Logics
- Semantics and Proof Theory
- Satisfiability Testing (SAT)
- Hardware Verification
- Product Configuration

- Description Logics
- Production Systems
- Cognitive Architectures

- Ontologies
- Semantic Web
- Summarization
- Video Games
- Tutoring Systems

- Propositional Logics
- First-Order Logics

Some Applications: you will know a little
Propositional (Definite Clauses) Logic: Syntax

We start from a restricted form of Prop. Logic:

Only two kinds of statements

- that a proposition is true
- that a proposition is true if one or more other propositions are true

\[ \neg (p_1 \lor p_2) \iff (p_3 \lor \neg p_3) \]

Definite clause is

either an atom

or

atom \leftarrow body

\[ p_3 \leftarrow p_1 \land p_2 \]
Lecture Overview

- Recap: Logic intro
- Propositional Definite Clause Logic: Semantics
- PDCL: Bottom-up Proof
Propositional Definite Clauses Semantics: Interpretation

Semantics allows you to relate the symbols in the logic to the domain you’re trying to model. An **atom** can be….

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If your domain can be represented by four atoms (propositions):

<table>
<thead>
<tr>
<th>q</th>
<th>p</th>
<th>s</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

So an interpretation is just a……possible world
PDC Semantics: Body

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 $I$.

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>$I_2$</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>$I_3$</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>$I_4$</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>$I_5$</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
**PDC Semantics: definite clause**

**Definition (truth values of statements cont’):** A rule \( h \leftarrow b \) is false in \( I \) if and only if \( b \) is true in \( I \) and \( h \) is false in \( I \).

<table>
<thead>
<tr>
<th>( I )</th>
<th>( p )</th>
<th>( q )</th>
<th>( r )</th>
<th>( s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
</tbody>
</table>

In other words: "if \( b \) is true I am claiming that \( h \) must be true, otherwise I am not making any claim"
PDC Semantics: Knowledge Base (KB)

- A knowledge base KB is true in I if and only if every clause in KB is true in I.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I₁</strong></td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

Which of the three KB below are True in I₁?

**A**
- p
- r
- s ← q ∧ p

**B**
- p
- q
- s ← q

**C**
- p
- q ← r ∧ s
A knowledge base $KB$ is true in $I$ if and only if every clause in $KB$ is true in $I$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$r$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

Which of the three $KB$ above are True in $I_1$? $KB_3$
Definition (truth values of statements cont’): A knowledge base $KB$ is true in $I$ if and only if every clause in $KB$ is true in $I$. 

$KB_1$:
- $p \lor \neg r$  
- $r \land q \land p$ 

$KB_2$:
- $p \lor q$  
- $s \leftarrow q$ 

$KB_3$:
- $q \leftarrow r \land s$  
- $p$ 

$p, q, r, s$ truth values:
- $T \quad T \quad T = F$  
- $F$ 

CPSC 322, Lecture 20
Models

Definition (model)
A model of a set of clauses (a KB) is an interpretation in which all the clauses are true.
Example: Models

\[ KB = \begin{cases} 
    p & \leftarrow q, \\
    q, \\
    r & \leftarrow s. 
\end{cases} \]

Which interpretations are models?

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>I_1</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>I_2</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>I_3</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>I_4</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>I_5</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
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, $KB \not\models G$ if there is no interpretation in which $KB$ is true and $G$ is false.
Example: Logical Consequences

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>I₁</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>I₂</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>I₃</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>I₄</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>I₅</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>I₆</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>I₇</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>I₈</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

Models

\[ KB = \{ \begin{align*}
p & \leftarrow q. \\ q. \\ r & \leftarrow s. 
\end{align*} \] 

\[ 2^4 = 16 \text{ interpretations in total, only 3 are models} \]

Which of the following is true?

1) \[ KB \models q, \ KB \models p, \ KB \models s, \ KB \models r \]

remaining 8 cannot be models because q
is false
Lecture Overview

- Recap: Logic intro
- Propositional Definite Clause Logic: Semantics
- PDCL: Bottom-up Proof
One simple way to prove that $G$ logically follows from a KB

- Collect all the models of the KB
- Verify that $G$ is true in all those models

Any problem with this approach? Intractable time complexity

- The goal of proof theory is to find proof procedures that allow us to prove that a logical formula follows from a KB avoiding the above
Soundness and Completeness

• If I tell you I have a **proof procedure for PDCL**
• What do I need to show you in order for you to trust my procedure?

  • \( KB \vdash G \) means \( G \) can be derived by my proof procedure from \( KB \).
  • 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 \( KB \not\models G \).

**Definition (completeness)**

A proof procedure is **complete** if \( KB \not\models G \) implies \( KB \vdash G \).
Bottom-up Ground Proof Procedure

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

If “$h \leftarrow b_1 \land \ldots \land b_m$” 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

\begin{itemize}
  \item select clause \( h \leftarrow b_1 \land \cdots \land b_m \) in $KB$
  \item such that \( b_i \in C \) for all \( i \), and \( h \notin C \);
  \item \( C := C \cup \{ h \} \)
\end{itemize}

until no more clauses can be selected.

$KB$: \[ e \leftarrow a \land b \quad a \quad b \quad r \leftarrow f \]
Bottom-up proof procedure: Example

KB. \( f \land e \)
\( f \land g \land z \)
\( a \land b \)

\( C := \{\}; \)

repeat

select clause “\( h \leftarrow b_1 \land \ldots \land b_m \)” 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.

Which one is correct?
A. \( KB \vdash \{z, q, a\} \)
B. \( KB \vdash \{r, z, b\} \)
C. \( KB \vdash \{q, a\} \)
Bottom-up proof procedure: Example

\[ z \leftarrow f \land e \]
\[ q \leftarrow f \land g \land z \]
\[ e \leftarrow a \land b \]

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

Which one is correct?

\[ KB \vdash \{ z, q, a \} \]
\[ KB \vdash \{ r, z, b \} \]
\[ KB \vdash \{ q, a \} \]
Bottom-up proof procedure: Example

\[ z \leftarrow f \land e \]
\[ q \leftarrow f \land g \land z \]
\[ e \leftarrow a \land b \]

\[ C = \{ +_1, b_1, a, e, z \} \]

\[ C := \{ \}; \]
repeat
    select clause "\[ h \leftarrow b_1 \land \ldots \land b_m \]" 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.

BU can derive \[ r \]
BU cannot derive \[ q, z \]

KB \( \not\vdash \) \( q \)
Learning Goals for today’s class

You can:

- Verify whether an interpretation is a model of a PDCL KB.
- Verify when a conjunction of atoms is a logical consequence of a knowledge base.
- Define/read/write/trace/debug the bottom-up proof procedure.
Next class

 stil section 5.2)

- Soundness and Completeness of Bottom-up Proof Procedure
- Using PDC Logic to model the electrical domain
- Reasoning in the electrical domain
Study for midterm (This Thurs)

Midterm: 6 short questions (8pts each) + 2 problems (26pts each)

- Study: textbook and inked slides

- Work on all practice exercises and revise assignments!

- While you revise the learning goals, work on review questions (posted on Connect) I may even reuse some verbatim 😊

- Also work on couple of problems (posted on Connect) from previous offering (maybe slightly more difficult) ⋮ but I’ll give you the solutions 😊
midterm (This Thurs)

• Midterm on June 8 – first block of class
  • Search
  • CSP
  • SLS
  • Planning
  • Possibly simple/minimal intro to logics