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

Computer Science cpsc322, Lecture 20

(Textbook Chpt 5.2)

February, 25, 2009

Lecture Overview

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

Logics as a R&R system

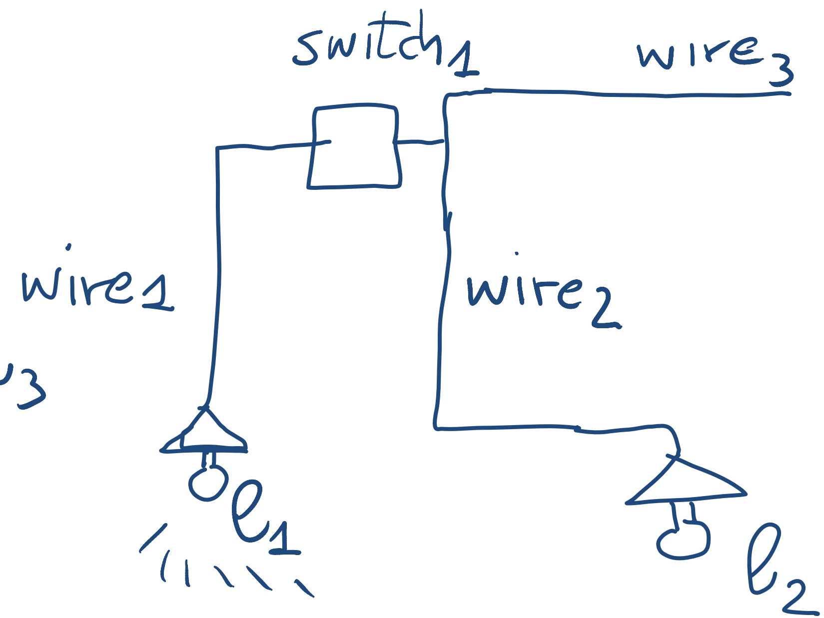
represent

- **formalize a domain**

$on\_l_1 \xleftarrow{\text{IF}} live\_w_1$.

$live\_w_1 \xleftarrow{\text{IF}} on\_sw_1 \overset{\text{AND}}{\wedge} live\_w_3$

. . . . .

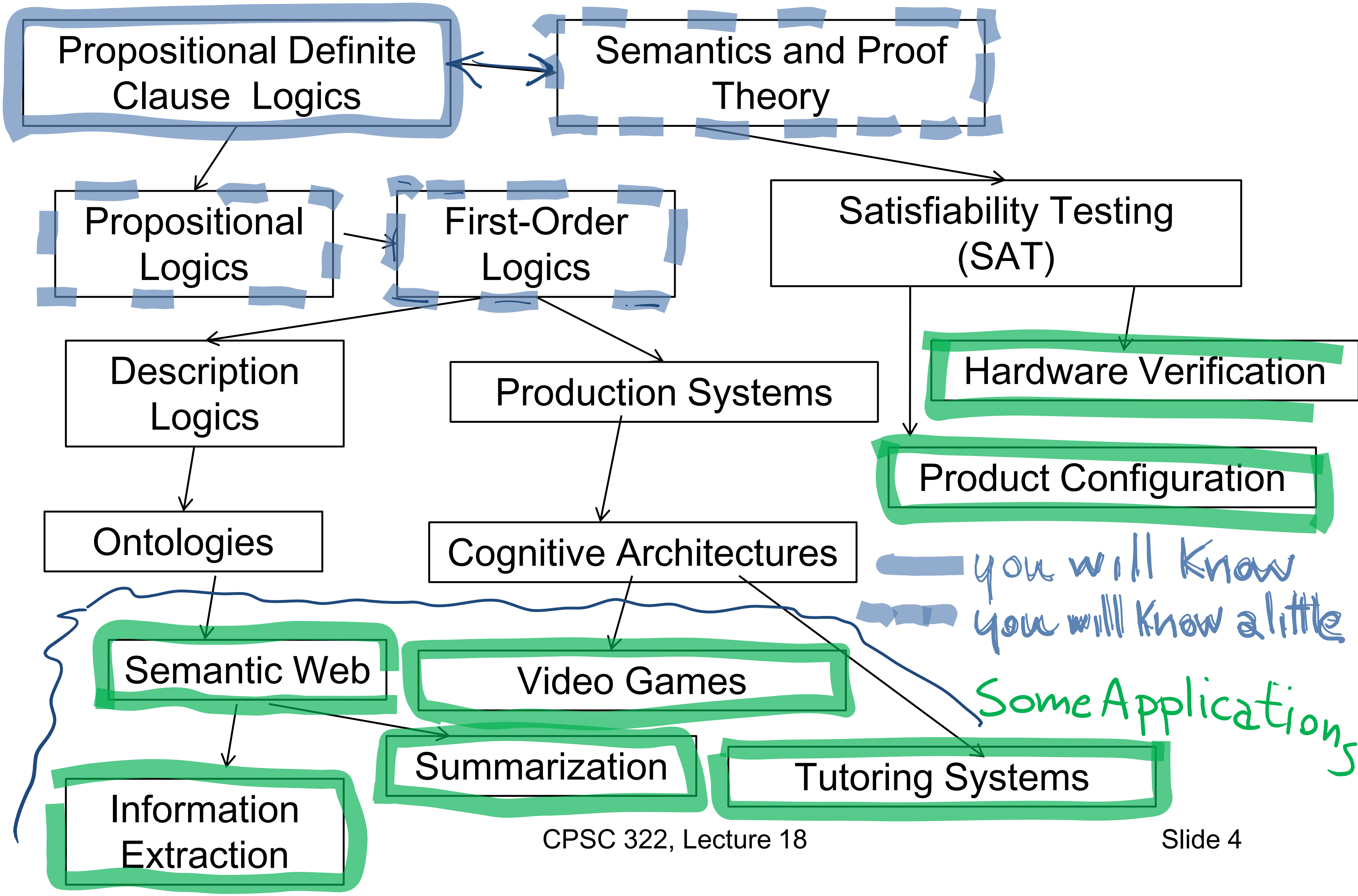


- **reason about it**

If the agent knows on_sw_1 and $live\_w_3$

it should be able to infer on_l_1

Logics in AI: Similar slide to the one for planning

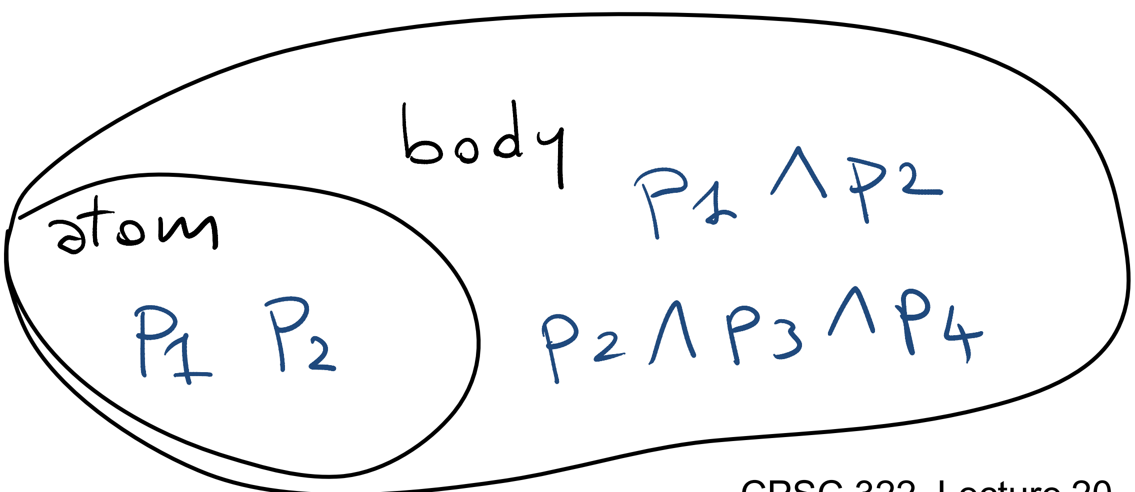


Propositional (Definite Clauses) Logic: Syntax

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



definite clause is either an atom

or atom ← body

e.g. $p_1 \leftarrow p_2 \wedge p_3$ ....

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.

atom can be T/F

> **Definition (interpretation)**
> An interpretation I assigns a truth value to each atom.

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

p	q	r	s
T	T	T	T
T	T	T	F
T	T	F	F

2^4 interpretations

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 $b_1 \wedge b_2$ is true in I if and only if b_1 is true in I and b_2 is true in I.

	p	q	r	s	$q \wedge r \wedge s$
I_1	*true*	*true*	*true*	*true*	T
I_2	*false*	*false*	*false*	*false*	F
I_3	*true*	*true*	*false*	*false*	F
I_4	*true*	*true*	*true*	*false*	F
I_5	*true*	*true*	*false*	*true*	F

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.

	p	q	r	s
I_1	*true*	*true*	*true*	*true*
I_2	*false*	*false*	*false*	*false*
I_3	*true*	*true*	*false*	*false*
I_4	*true*	*true*	*true*	*false*
…..	false …..	…..	….	true ….

$p \leftarrow s$

T

T

T

T

F

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

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.

p	q	r	s
T	T	F	F

I

KB_3

$q \leftarrow r \wedge s$

p

T

KB_1

- p ✓
- r ×
- $s \leftarrow q \wedge p$

F

KB_2

p ✓

q ✓

$s \leftarrow q$ ×

F

Models

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

Example: Models

$$\underline{KB} = \begin{cases} p \leftarrow q. \Leftarrow \\ \underline{q.} \\ r \leftarrow s. \end{cases}$$

	p	q	r	s	
I_1	*true*	*true*	*true*	*true*	M
I_2	~~*false*~~	~~*false*~~	~~*false*~~	~~*false*~~	X q is false
I_3	*true*	*true*	*false*	*false*	M
I_4	*true*	*true*	*true*	*false*	M
I_5	~~*true*~~	~~*true*~~	~~*false*~~	~~*true*~~	X r ← s is false

Which interpretations are models?

Logical Consequence

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*.

Example: Logical Consequences

	p	q	r	s	
I_1	*true*	*true*	*true*	*true*	← M
I_2	*true*	*true*	*true*	*false*	M
I_3	*true*	*true*	*false*	*false*	M
I_4	~~*true*~~	~~*true*~~	~~*false*~~	~~*true*~~	
I_5	~~*false*~~	~~*true*~~	~~*true*~~	~~*true*~~	
I_6	~~*false*~~	~~*true*~~	~~*true*~~	~~*false*~~	
I_7	~~*false*~~	~~*true*~~	~~*false*~~	~~*false*~~	
I_8	~~*false*~~	~~*true*~~	~~*false*~~	~~*true*~~	
…	….	…	…	…	← the remaining 8 cannot be models because q is false

$$\boxed{KB = \begin{cases} p \leftarrow q. \\ q. \\ r \leftarrow s. \end{cases}}$$

$2^4 = 16$ interpretations in total

ONLY 3 are models

Which of the following is true?

- $KB \models q$, $KB \models p$, $KB \not\models s$, $KB \not\models r$

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

you have to check all the 2^n interpretations!

- 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 form a KB avoiding the above

by "syntactic" processing of the KB

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? that it is sound & complete
  - KB ⊢ G means ~~*g*~~ G can be derived by my proof procedure from *KB*.
- Recall *KB* ⊨ *G* means ~~*g*~~ G is true in all models of *KB*.

Definition (soundness)

A proof procedure is sound if *KB* ⊢ *G* implies *KB* ⊨ *G*.

Definition (completeness)

A proof procedure is complete if *KB* ⊨ *G* implies *KB* ⊢ *G*.

Bottom-up Ground Proof Procedure

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

$\boxed{p \leftarrow q \qquad q} \qquad p$

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
  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.

Bottom-up proof procedure: Example

KB

$z \leftarrow f \wedge e$

$q \leftarrow f \wedge g \wedge z$

$e \leftarrow a \wedge b$

a

b

r

f

$C = \{\}$; $= \{f\}$; $= \{f, r\}$; $= \{f, r, b\}$; $= \{f, r, b, a\}$; $= \{f, r, b, a, e\}$; $\{f, r, b, a, e, z\}$ FINAL

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.

r? q? z?

$KB \vdash_{BU} z$ $KB \vdash_{BU} r$ $KB \nvdash_{BU} q$

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.

  G

  $KB \overset{?}{\models} G$
- Define/read/write/trace/debug the **bottom-up proof procedure**.

Next class

(still section 5.2)

- Soundness and Completeness of Bottom-up Proof Procedure
- Using PDC Logic to model the electrical domain
- Reasoning in the electrical domain