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

Computer Science cpsc322, Lecture 20

(Textbook Chpt 5.1.2 - 5.2.2)

Oct, 23, 2013

Lecture Overview

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

Logics as a R&R system

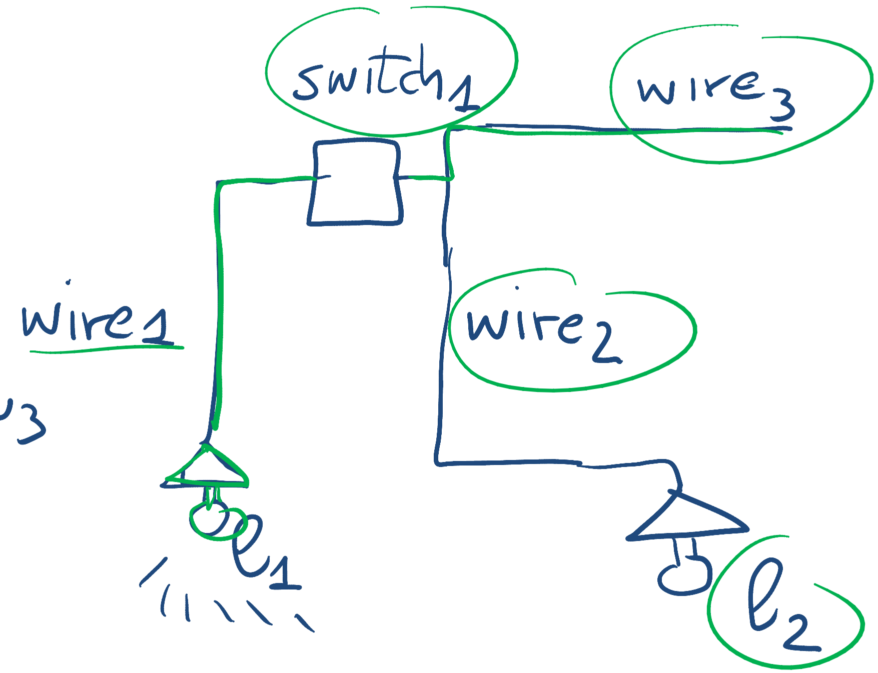
Represent

- **formalize a domain**

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

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

. . . . .

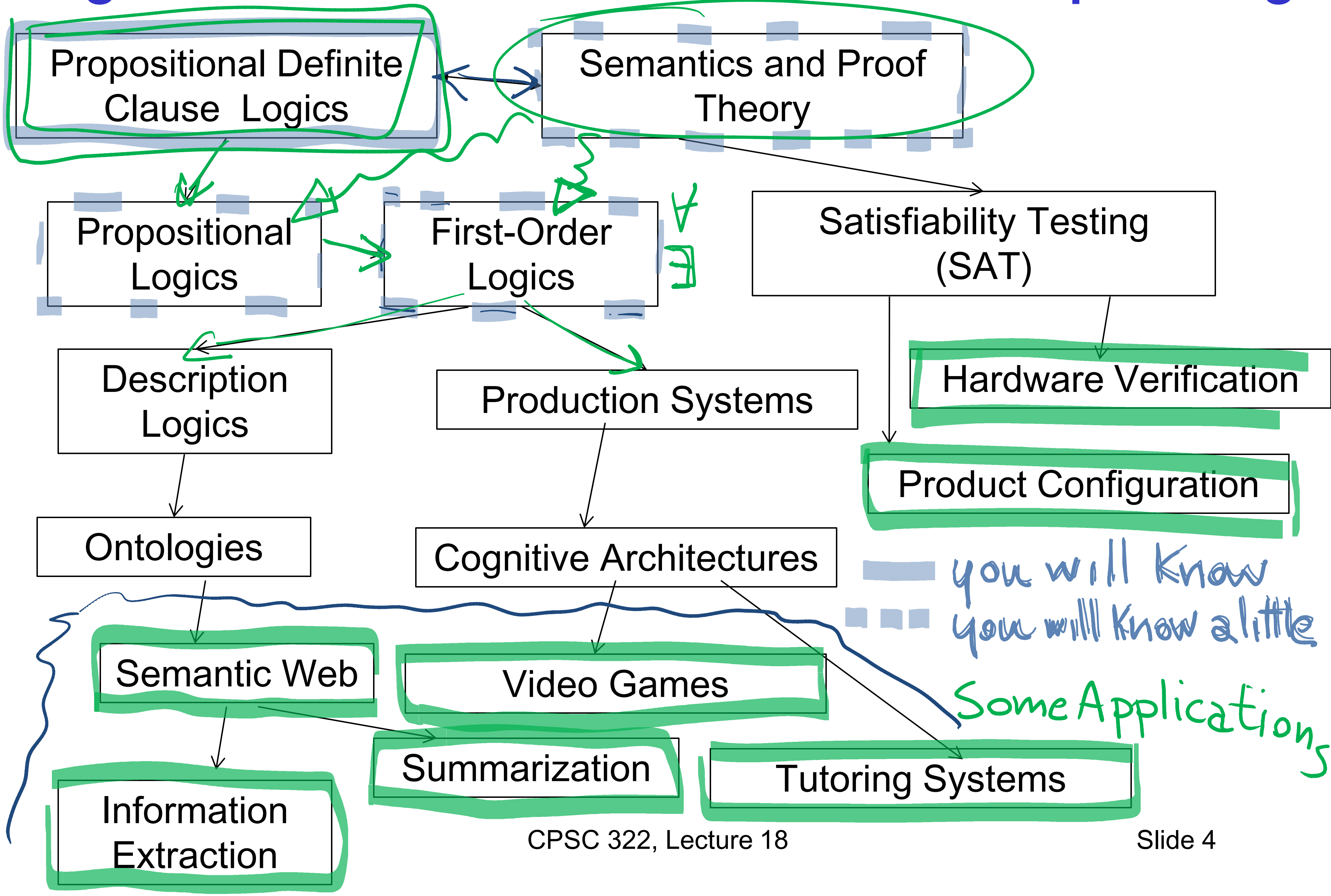


- **reason about it**

If the agent knows on_sw1 and live_w3

it should be able to infer $\boxed{on\_l_1}$

Logics in AI: Similar slide to the one for planning



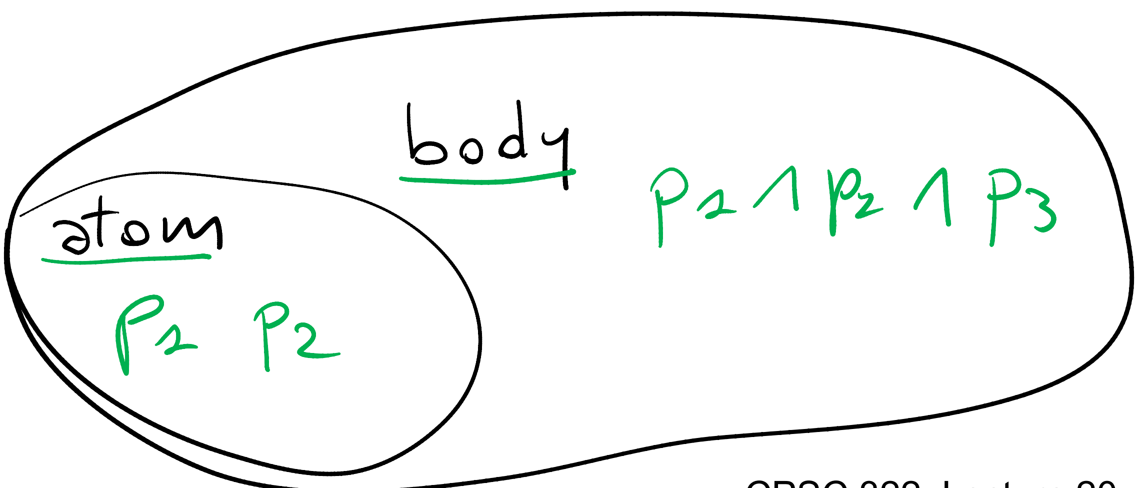
Propositional (Definite Clauses) Logic: Syntax

We start from a restricted form of Prop. Logic:

$\neg(p_1 \vee p_2) \Leftrightarrow (p_3 \vee \neg p_5)$

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

$p_3 \leftarrow p_1 \wedge p_2 \ldots$

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….. T F

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

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

q p s r

T T F F

2^4

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	$p \wedge r$	$p \wedge r \wedge s$
I_1	*true*	*true*	*true*	*true*	T	T
I_2	*false*	*false*	*false*	*false*	F	F
I_3	*true*	*true*	*false*	*false*	F	F
I_4	*true*	*true*	*true*	*false*	T	F
I_5	*true*	*true*	*false*	*true*	F	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	$p \leftarrow s$	$s \leftarrow q \wedge r$
I_1	*true*	*true*	*true*	*true*	T	T
I_2	*false*	*false*	*false*	*false*	T	T
I_3	*true*	*true*	*false*	*false*	T	T
I_4	*true*	*true*	*true*	*false*	T	F
…..	F ….	…..	….	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 (KB)

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

	p	q	r	s
I_1	*true*	*true*	*false*	*false*

iclicker

Which of the three KB below are True in I_1 ?

A

p
r
$s \leftarrow q \wedge p$

B

p
q
$s \leftarrow q$

C

p
$q \leftarrow r \wedge s$

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.

	p	q	r	s
I_1	*true*	*true*	*false*	*false*

KB_1

p
r
$s \leftarrow q \wedge p$

KB_2

p
q
$s \leftarrow q$

KB_3

p
$q \leftarrow r \wedge s$

Which of the three KB above are True in I_1 ? KB_3

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

T

KB_3

$q \leftarrow r \wedge s$

p

F

KB_1

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

F

KB_2

p ✓

q ✓

$s \leftarrow q$ ✗

Models

Definition (model)

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

Example: Models

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

	p	q	r	s	
I_1	*true*	*true*	*true*	*true*	M
I_2	*false*	*false*	*false*	*false*	X
I_3	*true*	*true*	*false*	*false*	M
I_4	*true*	*true*	*true*	*false*	M
I_5	*true*	*true*	$\boxed{\textit{false} \quad \textit{true}}$		X

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*
I_2	*true*	*true*	*true*	*false*
I_3	*true*	*true*	*false*	*false*
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*
...	....	...	...	...

Models

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

$2^4 = 16$ interpretations in total, only 3 are models

remaining 8 cannot be models because q is false

F

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

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

a set of atoms P1, P2, . . .

Any problem with this approach?

you have to check all the 2^n interpretations

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

is logically entailed by

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 $KB \models G$.

Definition (completeness)

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

Bottom-up Ground Proof Procedure

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

$q \leftarrow p$, p we can derive $\Rightarrow q$

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.

KB: $e \leftarrow a \wedge b$ a b $r \leftarrow f$

Bottom-up proof procedure: Example

BU

KB.

$z \leftarrow f \wedge e$

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

$e \leftarrow a \wedge b$

a

b

r

f

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.

Which one is correct?

A. $KB \vdash \{z, q, a\}$

B. $KB \vdash \{r, z, b\}$

C. $KB \vdash \{q, a\}$

iclicker.

Bottom-up proof procedure: Example

BU

$z \leftarrow f \wedge e$

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

$e \leftarrow a \wedge b$

a

b

r

f

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.

Which one is correct?

$KB \vdash \{z, q, a\}$

$KB \vdash \{r, z, b\}$

$KB \vdash \{q, a\}$

Bottom-up proof procedure: Example

BU

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

$z \leftarrow f \wedge e$

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

$e \leftarrow a \wedge b$

a

b

r

f

$C := \{\}$;
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.

BU can derive $r \wedge z$

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

r, z

BU cannot derive

$KB \nvdash_{BU} q$

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.
- 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

Study for midterm (Mon Oct 28)

Midterm: ~6 short questions (*10pts each*) + 2 problems (*20pts 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 ☺