CPSC 322: Introduction to Artificial Intelligence

Logics: Soundness, Completeness, and proofs

Textbook reference: [5.3.2]

Instructor: Varada Kolhatkar
University of British Columbia

Credit: These slides are adapted from the slides of the previous offerings of the course. Thanks to all instructors for creating and improving the teaching material and making it available!

Announcements

- Midterm grades have been released. Refer to Piazza post #321.
    - Most of you did pretty well! 🎉
    - Mean = 82%; Median = 84.29%
- Assignment 3 has been released.
    - Due date: **Nov 11, 11:59 PM**
- Final exam scheduled: **Dec 9 at 7:00pm**

Lecture outline

- Recap: PDCL (~10 mins) 👈
- Proofs intro (~5 mins)
- Soundness and completeness (~15 mins)
- Bottom-up proof procedures (~30 mins)
- Class activity (electric environment) (~15 mins)

PDCL semantics: Models

A **knowledge base** (KB) is a set of propositional definite clauses (PDC).

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

PDCL semantics: Models

A **knowledge base** (KB) is a set of propositional definite clauses (PDC).

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

Which interpretations are models?

	p	q	r	s	$p \leftarrow q$	$r \leftarrow s$	KB	Model?
I_1	T	T	T	T	T	T	T	✓
I_2	F	F	F	F	T	T	F	
I_3	T	T	F	F	T	T	T	✓
I_4	T	T	T	F	T	T	T	✓
I_5	T	T	F	T	T	F	F	

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

PDCL semantics: Knowledge Base (KB)

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

	p	q	r	s
K	T	T	F	F

Which of the following KBs below are true in K?

A.

p
r
$s \leftarrow p \wedge q$

B.

p
r
$s \leftarrow p$

C.

p
$q \leftarrow r \wedge s$

PDCL semantics: Knowledge Base (KB)

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

Which of the following KBs below are true in *K*?

	p	q	r	s
K	T	T	F	F

A.

p
r
$s \leftarrow p \wedge q$

B.

p
r
$s \leftarrow p$

C.

p
$q \leftarrow r \wedge s$

✅

PDCL semantics: Entailment

Definition: If KB is a knowledge base and g is a proposition, g is a **logical consequence** of KB, written as $KB \models g$, 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.

Entailment is a semantic notion.

Recap: PDCL (pair-share)

Consider the domain represented by three propositions: p, q, r

r	q	p	$p \leftarrow q \wedge r$	KB1	KB2
T	T	T	T	T	T
T	T	F	F	F	F
T	F	T	T	F	F
T	F	F	T	F	F
F	T	T	T	F	T
F	T	F	T	F	T
F	F	T	T	F	F
F	F	F	T	F	F

KB1

$r\,.$

$q\,.$

$p \leftarrow q \wedge r$

KB2

$q\,.$

$p \leftarrow q \wedge r$

Which ones are models for KB1 and KB2?

Is $g = p \wedge q$ is logically entailed by KB1?

Is $g = p \wedge q$ is logically entailed by KB2?

Recap: PDCL (pair-share)

Consider the domain represented by three propositions: p, q, r

r	q	p	$p \leftarrow q \wedge r$	KB1	KB2
T	T	T	T	T	T
T	T	F	F	F	F
T	F	T	T	F	F
T	F	F	T	F	F
F	T	T	T	F	T
F	T	F	T	F	T
F	F	T	T	F	F
F	F	F	T	F	F

KB1

$r.$

$q.$

$p \leftarrow q \wedge r$

KB2

$q.$

$p \leftarrow q \wedge r$

Which ones are models for KB1 and KB2?

Is $g = p \wedge q$ is logically entailed by KB1? ✅

Is $g = p \wedge q$ is logically entailed by KB2? ❌

Today: Learning outcomes

From this lecture, students are expected to be able to:

- Explain soundness and completeness of a proof procedure.
- Define/read/write/trace/debug the Bottom-Up (BU) proof procedure.
- Prove that the BU proof procedure is sound and complete.
- Model a relatively simple domain with propositional definite clause logic (PDCL).

Lecture outline

- Recap: PDCL (~10 mins)
- Proofs intro (~5 mins) 👈🏻
- Soundness and completeness (~15 mins)
- Bottom-up proof procedures (~30 mins)
- Class activity (electric environment) (~15 mins)

PDCL proofs: Motivation

How to compute the logical consequences of a knowledge base?

The problem of **deduction** is to determine if some proposition is a logical consequence of a knowledge base.

Proofs or derivations

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 **proved** or **derived** from knowledge base KB using the procedure.

Syntactic notion

Contrast it with logical consequence

If KB is a knowledge base and g is a proposition, g is a **logical consequence** of KB, written as $KB \models g$, if g is true in every model of KB.

Semantic notion

Proof procedures

A **proof procedure** is an algorithm for deriving consequences of a knowledge base.

Example: simple proof procedure S
Enumerate all **interpretations**.
Identify all **models**.
$KB \vdash_S g$ if g holds in all such models.

Quality of a proof procedure

- 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?
- A proof procedure's quality can be judged by whether it computes what it is meant to compute.
  - In other words when it is **sound** and **complete**.
  - You cannot prove anything that is false.
  - You can prove everything that is true.

Lecture outline

- Recap: PDCL (~10 mins)
- Proofs intro (~5 mins)
- Soundness and completeness (~15 mins) 👈
- Bottom-up proof procedures (~30 mins)
- Class activity (electric environment) (~15 mins)

Soundness

Recall $KB \vdash g$ means g can be derived or proved from KB.
Recall $KB \models g$ means g is true in all models of KB.

Definition: Soundness
A proof procedure is **sound** with respect to a semantics if everything that can be derived from a knowledge base is a **logical consequence** of the knowledge base. That is, $KB \vdash g$ implies $KB \models g$.

Everything **derived** from a **sound** proof procedure is **entailed** by the KB.

Soundness example

<u>**Example:**</u> simple proof procedure S
Enumerate all **interpretations**.
Identify all **models**.
$KB \vdash_S g$ if g holds in all such models.

Is the proof procedure S sound?

Can we say that if g can be derived by the procedure S ($KB \vdash_S g$) then g is true in all models of KB ($KB \models g$)?

Completeness

Recall $KB \vdash g$ means g can be derived or proved from KB.
Recall $KB \models g$ means g is true in all models of KB.

Definition: Completeness
A proof procedure is **complete** with respect to a semantics if there is a proof of each logical consequence of the knowledge base. That is $KB \models g$ implies $KB \vdash g$.

Everything **entailed** by the KB is **derived** from a **complete** proof procedure.

Completeness example

<u>Example</u>: simple proof procedure S
Enumerate all **interpretations**.
Identify all **models**.
$KB \vdash_S g$ if g holds in all such models.

Is the proof procedure S complete?

Can we say that if g is true in all models of KB ($KB \models g$), g can be derived by the procedure S ($KB \vdash_S g$)?

More examples: soundness and completeness

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

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

Consider a proof procedure U which derives every atom in KB. That is for any g that appears in KB, $KB \vdash_U g$.

Is the proof procedure U sound?

Is the proof procedure U complete?

More examples: soundness and completeness

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

Consider a proof procedure U which derives every atom in KB. That is for any g that appears in KB, $KB \vdash_U g$.

KB

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

Is the proof procedure U sound?

<u>Counter example:</u>

Consider KB. The procedure derives every atom in KB, which means it will derive $\{p, q\}$. If the procedure were sound, $KB \vdash q$ would imply $KB \models q$, which means for every model of KB, q should be true, which is not the case.

More examples: soundness and completeness

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

Consider a proof procedure U which derives every atom in KB. That is for any g that appears in KB, $KB \vdash_U g$.

Is the proof procedure U sound? ✗

<u>Counter example:</u>

Consider KB. The procedure derives every atom in KB, which means it will derive $\{p, q\}$. If the procedure were sound, $KB \vdash q$ would imply $KB \models q$, which means for every model of KB, q should be true, which is not the case.

KB

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

p	q	$p \leftarrow q$	Model?
T	T	T	✓
T	F	T	✓
F	T	F	
F	F	T	✓

More examples: soundness and completeness

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

Consider a proof procedure U which derives every atom in KB. That is for any g that appears in KB, $KB \vdash_U g$.

Is the proof procedure U complete?

It is complete because it will not miss any atom,
as we will be considering every atom.

More examples: soundness and completeness

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

Consider a proof procedure U which derives every atom in KB. That is for any g that appears in KB, $KB \vdash_U g$.

Is the proof procedure U complete? ✅

It is complete because it will not miss any atom, as we will be considering every atom.

More examples: soundness and completeness

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

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

Consider a proof procedure I which derives nothing. There is no atom g such that $KB \vdash_I g$.

Is the proof procedure I sound?

Is the proof procedure I complete?

More examples: soundness and completeness

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

Consider a proof procedure I which derives nothing. There is no atom g such that $KB \vdash_I g$.

Is the proof procedure I sound?
Yes because it derives nothing.

More examples: soundness and completeness

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

Consider a proof procedure I which derives nothing. There is no atom g such that $KB \vdash_I g$.

Is the proof procedure I sound? ✅
Yes because it derives nothing.

More examples: soundness and completeness

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

Consider a proof procedure I which derives nothing. There is no atom g such that $KB \vdash_I g$.

Is the proof procedure I complete?
No because it will miss deriving the atoms that are entailed by the knowledge base.
<u>Counter example:</u>
Consider knowledge base KB. Here $KB \models \{p, q\}$ but the procedure won't derive them.

KB

$p \leftarrow q$
q

More examples: soundness and completeness

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

Consider a proof procedure I which derives nothing. There is no atom g such that $KB \vdash_I g$.

Is the proof procedure I complete? ❌
No because it will miss deriving the atoms that are entailed by the knowledge base.
<u>Counter example:</u>
Consider knowledge base KB. Here $KB \models \{p, q\}$ but the procedure won't derive them.

KB

$p \leftarrow q$
q

Soundness and completeness

Let's consider these two "proof procedures" for PDCL

X. $C_X = \{$all clauses in KB with empty bodies$\}$
Y. $C_Y = \{$all atoms in the knowledge base$\}$

Which of the following statement is correct?

A. Both X and Y are sound and complete

B. X is sound only and Y is complete only

C. X is complete only and Y is sound only

KB

$a \leftarrow e \land g$
$b \leftarrow f \land g$
$c \leftarrow e$
$f \leftarrow c$
$e\,.$
$d\,.$

Soundness and completeness

Let's consider these two "proof procedures" for PDCL

X. $C_X = \{$all clauses in KB with empty bodies$\}$
Y. $C_Y = \{$all atoms in the knowledge base$\}$

Which of the following statement is correct?

A. Both X and Y are sound and complete

B. X is sound only and Y is complete only ✅

C. X is complete only and Y is sound only

KB

$$
\boxed{\begin{array}{l}
a \leftarrow e \wedge g \\
b \leftarrow f \wedge g \\
c \leftarrow e \\
f \leftarrow c \\
e\,. \\
d\,.
\end{array}}
$$

Lecture outline

- Recap: PDCL (~10 mins)
- Proofs intro (~5 mins)
- Soundness and completeness (~15 mins)
- Bottom-up proof procedures (~30 mins) 👈
- Class activity (electric environment) (~15 mins)

PDCL bottom-up proofs: motivation

The simple proof procedure S is sound and complete.

<u>Example</u>: simple proof procedure S
Enumerate all **interpretations**.
Identify all **models**.
$KB \vdash_S g$ if g holds in all such models.

Any problem with this approach?

PDCL bottom-up proofs: motivation

The simple proof procedure S is **sound** and **complete**.

Example: simple proof procedure S
Enumerate all **interpretations**.
Identify all **models**.
$KB \vdash_S g$ if g holds in all such models.

Any problem with this approach?

- Very much like the generate-and-test approach for CSPs

- Sound and complete, but there are a lot of interpretations. For n propositions, there are 2^n interpretations.

Bottom-up vs. top-down

- Need efficient ways to construct proofs for propositional definite clauses
- We will look at two such ways:
  - **a bottom-up procedure**: Going forward (forward chaining)
  - **a top-down procedure**: Given a query, going backward

Bottom-up proof procedure (BU)

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

If "$h \leftarrow a_1 \wedge \ldots \wedge a_m$" is a definite clause in the knowledge base, and each a_i has been derived, then h can be derived.

This rule also covers the case when m = 0.

Bottom-up proof procedure (BU)

$C := \{\}$;
repeat
   select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
      such that $a_i \in C$ for all i, and $h \notin C$;
   $C := C \cup \{h\}$
until no more clauses can be selected.

$KB \vdash g$ **if** $g \subseteq C$ at the end of this procedure

BU: Example

$C := \{\}$;
repeat
  select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
    such that $a_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected.

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

$KB \vdash g$ **if** $g \subseteq C$ at the end of this procedure

BU: Example

$C := \{\}$;
repeat
    select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
        such that $a_i \in C$ for all i, and $h \notin C$;
    $C := C \cup \{h\}$
until no more clauses can be selected.

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

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

$KB \vdash g$ **if** $g \subseteq C$ at the end of this procedure

BU (pair-share if time permits)

$C := \{\}$;
repeat
  select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
    such that $a_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected.

KB

$z \leftarrow f \wedge e$
$q \leftarrow f \wedge g \wedge z$
$e \leftarrow a \wedge b$
a
b
r
f

$\{\}$

BU (pair-share)

$C := \{\}$;
repeat
  select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
    such that $a_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected.

KB

$z \leftarrow f \wedge e$
$q \leftarrow f \wedge g \wedge z$
$e \leftarrow a \wedge b$
a
b
r
f

$\{\}$
$\{a, b, f, r\}$
$\{a, b, e, f, r\}$
$\{a, b, e, f, r, z\}$
Done!

Bottom-up proof procedure

iclicker.

$C := \{\}$;
repeat
  select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
    such that $a_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected.

Which of the following is correct?

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

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

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

KB

$z \leftarrow f \wedge e$
$q \leftarrow f \wedge g \wedge z$
$e \leftarrow a \wedge b$
a
b
r
f

Bottom-up proof procedure

iclicker.

$C := \{\}$;
repeat
  select clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB
    such that $a_i \in C$ for all i, and $h \notin C$;
  $C := C \cup \{h\}$
until no more clauses can be selected.

Which of the following is correct?

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

B. $KB \vdash \{r, z, b\}$ ✅

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

KB

$z \leftarrow f \wedge e$
$q \leftarrow f \wedge g \wedge z$
$e \leftarrow a \wedge b$
a
b
r
f

Soundness of BU procedure

A proof procedure is **sound** if $KB \vdash g$ implies $KB \models g$.

Is BU sound?

BU is **sound** if all the atoms in C are logically entailed by KB.

Proof: Soundness of BU procedure

Suppose it's not the case.

1. Let h be the first atom added to C that is not entailed by KB. That is, it's not true in every model of KB.

2. Suppose h isn't true in model M of KB.

3. Since h was added to C, there must be a clause in KB of the form: $h \leftarrow a_1 \wedge \ldots \wedge a_m$, where each a_i is true in M.

4. But h is false in M. So the clause is false in M. Therefore M is not a model.

5. Contradiction! So no such h exists.

Completeness of BU procedure

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

Is BU complete?

Completeness of BU

Sketch of our proof

1. Suppose $KB \models g$. Then g is true in all models of KB.

2. Thus g is true in a particular model of KB.

3. We'll define a particular model such that if g is true in that model, g is proved by the BU algorithm.

4. Thus $KB \vdash g$

Completeness of BU: Expand step 3

3. We'll define a particular model such that if g is true in that model, g is proved by the BU algorithm.

   3.1. We'll define a particular interpretation I such that g is true in I iff g is proved by the BU algorithm

   3.2. We'll then show that I is a model

Completeness of BU: Step 3.1

Observe that the C generated at the end of the BU algorithm is a **fixed point**; further applications of our rule of derivation will not change C!

The **minimal model** is the interpretation in which every element of BU's fixed point C is true and every other element is false.

KB

$$\begin{array}{l} a \leftarrow e \wedge g \\ b \leftarrow f \wedge g \\ c \leftarrow e \\ f \leftarrow c \wedge e \\ e\,. \\ d\,. \end{array}$$

$$C = \{e, d, c, f\}$$

	a	*b*	*c*	*d*	*e*	*f*	*g*
I	F	F	T	T	T	T	F

Completeness of BU: step 3.2

Claim: The minimal model I is the model of KB.

- Assume that I is not a model of KB
- Then there must exist some clause $h \leftarrow a_1 \wedge \ldots \wedge a_m$ in KB with $m \geq 0$, which is false in I.
- The only way this can occur is $a_1 \wedge \ldots \wedge a_m$ are true in I (i.e., in C) and h is false in I (i.e., is not in C).
- But if each a_i was in C, BU would have added h to C as well.
- So there can be no clause in KB that is false in I

Completeness of BU: proof summary

If $KB \models g$. Then $KB \vdash_{BU} g$.

1. Suppose $KB \models g$
2. Then g is true in all models of KB
3. Thus g is true in the minimal model
4. Thus $g \subseteq C$
5. Thus g is proved by BU
6. Which means $KB \vdash_{BU} g$

Summary for bottom-up proof procedure

- BU is **sound**; it derives **only** atoms that logically follow from KB
- BU is **complete**; it derives **all** atoms that logically follow from KB
- Together: it derives **exactly** the atoms that logically follow from KB

Summary for bottom-up proof procedure

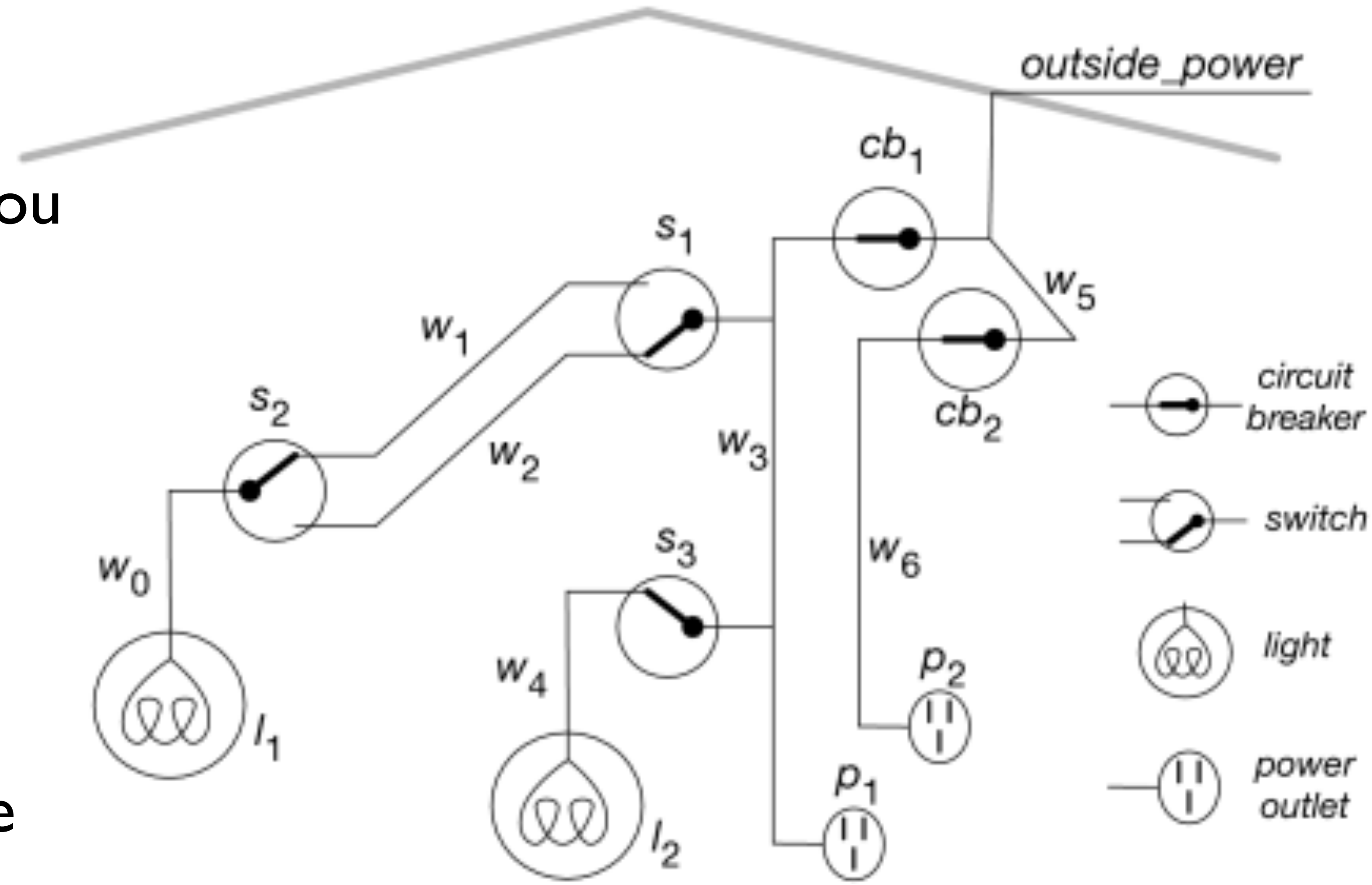
- BU is **sound** and **complete.**
- And, it is **efficient**!
    - **Linear** in the number of clauses in KB
    - Each clause is used maximally once by BU

Lecture outline

- Recap: PDCL (~10 mins)
- Proofs intro (~5 mins)
- Soundness and completeness (~15 mins)
- Bottom-up proof procedures (~30 mins)
- Class activity (electric environment) (~15 mins) 👈

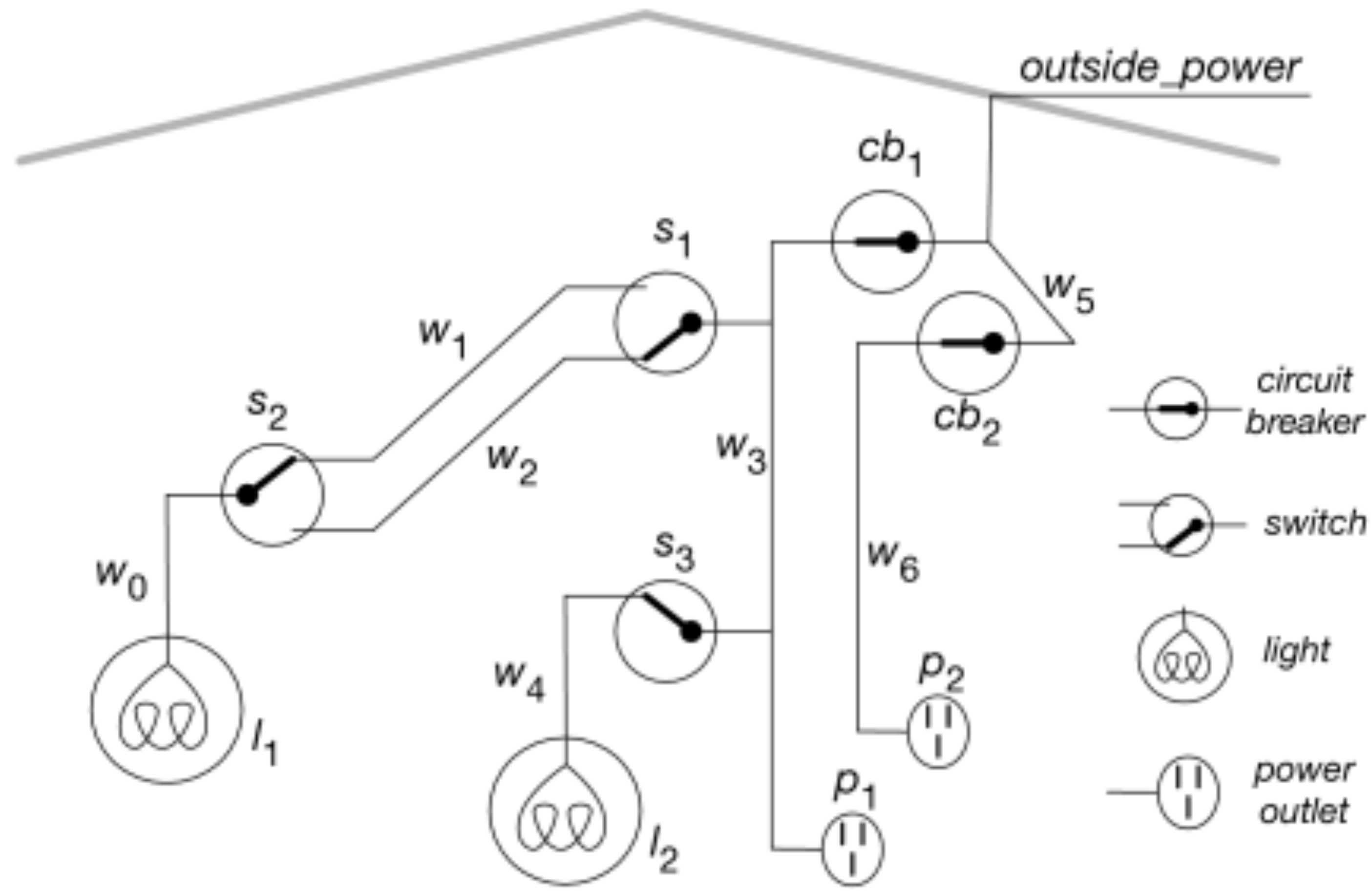
Example: Electric environment

1. Begin with a task domain.
2. Distinguish those things you want to talk about (the ontology)
3. Choose symbols in the computer to denote propositions
4. Tell the system knowledge about the domain
5. Ask the system whether new statements about the domain are true or false



Class activity: Let's define relevant propositions

- For each wire w
- For each circuit breaker cb
- For each switch s
- For each light l
- For each outlet p



How many interpretations?

Intended interpretation (time permitting)

- User chooses task domain: **intended interpretation**. This is the interpretation of the symbols the user has in mind.

- User tells the system clauses (the knowledge base KB).

- Each clause is true in the user's intended interpretation. Thus, the intended interpretation is a model.

- **The computer does not know the intended interpretation.** But if it can derive something that's true in all models, then it is true in the intended interpretation

Intended interpretation (time permitting)

If $KB \models g$, then

- g is true in the intended interpretation
- g is true in every model of KB (by definition)

Notice there is always at least one model of any PDCL theory!
Q: What is it?
A: The interpretation with every atom true is always a model.

Coming up

Top-down proofs and datalog

5.3.2 Proofs

13.3 Datalog: A Relational Rule Language