Propositional Definite Clause Logic: Syntax, Semantics, R&R and Proofs

Computer Science cpsc322, Lecture 7

(Textbook Chpt 5.1- 5.2)

May, 29, 2012

Lecture Overview

- **Recap**: Logic intro and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
  - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

Logics as a R&R system

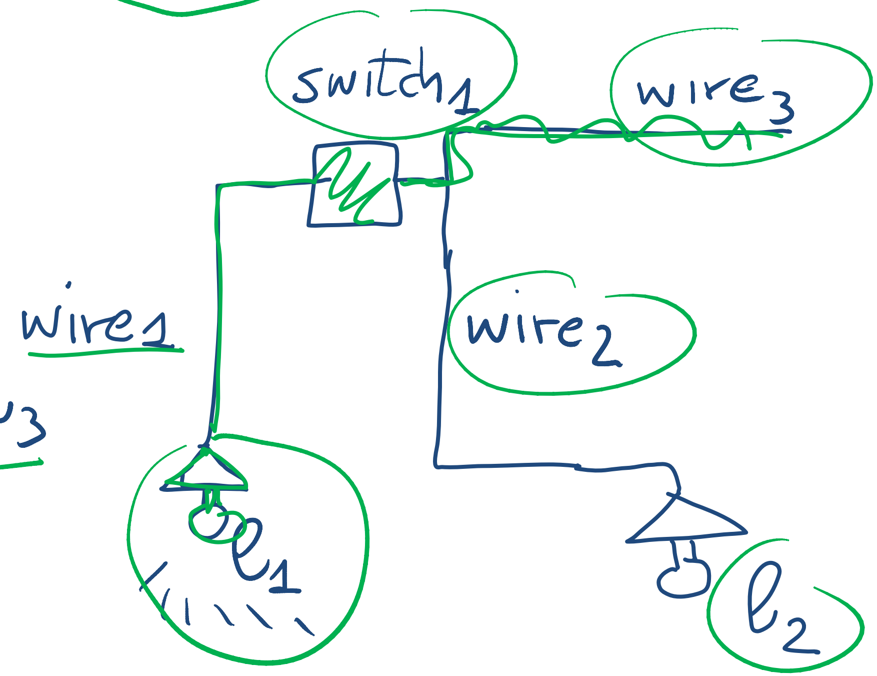
Represent

- formalize a domain

on_l_1 ← IF live_w_1

live_w_1 ← IF on_sw_1 AND ∧ live_w_3

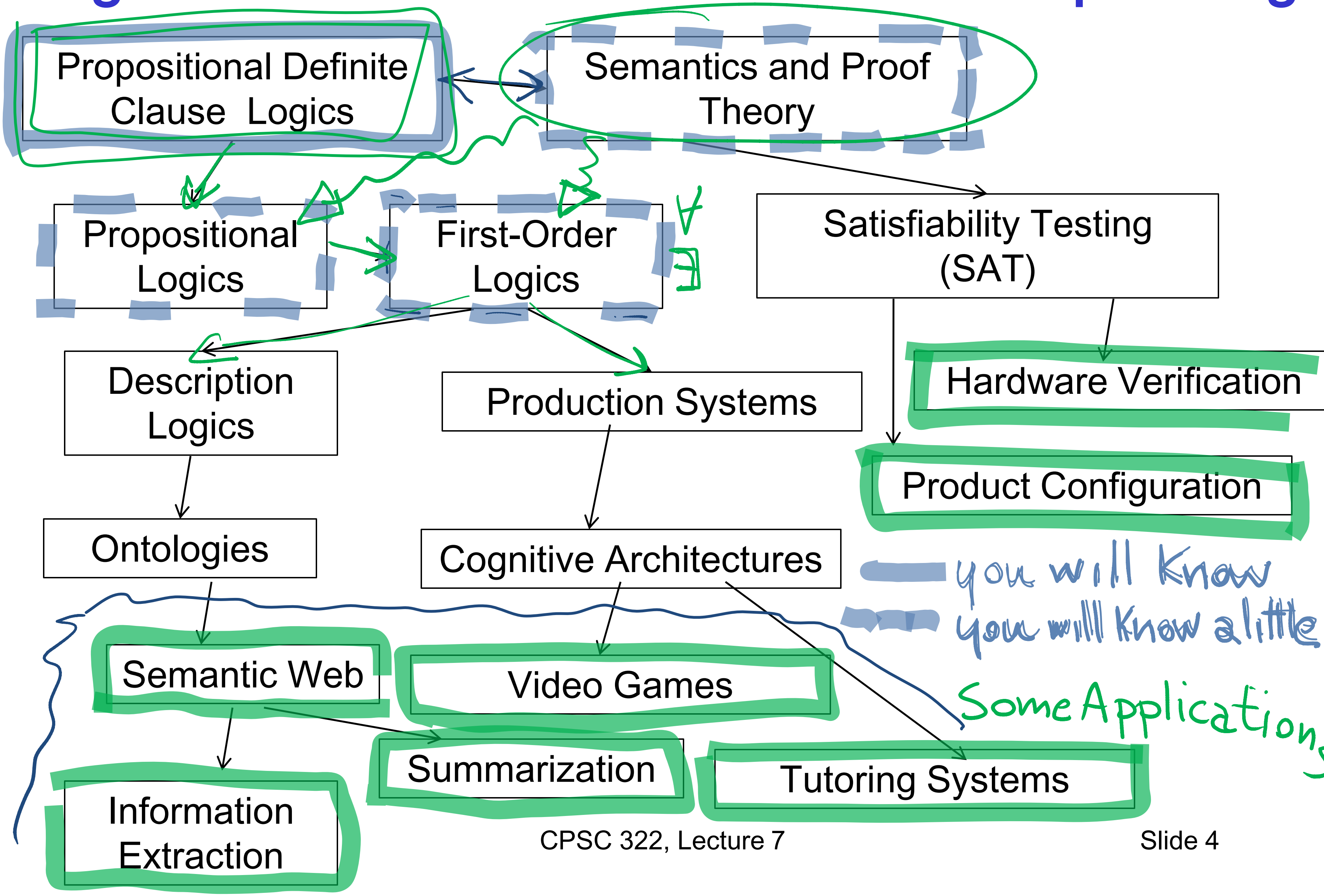
. . . . . .



- 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



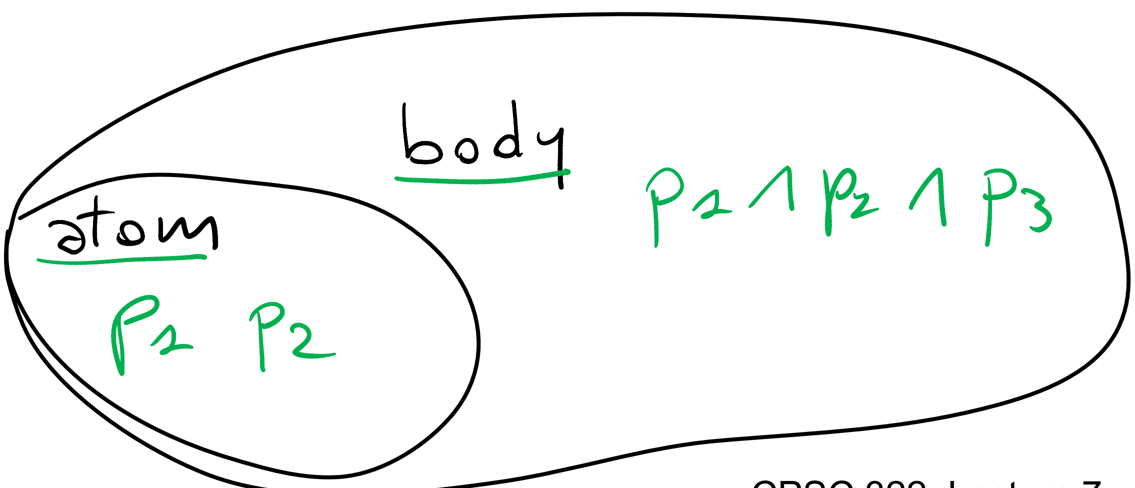
Propositional (Definite Clauses) Logic: Syntax

We start from a restricted form of Prop. Logic:

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

Only two kinds of statements

- that a $\boxed{\text{proposition is true}}$
- that a proposition is true if one or more other propositions are true



definite clause is
either an $\boxed{\text{atom}}$
or $\boxed{\text{atom} \leftarrow \text{body}}$

$p_3 \leftarrow p_1 \wedge p_2 \ldots$

Lecture Overview

- **Recap**: Logic intro and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
  - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

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
I_1	*true*	*true*	*true*	*true*
I_2	*false*	*false*	*false*	*false*
I_3	*true*	*true*	*false*	*false*
I_4	*true*	*true*	*true*	*false*
I_5	*true*	*true*	*false*	*true*

$p \wedge r$	$p \wedge r \wedge s$
T	T
F	F
F	F
T	F
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*

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

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
$s \leftarrow q \land p$

KB_2

p
q
$s \leftarrow q$

KB_3

p
$q \leftarrow r \land s$

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

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*	*false*	*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*~~
...	....	... F	...	...

Models

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

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

remaining 8 cannot be models because q is false

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 and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
    - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

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***:

$q \leftarrow p,\quad p \xRightarrow{\text{we can derive}} q$

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.

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

BU can derive $r \wedge z$

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.

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

BU cannot derive

$KB \nvdash_{BU} q$

Lecture Overview

- **Recap**: Logic intro and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
  - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

Soundness of bottom-up proof procedure

Generic Soundness of proof procedure:

If G can be proved by the procedure (KB ⊢ G)

then G is logically entailed by the KB (KB ⊨ G)

For Bottom-Up proof

if $G \subseteq C$ at the end of procedure

then G is logically entailed by the KB

So BU is sound, if all the atoms in C……

are logically entailed by the KB

Soundness of bottom-up proof procedure

Suppose this is not the case.

1. Let *h* be the first atom added to *C* that is not entailed by KB (i.e., that'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 form: $h \leftarrow b_1 \wedge \ldots \wedge b_m$
4. Each b_i is true in *M (because of 1.)*. *h* is false in *M.* So…… the clause is false in M
5. Therefore M is not a model
6. Contradiction! thus no such *h* exists.

Learning Goals for today's class – part1

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**.
- Prove that BU proof procedure is **sound**

Lecture Overview

- **Recap**: Logic intro and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
  - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

30 mins BREAK

Completeness of Bottom Up

Generic Completeness of proof procedure:

If G is logically entailed by the KB ($KB \models G$)

then G can be proved by the procedure ($KB \vdash G$)

$G \subseteq C$

Sketch of our proof:

1. Suppose $KB \models G$. Then G is true in all models of KB.
2. Thus G is true in any particular model of KB
3. We will define a model so that if G is true in that model, G is proved by the bottom up algorithm. $G \subseteq C$
4. Thus $KB \vdash_{BU} G$.

Let's work on step 3

3. We will define a model so that if G is true in that model, G is proved by the bottom up algorithm.

$G \subseteq C$

3.1 We will define an interpretation $\mathcal{I}$ so that if G is true in $\mathcal{I}$, G is proved by the bottom up algorithm.

$G \subseteq C$

3.2 We will then show that ……… $\mathcal{I}$ is a model

Let's work on step 3.1

3.1 Define interpretation I so that if G is true in I, Then $G \subseteq C$.

Let I be the interpretation in which every element of C is true and every other atom is false.

$a \leftarrow e \wedge g.$
$b \leftarrow f \wedge g.$
$c \leftarrow e.$
$f \leftarrow c \wedge e.$
$e.$
$d.$

F F T T T T F
$\{a, b, c, d, e, f, g\}$

C

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

Let's work on step 3.2

Claim: I is a model of *KB* *(we'll call it the* ***minimal model****)*.

Proof: **Assume** that I is not a model of *KB*.

- **Then** there must exist some clause $h \leftarrow b_1 \land \dots \land b_m$ in *KB* (having zero or more b_i's) which is false in I.
- The only way this can occur is if $b_1 \dots b_m$ are true in I (i.e., are in C) and h is false in I (i.e., is not in C)
- **But** if each b_i belonged to C, Bottom Up would have added h to C as well.
- So, there can be no clause in the KB that is false in interpretation I (which implies the claim :-)

Completeness of Bottom Up
(proof summary)

If $KB \models G$ then $KB \vdash_{BU} G$

- Suppose $KB \models G$.
- Then G is true in all the models
- Thus G is true in the minimal model
- Thus $G \subseteq C$
- Thus G is proved by…. BU
- ~~Thus~~ $KB \vdash_{BU} G$

i.e.

Soundness

$KB \models G$

BU

$KB \vdash_{BU} G$

Completeness

RELATION BETWEEN SOUNDNESS & COMPLETENESS

Soundness & completeness of proof procedures

- A proof procedure X is sound …

$KB \vdash_X G \Rightarrow KB \models G$

- A proof procedure X is complete….

$KB \models G \Rightarrow KB \vdash_X G$

- BottomUp for PDCL is $\leftarrow$

sound & complete

- We proved this in general even for domains represented by **thousands of propositions** and corresponding **KB with millions of definite clauses** !

An exercise for you $BU\ C = \{d, e, c, f\}$

Let's consider these two alternative proof procedures for PDCL

KB

$a \leftarrow e \wedge g.$

$b \leftarrow f \wedge g.$

$c \leftarrow e.$

$f \leftarrow c$

$e.$

$d.$

A. C_A = {All clauses in KB with empty bodies} $= \{e, d\}$

B. C_B = {All atoms in the knowledge base}

$\{e\ d\ f\ c\ \overset{*}{g}\ \overset{*}{a}\}$

Both A and B are sound and complete

Both A and B are neither sound nor complete

A is sound only and B is complete only $\Leftarrow$

A is complete only and B is sound only

Can you think of a proof procedure for PDCL….

A: $C_A = \{$all clauses with empty bodies$\}$

$KB \vdash_A G \quad G \subseteq C_A \subseteq C_{BU}$

B: $C_B = \{$all atoms of $KB\}$

$KB \vdash_B G \quad G \subseteq C_B \quad C_{BU} \subseteq C_B$

KB

- $a \leftarrow e \wedge g.$
- $b \leftarrow f \wedge g.$
- $c \leftarrow e.$
- $f \leftarrow c \wedge e.$
- $e.$
- $d.$

C_A

$C_B \{a\ b\ c\ d\ e\ f\ g\}$

- That is sound but not complete?

$KB \vdash_A G \Rightarrow KB \models G$

$\Rightarrow G \subseteq C_A \subseteq C_{BU} \Rightarrow KB \vdash_{BU} G \Rightarrow KB \models G$

soundness of BU

- That is complete but not sound?

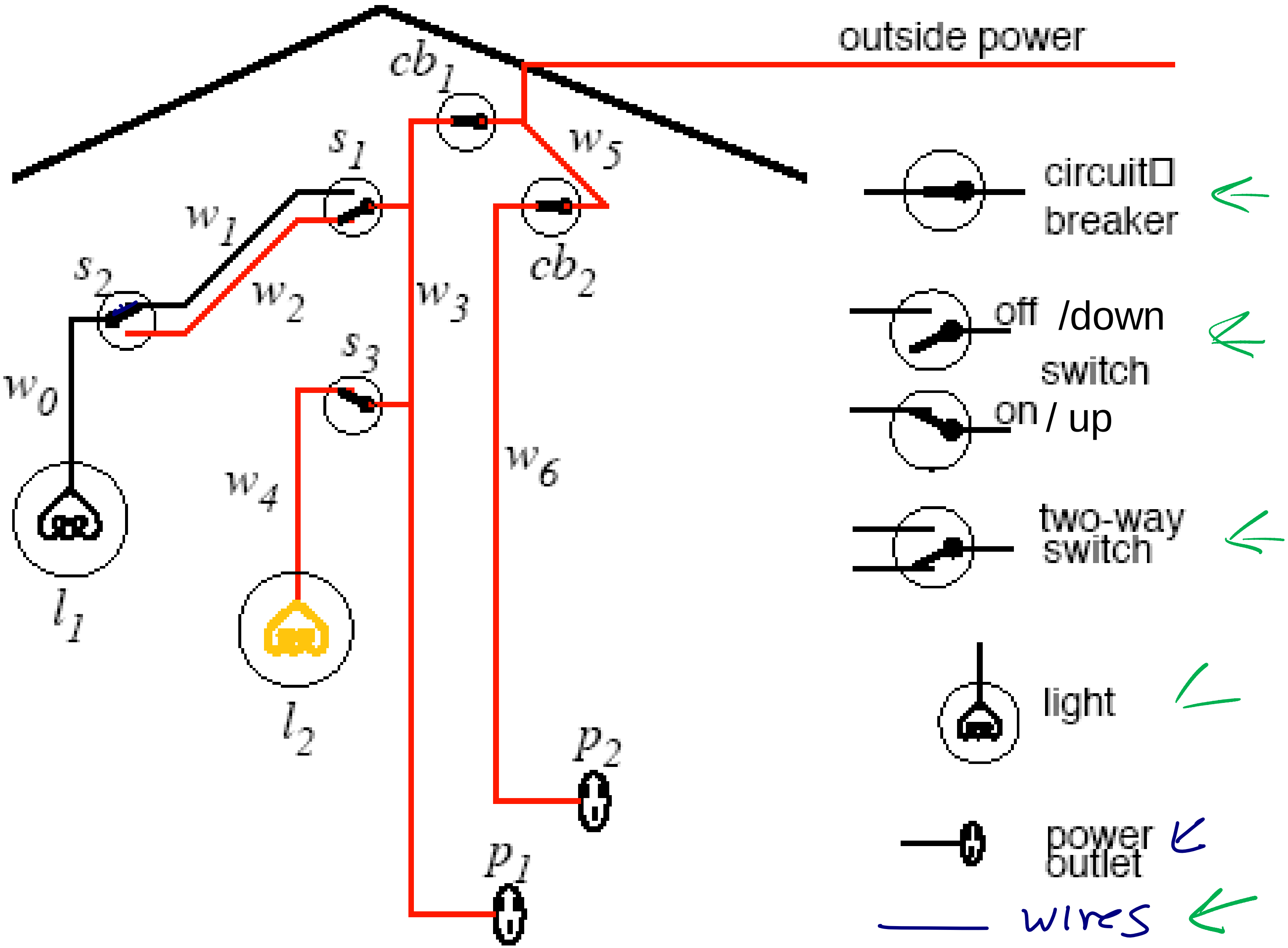
$KB \models G \Rightarrow KB \vdash_B G$

$KB \models G \Rightarrow KB \vdash_{BU} G \Rightarrow G \subseteq C_{BU} \Rightarrow G \subseteq C_B \Rightarrow KB \vdash_B G$

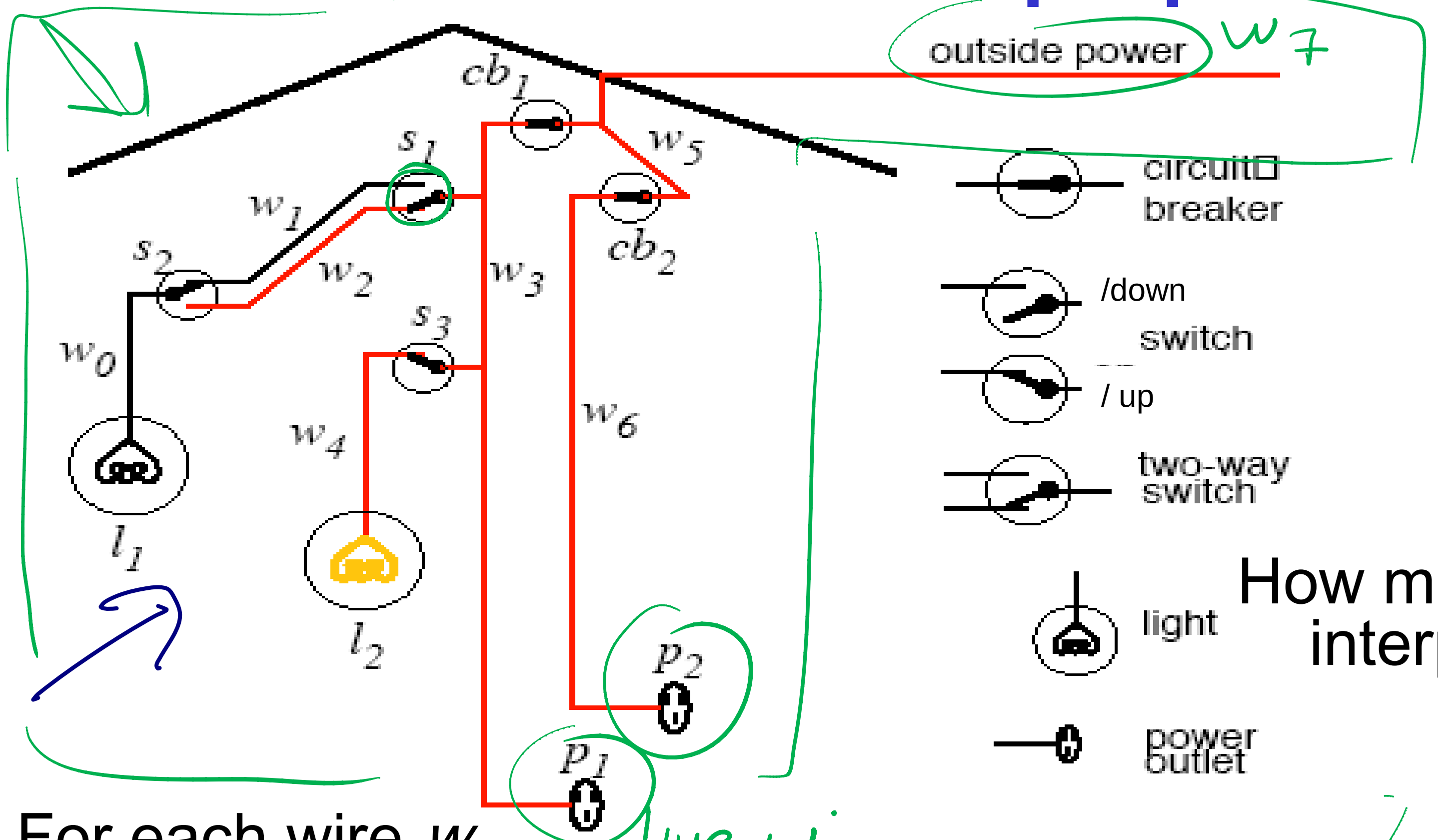
Lecture Overview

- **Recap**: Logic intro and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
  - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

Electrical Environment



Let's define relevant propositions



2^{19}

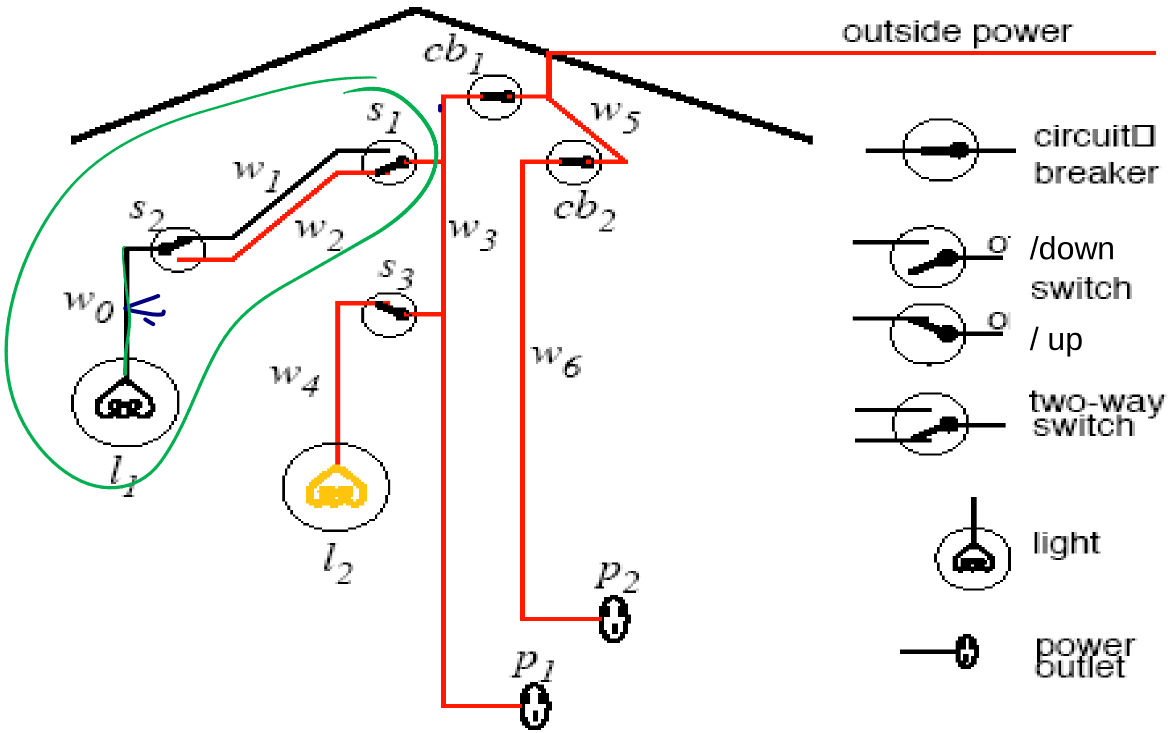
How many interpretations? $\sim 5*10^5$

- For each wire *w* live_wi
- For each circuit breaker *cb* ok_cbi
- For each switch *s* up_si, down_si
- For each light ℓ live_li
- For each outlet *p* live_p

19

- 7
- 2
- 3 x 2
- 2
- 2

Let’s now tell system knowledge about how the domain works



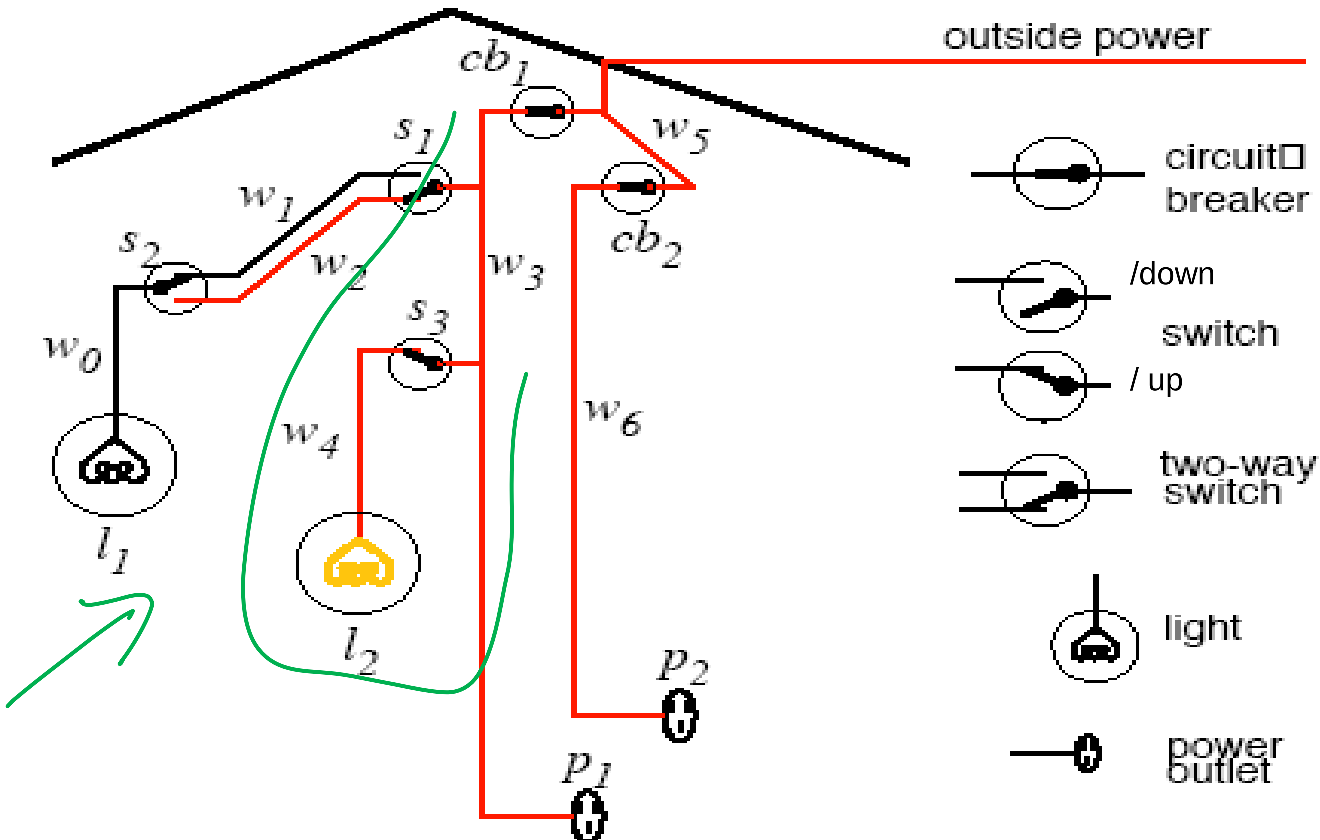
$live\_l_1 \leftarrow live\_w_0$

$live\_w_0 \leftarrow up\_s_2 \wedge live\_w_1$

$live\_w_0 \leftarrow down\_s_2 \wedge live\_w_2$

$live\_w_1 \leftarrow up\_s_1 \wedge live\_w_3$

More on how the domain works….



$live\_w_2 \leftarrow live\_w_3 \wedge down\_s_1.$

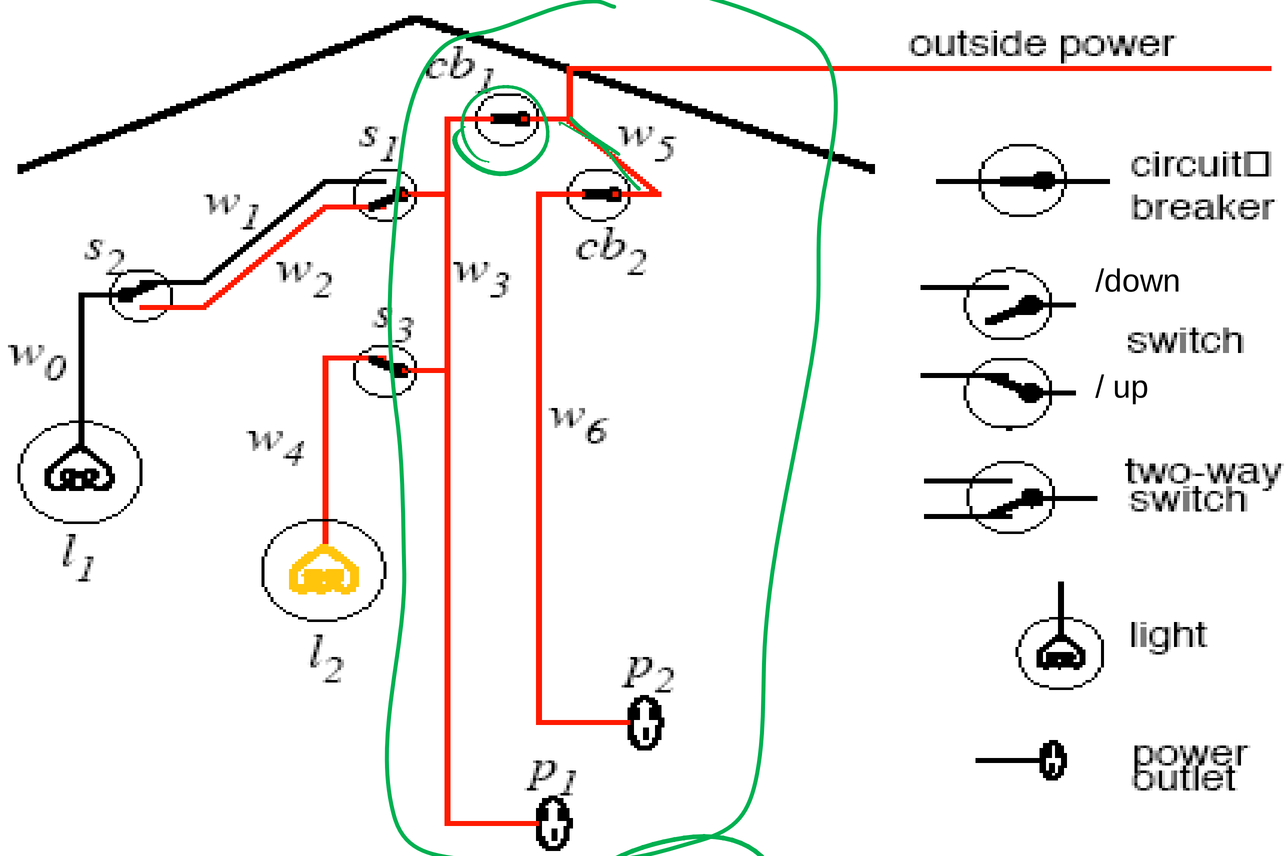
$live\_l_2 \leftarrow live\_w_4.$

$live\_w_4 \leftarrow live\_w_3 \wedge up\_s_3.$

$live\_p_1 \leftarrow live\_w_3.$

.

More on how the domain works….



$live\_w_3 \leftarrow live\_w_5 \land ok\_cb_1.$

$live\_p_2 \leftarrow live\_w_6.$

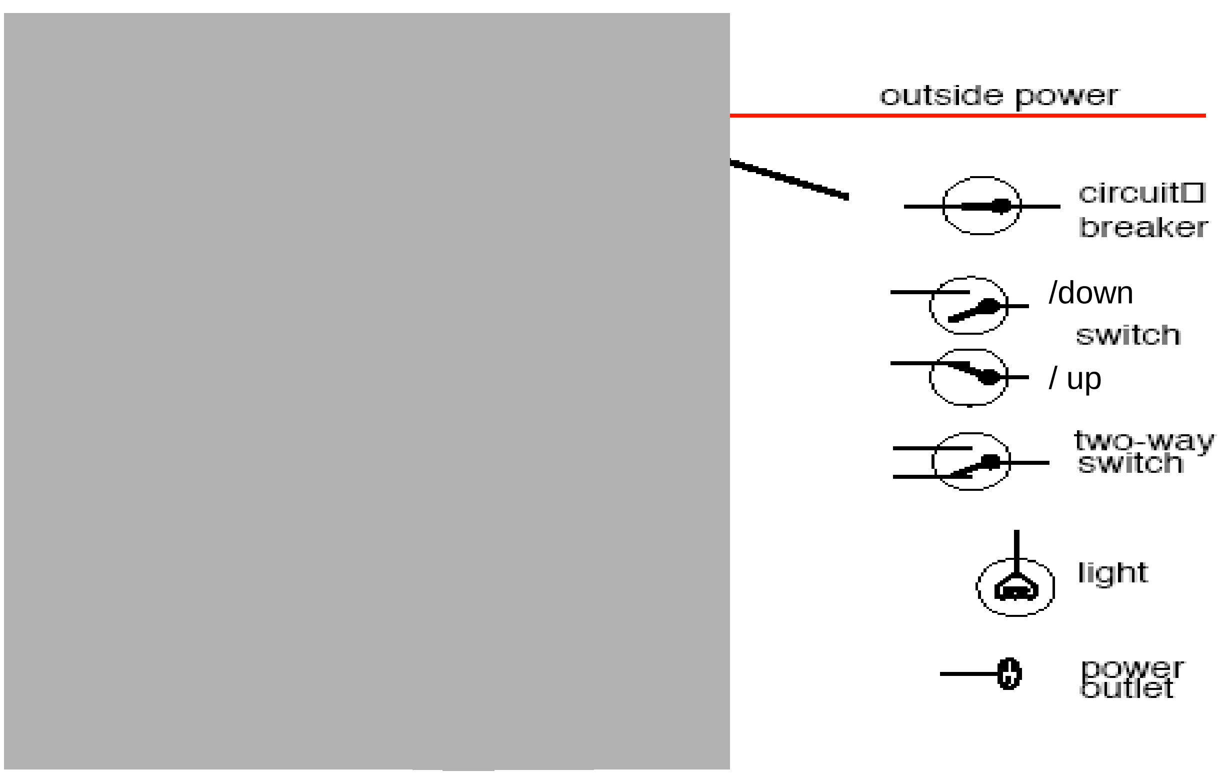
$live\_w_6 \leftarrow live\_w_5 \land ok\_cb_2.$

$live\_w_5 \leftarrow live\_outside.$

What else we may know about this domain?

- That some simple propositions are true

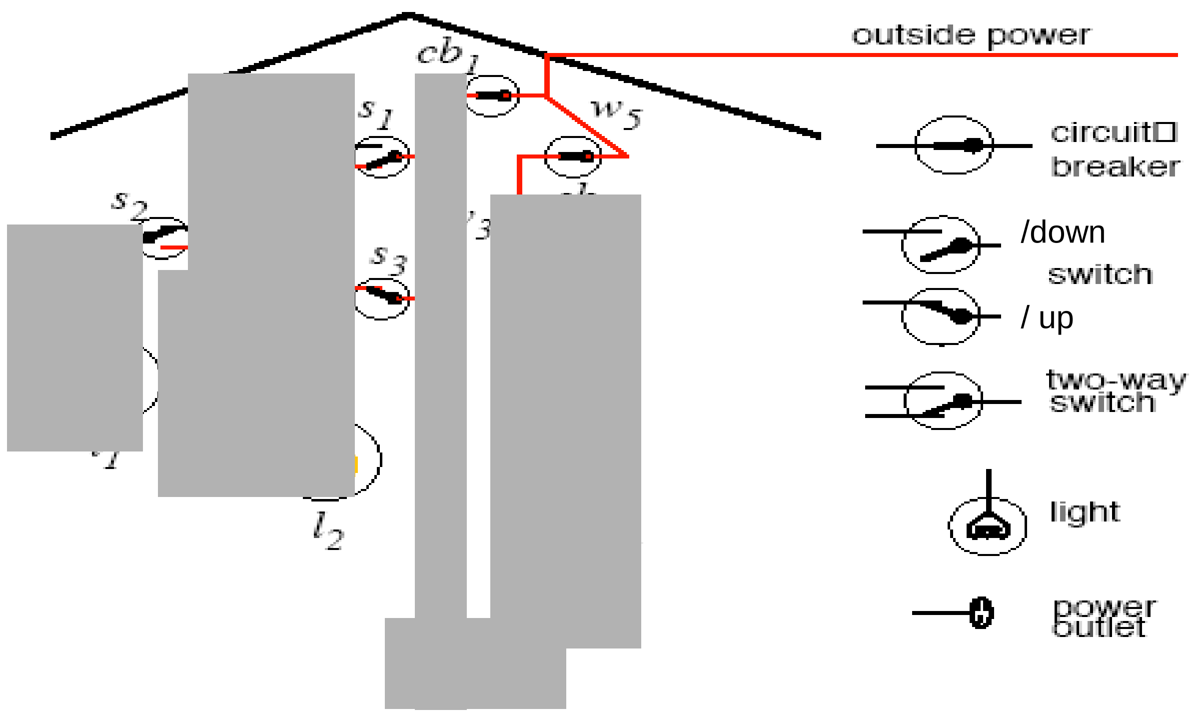
live_outside.



What else we may know about this domain?

- That some additional simple propositions are true

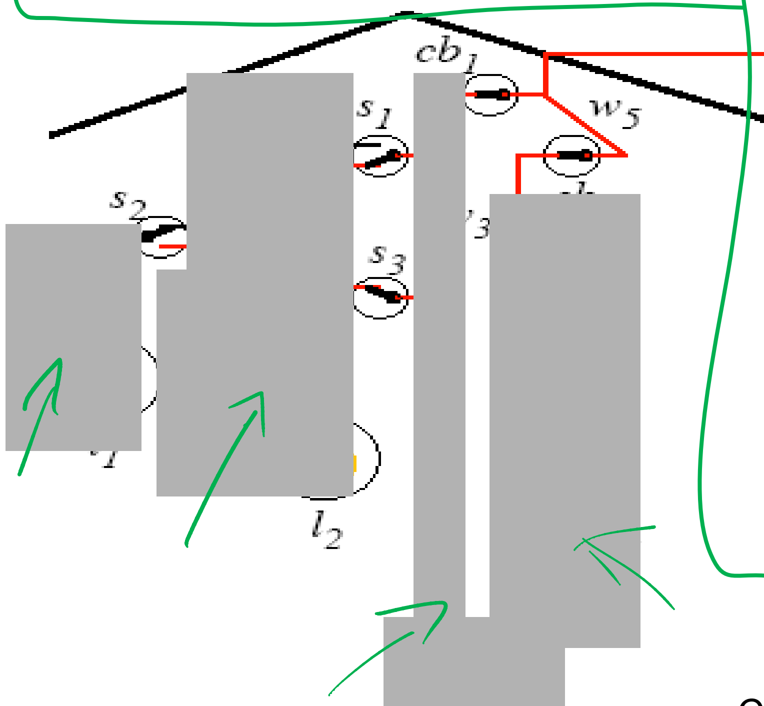
$down\_s_1$. up_s_2. up_s_3. ok_cb_1. ok_cb_2. $live\_outside$.



All our knowledge….. KB

$down\_s_1.$
$up\_s_2.$
$up\_s_3.$
$ok\_cb_1.$
$ok\_cb_2.$
$live\_outside$

$live\_l_1 \leftarrow live\_w_0$
$live\_w_0 \leftarrow live\_w_1 \wedge up\_s_2.$
$live\_w_0 \leftarrow live\_w_2 \wedge down\_s_2.$
$live\_w_1 \leftarrow live\_w_3 \wedge up\_s_1.$
$live\_w_2 \leftarrow live\_w_3 \wedge down\_s_1.$
$live\_l_2 \leftarrow live\_w_4.$
$live\_w_4 \leftarrow live\_w_3 \wedge up\_s_3.$
$live\_p_1 \leftarrow live\_w_3.$
$live\_w_3 \leftarrow live\_w_5 \wedge ok\_cb_1.$
$live\_p_2 \leftarrow live\_w_6.$
$live\_w_6 \leftarrow live\_w_5 \wedge ok\_cb_2.$
$live\_w_5 \leftarrow live\_outside.$



What Semantics is telling us

- Our KB (all we know about this domain) is going to be true only in a subset of all possible ____2^{19}____ interpretations

- What is logically entailed by our KB are all the propositions that are true in all those models

- This is what we should be able to derive given a sound and complete proof procedure

If we apply the bottom-up (BU) proof procedure

$down\_s_1.$
$up\_s_2.$
$up\_s_3.$
$ok\_cb_1.$
$ok\_cb_2.$
$live\_outside.$

C

$live\_l_1 \leftarrow live\_w_0$
$live\_w_0 \leftarrow live\_w_1 \wedge up\_s_2.$
$live\_w_0 \leftarrow live\_w_2 \wedge down\_s_2.$
$live\_w_1 \leftarrow live\_w_3 \wedge up\_s_1.$
$live\_w_2 \leftarrow live\_w_3 \wedge down\_s_1.$
$live\_l_2 \leftarrow live\_w_4.$
$live\_w_4 \leftarrow live\_w_3 \wedge up\_s_3.$
$live\_p_1 \leftarrow live\_w_3..$
$live\_w_3 \leftarrow live\_w_5 \wedge ok\_cb_1.$
$live\_p_2 \leftarrow live\_w_6.$
$live\_w_6 \leftarrow live\_w_5 \wedge ok\_cb_2.$
$live\_w_5 \leftarrow live\_outside.$

BU generates C

all the atoms added to C are in green

$\boxed{live\_l_2}$? ✓ → $live\_l_2 \in C \Rightarrow KB \vdash_{BU} live\_l_2 \Rightarrow KB \models live\_l_2$

$\boxed{live\_l_1}$ ✗ which is not the case for $live\_l_1$

Lecture Overview

- **Recap**: Logic intro and Propositional Definite Clause (PDCL)
- **Semantics** (interpretation, model, logical consequence)
- **Bottom-up** Proof Procedure for PDCL
  - Soundness and Completeness
- Using PDCL for **R&R in a domain** (Electrical System)
- **Top-Down** Proof Procedure for PDCL

Bottom-up vs. Top-down

Bottom-up

KB → C

G is proved if $G \subseteq C$

When does BU look at the query? g

In every loop iteration

Never

At the end

At the beginning

Bottom-up vs. Top-down

- **Key Idea of top-down:** search backward from a query g to determine if it can be derived from *KB*.

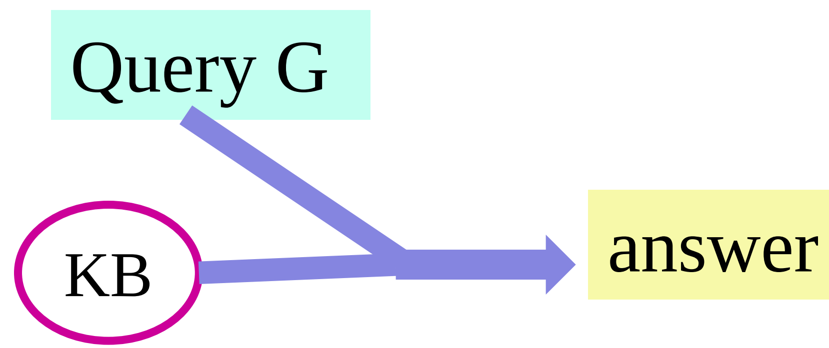
Bottom-up

g is proved if $G \subseteq C$

When does BU look at the query G?

- At the end

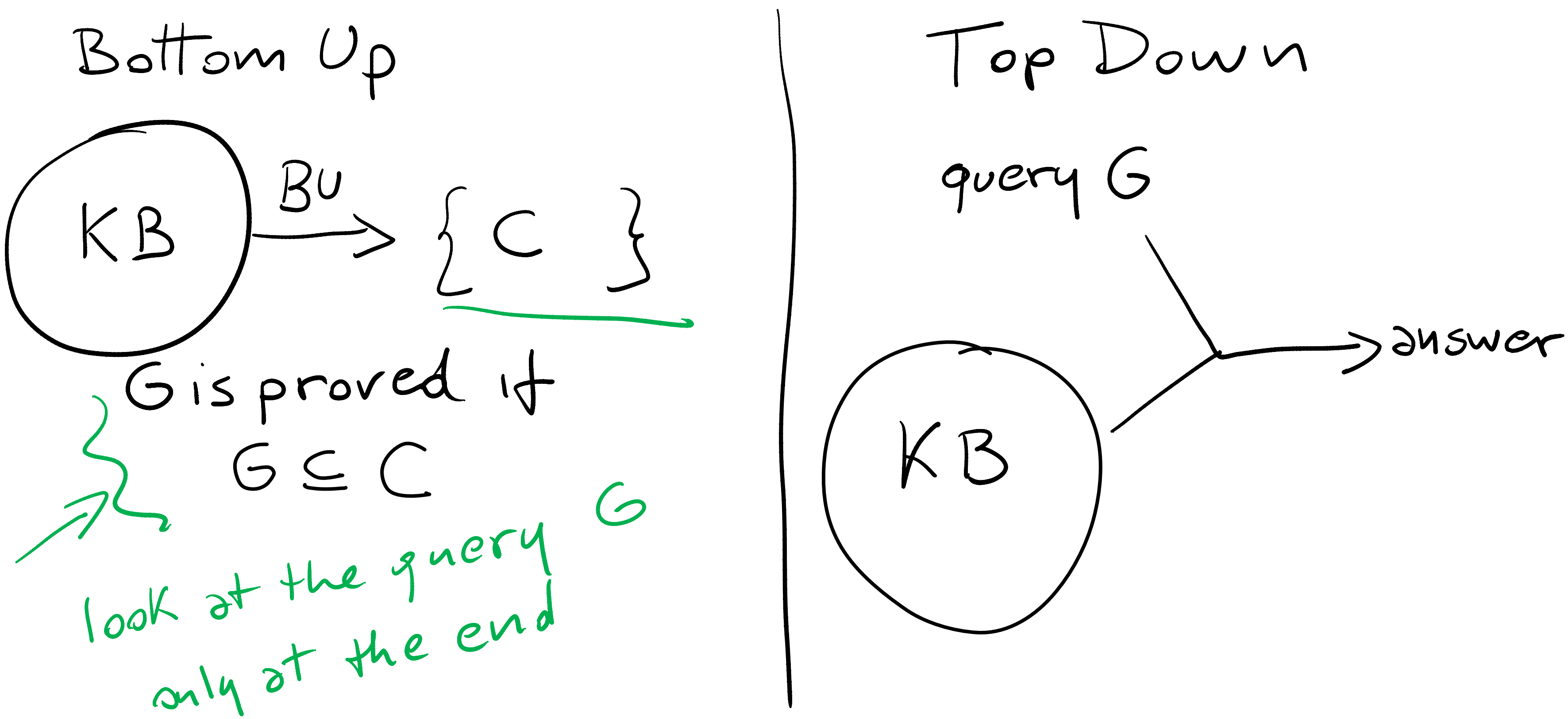
Top-down



TD performs a backward search starting at G

Top-down Ground Proof Procedure

Key Idea: search backward from a query *G* to determine if it can be derived from *KB*.



Top-down Proof Procedure: Basic elements

Notation: An answer clause is of the form:

$yes \leftarrow a_1 \land a_2 \land \ldots \land a_m$ G

Express query as an answer clause (e.g., query $a_1 \land a_2 \land \ldots \land a_m$)

$yes \leftarrow a_2 \land \ldots \land a_m$

Rule of inference (called SLD Resolution)

Given an answer clause of the form:

$\boxed{yes \leftarrow a_1 \land a_2 \land \ldots \land a_m}$

KB

and the clause:

$a_i \leftarrow b_1 \land b_2 \land \ldots \land b_p$

You can generate the answer clause

$yes \leftarrow a_1 \land \ldots \land a_{i-1} \land b_1 \land b_2 \land \ldots \land b_p \land a_{i+1} \land \ldots \land a_m$

Rule of inference: Examples

Rule of inference (called SLD Resolution)

Given an answer clause of the form:

$$yes \leftarrow a_1 \land a_2 \land \ldots \land a_m$$

KB

and the clause:

$$a_i \leftarrow b_1 \land b_2 \land \ldots \land b_p$$

You can generate the answer clause

$$yes \leftarrow a_1 \land \ldots \land a_{i-1} \land b_1 \land b_2 \land \ldots \land b_p \land a_{i+1} \land \ldots \land a_m$$

KB clause

$\boxed{yes \leftarrow b \land c.}$ $\quad b \leftarrow k \land f.$ $\quad \Rightarrow yes \leftarrow k \land f \land c$

KB

$yes \leftarrow e \land f.$ $\quad e. \leftarrow$ $\quad \Rightarrow yes \leftarrow f$

(successful) Derivations

- An answer is an answer clause with $m = 0$. That is, it is the answer clause $yes \leftarrow .$

G

- A (successful) derivation of query "$?q_1 \wedge \ldots \wedge q_k$" from KB is a sequence of answer clauses $\gamma_0, \gamma_1, \ldots, \gamma_n$ such that
  - γ_0 is the answer clause $yes \leftarrow q_1 \wedge \ldots \wedge q_k$
  - γ_i is obtained by resolving γ_{i-1} with a clause in KB, and
  - γ_n is an answer. $yes \leftarrow .$

- An unsuccessful derivation…..

$yes \leftarrow a \wedge b$

Example: derivations

$\checkmark$

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

KB

Query: a `(two ways)`

$\boxed{yes \leftarrow a.}$

$\quad '' \leftarrow b \wedge c$

$\quad '' \leftarrow k \wedge f \wedge c$

K cannot be eliminated so will Fail

$yes \leftarrow a.$

$\quad '' \leftarrow e \wedge f$

$\quad '' \leftarrow f$

$\quad '' \leftarrow c$

$\quad '' \leftarrow e$

$yes \leftarrow \ldots$

Query: b `(`*k*`, `*f*` different order)`

$yes \leftarrow b.$

Learning Goals for today's class – part 2

You can:

- Prove that BU proof procedure is **complete**
- Model a relatively simple domain with propositional definite clause logic (PDCL)
- Trace query derivation using SLD resolution rule of inference

Course Big Picture

Environment

Problem		Deterministic	Stochastic
Static	Constraint Satisfaction	Arc Consistency *Vars + Constraints* Search → for CSP SLS	
Static	Inference	→ CSP *Logics* Search → for Inference	*Belief Nets* Var. Elimination
Sequential	Planning	*STRIPS* → CSP Search → for complex planning	*Decision Nets* Var. Elimination *Markov Processes* Value Iteration

Representation

Reasoning Technique

Next Class

- Finish Logics (Datalog) (12.3)
- MAJOR TRANSITION

STOCHASTIC ENVIRONMENTS

- Probability Theory and Conditional Probability (6.1)

(Propositional) Logic: Key ideas

Given a domain that can be represented with **n propositions** you have …… interpretations (possible worlds) 2^n

If you do not know anything you can be in any of those

If you know that some **logical formulas** are true (your KB ……). You know that you can be only in …….. interpretations in which the KB is true (i.e. the models of KB)

It would be nice to know what else is true in all those… models what is logically entailed

PDCL syntax / semantics / proofs

Domain can be represented by three propositions: *p, q, r*

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

Interpretations?

r	q	p
T	T	T
~~T~~	~~T~~	~~F~~
~~T~~	~~F~~	~~T~~
~~T~~	~~F~~	~~F~~
~~F~~	~~T~~	~~T~~
~~F~~	~~T~~	~~F~~
~~F~~	~~F~~	~~T~~
~~F~~	~~F~~	~~F~~

Models?

What is logically entailed ? r, q, p

$G \subseteq C$

Prove $G = \boxed{(q \wedge p)}$

$C = \{q, r, p\}$

$\Updownarrow$

$KB \vdash_{BU} G$

PDCL syntax / semantics / proofs

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

Interpretations

r	q	p
T	T	T
~~T~~	~~T~~	~~F~~
~~T~~	~~F~~	~~T~~
~~T~~	~~F~~	~~F~~
F	T	T
F	T	F
~~F~~	~~F~~	~~T~~
~~F~~	~~F~~	~~F~~

Models

What is logically entailed?

$C = \{q\}$

Prove $G = (q \wedge p)$

$G \notin C$ $\qquad KB \nvdash_{BU} G$

To Define a Logic We Need

- Syntax: specifies the symbols used, and how they can be combined to form legal sentences
  - **Knowledge base** is a set of sentences in the language
- Semantics: specifies the meaning of symbols and sentences
- Reasoning theory or proof procedure: a specification of how an answer can be produced.
  - **Sound**: only generates correct answers with respect to the semantics
  - **Complete**: Guaranteed to find an answer if it exists

Propositional Definite Clauses: Syntax

Definition (atom)

An **atom** is a symbol starting with a lower case letter

Examples: p_1; $live\_l_1$

Definition (body)

A **body** is an atom or is of the form $b_1 \land b_2$ where b_1 and b_2 are bodies.

Examples: $p_1 \land p_2$; $ok\_w_1 \land live\_w_0$

Definition (definite clause)

A **definite clause** is

- an atom or
- a rule of the form $h \leftarrow b$ where h is an atom ("head") and b is a body. (Read this as "h if b".)

Examples: $p_1 \leftarrow p_2$; $live\_w_0 \leftarrow live\_w_1 \land up\_s_2$

Definition (KB)

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

PDC Semantics: Example for models

Definition (model)
A **model** of a knowledge base KB is an interpretation in which every clause in KB is true.

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

Which of the interpretations below are models of KB?

	p	q	r	s		
I_1	T	T	T	T	yes	no
I_2	F	F	F	F	yes	no
I_3	T	T	F	F	yes	no
I_4	T	T	T	F	yes	no
I_5	F	T	F	T	yes	no

PDC Semantics: Example for models

Definition (model)
A **model** of a knowledge base KB is an interpretation in which every clause in KB is true.

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

Which of the interpretations below are models of KB?

	p	q	r	s	$p \leftarrow q$	q	$r \leftarrow s$	Model of KB
I_1	T	T	T	T	T	T	T	yes no
I_2	F	F	F	F		F		yes no
I_3	T	T	F	F	T	T	T	yes no
I_4	T	T	T	F				yes no
I_5	F	T	F	T				yes no

PDC Semantics: Example for models

Definition (model)
A **model** of a knowledge base KB is an interpretation in which every clause in KB is true.

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

Which of the interpretations below are models of KB?
All interpretations where KB is true: I_1, I_3, and I_4

	p	q	r	s	$p \leftarrow q$	q	$r \leftarrow s$	Model of KB
I_1	T	T	T	T	T	T	T	yes
I_2	F	F	F	F	T	F	T	no
I_3	T	T	F	F	T	T	T	yes
I_4	T	T	T	F	T	T	T	yes
I_5	F	T	F	T	F	T	F	no

PDCL Semantics: Logical Consequence

Definition (model)
A **model** of a knowledge base KB is an interpretation in which every clause in KB is 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

$$KB = \left\{ \begin{array}{l} p \leftarrow q \\ q \\ r \leftarrow s \end{array} \right.$$

Which of the following are true?

$KB \models p$ $KB \models q$ $KB \models r$ $KB \models s$

PDCL Semantics: 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 ⊧ g,
if g is true in every model of KB

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

Which of the following are true?

KB ⊧ p

KB ⊧ q

If KB is true, then q is true. Thus KB ⊧ q.

If KB is true then both q and p ← q are true,
so p is true ("p if q"). Thus KB ⊧ p.

There is a model where r is false, likewise for s

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*
I_9	*true*	*false*	*true*	*true*
I_{10}	*true*	*false*	*true*	*false*
I_{11}	*true*	*false*	*false*	*false*
I_{12}	*true*	*false*	*false*	*false*
…..	……	…..	……	…….

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

Which of the following is true?

- $KB \models q$, T
- $KB \models p$, T
- $KB \models s$, F
- $KB \models r$ F

An exercise for you

Let's consider these two alternative proof procedures for PDCL

A. C_A = {All clauses in KB with empty bodies}

B. C_B = {All atoms in the knowledge base}

KB

$a \leftarrow e \wedge g.$

$b \leftarrow f \wedge g.$

$c \leftarrow e.$

$f \leftarrow c$

$e.$

$d.$

Both A and B are sound and complete

Both A and B are neither sound nor complete

A is sound only and B is complete only

A is complete only and B is sound only

An exercise for you

Let's consider these two alternative proof procedures for PDCL

A. C_A = {All clauses in KB with empty bodies}

B. C_B = {All atoms in the knowledge base}

KB

$a \leftarrow e \wedge g.$

$b \leftarrow f \wedge g.$

$c \leftarrow e.$

$f \leftarrow c$

$e.$

$d.$

A is sound only and B is complete only

Bottom-up vs. Top-down

Bottom-up

g is proved if g $\in$ C

When does BU look at the query? g

In every loop iteration

Never

At the end

At the beginning

Bottom-up vs. Top-down

- **Key Idea of top-down:** search backward from a query g to determine if it can be derived from *KB*.

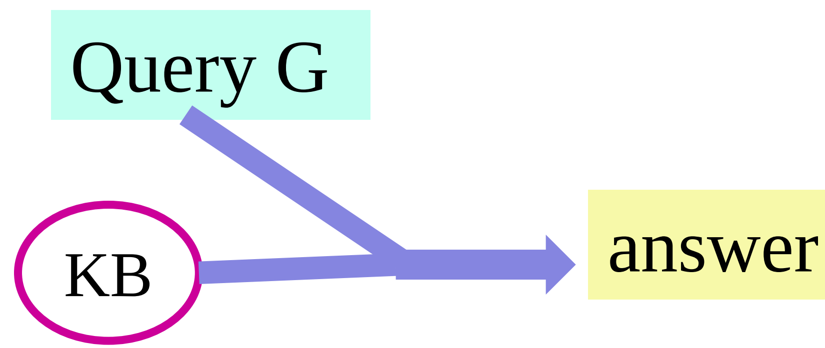
Bottom-up

g is proved if g $\in$ C

When does BU look at the query g?

- Never
- It derives the same C regardless of the query

Top-down



TD performs a backward search starting at g