Logic: Domain Modeling /Proofs + Top-Down Proofs

Computer Science cpsc322, Lecture 22

(Textbook Chpt 5.2)

June, 8, 2017

Lecture Overview

- Recap
- Using Logic to Model a Domain (Electrical System)
- Reasoning/Proofs (in the Electrical Domain)
- Top-Down Proof Procedure

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….

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

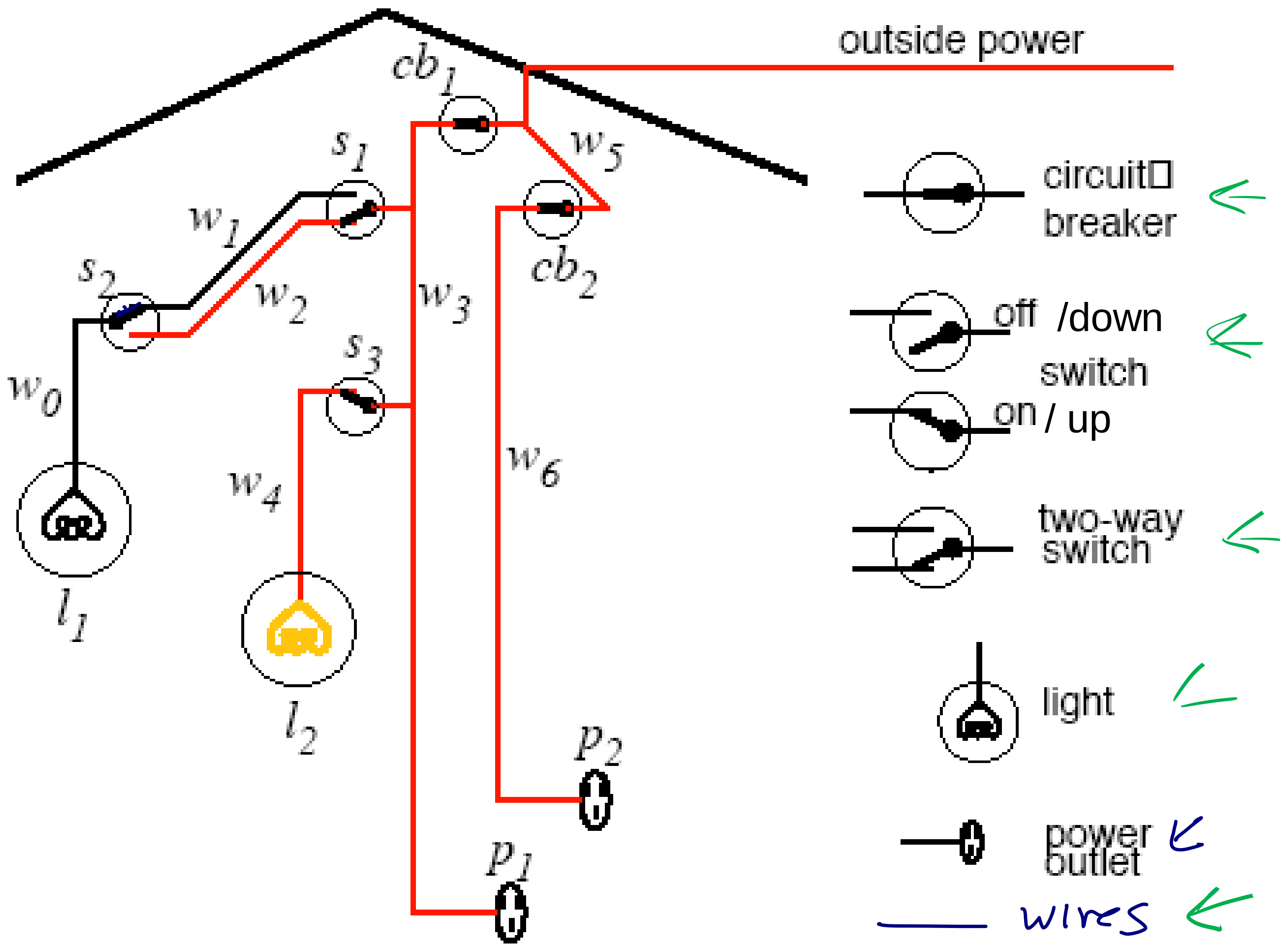
- BottomUp for PDCL is ← sound & complete

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

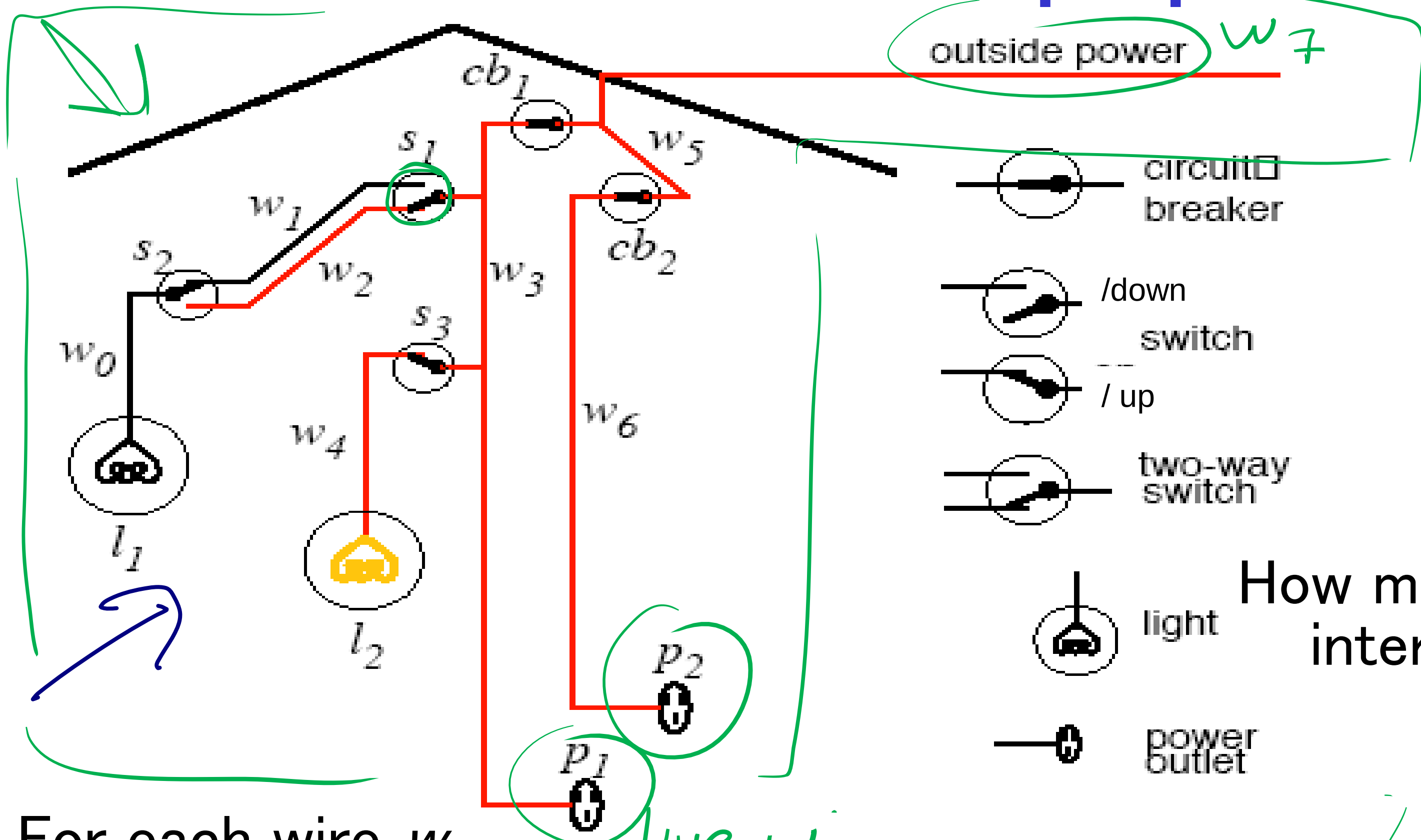
Lecture Overview

- Recap
- Using PDCL Logic to Model a Domain (Electrical System)
- Reasoning/Proofs (in the Electrical Domain)
- Top-Down Proof Procedure

Electrical Environment



Let's define relevant propositions

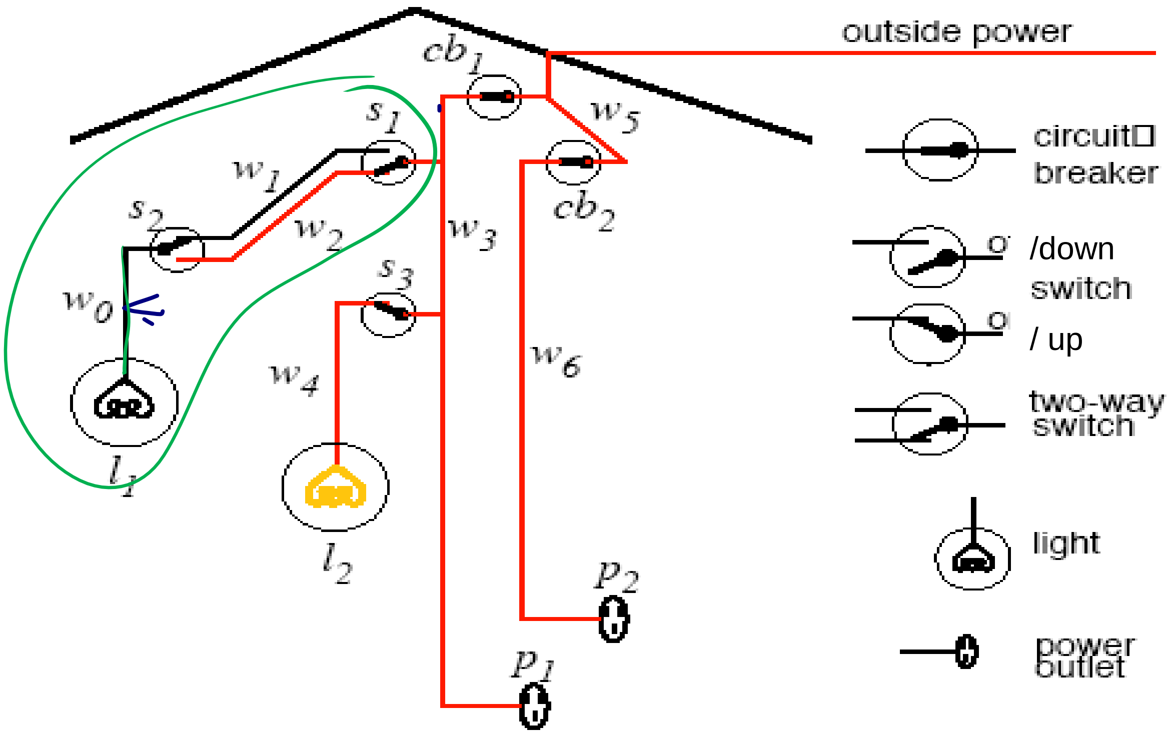


How many interpretations?

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

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

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



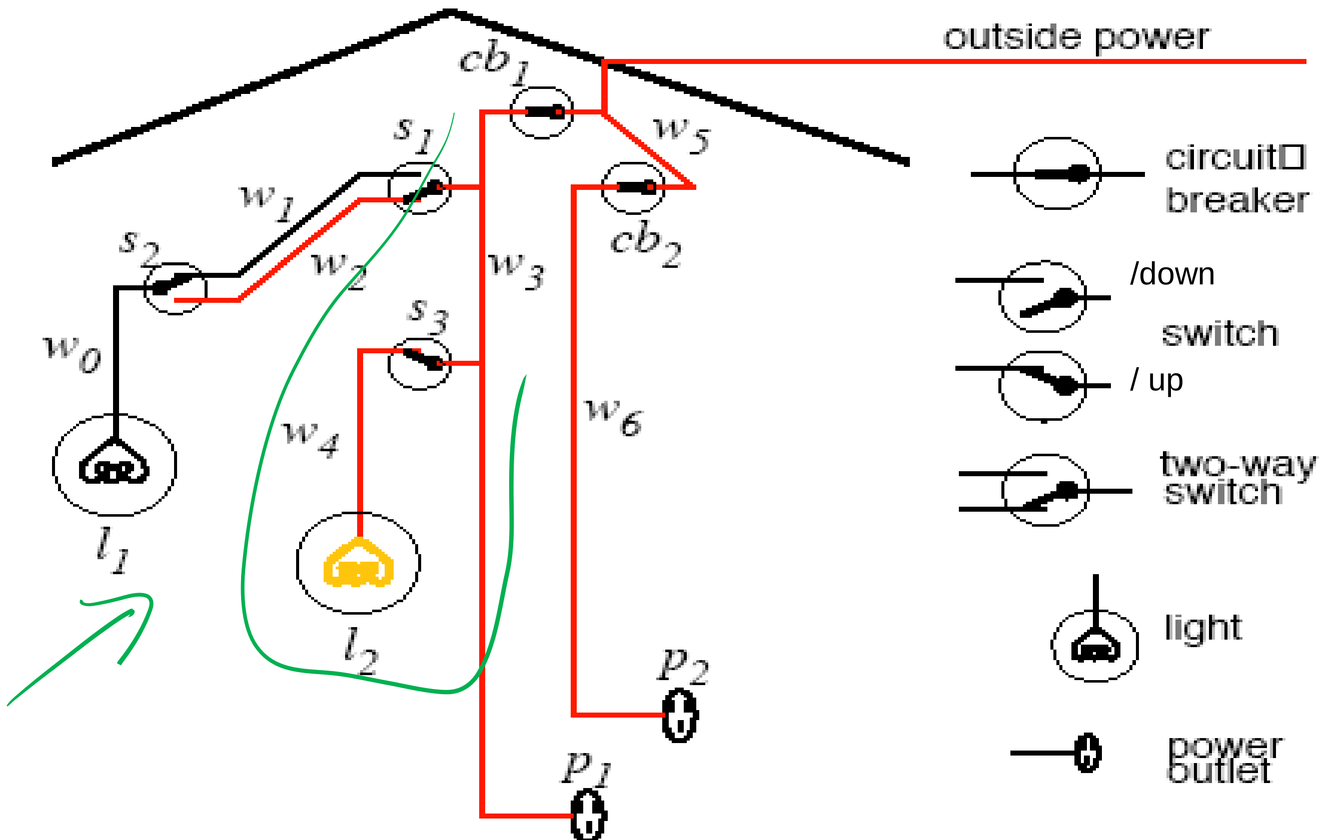
$live\_l_1 \leftarrow live\_w_0$

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

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

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

More on how the domain works….



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

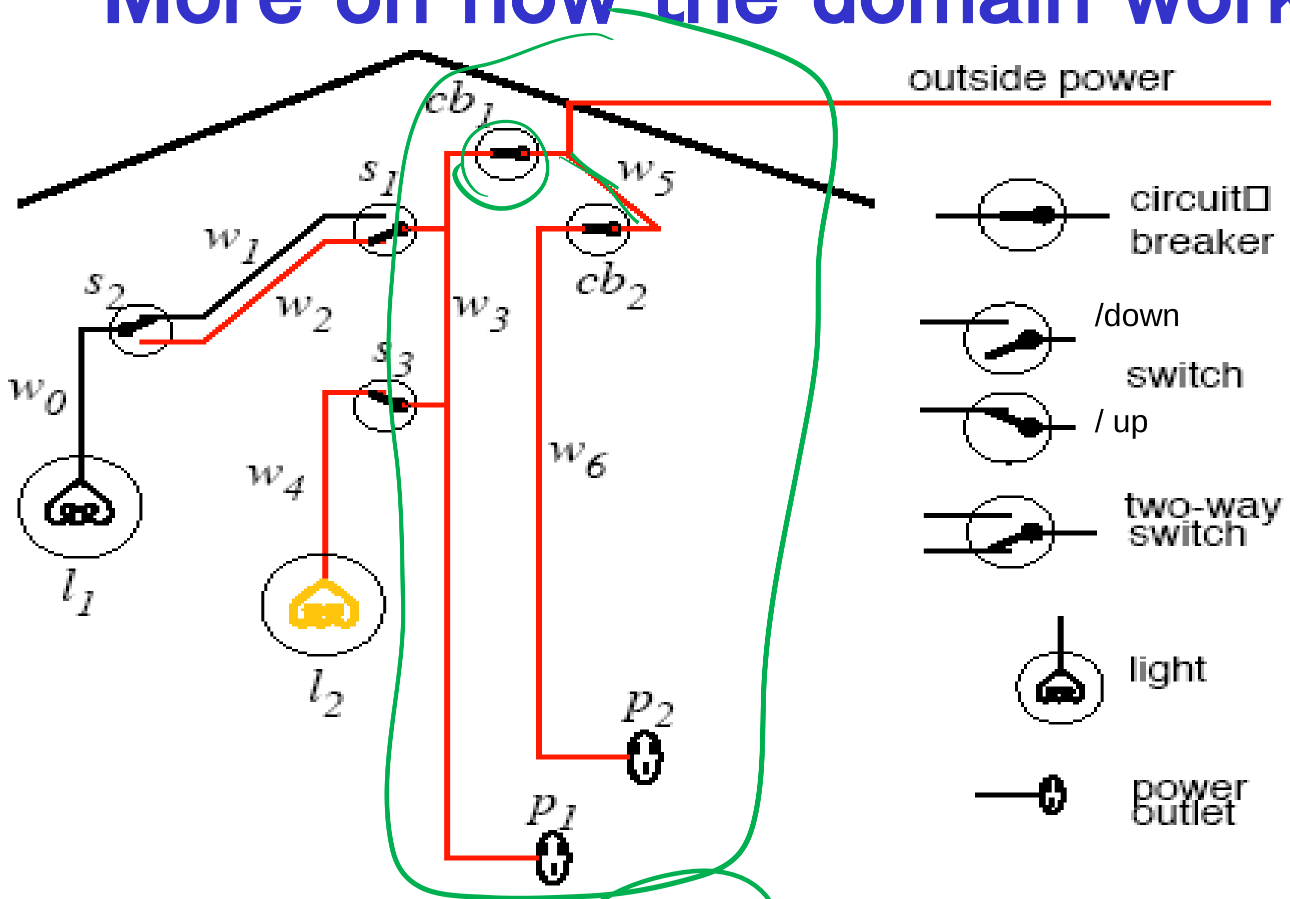
$live\_l_2 \leftarrow live\_w_4.$

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

$live\_p_1 \leftarrow live\_w_3.$

.

More on how the domain works….



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

$live\_p_2 \leftarrow live\_w_6.$

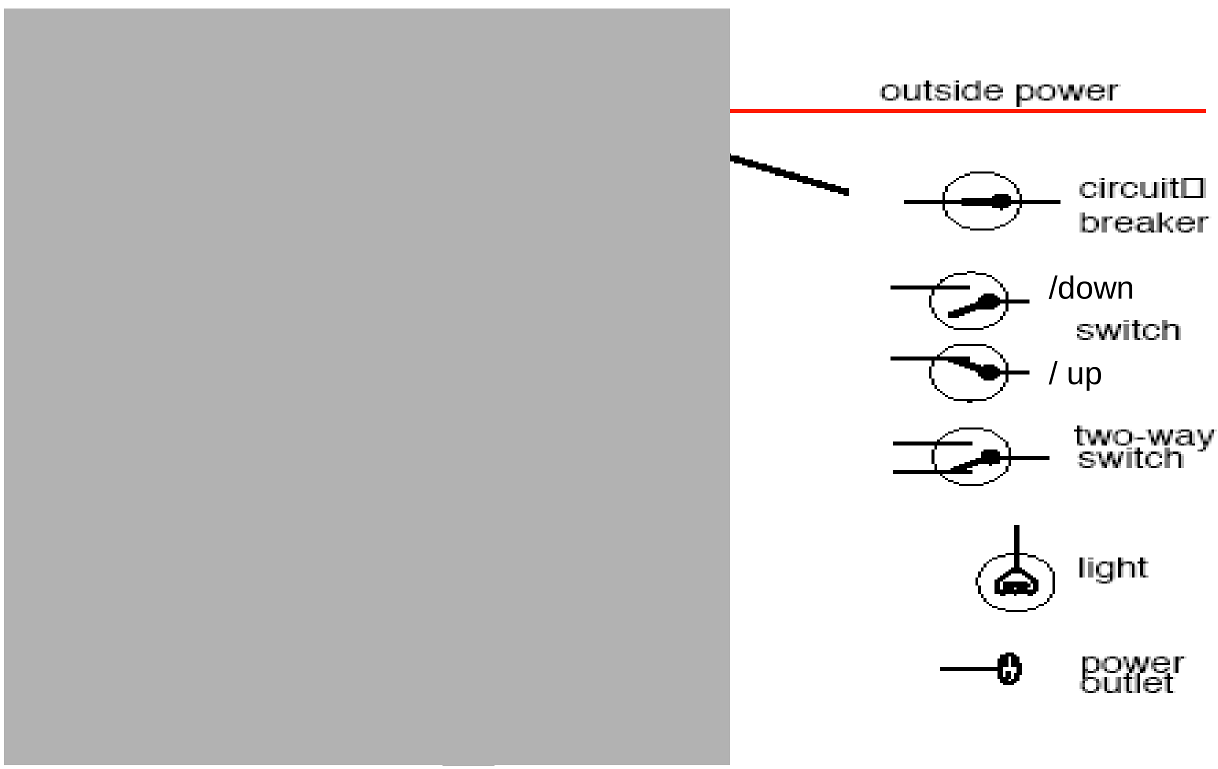
$live\_w_6 \leftarrow live\_w_5 \wedge ok\_cb_2.$

$live\_w_5 \leftarrow live\_outside.$

What else we may know about this domain?

- That some simple propositions are true

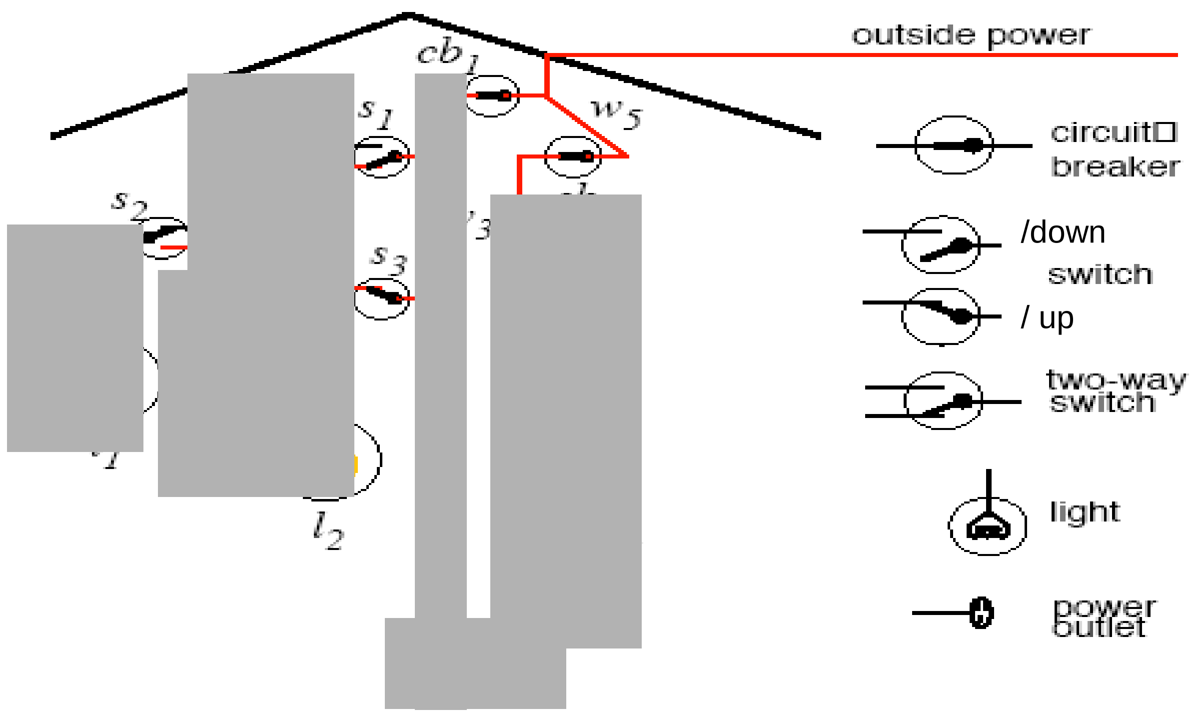
  live_outside.



What else we may know about this domain?

- That some additional simple propositions are true

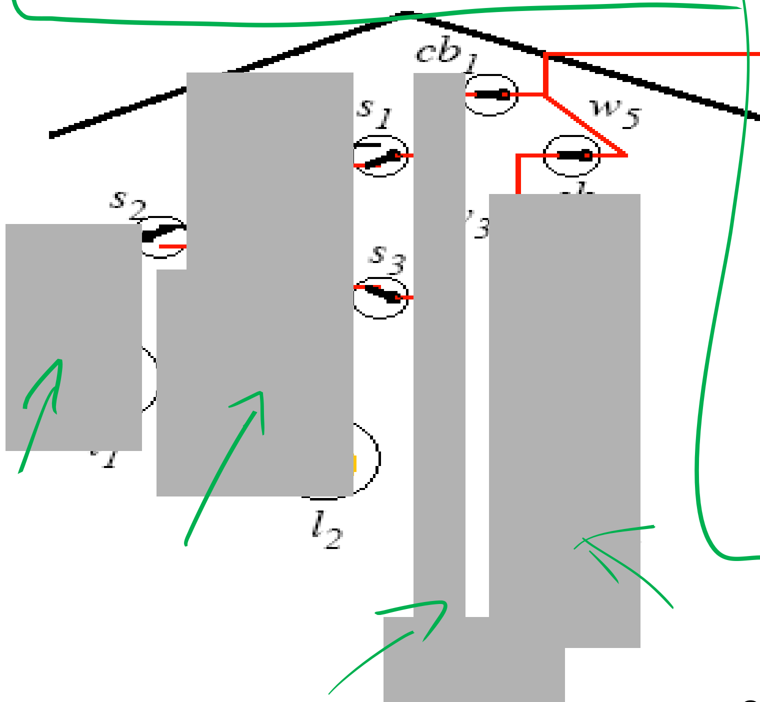
$down\_s_1.$ $up\_s_2.$ $up\_s_3.$ $ok\_cb_1.$ $ok\_cb_2.$ $live\_outside.$



All our knowledge….. KB

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

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



Lecture Overview

- Recap
- Using Logic to Model a Domain (Electrical System)
- Reasoning/Proofs (in the Electrical Domain)
- Top-Down Proof Procedure

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 ________ interpretations 2^{19}

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

  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_s₁.
up_s₂.
up_s₃.
ok_cb₁.
ok_cb₂.
live_outside

C

BU generates C — all the atoms added to C are in green

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

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

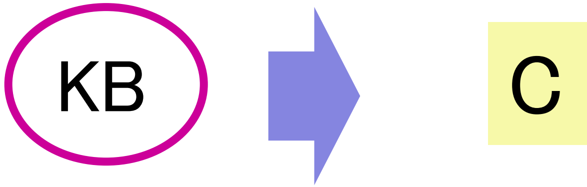
$live\_l_1$ ✗ which is not the case for $live\_l_1$

Lecture Overview

- Recap
- Using Logic to Model a Domain (Electrical System)
- Reasoning/Proofs (in the Electrical Domain)
- Top-Down Proof Procedure

Bottom-up vs. Top-down

Bottom-up



G is proved if $G \subseteq C$

When does BU look at the query G ?

A. In every loop iteration

B. Never

C. Only at the end

D. Only 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*.

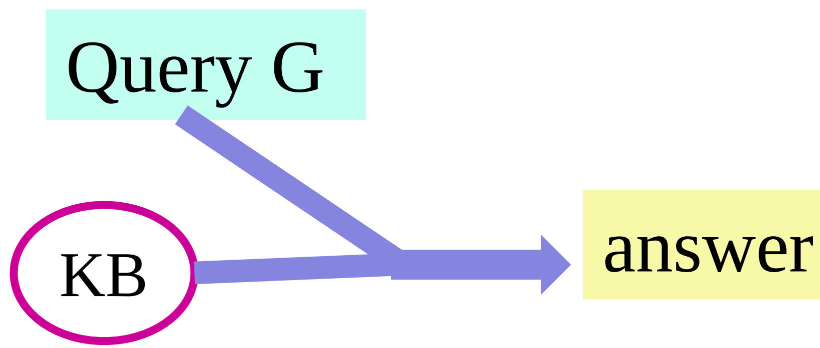
Bottom-up

G is proved if $G \subseteq C$

When does BU look at the query G?

- At the end

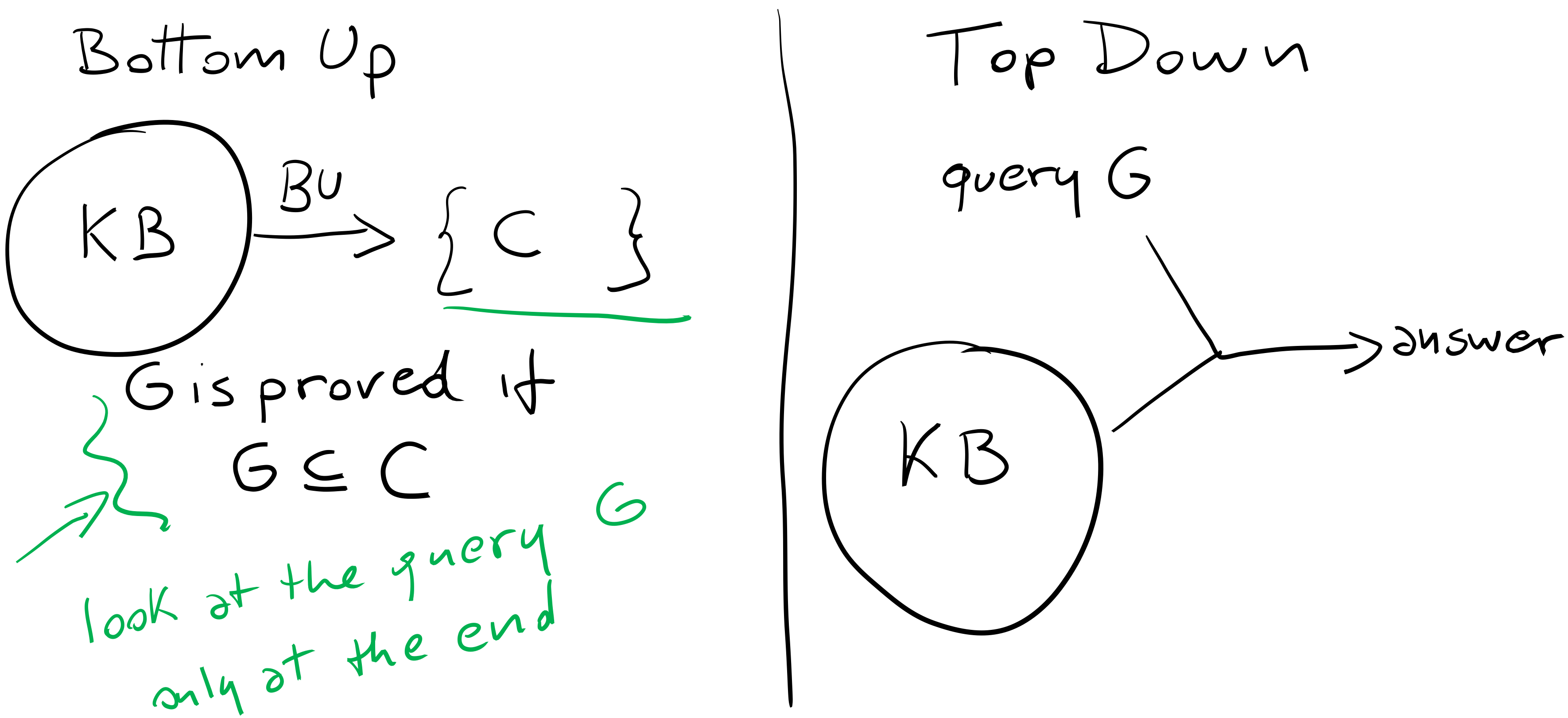
Top-down



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:

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

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

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

Rule of inference (called SLD Resolution)

Given an answer clause of the form:

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

and the clause:

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

You can generate the answer clause

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

Rule of inference: Examples

Rule of inference (called SLD Resolution)

Given an answer clause of the form:

$$yes \leftarrow a_1 \wedge a_2 \wedge \cdots \wedge a_m$$

and the KB clause:

$$a_i \leftarrow b_1 \wedge b_2 \wedge \cdots \wedge b_p$$

You can generate the answer clause

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

KB clause

$yes \leftarrow b \wedge c.$ $\quad b \leftarrow k \wedge f.$ $\Rightarrow yes \leftarrow k \wedge f \wedge c$

KB

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

(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 "$?q_1 \wedge \cdots \wedge q_k$" from KB is a sequence of answer clauses $\gamma_0, \gamma_1, \cdots, \gamma_n$ such that
  - γ_0 is the answer clause $yes \leftarrow q_1 \wedge \cdots \wedge q_k$
  - 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

KB:

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

$c \leftarrow e.$ $d \leftarrow k.$ $e.$

$f \leftarrow j \wedge e.$ $f \leftarrow c.$ $j \leftarrow c.$

Query: a (two ways)

$yes \leftarrow a.$

" $\leftarrow b \wedge c$

" $\leftarrow k \wedge f \wedge c$

k cannot be eliminated so will fail

$yes \leftarrow a.$

" $\leftarrow e \wedge f$

" $\leftarrow f$

" $\leftarrow c$

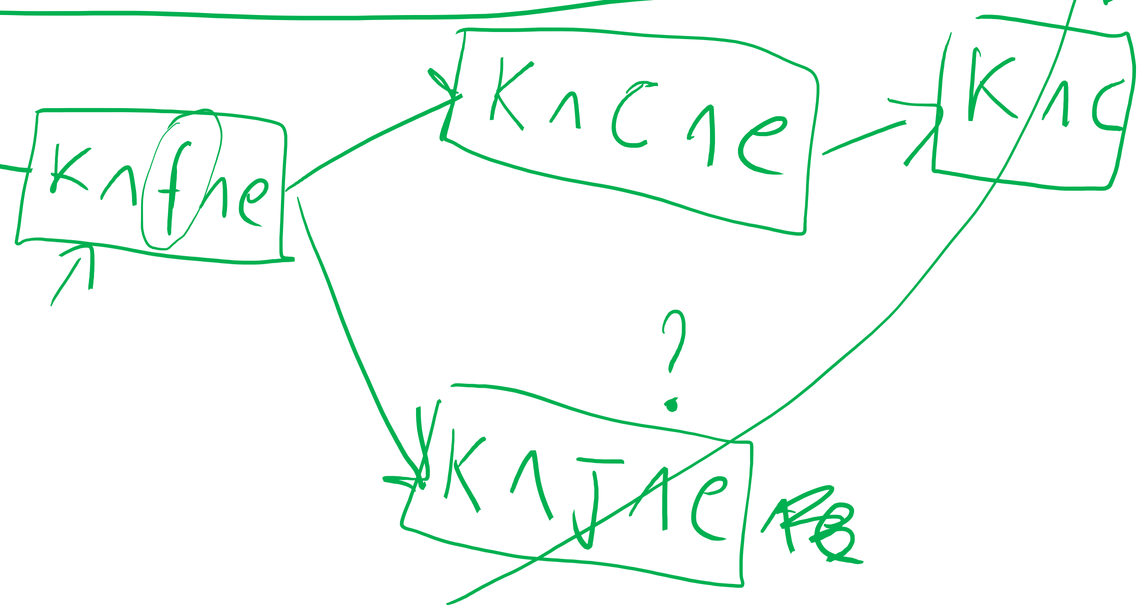
" $\leftarrow e$

$yes \leftarrow$ . . . .

Example: derivations

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

Query: $b \wedge e$



A. Provable by TD

B. It depends

C. Not Provable by TD

Course Big Picture

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

Representation

Reasoning Technique

Standard Search vs. Specific R&R systems

Constraint Satisfaction (Problems):

- State: assignments of values to a subset of the variables
- Successor function: assign values to a "free" variable
- Goal test: set of constraints
- Solution: possible world that satisfies the constraints
- Heuristic function: *none (all solutions at the same distance from start)*

Planning :

- State possible world
- Successor function states resulting from valid actions
- Goal test assignment to subset of vars
- Solution sequence of actions
- Heuristic function empty-delete-list (solve simplified problem)

Logical Inference

- State answer clause
- Successor function states resulting from substituting one atom with all the clauses of which it is the head
- Goal test empty answer clause
- Solution start state
- Heuristic function ….. *(next time)*

Learning Goals for today's class

You can:

- Model a relatively simple domain with propositional definite clause logic (PDCL)
- Trace query derivation using SLD resolution rule of inference