Logic: Domain Modeling /Proofs + Top-Down Proofs

Computer Science cpsc322, Lecture 22

(Textbook Chpt 5.2)

March, 8, 2010

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** !

Can you think of a proof procedure for PDCL….

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

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

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

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

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?

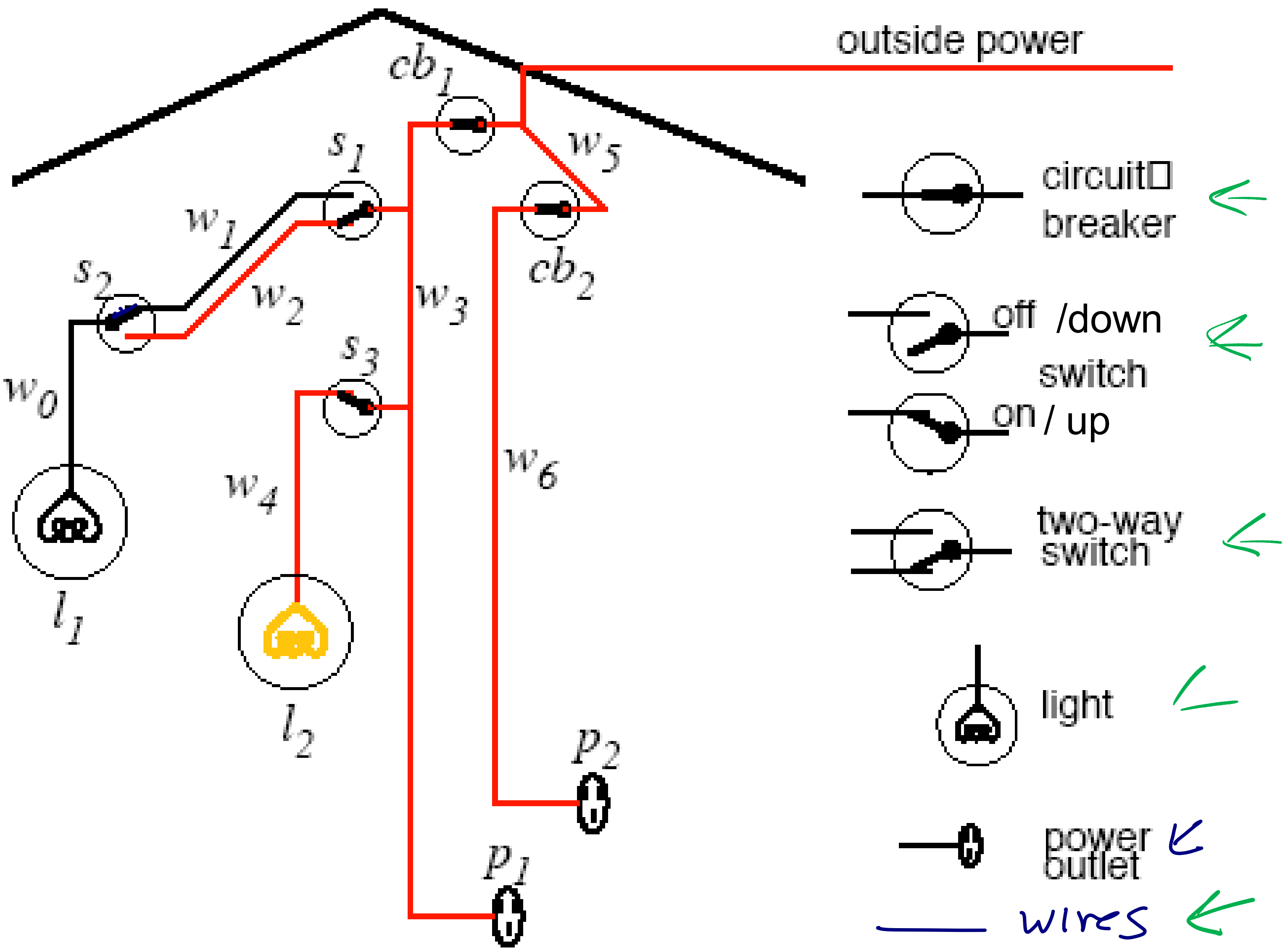
$KB \models G \Rightarrow KB \vdash_B G$

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

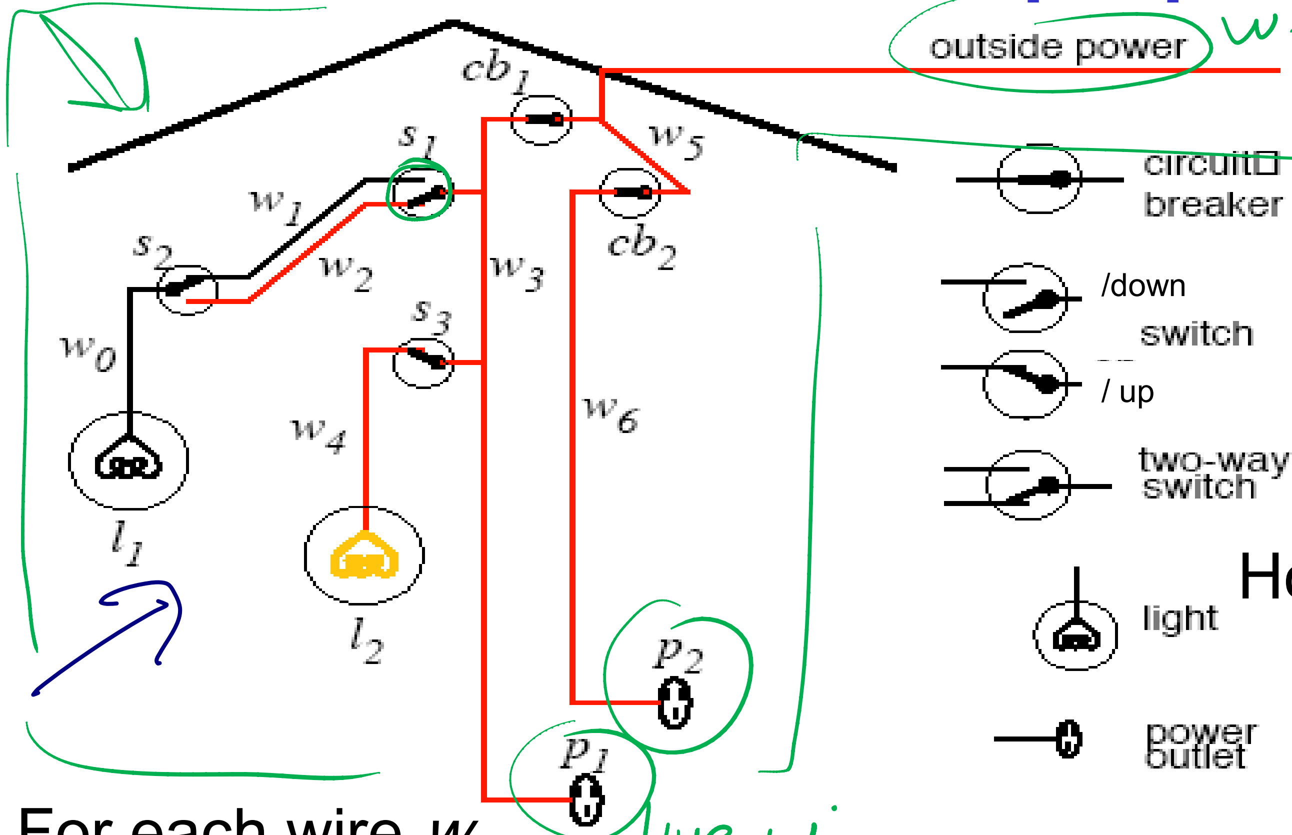
Lecture Overview

- Recap
- 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

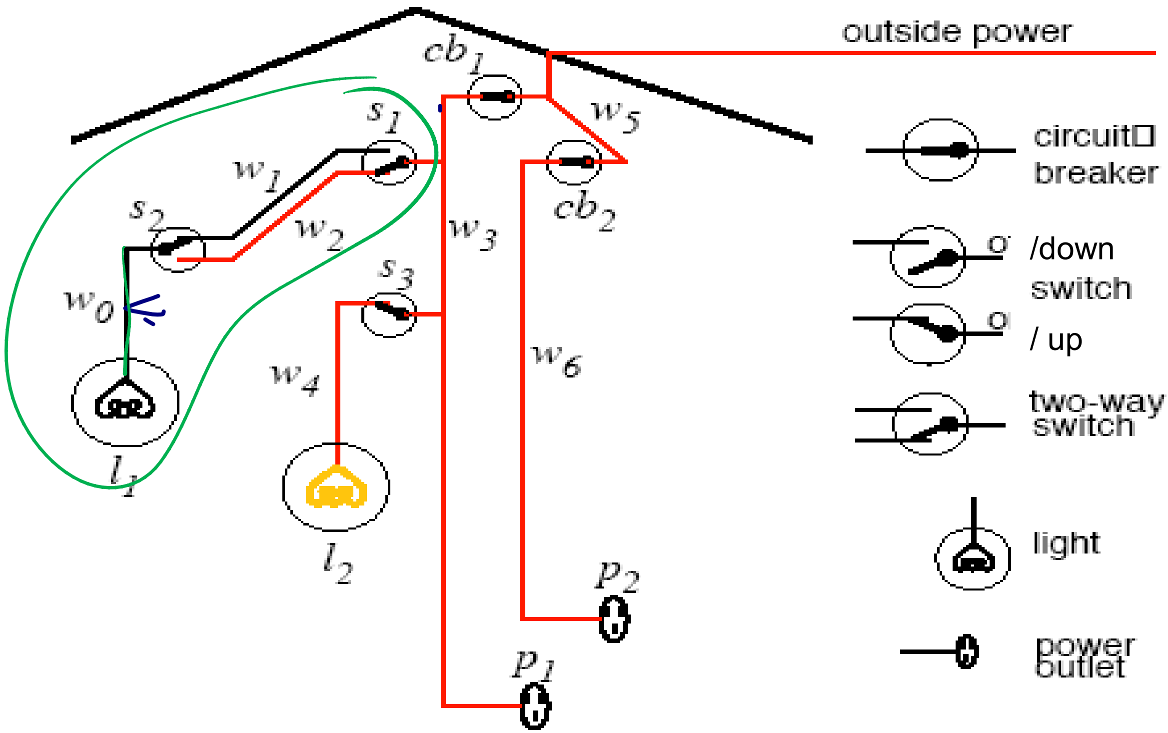


How many interpretations?

- 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



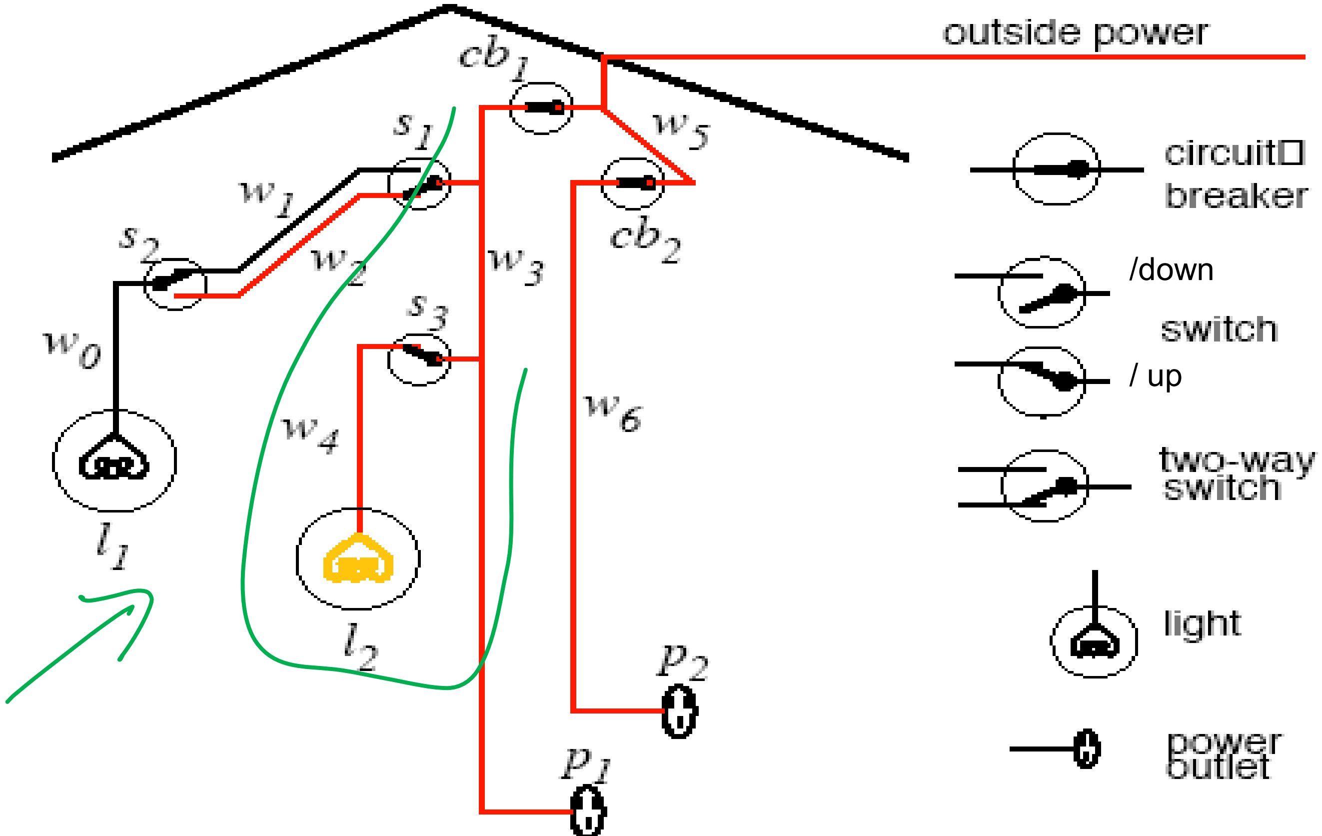
$live\_l_1 \leftarrow$ live_w0

$live\_w_0 \leftarrow$ up_s2 ∧ live_w1

$live\_w_0 \leftarrow$ down_s2 ∧ live_w2

$live\_w_1 \leftarrow$ up_s1 ∧ live_w3

More on how the domain works….



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

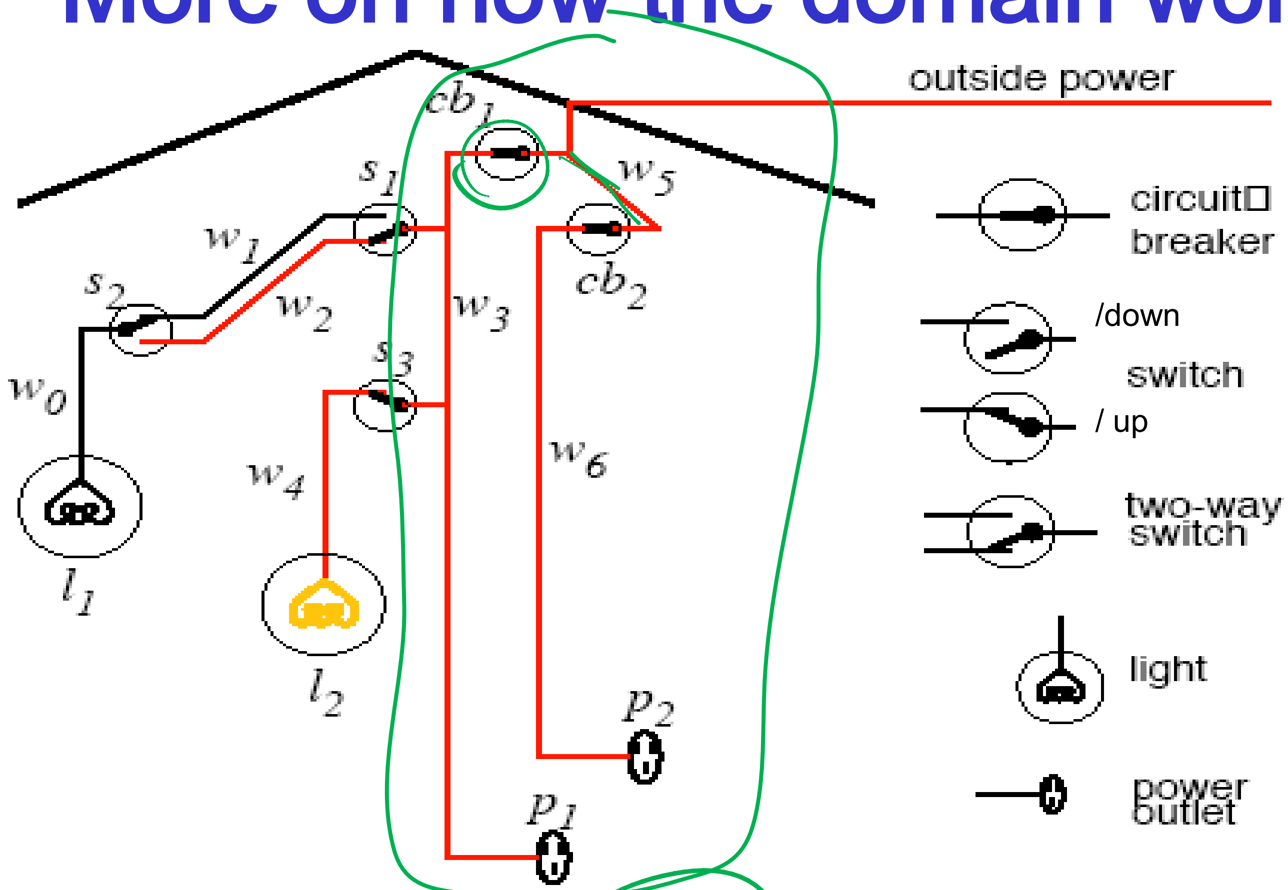
$live\_l_2 \leftarrow live\_w_4.$

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

$live\_p_1 \leftarrow live\_w_3.$

.

More on how the domain works….



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

$live\_p_2 \leftarrow live\_w_6.$

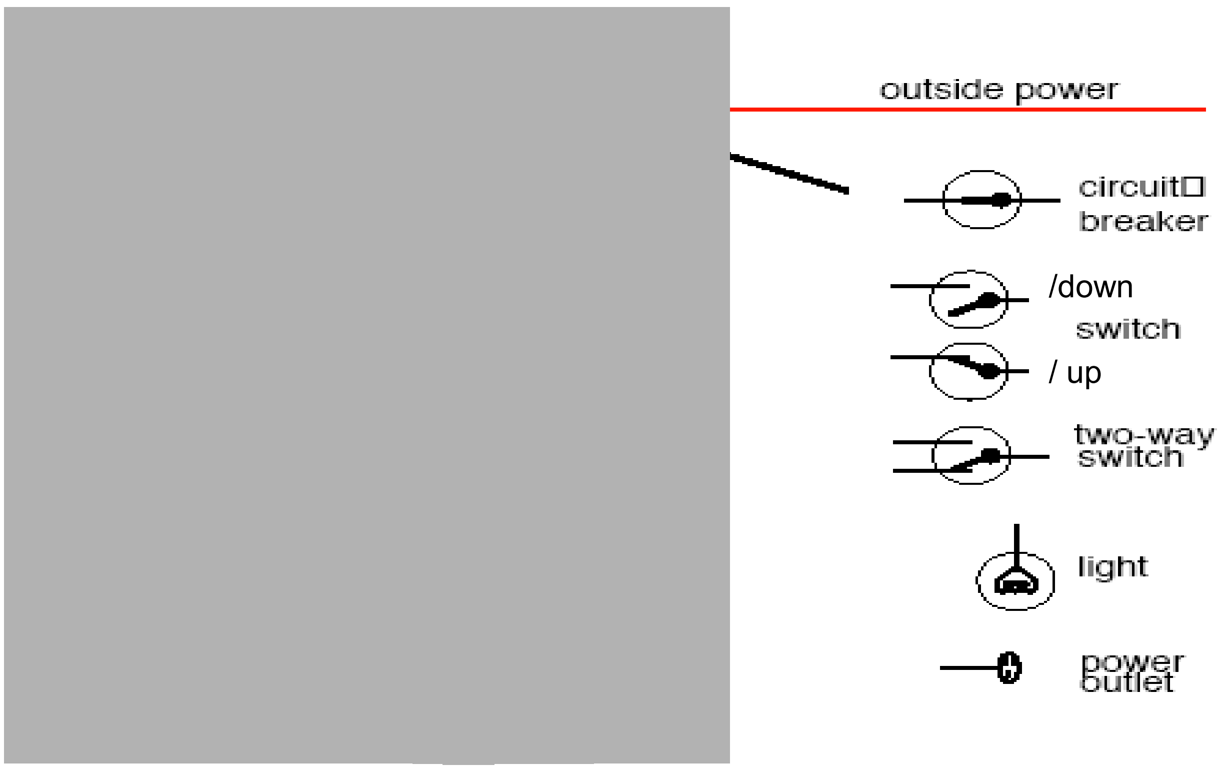
$live\_w_6 \leftarrow live\_w_5 \land ok\_cb_2.$

$live\_w_5 \leftarrow live\_outside.$

What else we may know about this domain?

- That some simple propositions are true

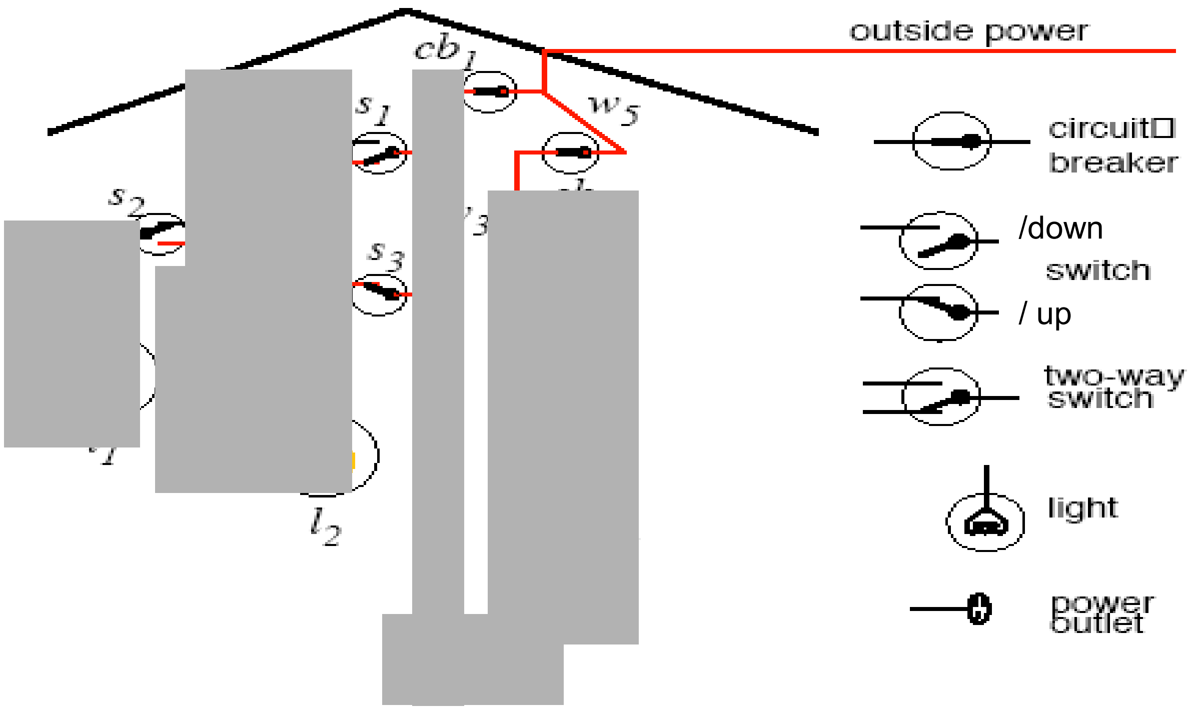
live_outside.



What else we may know about this domain?

- That some additional simple propositions are true

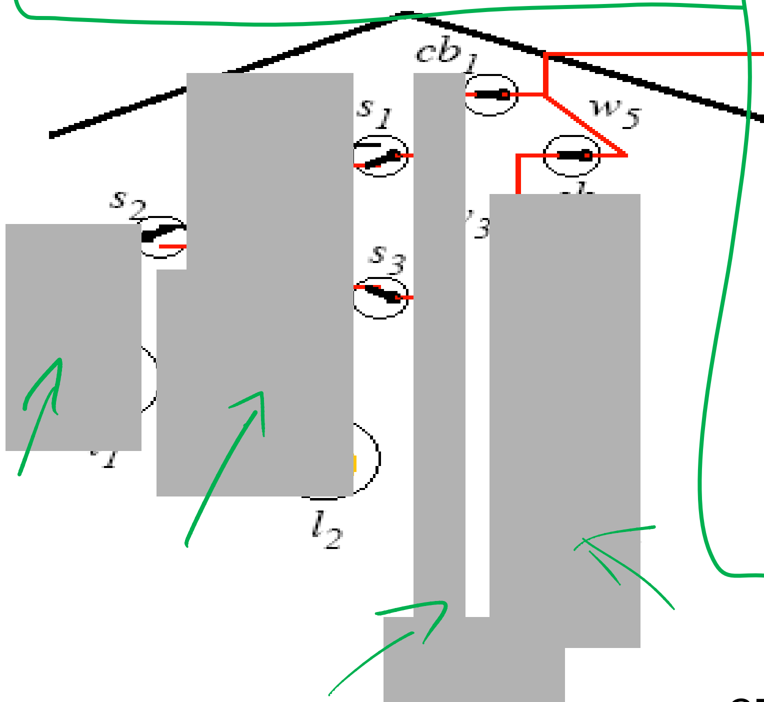
down_s$_1$. *up_s*$_2$. *up_s*$_3$. *ok_cb*$_1$. *ok_cb*$_2$. *live_outside*.



All our knowledge…..

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

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



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 2^{19} __________interpretations
- 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

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

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

C

BU generates C

all the atoms added to C are in green

$live\_l_2$? ✓

$live\_l_1$ ✗

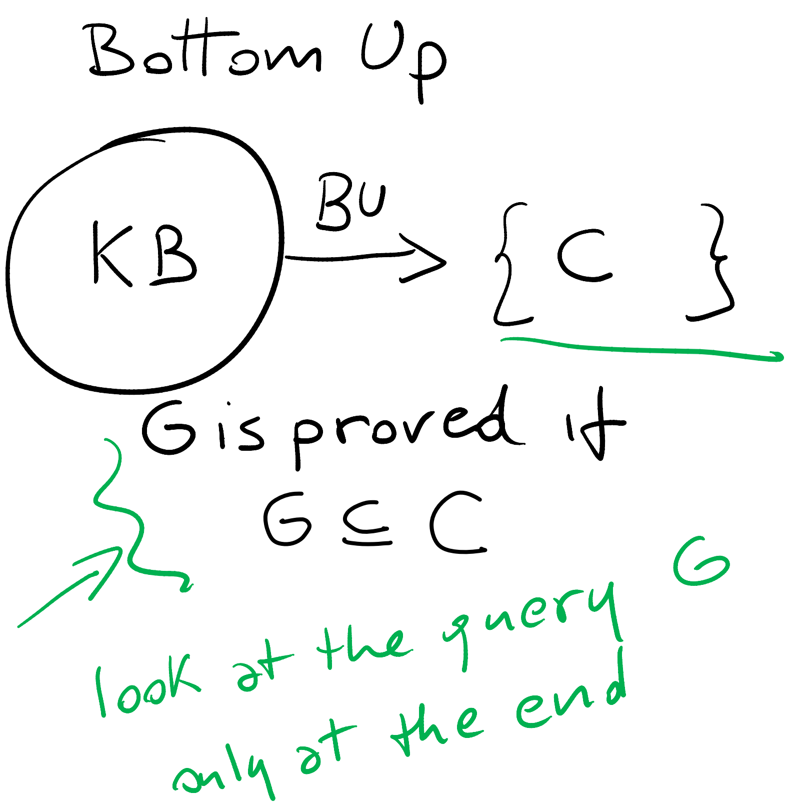
$live\_l_2 \in C \Rightarrow KB \vdash_{BU} live\_l_2 \Rightarrow KB \models live\_l_2$ which is not the case for $live\_l_1$

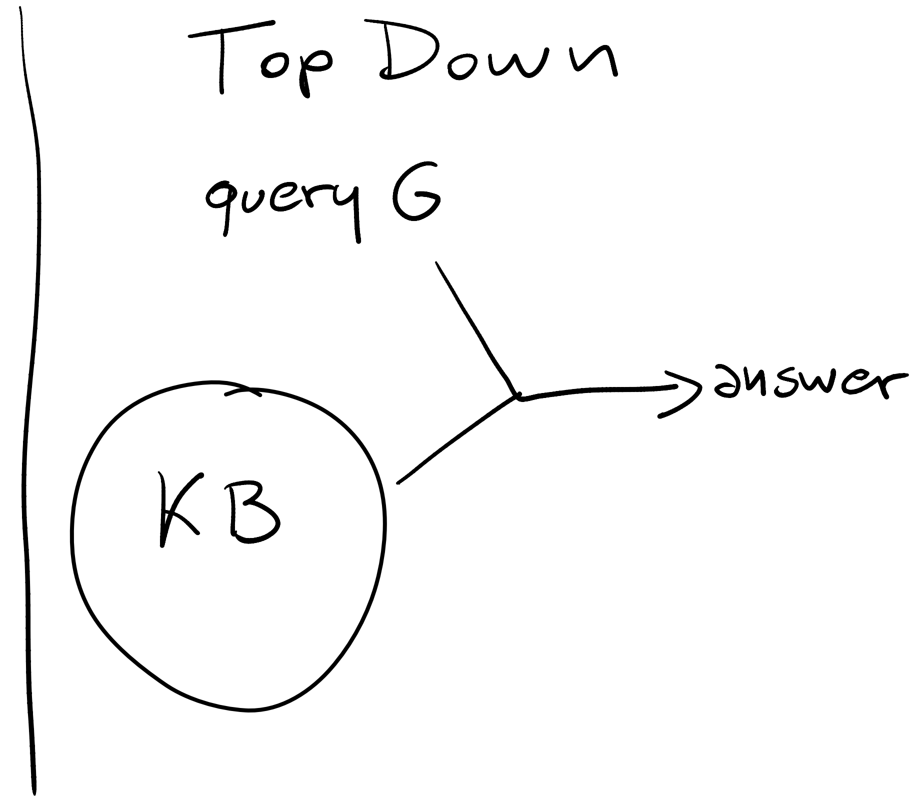
Lecture Overview

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

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:

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

G

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

$yes \leftarrow a_2 \wedge \ldots \wedge a_m$

Rule of inference (called SLD Resolution)

Given an answer clause of the form:

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

KB

and the clause:

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

You can generate the answer clause

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

Rule of inference: Examples

Rule of inference (called SLD Resolution)
Given an answer clause of the form:

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

and the KB clause:

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

You can generate the answer clause

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

KB clause

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

KB

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

(successful) Derivations

- An answer is an answer clause with $m = 0$. That is, it is the answer clause $yes \leftarrow .$
- A (successful) derivation of query "$?q_1 \land \ldots \land q_k$" from KB is a sequence of answer clauses $\gamma_0, \gamma_1, \ldots, \gamma_n$ such that
  - γ_0 is the answer clause $yes \leftarrow q_1 \land \ldots \land q_k$
  - 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…..

$yes \leftarrow a \land b$

Example: derivations

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

KB

Query: a `(two ways)`

$yes \leftarrow a.$

" $\leftarrow b \wedge c$

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

K cannot be eliminated so will Fail

$yes \leftarrow a.$

" $\leftarrow e \wedge f$

" $\leftarrow f$

" $\leftarrow c$

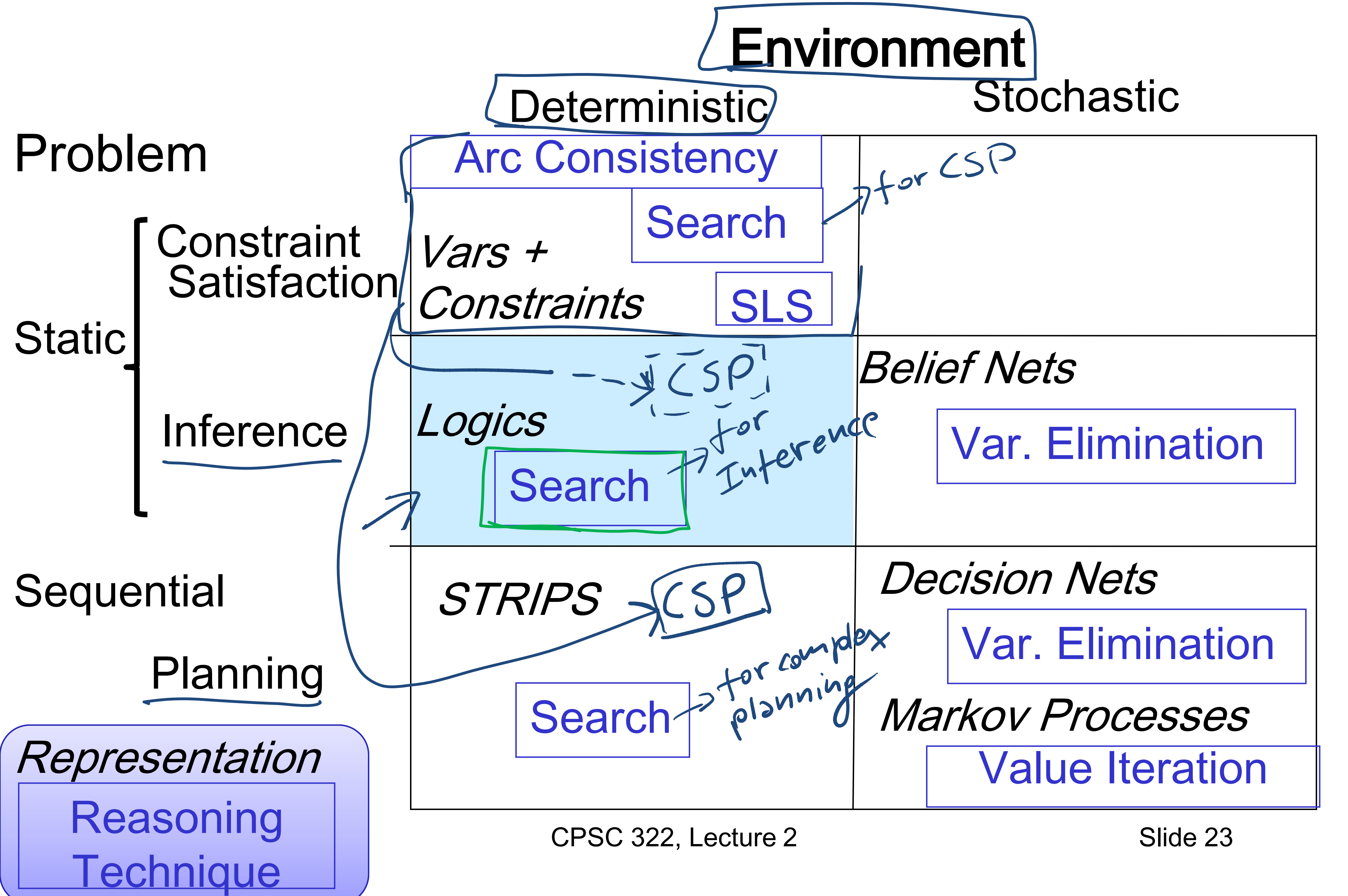
" $\leftarrow e$

$yes \leftarrow$ . . . . .

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

$yes \leftarrow b.$

Course Big Picture



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 number of atoms in given state

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

Midterm: this Wed March 10

Midterm: 6 short questions (8 *pts each*) + 2 problems (*26 pts each)*
+ 5 bonus points

- Study: textbook and **inked** slides
- Work on **all** practice exercises and **revise assignments**!
- While you revise the **learning goals**, work on **review questions (posted)** I may even reuse some verbatim ☺
- I have also posted a **couple of problems** from previous offering (maybe slightly more difficult /inappropriate for you because they were not informed by the learning goals) … but you have the solutions ☺

Midterm: this Wed March 4

Tue 3 office hours
<2 - 4>

SAME ROOM – 1.5 hours

~10 short questions (~6pts each) + 2 problems (~20pts each)

- Study: textbook and **inked** slides
- Work on **all** practice exercises
- Work-on/Study the posted **learning goals**, **review questions** (I may even reuse some verbatim ☺), **two problems** from previous offering (solutions also posted ☺)