Propositional Logic Intro, Syntax

Computer Science cpsc322, Lecture 19

(Textbook Chpt 5.0-5.1)

February, 23, 2009

Lecture Overview

- **Recap Planning**
- Logic Intro
- Propositional Definite Clause Logic: Syntax

Recap Planning

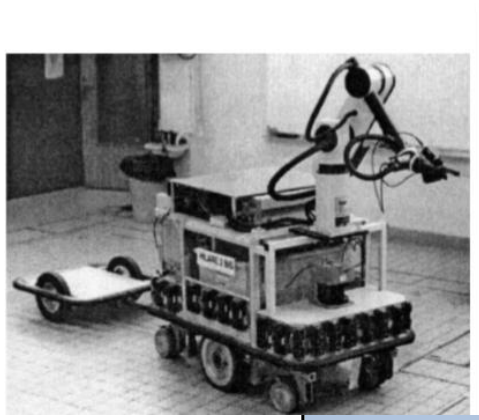
- Represent possible actions with …..STRIPS
- Plan can be found by….. search
- Or can be found by mapping planning problem into…CSP

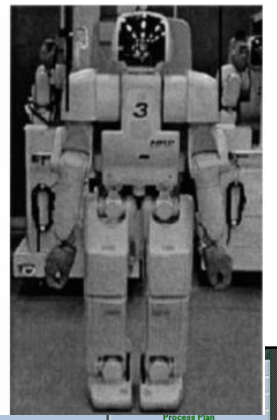
Solve planning as CSP: pseudo code

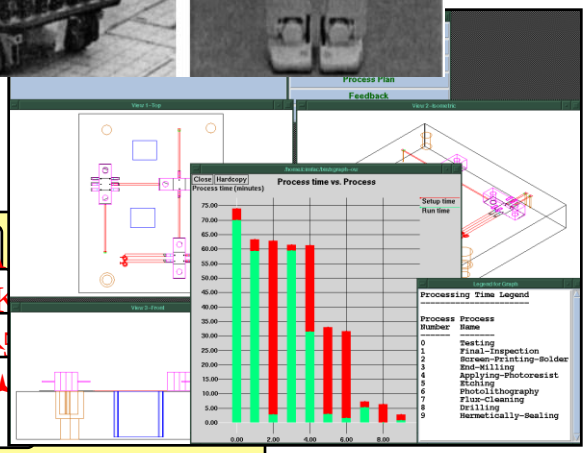
```
horizon = 0
while not solved
    map STRIPS to CSP with horizon
    solve CSP -> solution
    if solution found then
        solved = true
    else
        horizon = horizon + 1
return solution
```

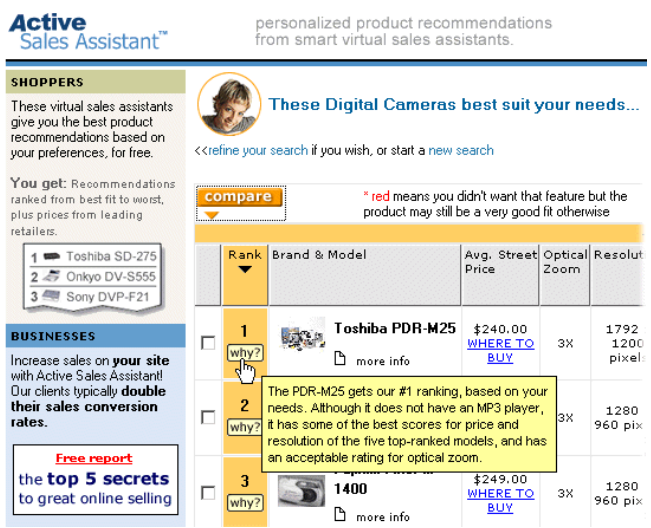
Now, do you know how to implement a planner for….

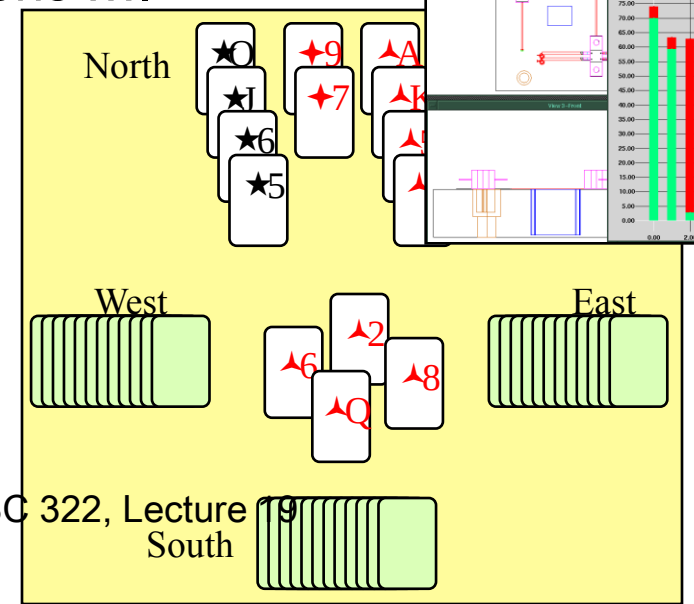
- Emergency Evacuation?
- Robotics?
- Space Exploration?
- Manufacturing Analysis?
- Games (e.g., Bridge)?
- Generating Natural language
  - Product Recommendations ….

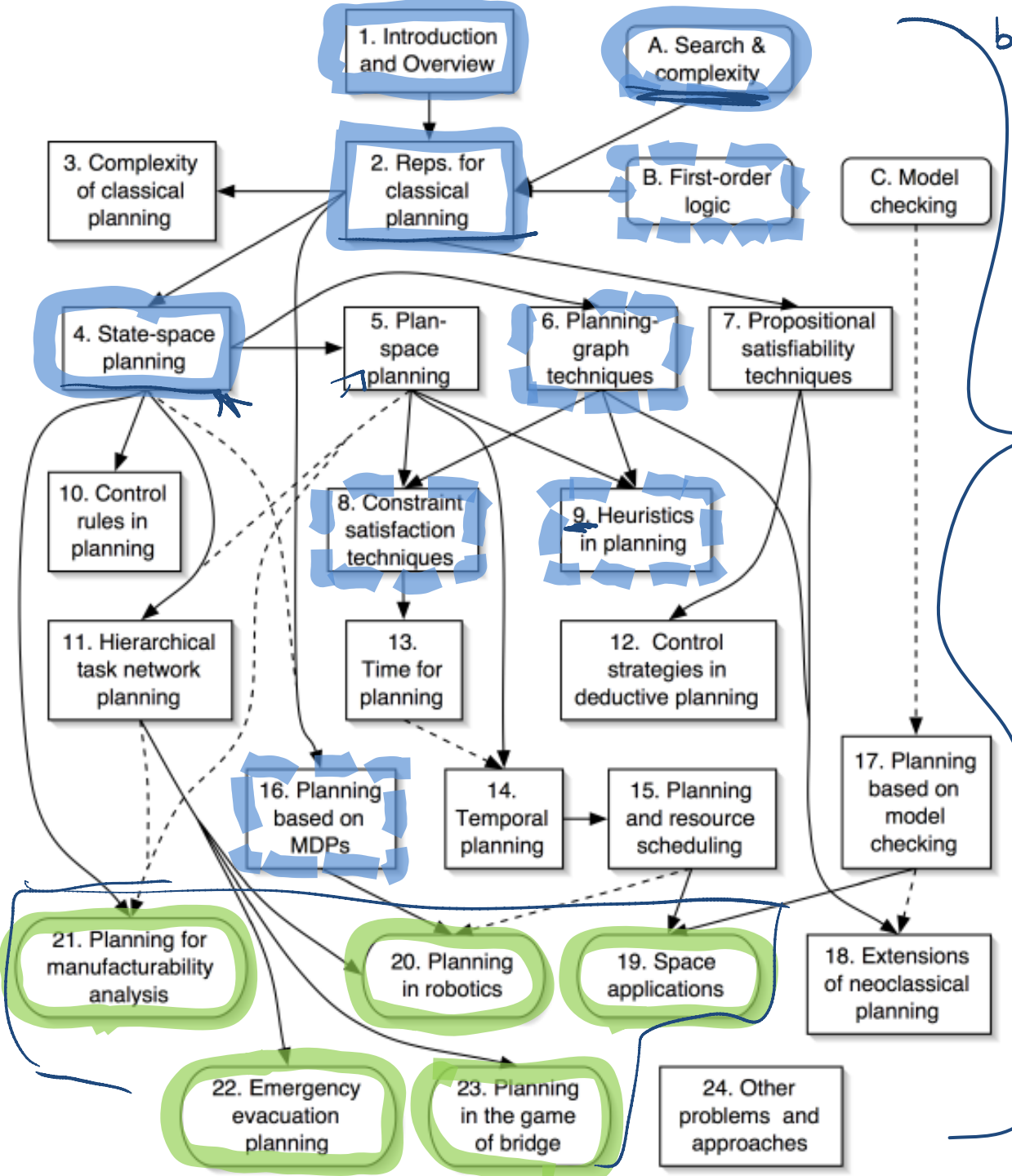












book chapters

No ☹, but you (will) know the key ideas ☺!

- Ghallab, Nau, and Traverso
  Automated Planning: Theory and Practice
  Morgan Kaufmann, May 2004
  ISBN 1-55860-856-7
- Web site:
  - ✓ http://www.laas.fr/planning

you know

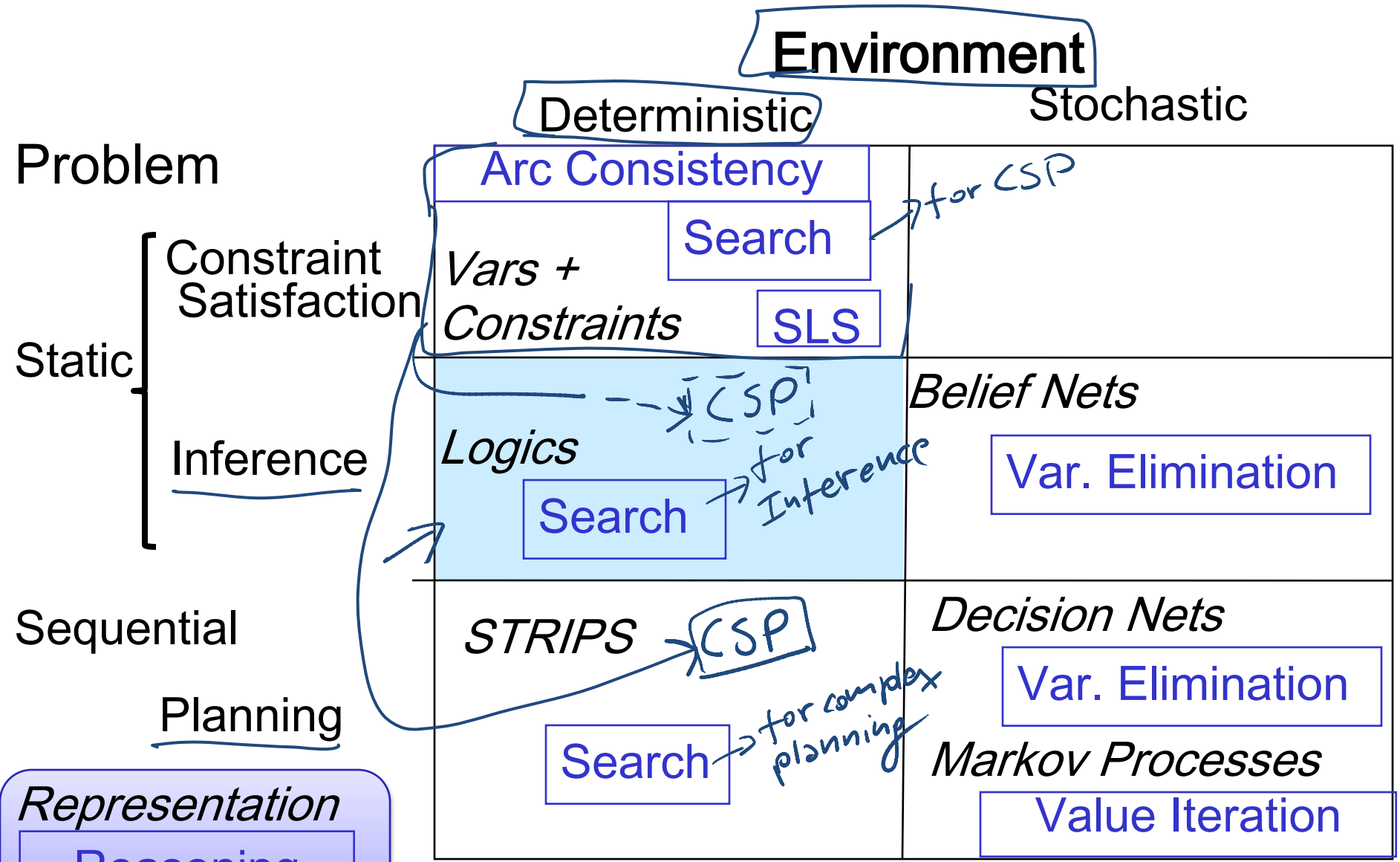
you know a little

Applications

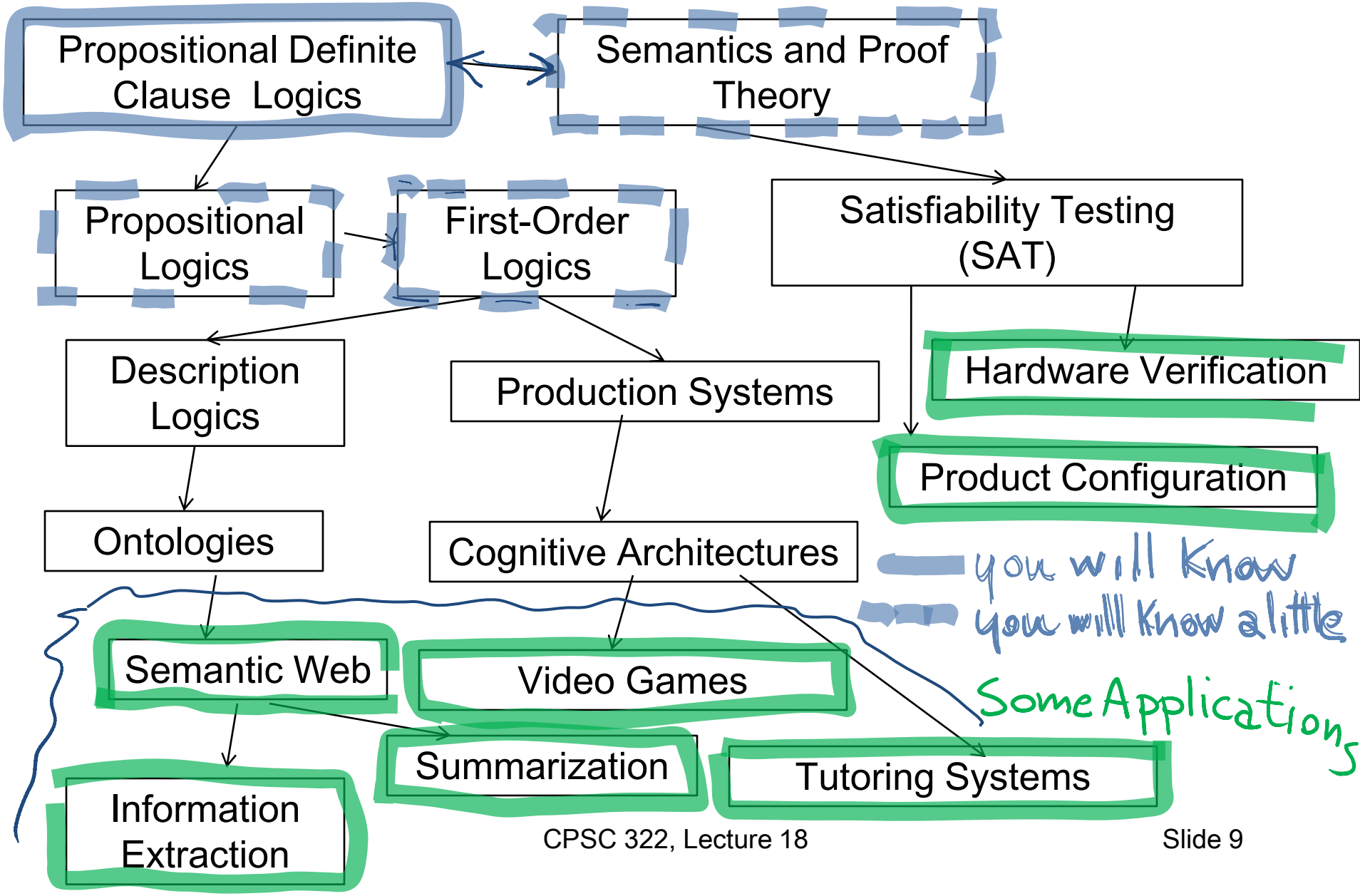
Lecture Overview

- Recap Planning
- **Logic Intro**
- Propositional Definite Clause Logic: Syntax

What is coming next ?



Logics in AI: Similar slide to the one for planning



What you already know about logic...

From programming: Some logical operators

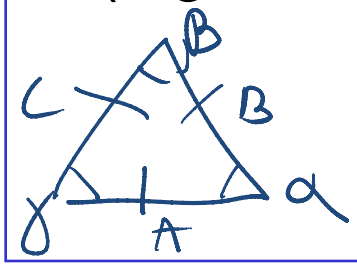
```
If ((amount > 0) && (amount < 1000)) || !(age < 30)
...
```

∧ and

∨ or ¬ not

You know what they mean in a "procedural" way

Logic is the language of Mathematics. To define formal structures (e.g., sets, graphs) and to proof statements about those

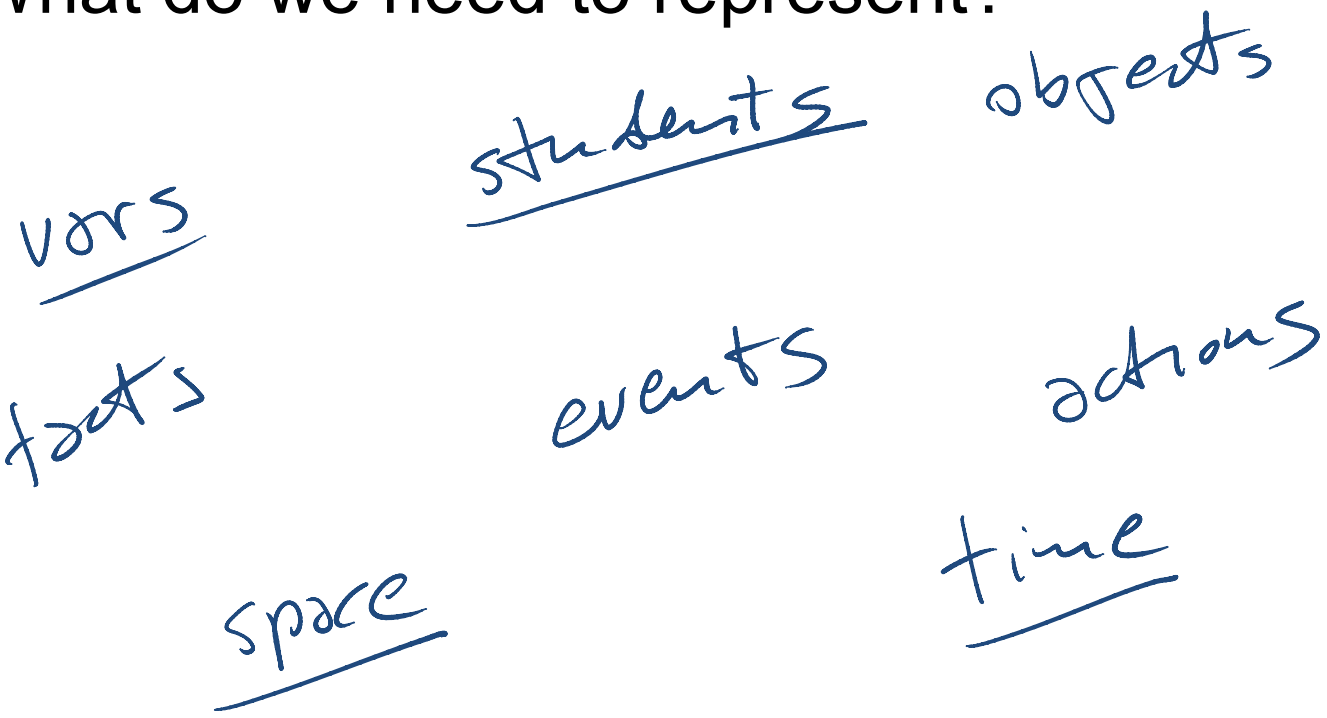


$\forall x\ triangle(x)\ \ A = B = C \iff \alpha = \beta = \gamma$

We are going to look at Logic as a **Representation and Reasoning System** that can be used to **formalize a domain (e.g., an electrical system, an organization)** and to **reason about it**

Logic: A general framework for representation & reasoning

- Let's now think about how to represent an environment about which we have only partial (but certain) information
- What do we need to represent?



Why Logics?

- **"Natural"** to express **knowledge** about the world

(more natural than a "flat" set of variables & constraints)

"Every 322 student will pass the midterm"

Course(c_1)
nameof(c_1, 322)
Midterm(m_1)
course_of(m_2, c_1)

$\forall x$ student(x) $\wedge$
registered(x, c_1)
$\Rightarrow$ pass(x, m_1, t) $\wedge$
$>(t, \text{Now})$

- It is easy to incrementally add knowledge
- It is easy to check and debug knowledge
- Provide language for asking complex queries
- Well understood formal properties

Propositional Logic

We will study the simplest form of Logic: Propositional

- The primitive elements are propositions: Boolean variables that can be {*true, false*}

No quantified vars

- The goal is to illustrate the basic ideas

- This is a starting point for more complex logics (e.g., first-order logic)

second order (quantify on predicates) (e.g. Induction principle)

Modal Logic Temporal Logic

- Boolean nature can be exploited for efficiency

Propositional logic: Complete Language

The **proposition** symbols p_1, p_2 … etc are sentences

- If S is a sentence, $\neg \mathbf{S}$ is a sentence (negation)
- If S_1 and S_2 are sentences, $\mathbf{S_1} \wedge \mathbf{S_2}$ is a sentence (conjunction)
- If S_1 and S_2 are sentences, $\mathbf{S_1} \vee \mathbf{S_2}$ is a sentence (disjunction)
- If S_1 and S_2 are sentences, $\mathbf{S_1} \Rightarrow \mathbf{S_2}$ is a sentence (implication)
- If S_1 and S_2 are sentences, $\mathbf{S_1} \Leftrightarrow \mathbf{S_2}$ is a sentence (biconditional)

sample sentence/formula

$$(p_1 \vee p_2) \Leftrightarrow \neg(p_3 \Leftrightarrow p_4) \vee p_5$$

Propositional Logics in practice

- Agent is told (perceives) some facts about the world
- Agent is told (already knows / learns) how the world works logical formulas
- Agent can answer yes/no questions about whether other facts must be true

Using Logics to make inferences…

1) Begin with a task domain.

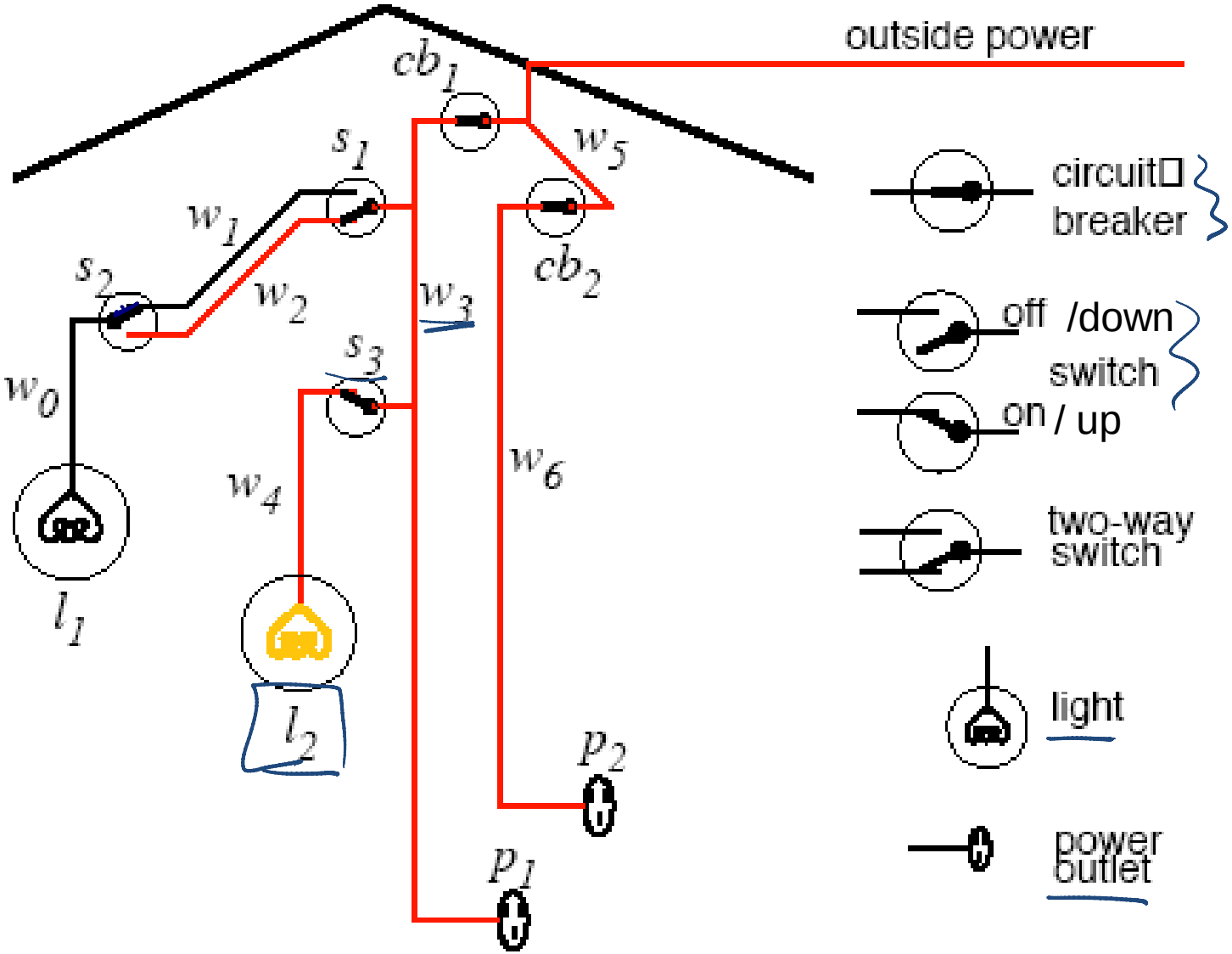
2) Distinguish those things you want to talk about (the ontology). s_2 l_2

3) Choose symbols in the computer to denote propositions on_s_2 on_l_2

4) Tell the system knowledge about the domain. $on\_l_2 \leftarrow on\_s_3 \wedge on\_w_3$

5) Ask the system whether new statements about the domain are true or false. on_l_1 ?

Electrical Environment



Lecture Overview

- Recap Planning
- Logic Intro
- **Propositional Definite Clause Logic: Syntax**

Propositional Definite Clauses

- **Propositional Definite Clauses:** our first logical representation and reasoning system.

(very simple!)

- Only two kinds of statements:
  - that a proposition is true
  - that a proposition is true if one or more other propositions are true

- Why still useful?
  - Adequate in many domains (with some adjustments)
  - Reasoning steps easy to follow by humans
  - Inference linear in size of your set of statements
  - Similar formalisms used in cognitive architectures

Propositional Definite Clauses: Syntax

Definition (atom)

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

Definition (body)

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

$(p_2 \wedge p_2) \wedge (p_3 \wedge p_4)$

Definition (definite clause)

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

Definition (KB)

A **knowledge base** is a set of definite clauses

clause 1
⋮
clause n

PDC Syntax: Examples

a) *ai_is_fun*

b) *ai_is_fun* ∨ *ai_is_boring*

c) *ai_is_fun* ← *learn_useful_techniques*

d) *ai_is_fun* ← *learn_useful_techniques* ∧ *notTooMuch_work*

e) *ai_is_fun* ∧ *relaxing_term* ← *getGoodGrade* ∧ *notTooMuch_work*

f) *ai_is_fun* ← *learn_useful_techniques* ∧ *getGoodGrade* ∧ *notTooMuch_work*

Do any of these statements mean anything?
Syntax doesn't answer this question.

Learning Goals for today's class

You can:

- Verify whether a logical statement belongs to the language of full propositional logics.

$\wedge \quad \vee \quad \neg \quad \rightarrow \quad \leftrightarrow$

- Verify whether a logical statement belongs to the language of propositional definite clauses.

a

b

$c \leftarrow a \wedge b$

$d \leftarrow c \wedge b \wedge f \quad \ldots$

Study for midterm (Wed March 4)

Midterm: ~10 short questions + 2 problems

~20 pts each

~6 pts each

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

Next class

- Definite clauses Semantics and Proofs (textbook 5.2)