(finish Planning)

Propositional Logic Intro, Syntax

Computer Science cpsc322, Lecture 19

(Textbook Chpt 5.1– 5.1.1 – 5.2)

June, 6, 2017
Lecture Overview

- Recap Planning
- Logic Intro

- Propositional Definite Clause Logic: Syntax
Recap Planning

• Represent possible actions with ...  
  \[ \text{STRIPS} \]

• Plan can be found by...  
  \[ \text{search} \]

• Or can be found by mapping planning problem into ...  
  \[ \text{CSP} \]
Solve planning as CSP: pseudo code

```
horizon = 0; solved = false

while not solved
  map STRIPS to CSP with horizon
  solve CSP -> solution

  if solution found then
    solved = true
  else
    horizon = horizon + 1

return solution
```
Planning as CSP

If the algorithm for planning as CSP stops and returns a solution plan of length \( k \), does it mean that there are no shorter solutions?

A. Yes  
B. No  
C. It depends ...
STRIPS to CSP applet

Allows you:

- to specify a planning problem in STRIPS
- to map it into a CSP for a given horizon
- the CSP translation is automatically loaded into the CSP applet where it can be solved

Practice exercise using STRIPS to CSP is available on AI-space
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 ...

---

**Active Sales Assistant**

**SHOPS**

These digital cameras best suit your needs...

- Toshiba SD-21
- Sony DSC-S35
- Sony DVP-F21

**BUSINESSES**

Increase sales on your site with Active Sales Assistant. You can also double your sales conversion rates.

Free expert
tips to great online selling.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Brand &amp; Model</th>
<th>Avg. Street Price</th>
<th>Optimal Zoom</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Toshiba PDR-M25</td>
<td>$249.00</td>
<td>3X</td>
<td>1600 x 960</td>
</tr>
<tr>
<td>2</td>
<td>more info</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>more info</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
No 😞, but you (will) know the key ideas 😊!

Ghallab, Nau, and Traverso

*Automated Planning: Theory and Practice*

Morgan Kaufmann, May 2004


Web site:

- [http://www.laas.fr/planning](http://www.laas.fr/planning)
- [http://projects.laas.fr/planning/](http://projects.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?

Environment

Deterministic

Arc Consistency

Search

Vars + Constraints

CSP

Logics

Search

STRIPS

CSP

Static

Constraint Satisfaction

Query

Problem

Stochastic

SLS

Belief Nets

Var. Elimination

Query

Sequential

Planning

Var. Elimination

Decision Nets

Markov Processes

Var. Elimination

Value Iteration

Reasoning Technique

Representation
Logics

- **Mostly only propositional**… This is the starting point for more complex ones ….

- **Natural** to express **knowledge** about the world
  - What is true (boolean variables)
  - How it works (logical formulas)

- Well understood formal properties

- Boolean nature can be exploited for efficiency

- ……
Logics in AI: Similar slide to the one for planning

Propositional Definite Clause Logics

First-Order Logics

Propositional Logics

Semantics and Proof Theory

Satisfiability Testing (SAT)

Description Logics

Production Systems

Ontologies

Hardware Verification

Cognitive Architectures

Product Configuration

Semantic Web

Video Games

Summarization

Tutoring Systems

Information Extraction

Some Applications

you will know a little bit...
What you already know about logic...

From programming: Some logical operators

If ((amount > 0) \&\& (amount < 1000)) \textbf{OR} !(age < 30)

... \textbf{AND}

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

We are going to look at Logic as a \textbf{Representation and Reasoning System} that can be used to \textbf{formalize a domain} (e.g., an electrical system, an organization) and to \textbf{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?

- objects
- events
- space
- time
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"

  $\text{Midterm}(m_1)$
  $\text{Course}(c_1)$
  $\text{Name-of}(c_1, 322)$
  $\text{Course-of}(m_1, c_1)$

  $(\forall z \ \text{Student}(z) \land \text{Registered}(z, c_1) \implies \text{pass}(m_1, z))$

  $(\land \text{Follows-advice}(z, \text{Slide}23))$

- 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
Complex Query

"Will Sue pass all her midterms?"

\( \forall c, m \ \text{registered}\ (Sue, c) \land \text{course-of}\ (m, c) \land ?\ \text{pass}(m, Sue) \)
Propositional Logic

We will study the simplest form of Logic: Propositional

- The primitive elements are propositions: Boolean variables that can be \{true, false\} \( p_1, p_2 \)
- The goal is to illustrate the basic ideas
- This is a starting point for more complex logics (e.g., first-order logic)
- Boolean nature can be exploited for efficiency.
Propositional logic: Complete Language

The proposition symbols $p_1, p_2 \cdots$ etc are sentences

- If $S$ is a sentence, $\neg S$ is a sentence (negation)
- If $S_1$ and $S_2$ are sentences, $S_1 \land S_2$ is a sentence (conjunction)
- If $S_1$ and $S_2$ are sentences, $S_1 \lor S_2$ is a sentence (disjunction)
- If $S_1$ and $S_2$ are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
- If $S_1$ and $S_2$ are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Sample Formula

$((p_2 \lor p_2) \land p_3) \Leftrightarrow ((p_2 \Rightarrow \neg p_4) \lor p_5)$
Propositional Logics in practice

- Agent is told (perceives) some facts about the world some propositions are true

- 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).

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.
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.  
\( \rho_1 \)

**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.  
\( \rho_2 \land \cdots \land \rho_n \)

**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.'')  
\( \rho_4 \leftarrow \rho_2 \land \cdots \land \rho_5 \)

**Definition (KB)**  
A **knowledge base** is a set of definite clauses.  
\[ \text{clause}_1, \ldots, \text{clause}_n \]
light_l1.  
light_l2.  
ok_l1.  
ok_l2.  
ok_cb1.  
ok_cb2.  
live_outside.

\[\text{definite clauses, KB}\]

\[\text{rules}\]

\[\text{atoms}\]
PDC Syntax: more examples

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 \( b \).’)

a) \( ai\_is\_fun \)

b) \( ai\_is\_fun \lor ai\_is\_boring \)

c) \( ai\_is\_fun \leftarrow learn\_useful\_techniques \)

d) \( ai\_is\_fun \leftarrow learn\_useful\_techniques \land notTooMuch\_work \)

e) \( ai\_is\_fun \leftarrow learn\_useful\_techniques \land \neg TooMuch\_work \)

f) \( ai\_is\_fun \leftarrow f(time\_spent, material\_learned) \)

g) \( srtsyj \leftarrow errt \land gffdgdfgd \)
PDC Syntax: more examples

Legal PDC clause  Not a legal PDC clause

a) ai_is_fun

b) ai_is_fun \lor \text{ai_is_boring}

c) ai_is_fun \leftarrow \text{learn_useful_techniques}

d) ai_is_fun \leftarrow \text{learn_useful_techniques} \land \text{notTooMuch_work}

e) ai_is_fun \leftarrow \text{learn_useful_techniques} \land \neg \text{TooMuch_work}

f) ai_is_fun \leftarrow f(\text{time_spent, material_learned})

g) srtsyj \leftarrow errt \land gffdgdgd

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.

• Verify whether a logical statement belongs to the language of propositional definite clauses.
Study for midterm (Thurs June 8)

Midterm: ~6 short questions (8pts each) + 2 problems (26pts each)

- Study: textbook and inked slides
- Work on all practice exercises and revise assignments!
- While you revise the learning goals, work on review questions — I may even reuse some verbatim 😊
- Work on the couple of problems from previous offering (maybe slightly more difficult) ⋯ but you have the solutions 😊
Next class

- Definite clauses Semantics and Proofs (textbook 5.1.2, 5.2.2)