Lecture 19

CS 6380 Artificial Intelligence

Advanced Topics in AI Representation and Modeling

Department of Computer Science and Engineering

Indian Institute of Technology Madras (IITM), India



This material is based on lecture slides for CSE 571: Artificial Intelligence by Siddharth Srivastava, ASU.

Course Logistics

Logistics

- HW3-Quiz moved to September 22, Tuesday
- HW3 Solutions to be released today
- (Maybe) No/online class on Thursday
- User Study on Robotics Representations
  - Does not have any effect on your grades.
  - Completely independent of the course
  - Participation is voluntary
  - Rs. 100 compensation for about 30-45 min of your time



Back to AI

The Story So Far

- In the last few lectures we saw our first AI problems, and how to solve them
- Discussion was restricted to "toy" examples
- Real world problems are a different beast
  - Just expressing the problem can be a challenging exercise!
- Today's lecture: advanced topics on expressing real world problems
  - And some wisdom of the ages

Logistics: Search Problem on Steroids

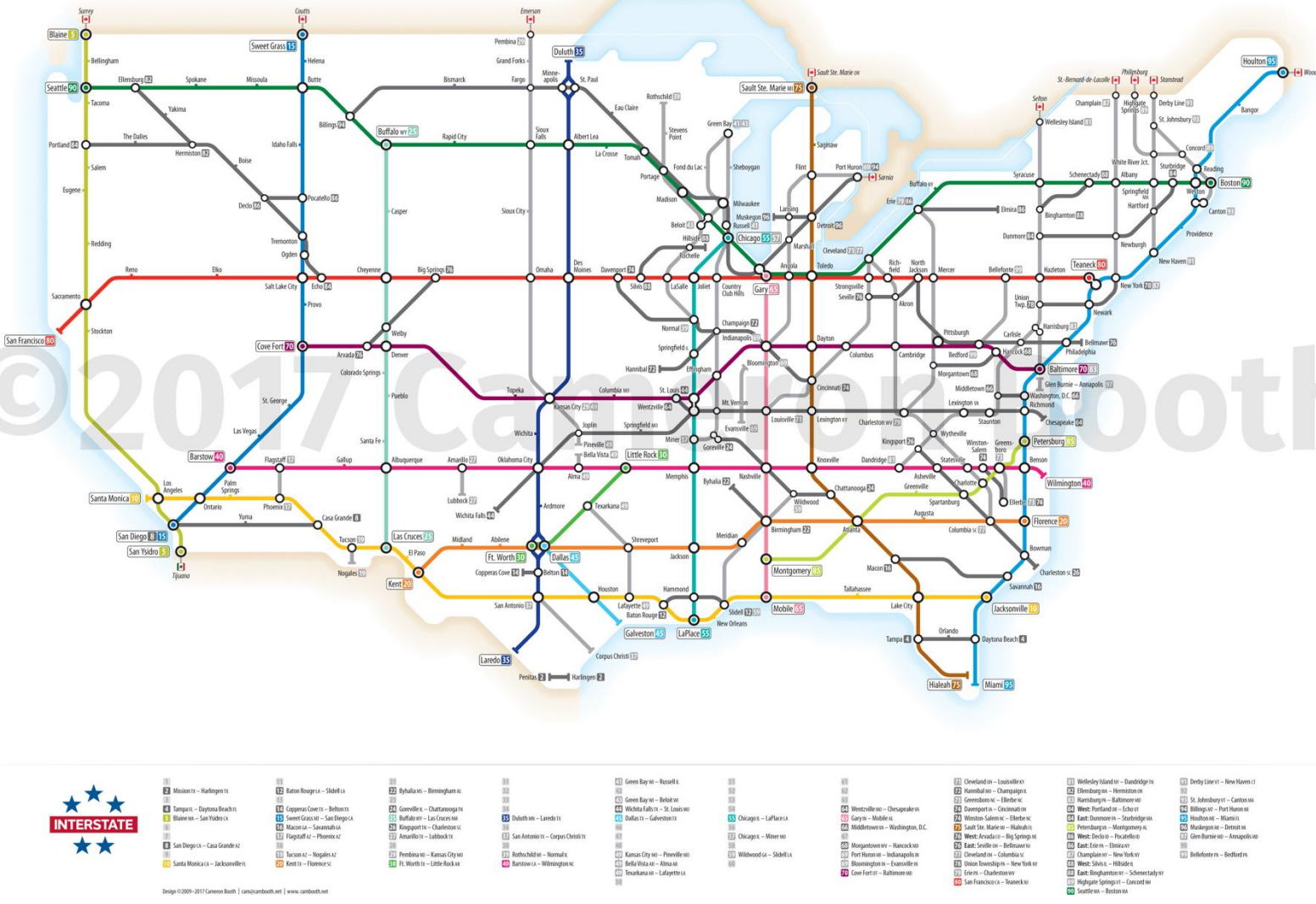
- Vertices
- Edges (directed?)
- Delivery service
  - Trucks, drivers, packages
- Actions:



https://www.cambooth.net/interstates-as-subway-diagram-2011-version/interstate_subway-larg

Logistics: Search Problem on Steroids

- Vertices
- Edges (directed?)
- Delivery service
  - Trucks, drivers, packages
- Actions:
  - Get a driver to drive a truck to a location
  - Load a package on to a truck
  - Unload a package from a truck
  - Refuel a truck
- Considerations
  - Packages may require refrigerated trucks
  - Trucks may require certain types of roads
  - Drivers need to sleep
  - …



https://www.cambooth.net/interstates-as-subway-diagram-2011-version/interstate_subway-larg

How Should we Express this Problem?

- Option 1: Naïve representation: draw the explicit state space graph
  - Each state is a configuration of trucks, packages, drivers along the map
  - n cities, p packages, t trucks, d drivers
  - $> n^{p+t+d}$ vertices
    - Small problem: 100 cities, 1000 packages: $> 100^{1000}$ vertices
    - Number of atoms in the universe $\sim 10^{82}$

How Should we Express this Problem?

- Option 1: Naïve representation: draw the explicit state space graph
  - Each state is a configuration of trucks, packages, drivers along the map
  - n cities, p packages, t trucks, d drivers
  - $> n^{p+t+d}$ vertices
    - Small problem: 100 cities, 1000 packages: $> 100^{1000}$ vertices
    - Number of atoms in the universe $\sim 10^{82}$
  - But that is not all… Actions correspond to edges in the graph
    - Need to work out edges corresponding to each action instance (load p1 onto t1 when t1 is in PHX)
    - Amount of fuel used needs to be calculated separately for each edge

How Should we Express this Problem?

- Option 1: Naïve representation: draw the explicit state space graph
  - Each state is a configuration of trucks, packages, drivers along the map
  - n cities, p packages, t trucks, d drivers
  - $> n^{p+t+d}$ vertices
    - Small problem: 100 cities, 1000 packages: $> 100^{1000}$ vertices
    - Number of atoms in the universe $\sim 10^{82}$
  - But that is not all… Actions correspond to edges in the graph
    - Need to work out edges corresponding to each action instance (load p1 onto t1 when t1 is in PHX)
    - Amount of fuel used needs to be calculated separately for each edge
- Problem: creating or maintaining such a representation
    - Some packages require refrigerated trucks (any of $\{t_1, t_{12}, t_{1232}, \dots\}$)
    - Iteratively prune each edge that violates this
    - Changes in fuel efficiency → redo the fuel usage for each edge

A Step Up: Factored Representation with a Successor Function

- States are factored into variables
- $\langle \mathrm{loc}_{\mathrm{p1}}, \mathrm{loc}_{\mathrm{p2}}, \ldots, \mathrm{loc}_{\mathrm{t1}}, \ldots \mathrm{loc}_{\mathrm{d2}}, \ldots, \mathrm{fuel}_{\mathrm{t1}}, \mathrm{fuel}_{\mathrm{t2}}, \ldots \rangle$
- Successor function can "focus" on only the relevant factors
  - Action "drive truck t1 from Phoenix to Ontario"
  - This action changes loc_{t1} : loc_{t1}=Ontario
  - Regardless of where the other trucks, drivers, packages are
  - One variable assignment equivalent to specifying edges from $O(ptd)$ vertices (states)



https://www.cambooth.net/interstates-as-subway-diagram-2011-version/interstate_subway-large/

Problems with Factored Representation

- Everything's fine until a new package, truck or driver shows up
- REDO
  - State vector representation
  - Successor functions for load, unload
  - (But the actions do the same thing they did to every other package!!)

$\langle \text{loc}_{p1}, \text{loc}_{p2}, \ldots, \text{loc}_{t1}, \ldots \text{loc}_{d2}, \ldots, \text{fuel}_{t1}, \text{fuel}_{t2}, \ldots \rangle$



https://www.cambooth.net/interstates-as-subway-diagram-2011-version/interstate_subway-large/

Is there A Better Way?

- Can we avoid spelling out each rule for each tuple of objects?

- Can we express knowledge in terms of properties of objects and their relationships, rather than their “names”?

Relational Representations

- Main problem in prior attempts: successor function, state expressed in terms of object *names*
  - But in general, effects of actions depend on object properties
- E.g., the action "load package p_1 onto truck t_1" can be executed if
  - Both are at the same location
  - $t_1$has space
  - These conditions are not specific to p_1 or t_1



Relational Representations

- Main problem in prior attempts: successor function, state expressed in terms of object *names*
  - But in general, effects of actions depend on object properties
- E.g., the action "load package p_1 onto truck t_1" can be executed if
  - Both are at the same location
  - $t_1$has space
  - These conditions are not specific to p_1 or t_1
- Can this "symmetry" in the problem be exploited?
  - How would you code up a simulator for this problem?



Relational Representations

- A state is defined using properties and relationships among objects
- Given a *domain*, fix a set of relations R and a set of functions F as its *vocabulary*
  - Suppose you have sets A and B
  - What is a relation between A and B?
  - What is a function from A to B?

$$loc(t_1) = Phoenix$$
$$loc(t_2) = Ontario$$



Relational Representations

- A state is defined using properties and relationships among objects
- Given a *domain*, fix a set of relations R and a set of functions F as its *vocabulary*
- Express the successor functions for each action in terms of these functions and relations
- Logistics domain:

$$R = \{In(Package, Truck), On(Package, Package)\}$$
$$F = \{Fuel(Truck), DriverOf(Truck), Location(Truck)\}$$

$$loc(t_1) = Phoenix$$
$$loc(t_2) = Ontario$$



Successor Functions for Relational Representations

$$R = \{In(Package, Truck), At(Package, Location)\}$$
$$F = \{Fuel(Truck), DriverOf(Truck), Location(Truck)\}$$

- Relations and functions specify properties that can be used while writing successor functions or constraints
- E.g., the effect of "load package p onto truck t" results in the following changes:
  - If p and t are at the same location,
    - $In(p, t)$ becomes True;
    - $At(p, l)$ becomes False for all l
  - Regardless of
    - object names
    - values of other properties and relationships between other objects

We used variables, quantification

A Comparison of the Three Representational Choices

$\text{At_}t_1 Phoenix_At_t_2 Ontario_\ldots$

Atomic Representation

- A state is a black-box
- String representations are used to alleviate the situation

$loc_{t_1} = Phoenix$
$loc_{t_2} = Ontario$

Factored Representation

- A state is a valuation of a fixed set of "state variables"
- Successor functions for each action can depend on a small set of variables
- Problem: dependent on object names, numbers

$loc(t_1) = Phoenix$
$loc(t_2) = Ontario$

Relational Representation

- A state is defined using properties and relationships among objects
- Specifying successor functions *seems to be easier* through the use of variables, quantifiers

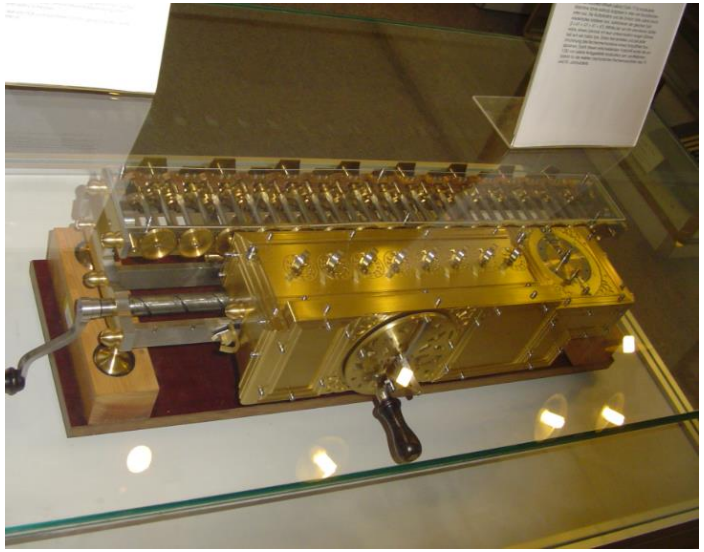
Underlying problem: building a formal system of reasoning with symbolic representations

A Centuries-Old Quest: Formalizing Knowledge and Reasoning

- Aristotle's formalization: propositional logic (384BC)
  - Formalize sentences as terms or propositions
- Leibniz developed a mechanical calculator (1672)







Leibniz's Pitch

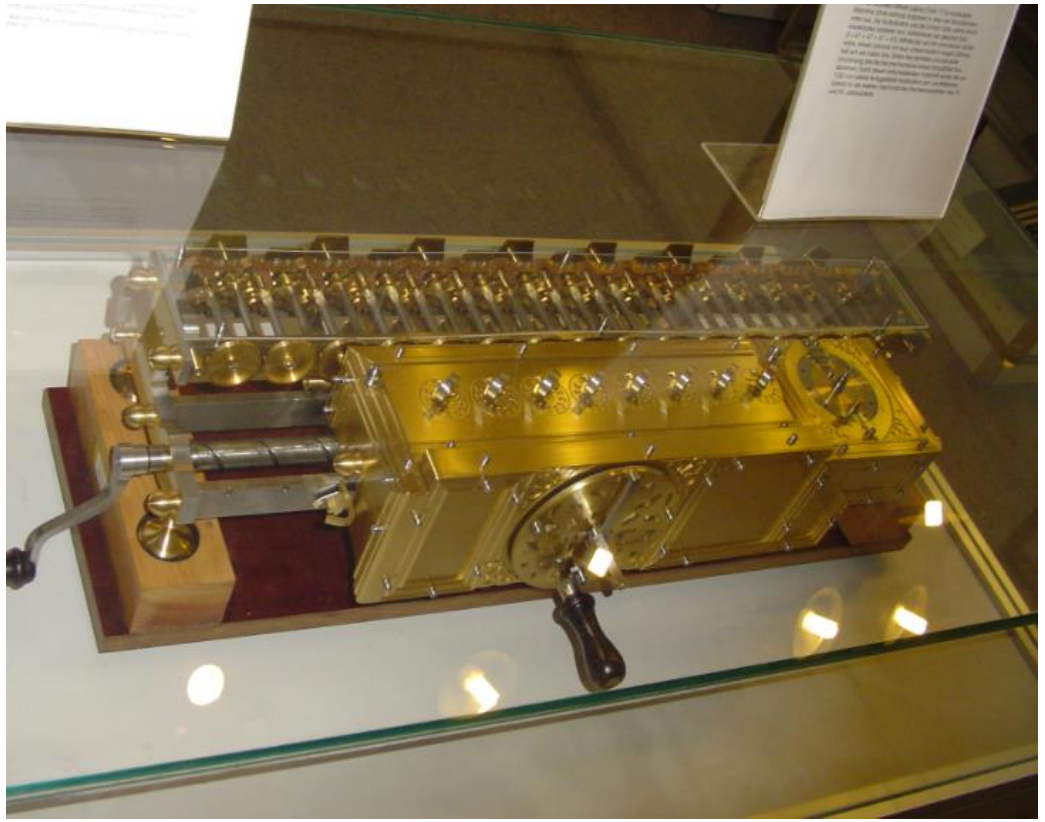
It could add, subtract, multiply, divide (max: 16 digits)

Imagine a time when no one thought a calculator would be useful.

How would you advertise it?

Leibniz's Sales pitch:

"....it will pay to extend as far as possible the major Pythagorean tables..."



AI Pioneers

- Leibniz's Big Idea:
  - Can we build a machine for reasoning: draw inferences mechanically?
  - Essence of AI!
- George Boole showed how propositional logic could be formalized as an algebra (1847)
  - Express concepts as terms (Aristotle)
  - Conjunction, disjunction as operators
  - Formalized logic as a form of mathematics

    "Laws of Thought"



Propositional Logic: Syntax

- Syntax: How to state something
- Semantics: What a statement means

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ 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)

Propositional Logic

- Syntax defines how to make well-structured statements

P_1 = "It rains a lot in Tempe"
P_2 ="The university in Tempe has more than 100 AI students"

- P_1 and P_2 are propositions
- Are they true?
- How can we determine truth?

Propositional Logic

- Syntax: how to make well-structured statements
- Semantics: determines which statements are true

P_1 = "It rains a lot in Tempe"
P_2 ="The university in Tempe has more than 100 AI students"

- P_1 and P_2 are propositions
- Are they true?
- How can we determine truth?

Dictionary Definitions

"Fact"

OED:
a thing that is known to be true, especially when it can be proved

M-W:
a piece of information presented as having objective reality

"Proposition"

Something offered for consideration or acceptance

Propositional Logic: Semantics

P_1= "It rains a lot in Tempe"

P_2="The university in Tempe has more than 100 AI students"

Models are like possible worlds/interpretations

- Model M_1 (notation: $[\varphi]_M$ denotes the truth value of φ under M)
  - $[P_1]_{M_1} = False$
  - $[P_2]_{M_1} = True$

- A model assigns a truth value to every proposition

Propositional Logic: Semantics

P_1 = "It rains a lot in Tempe"

P_2 = "The university in Tempe has more than 100 AI students"

Models are like possible worlds/interpretations

- Model M_1
  - $[P_1]_{M_1} = False$
  - $[P_2]_{M_1} = True$
- Another model: M_2
  - $[P_1]_{M_2} = True$
  - $[P_2]_{M_2} = False$



The vale of Tempe (note: no university)

- A model assigns a truth value to every proposition

Propositional Logic: Semantics

- Semantics of propositional logic tell us how to evaluate the truth value of a sentence under a given model

$\neg S$	is true iff	S	is false			
$S_1 \wedge S_2$	is true iff	S_1	is true **and**	S_2	is true	
$S_1 \vee S_2$	is true iff	S_1	is true **or**	S_2	is true	
$S_1 \Rightarrow S_2$	is true iff	S_1	is false **or**	S_2	is true	
i.e.,	is false iff	S_1	is true **and**	S_2	is false	
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true **and**	$S_2 \Rightarrow S_1$	is true	

P_1 = "It rains a lot in Tempe"

P_2 = "The university in Tempe has >100 AI students"

- Model M_1:
  - $[P_1]_{M_1} = False$
  - $[P_2]_{M_1} = True$
- $[\, P_1 \Longrightarrow P_2 \,]_{M_1} = ?$
- $[\, P_1 \Longrightarrow \neg P_2 \,]_{M_1} = ?$

Propositional Logic: Semantics

- Given a model, use a recursive process to evaluate the truth value of a sentence

P_1 = "It rains a lot in Tempe"
P_2 = "The university in Tempe has >100 AI students"

$\neg S$	is true iff	S	is false				
$S_1 \wedge S_2$	is true iff	S_1	is true **and**	S_2	is true		
$S_1 \vee S_2$	is true iff	S_1	is true **or**	S_2	is true		
$S_1 \Rightarrow S_2$	is true iff	S_1	is false **or**	S_2	is true		
i.e.,	is false iff	S_1	is true **and**	S_2	is false		
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true **and**	$S_2 \Rightarrow S_1$	is true		

- A proposition is a fact (aka "valid" in logic) iff it is true in every possible model

Validity in Propositional Logic

- In mathematical logic, validity supersedes truth
- A sentence is valid if it is true in all possible models
  - You are familiar with truth tables: every row in a truth table is a model

- Scientific notions of truth try to achieve this by conducting experiments
  - Like testing under different models, but sometimes not possible to enumerate all possibilities

Model Theoretic "Truth"

- Empirical truth relies on sampling, or conducting multiple observations to determine truth of propositions
- But science goes beyond tabulating observations
  - Main role of science: predictions!
    - In AI: what would happen if self-driving car changed lanes now?
  - Empirical determination of truth is not sufficient for making predictions
  - Need to build a theory: inductive reasoning
  - Then use the theory to make predictions: deductive reasoning
  - Science = predictive, empirically verifiable theory

Theories in Mathematical Logic

- Mathematical logic also allows the expression of theories, and automation of reasoning – the core of what an AI agent needs to do

- Theory = collection of logical sentences (a conjunction of formulas)
- E.g., P_1: "AI is a hot topic"; P_2: "A university in town is good in AI"

"P_3: town has hot summers"

Theory:

$$T = \{P_1 \wedge P_3 \rightarrow P_2\}$$

Theories in Mathematical Logic

- E.g., P_1: "AI is a hot topic"; P_2: "A university in town is good in AI"

"P_3: town has hot summers"

Theory:

$$T = \{P_1 \wedge P_3 \rightarrow P_2\}$$

- A theory T is said to be consistent with a model M iff $M \models T$
- A theory is said to be logically consistent iff it doesn't imply a contradiction
  - A contradiction is any statement of the form $P \wedge \neg P$

Theories in Mathematical Logic

$$T = \{P_1 \wedge P_3 \rightarrow P_2\}$$

- A theory is said to be consistent with a model M iff $M \vDash T$
- A theory is said to be logically consistent iff it doesn't imply a contradiction
  - A contradiction is any statement of the form $P \wedge \neg P$

$$T' = \{P_1 \wedge P_3 \rightarrow P_2, \qquad P_1 \rightarrow \neg\, P_2\}$$

Is T' logically consistent?

Theories in Mathematical Logic

- E.g., P_1: "AI is a hot topic"; P_2: "A university in town is good in AI"

"P_3: town has hot summers"

Theory:

$$T = \{P_1 \wedge P_2 \rightarrow P_3\}$$

Is T consistent with these models?

- $M_{Tempe,AZ} : P_1 \leftarrow True; P_2 \leftarrow True; P_3 \leftarrow True$
- $M_{Tempe,Greece}: P_1 \leftarrow True; P_2 \leftarrow False; P_3 \leftarrow False$
- $M_{Toronto}$:P_1: $\leftarrow True; P_2 \leftarrow True; P_3 \leftarrow False$

Notation:
If $[\varphi]_M = true$, we say M "models" φ
"$M \vDash \varphi$"

A theory T is said to be consistent with a model M iff $M \vDash T$

We now know how to create representations...
How do we use them?