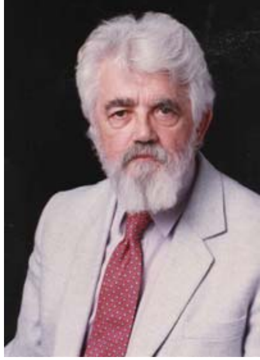
Artificial Intelligence

(Informal Introduction)

Q&A

Father of AI:
John McCarthy
(September 4, 1927 – October 23, 2011)



Q. What is artificial intelligence?

A. Science and engineering of making intelligent machines, especially intelligent computer programs.

Q. Yes, but what is intelligence?

A. Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

Q. Isn't there a solid definition of intelligence that doesn't depend on relating it to human intelligence?

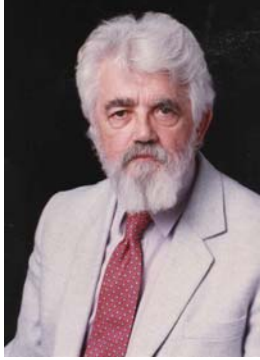
A. Not yet. The problem is that we cannot yet characterize in general what kinds of computational procedures we want to call intelligent.

Q&A

Father of AI:

John McCarthy

(September 4, 1927 – October 23, 2011)



Q. When did AI research start?

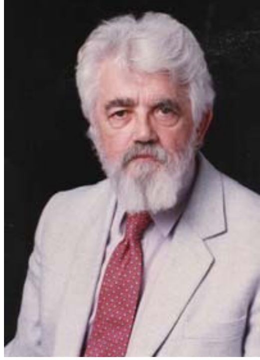
A. After WWII, a number of people independently started to work on intelligent machines. Alan Turing may have been the first. He gave a lecture on it in 1947. He also may have been the first to decide that AI was best researched by programming computers rather than by building machines. By the late 1950s, there were many researchers on AI, and most of them were basing their work on programming computers.

Q&A

Father of AI:

John McCarthy

(September 4, 1927 – October 23, 2011)



Q. What is the Turing test?

A. Alan Turing's in 1950 discussed conditions for considering a machine to be intelligent. He argued that if the machine could successfully pretend to be human to a knowledgeable observer then you certainly should consider it intelligent.

Q. What about chess?

A. Chess programs now play at grandmaster level, but they do it with limited intellectual mechanisms compared to those used by a human chess player, substituting large amounts of computation for understanding.

Some AI Topics

Topics I will cover:

- Search
- Logic
- Machine Learning

Other topics:

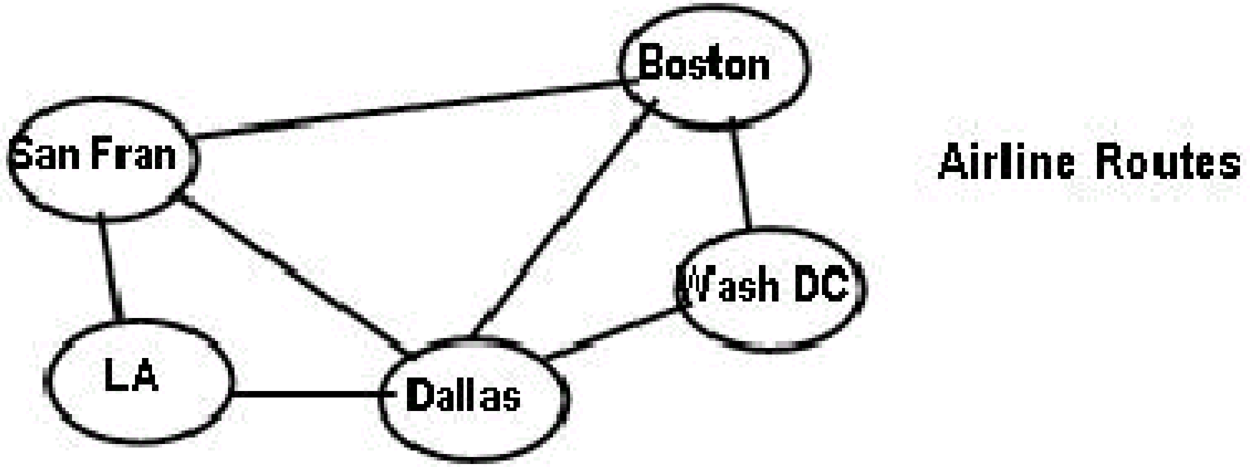
- Computer Vision
- Robotics
- Natural Language Processing
- Etc.

Search

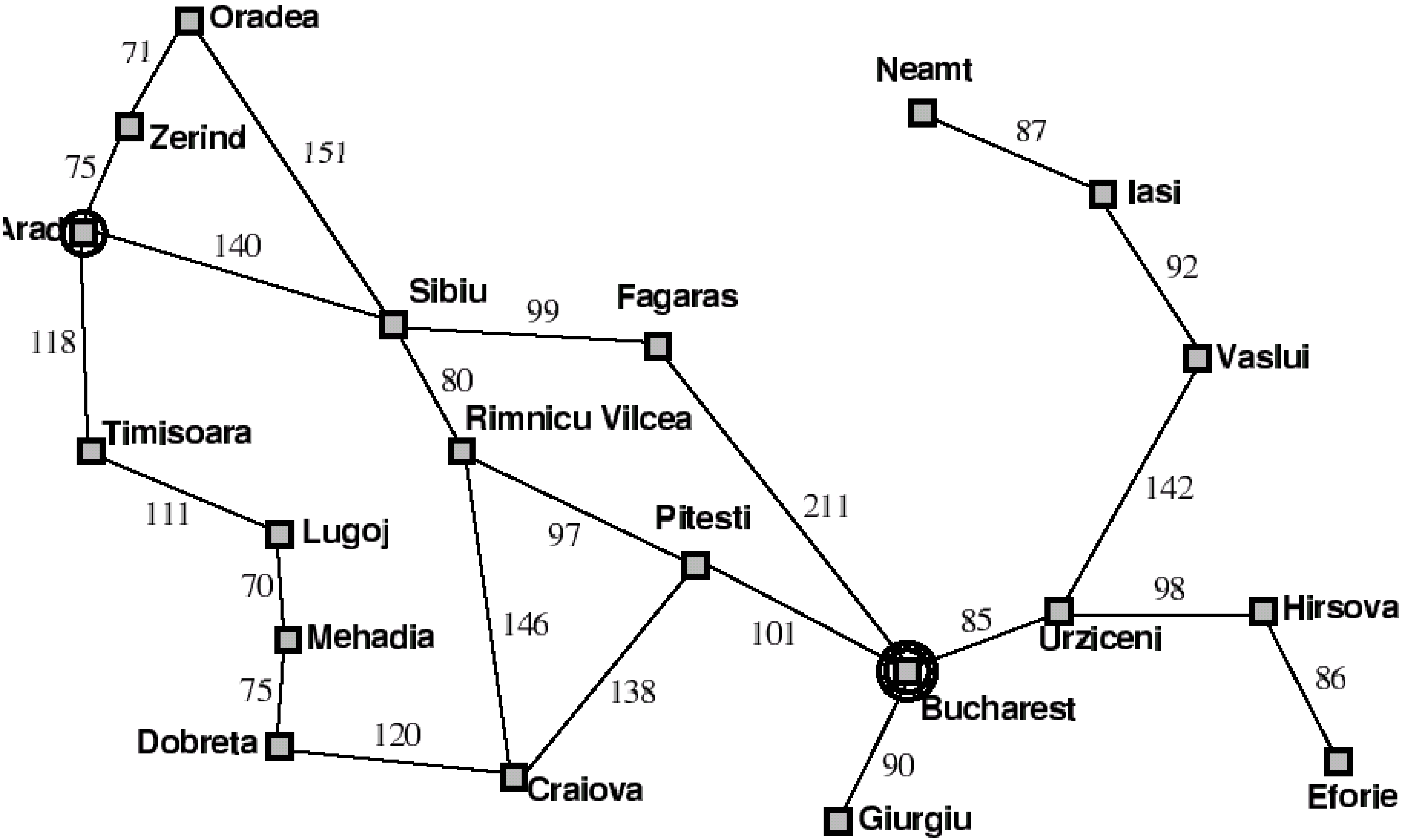
Systematic exploration of alternatives for reaching a goal.

Graphs

- Graphs are everywhere; E.g., think about road networks or airline routes or computer networks.
- In all of these cases we might be interested in finding a path through the graph that satisfies some property.



Romania graph



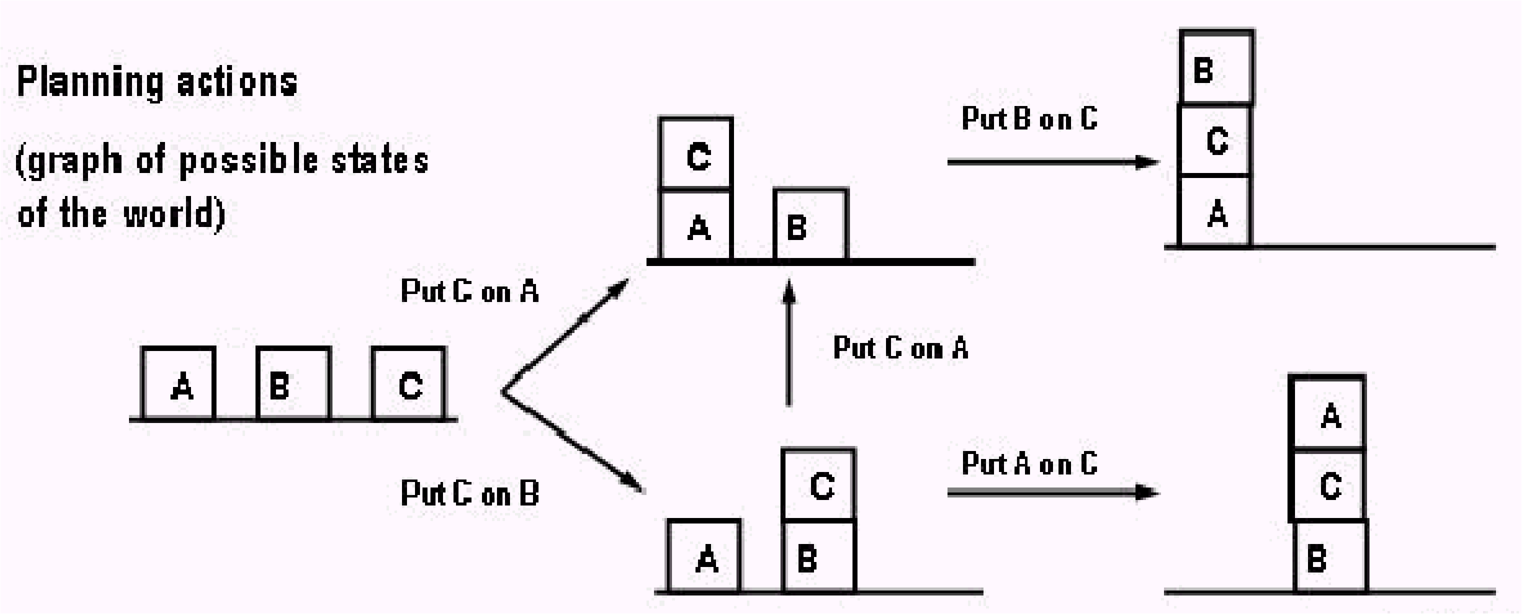
Formulating the problem

- On holiday in Romania; currently in Arad.
- Flight leaves tomorrow from Bucharest.

- Formulate goal:
  - be in Bucharest
- Formulate problem:
  - states: various cities
  - actions: drive between cities
- Find solution:
  - sequence of cities, e.g., Arad, Sibiu, Fagaras, Bucharest

Another graph example

- However, graphs can also be much more **abstract**.



- A **path** through such a graph (from a start node to a goal node) is a "plan of action" to achieve some desired goal.
- It's this type of graph that is of more general interest in AI.

Formally…

A problem is defined by four items:

1. initial state e.g., "at Arad"

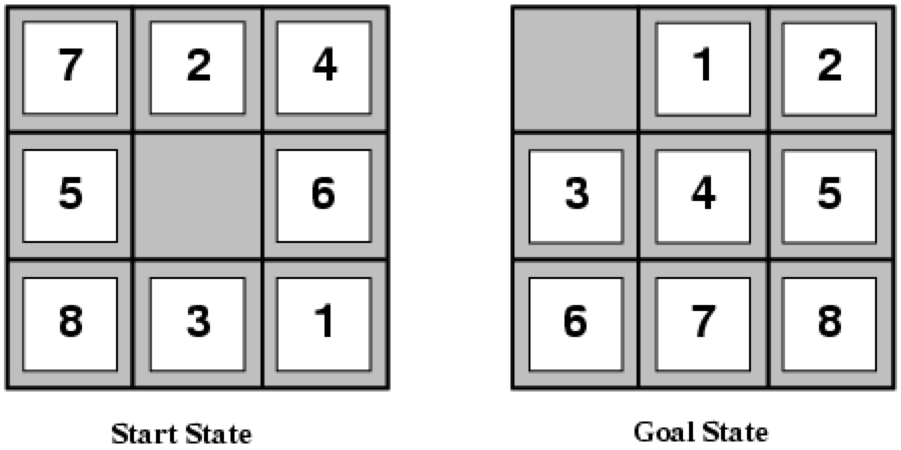
2. actions and successor function *S:* = set of **action-state** tuples
   – e.g., *S(Arad) = {(goZerind, Zerind), (goTimisoara, Timisoara), …, (goSilbiu, Silbiu)}*

3. goal test, can be
   – explicit, e.g., *x* = "at Bucharest"
   – implicit, e.g., *Checkmate(x)*

4. path cost (additive)
   – e.g., sum of distances, or number of actions executed, etc.

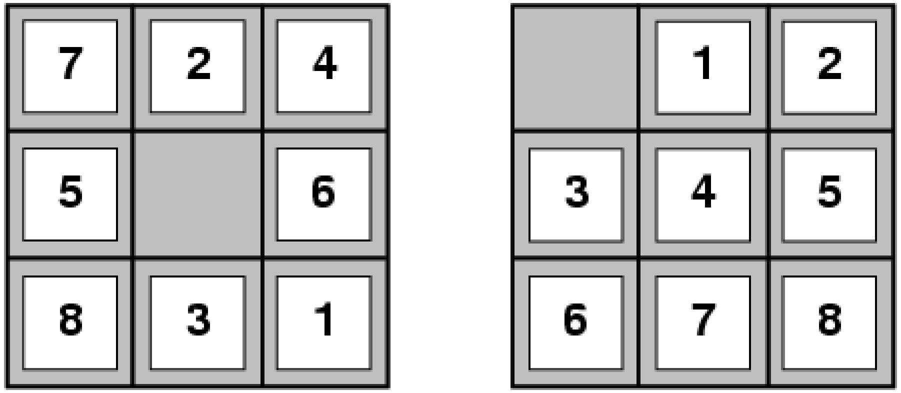
- **A solution is a sequence of actions leading from the initial state to a goal state**

Example: The 8-puzzle



- states?
- actions?
- goal test?
- path cost?

Example: The 8-puzzle

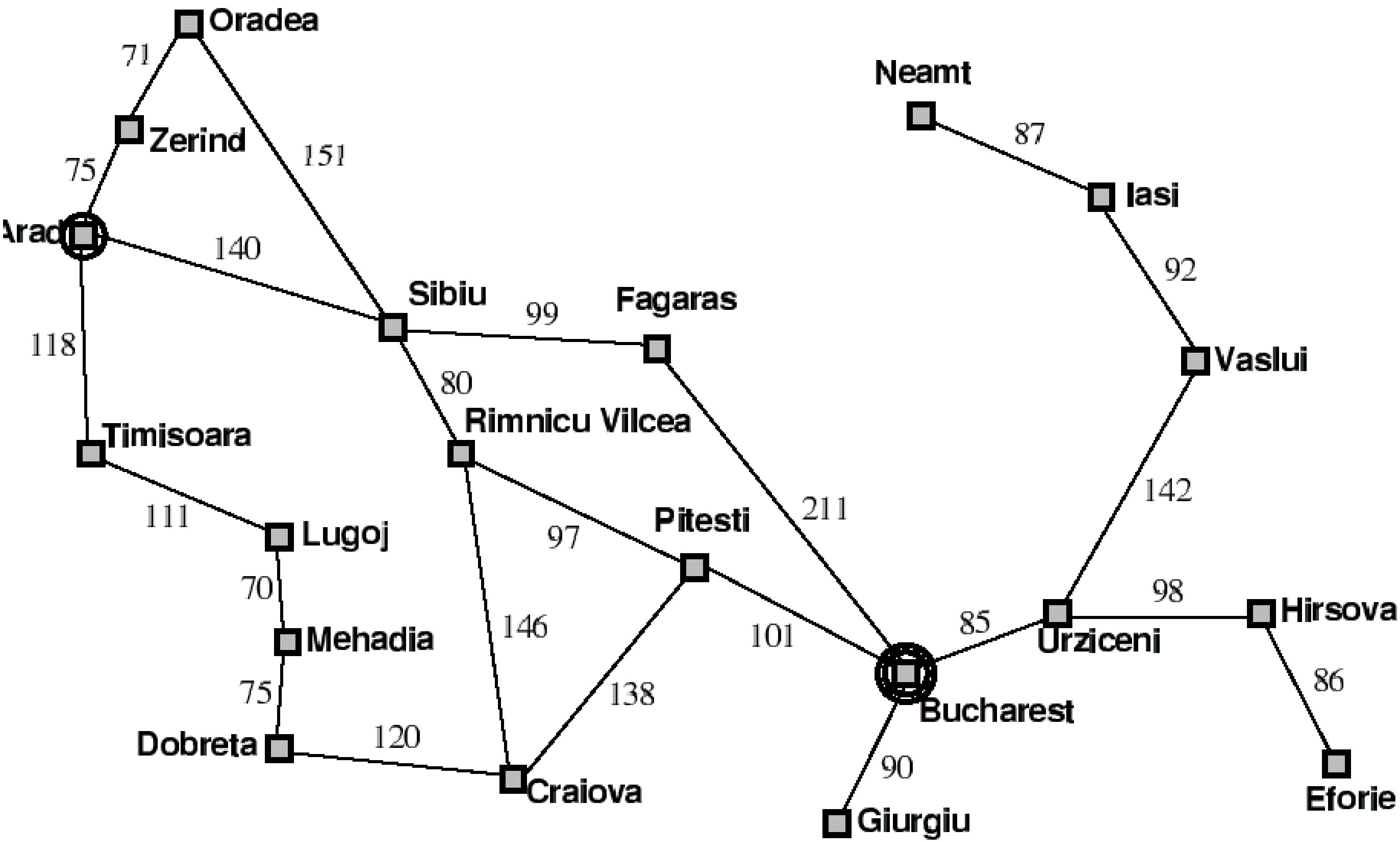


Start State

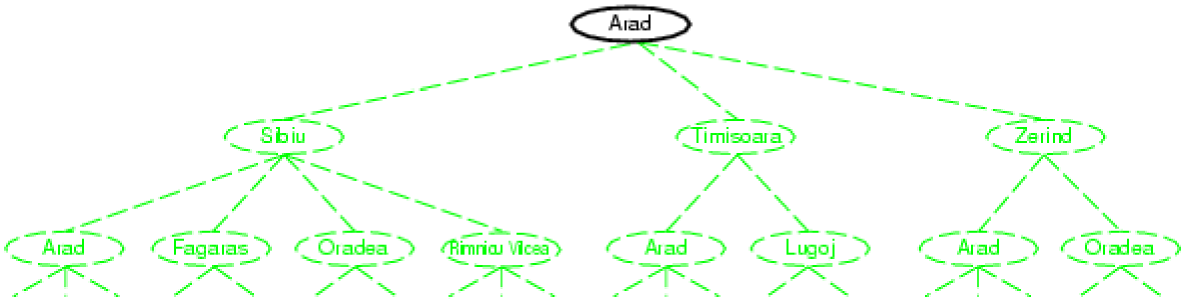
Goal State

- states? locations of tiles
- actions? move blank left, right, up, down
- goal test? = goal state (given)
- path cost? 1 per move

Romania graph

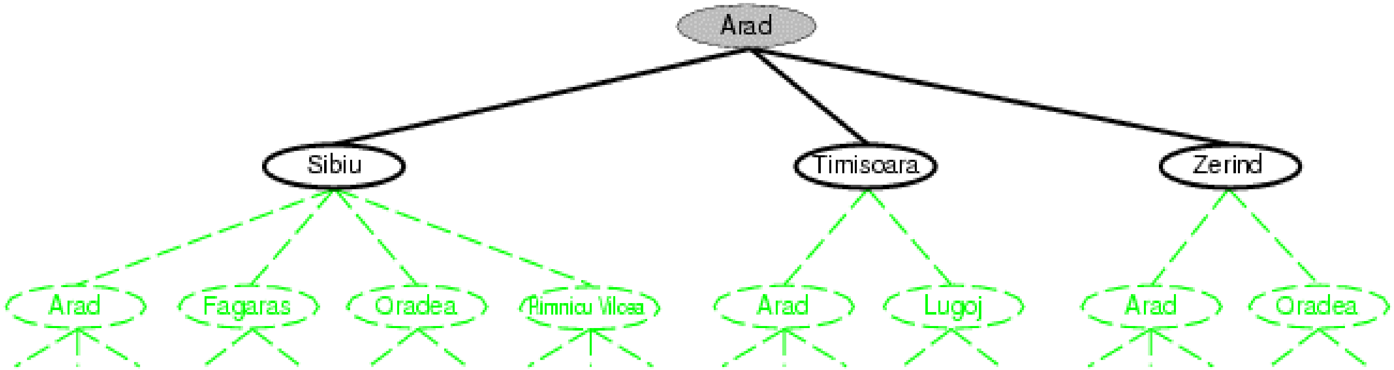


Tree search example

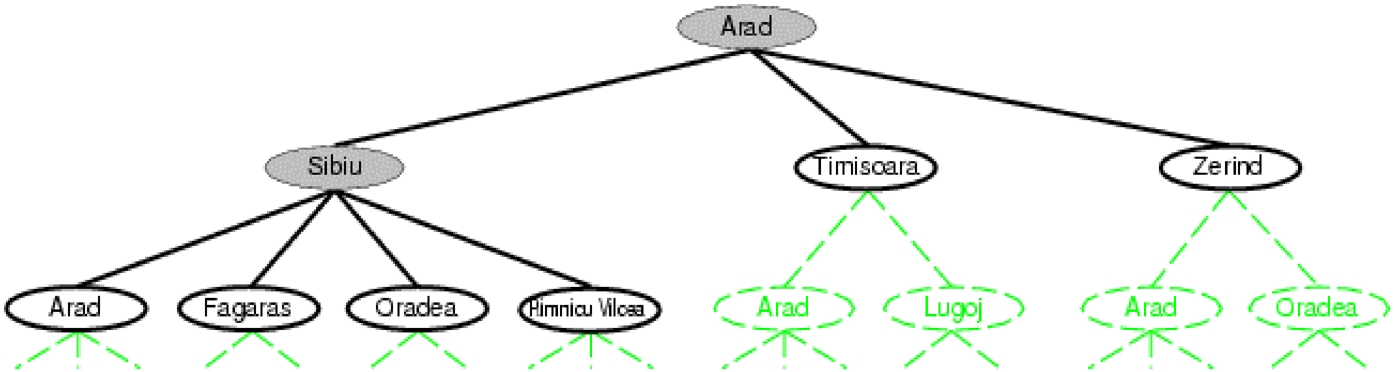


- Basic idea:
  - Explore state space by generating successors of already-explored states (i.e. expanding states)
  - In effect we have a tree of exploration

Tree search example



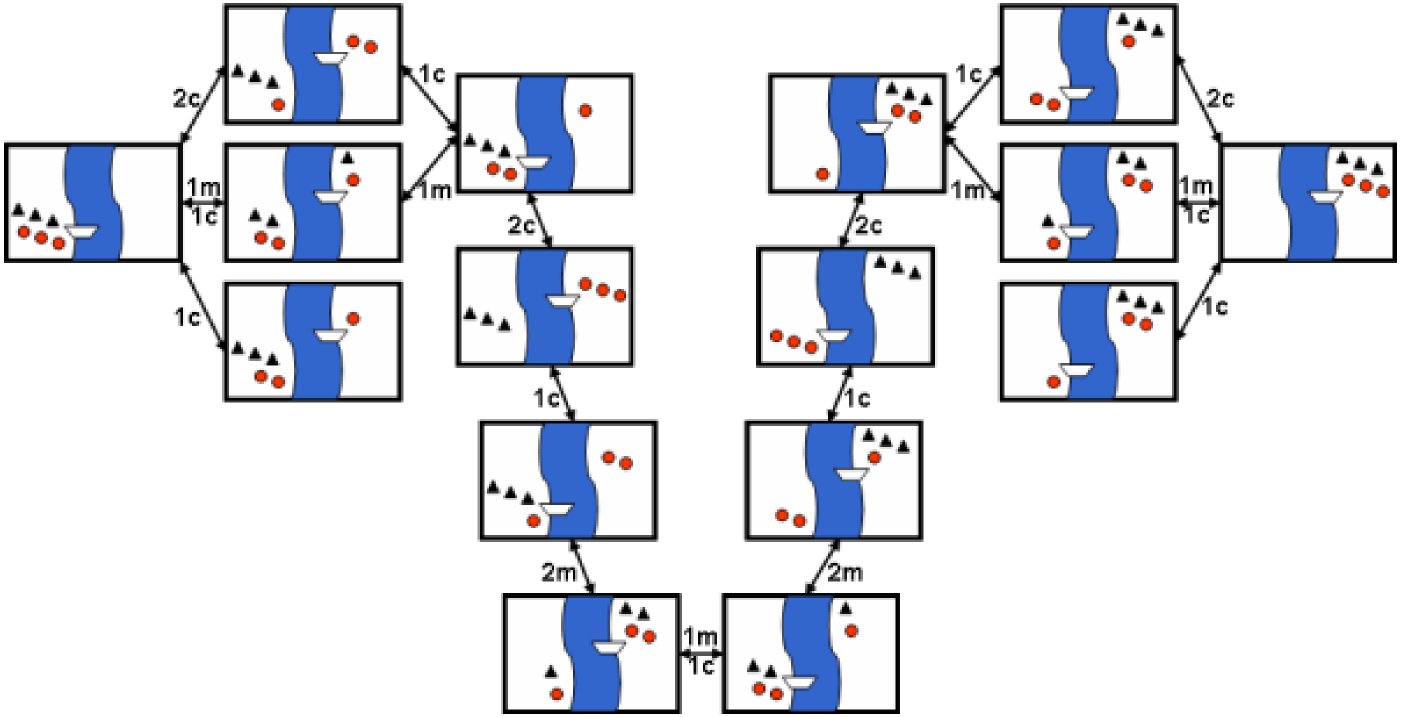
Tree search example



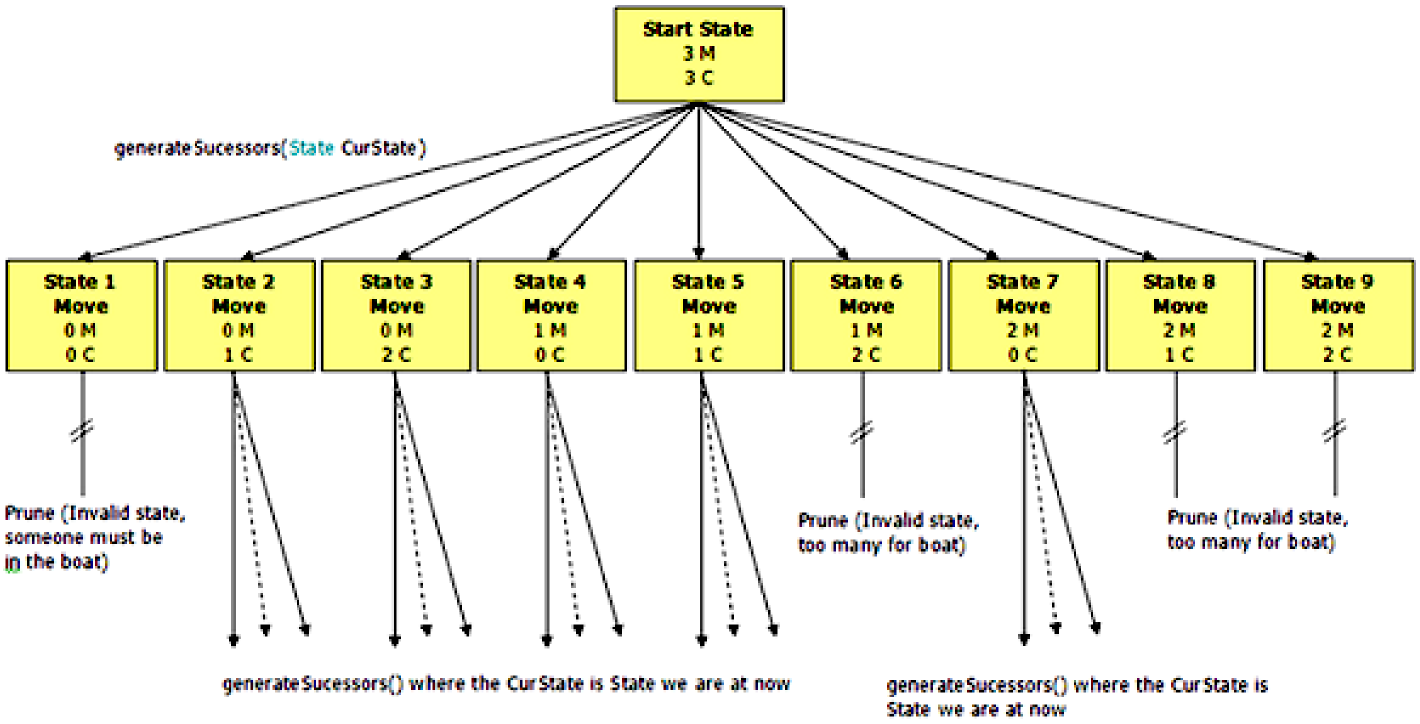
Missionaries and Cannibals

- Help 3 cannibals and 3 missionaries to move to the other side of the lake. Note that when there are more cannibals on one side of the lake than missionaries, the cannibals eat them. So you need to think logically and strategically in order to get them all across to the other side safely.
- http://www.learn4good.com/games/puzzle/boat.htm

M&C Graph



Missionaries and Cannibals



Logic

Human Logic

- Humans are, among other things, information processors.

- We acquire information about the world and use this information to further our ends.

- One of the strengths of human information processing is our ability to represent and manipulate logical information.

A simple puzzle

- We are given some facts about the arrangement of five blocks in a stack, and we are asked to determine their exact arrangement.

Premises

- *The red block is on the green block.*
- *The green block is somewhere* ***above*** *the blue block.*
- *The green block is* ***not*** *on the blue block.*
- *The yellow block is on the green block* ***or*** *the blue block.*
- *There is* ***some*** *block on the black block.*

Conclusions:

- *The red block is on the green block.*
- *The green block is on the yellow block.*
- *The yellow block is on the blue block.*
- *The blue block is on the black block.*
- *The black block is directly on the table.*

Aristotle (384-322 B.C.E.)

- The concept of proof, in order to be meaningful, requires that we be able to recognize certain reasoning steps as immediately obvious.

- One of Aristotle's great contributions to philosophy was his recognition that what makes a step of a proof immediately obvious is its **form** rather than its content.
- *It does not matter whether you are talking about blocks or stocks or automobiles. What matters is the structure of the facts with which you are working.*

Examples

All Accords are Hondas.
All Hondas are Japanese.
Therefore, all Accords are Japanese.

All borogoves are slithy toves.
All slithy toves are mimsy.
Therefore, all borogoves are mimsy.

Lewis Carroll's poem "Jabberwocky"

- What's more, in order to reach these conclusion, we do not need to know anything about *Hondas* and *Accords* or *borogoves* and *slithy toves* or what it means to be *mimsy*.
- What is interesting about these examples is that they share the same reasoning structure, with respect to the **pattern** shown below.

All x are y.
All y are z.
Therefore, all x are z.

Question

- Which patterns are sound?

Unsound Patterns

Pattern

All x are y.

Some y are z.

Therefore, some x are z.

Good Instance

All Toyotas are Japanese cars.

Some Japanese cars are made in America.

Therefore, some Toyotas are made in America.

Not-So-Good Instance

All Toyotas are cars.

Some cars are Porsches.

Therefore, some Toyotas are Porsches.

Formal Logic

1. Formal language for encoding information

2. Legal transformations

Logic Problem

If Amy loves Pat, then Amy loves Quincy.
If it is Monday and raining, then Amy loves Pat or Quincy.

If it is Monday and raining, does Amy love Quincy?

Formalization

Simple sentences:

Amy loves Pat.

loves(amy, pat)

Amy loves Quincy.

loves(amy, quincy)

It is Monday.

ismonday

It's raining.

raining

Premises:

If Amy loves Pat, Amy loves Quincy.

$loves(amy,pat) \Rightarrow loves(amy,quincy)$

If it Monday and raining, Amy loves Pat or Quincy.

$ismonday \wedge raining \Rightarrow loves(amy,pat) \vee loves(amy,quincy)$

Question:

If it is Monday and raining, does Amy love Quincy?

$\boldsymbol{ismonday \wedge raining \Rightarrow loves(amy,quincy)\ ?}$

Rule of Inference (Resolution)

$$\frac{\begin{array}{l} p_1 \wedge \ldots \wedge p_k \Rightarrow q_1 \vee \ldots \vee q_l \\ r_1 \wedge \ldots \wedge r_m \Rightarrow s_1 \vee \ldots \vee s_n \end{array}}{p_1 \wedge \ldots \wedge p_k \wedge r_1 \wedge \ldots \wedge r_m \Rightarrow q_1 \vee \ldots \vee q_l \vee s_1 \vee \ldots \vee s_n}$$

- If p_i on the left hand side of one sentence is the same as q_j in the right hand side of the other sentence, it is okay to drop the two symbols.
- A sentence

  $\Rightarrow p$

  asserts that p is true (i.e. a fact)
- A sentence

  $p \Rightarrow$

  asserts that p is false

Solving Amy's love life…

$$\frac{\begin{array}{l} p_1 \wedge \ldots \wedge p_k \Rightarrow q_1 \vee \ldots \vee q_l \\ r_1 \wedge \ldots \wedge r_m \Rightarrow s_1 \vee \ldots \vee s_n \end{array}}{p_1 \wedge \ldots \wedge p_k \wedge r_1 \wedge \ldots \wedge r_m \Rightarrow q_1 \vee \ldots \vee q_l \vee s_1 \vee \ldots \vee s_n}$$

- If p_i on the left hand side of one sentence is the same as q_j in the right hand side of the other sentence, drop the two symbols.

$$\frac{\begin{array}{l} \mathit{loves(amy,pat)} \Rightarrow \mathit{loves(amy,quincy)} \\ \mathit{ismonday} \wedge \mathit{raining} \Rightarrow \mathit{loves(amy,pat)} \vee \mathit{loves(amy,quincy)} \end{array}}{\cancel{\boldsymbol{\mathit{loves(amy,pat)}}} \wedge \mathit{ismonday} \wedge \mathit{raining} \Rightarrow \mathit{loves(amy,quincy)} \vee \cancel{\boldsymbol{\mathit{loves(amy,pat)}}} \vee \mathit{loves(amy,quincy)}}$$

$$\mathit{ismonday} \wedge \mathit{raining} \Rightarrow \mathit{loves(amy,quincy)} \vee \mathit{loves(amy,quincy)}$$

$$\boldsymbol{\mathit{ismonday} \wedge \mathit{raining} \Rightarrow \mathit{loves(amy,quincy)}}$$

History

- Theorem Provers have come up with novel mathematical results.
- The most famous of these concerns Robbins Algebra.

- In 1933, E. V. Huntington presented the following basis for Boolean algebra:
  - $x + y = y + x$. [commutativity]
  - $(x + y) + z = x + (y + z)$. [associativity]
  - $n(n(x) + y) + n(n(x) + n(y)) = x$. [Huntington equation]

- Shortly thereafter, Herbert Robbins conjectured that the Huntington equation can be replaced with a simpler one:
  - $n(n(x + y) + n(x + n(y))) = x$. [Robbins equation] a

- Robbins said that he worked on the problem for some time, and then passed it on one of the century's most famous logicians, Dr. Albert Tarski of Stanford University. Tarski, who is now dead, worked on the problem, included it in a book, and handed it out to graduate students and visitors.

- Burris (University of Waterloo), for example said that Tarski suggested the problem to him in the early 1970s, while he was visiting Stanford for a couple of months. Tarski, he said, "liked to throw out challenging problems to people passing through."

- While mathematicians were batting around Robbins's problem, computer scientists were striving to see if they could get computers to reason.
- Among them was Wos, who started working on automated reasoning in the 1960s. It was a time when computers were primitive, clunky and slow, and researchers were divided on how to proceed.

- On October 10, 1996, after eight days of computation, EQP (a version of OTTER Theorem Prover) found a proof.

Nowadays

- AMD, Intel and others use automated theorem proving to verify that division and other operations are correctly implemented in their processors.

Learning

Learning

We can think of at least three different problems being involved in learning:

- memory
- averaging
- generalization

Example problem

(Adapted from Leslie Kaelbling's example in the MIT courseware)

- Imagine I'm trying predict whether my neighbor is going to drive into work, so I can ask for a ride.

- Whether she drives into work seems to depend on the following attributes of the day:
  - temperature
  - expected precipitation
  - day of the week
  - what she's wearing

Memory

- Okay. Let's say we observe our neighbor on three days:

Temp	**Precip**	**Day**	**Clothes**	
25	None	Sat	Casual	**Walk**
-5	Snow	Mon	Casual	**Drive**
15	Snow	Mon	Casual	**Walk**

Memory

- Now, we find ourselves on a snowy “-5” degree Monday, and the neighbor is wearing casual clothes.
- Do you think she's going to drive?

Temp	**Precip**	**Day**	**Clothes**	
25	None	Sat	Casual	**Walk**
-5	Snow	Mon	Casual	**Drive**
15	Snow	Mon	Casual	**Walk**
-5	Snow	Mon	Casual	

Memory

- Standard answer in this case is "yes".
  - This day is just like one of the ones we've seen before, and so it seems like a good bet to predict "yes."
- This is the most rudimentary form of learning, which is just to memorize the things you've seen before.

Temp	**Precip**	**Day**	**Clothes**	
25	None	Sat	Casual	**Walk**
-5	Snow	Mon	Casual	**Drive**
15	Snow	Mon	Casual	**Walk**
-5	Snow	Mon	Casual	**Drive**

Noisy Data

- Things aren't always as easy as they were in the previous case. What if you get this set of noisy data?

Temp	Precip	Day	Clothes	
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Drive**
25	None	Sat	Casual	**Drive**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**?**

- We have certainly seen this case before, but the problem is that it has had different answers. Our neighbor is not entirely reliable.

Averaging

- One strategy would be to predict the majority outcome.

Temp	**Precip**	**Day**	**Clothes**	
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Drive**
25	None	Sat	Casual	**Drive**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**
25	None	Sat	Casual	**Walk**

Generalization

- Dealing with previously unseen cases
- Will she walk or drive?

Temp	Precip	Day	Clothes	
22	None	Fri	Casual	**Walk**
3	None	Sun	Casual	**Walk**
10	Rain	Wed	Casual	**Walk**
30	None	Mon	Casual	**Drive**
20	None	Sat	Formal	**Drive**
25	None	Sat	Casual	**Drive**
-5	Snow	Mon	Casual	**Drive**
27	None	Tue	Casual	**Drive**
24	Rain	Mon	Casual	**?**

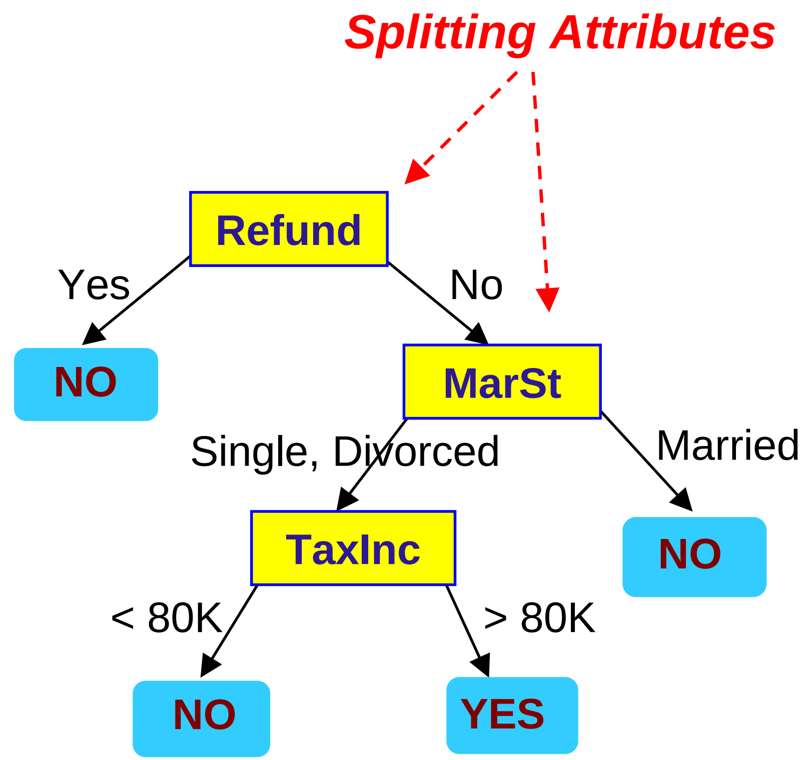
We might plausibly make any of the following arguments:

- She's going to walk because it's raining today and the only other time it rained, she walked.
- She's going to drive because she has always driven on Mondays…

Example of a Decision Tree

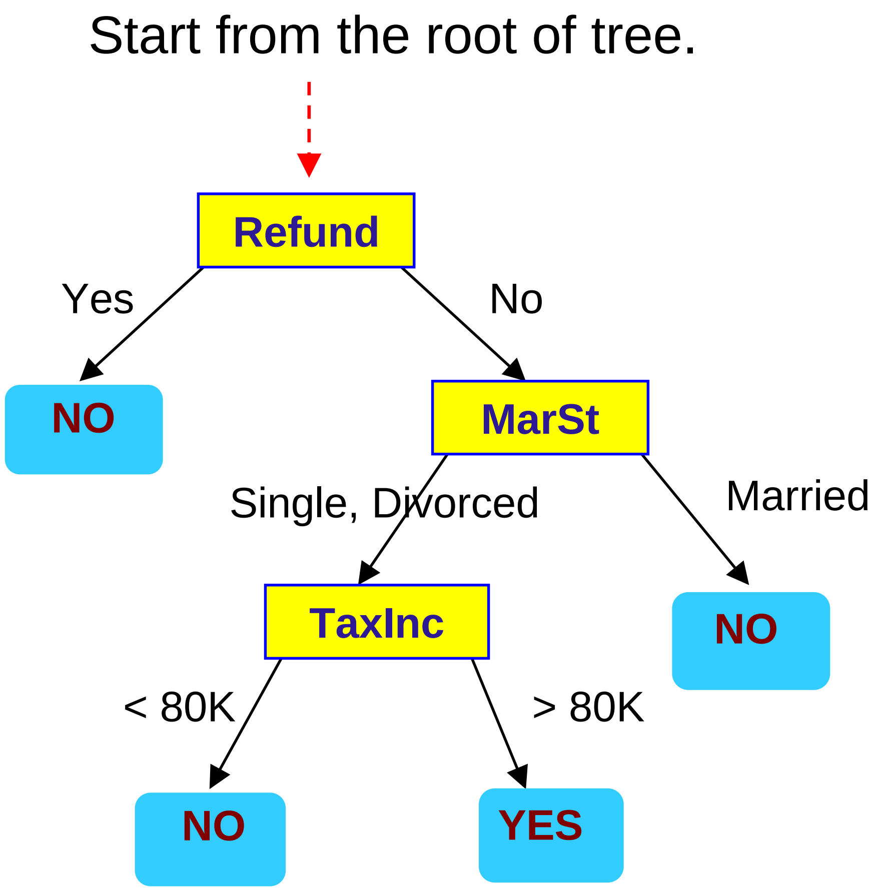
	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Training Data



Model: Decision Tree

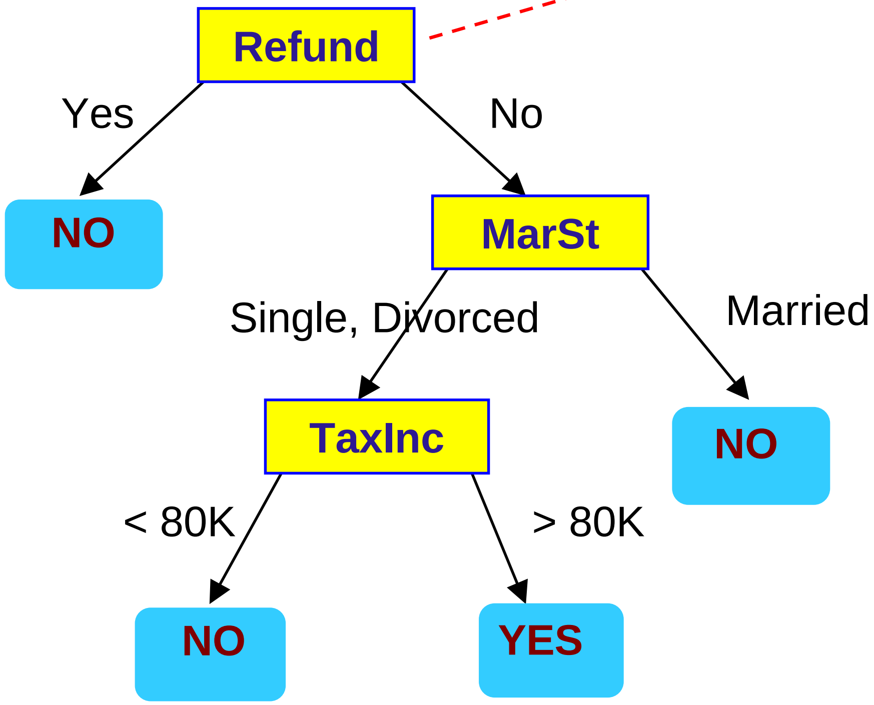
Apply Model to Data



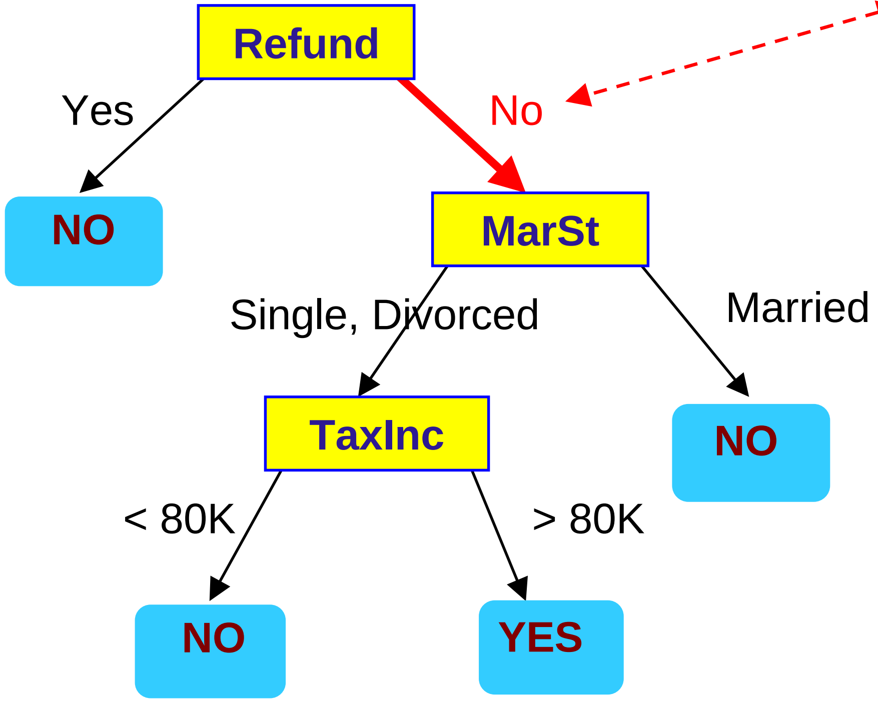
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	**?**

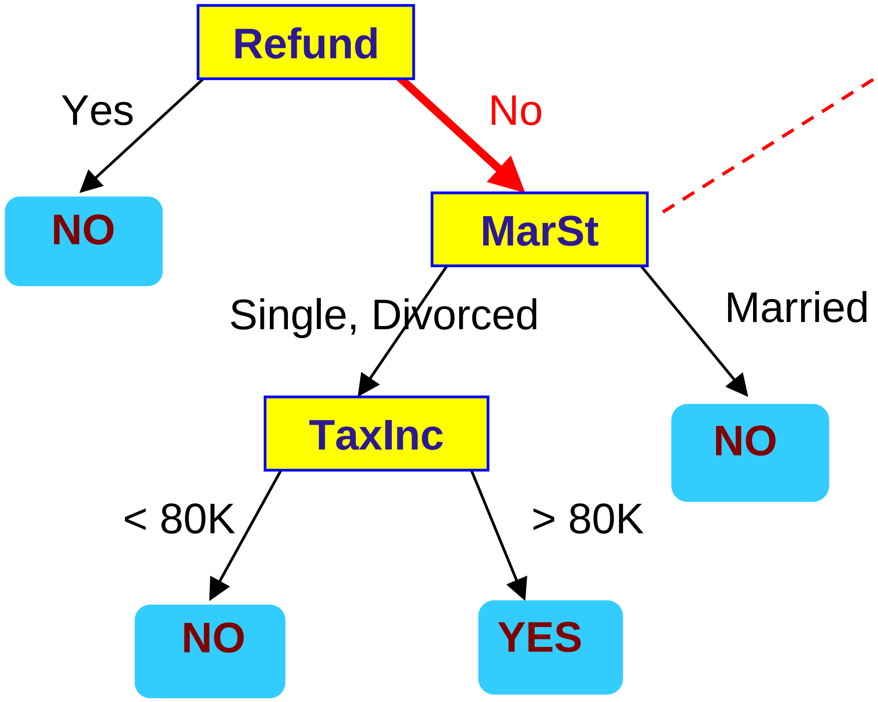


Apply Model to Data



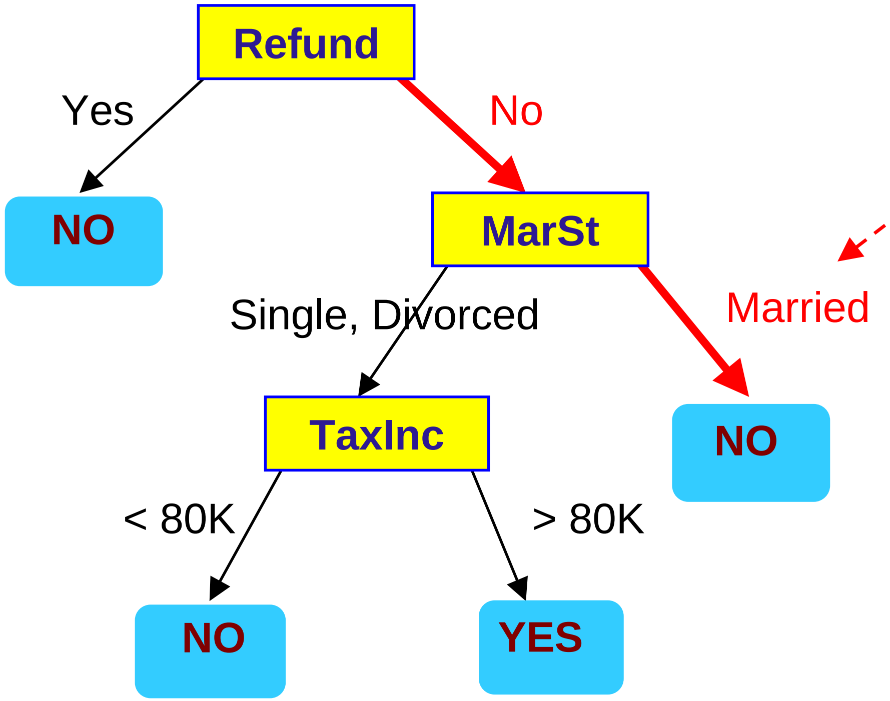
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Data



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

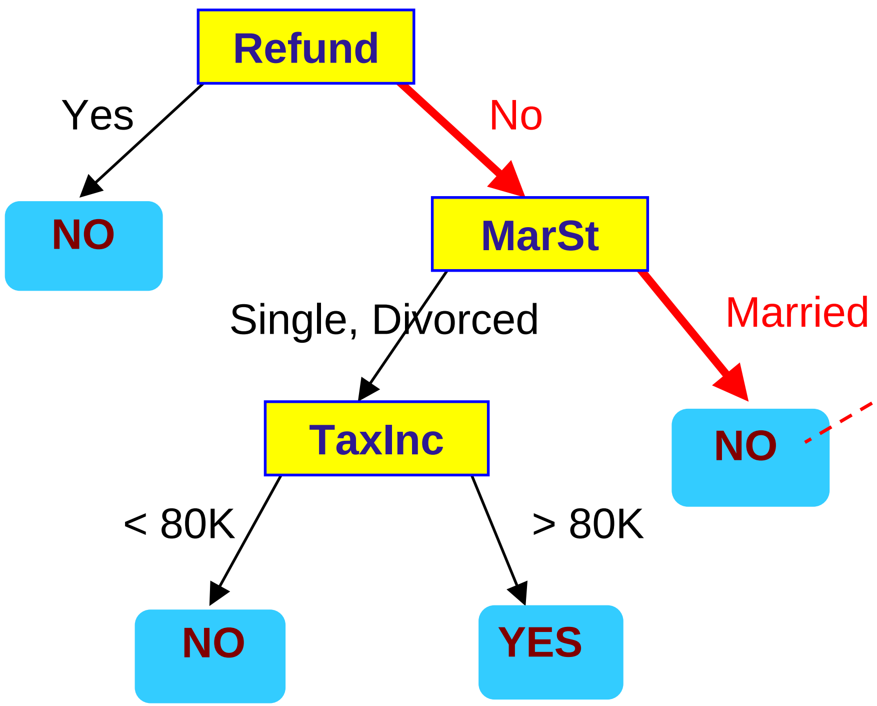
Apply Model to Data



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	**?**

Apply Model to Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

Classification: Fraud Detection

Goal: Predict fraudulent cases in credit card transactions.

Approach:

- Collect data about past transactions
  - when does a customer buy,
  - what does he buy,
  - where does he buy, etc.
- Label some past transactions as **fraud** or **fair** transactions.
- Learn a model for the class of the transactions.
- Use this model to detect fraud by observing credit card transactions on an account.

Classification: Direct Marketing

Goal: Reduce cost of mailing by *targeting* a set of consumers likely to buy a new cell-phone product.

Approach:

- Use the data for a similar product introduced before.
- We know which customers decided to buy and which decided otherwise.
- Collect various demographic, lifestyle, and other related information about customers.
- Learn a classifier model.

Finding Associations

The Market-Basket Model

- A large set of *items*, e.g., things sold in a supermarket.
- A large set of *baskets*, each of which is a small set of the items, e.g., the things one customer buys on one day.

Fundamental problem

- Learn sets of items that are often bought together.

Application…

- If a large number of baskets contain both hot dogs and mustard, we can use this information in several ways. How?

Beer and Diapers

What's the explanation here?

Or

On-Line Purchases

- Amazon.com: several million different items for sale, and several tens of millions of customers.

- Basket = Customer, Item = Book, DVD, etc.
  - Motivation: Learn what items are bought together.

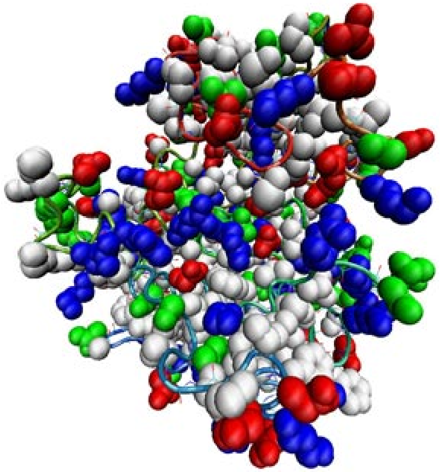
- Basket = Book, DVD, etc. Item = Customer
  - Motivation: Find out similar customers.

- Result: Use for recommender systems.

Words and Documents

- Baskets = sentences; items = words in those sentences.
  - Find words that appear together unusually frequently, i.e., learn linked concepts.
- Baskets = sentences, items = documents containing those sentences.
  - Items that appear together too often could represent plagiarism.

Genes



- Baskets = people; items = genes or blood-chemistry factors.
  - Has been used to learn combinations of genes that result in diabetes

That's all so far -- questions?