

K-12 AI Playground

David S. Touretzky, Christina Gardner-McCune,
and Jason Huang

Supported by NSF DRL-1846073.



Overview

- Introductions
- AI4K12 and Five Big Ideas in AI
- ReadyAI Curricula & WAICY Competition
- Part I: Interactive Browser-Based AI demos
- Part II: Unplugged Activities
- Break
- Part III: Cognimates / Machine Learning for Kids
- Break
- Part IV: Cozmo + Calypso
- Additional Resources and Next Steps

Your Workshop Presenters



Dave Touretzky, Research Professor of Computer Science
Carnegie Mellon University, Pittsburgh, PA



Christina Gardner-McCune, Assistant Professor
Computer & Information Science & Engineering Department
University of Florida, Gainesville, FL



Jason Huang, Program Director, ReadyAI.org
Pittsburgh, PA

Please Introduce Yourselves

- Who teaches K-2?
- Who teaches 3-5?
- Who teaches 6-8?
- Who teaches 9-12?
- Who doesn't fit these categories?

- Please introduce yourselves to the other people at your table.

The AI for K-12 Initiative



- Developing national guidelines for teaching AI in K-12
 - Modeled after the CSTA standards for computing education.
 - Organizing framework for the guidelines: “Five Big Ideas in AI”
 - Four grade bands: K-2, 3-5, 6-8, and 9-12
 - What should students know?
 - What should students be able to do?
- Developing a curated AI resource directory for K-12 teachers.
- Fostering a community of AI resource developers.
- Web site: <http://AI4K12.org>

The AI4K12 Initiative is a joint project of:

AAAI (Association for the Advancement of Artificial Intelligence)



Association for the
Advancement of Artificial Intelligence

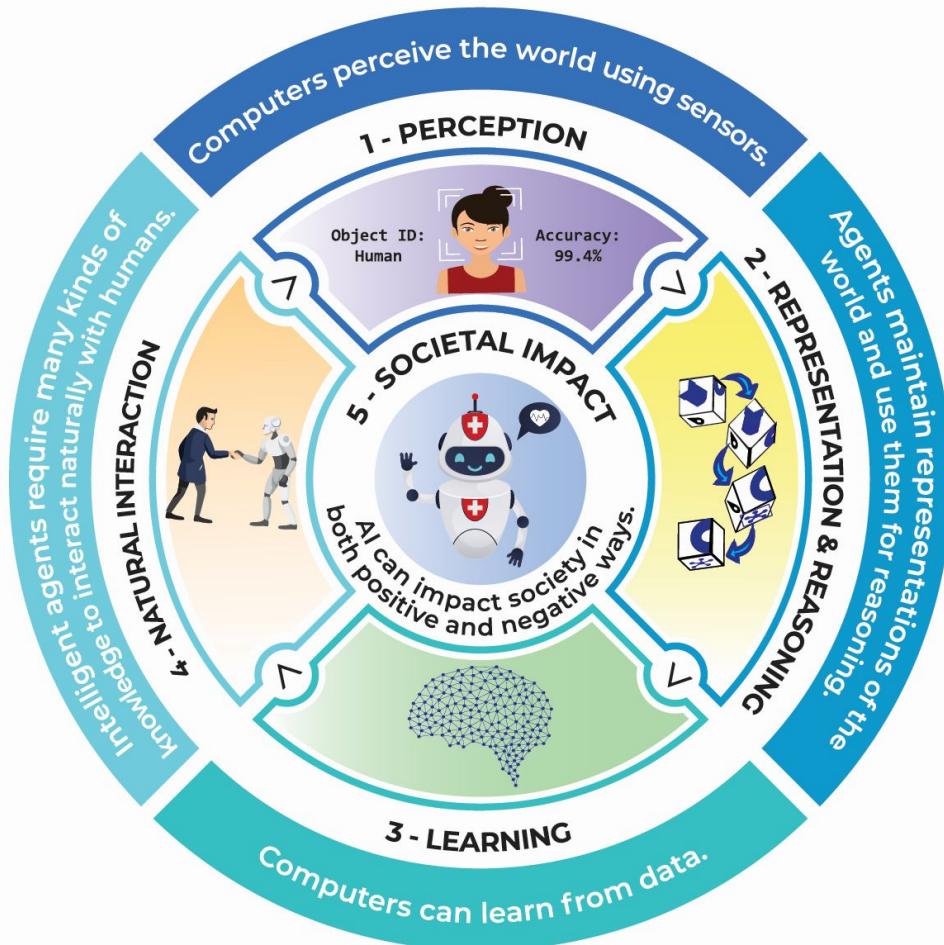
CSTA (Computer Science Teachers Association)



With funding from National Science Foundation ITEST Program (DRL-1846073)

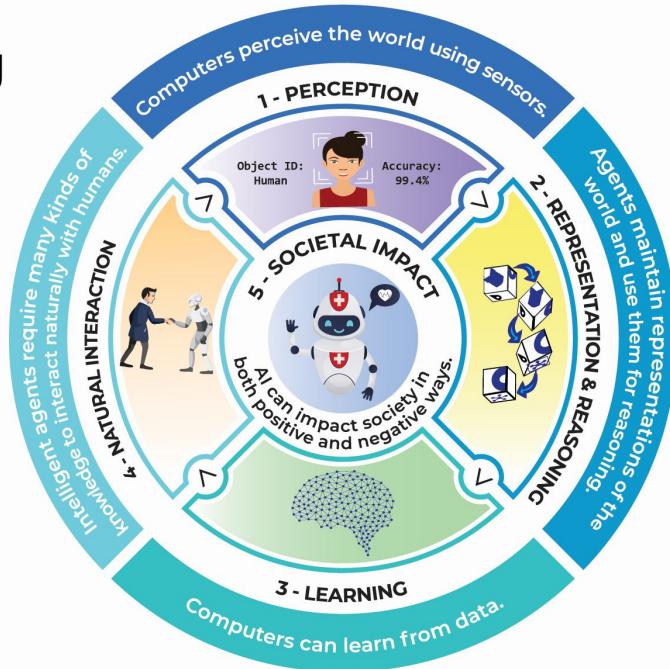
Carnegie Mellon University
School of Computer Science

Five Big Ideas in Artificial Intelligence



Five Big Ideas in AI

- Perception:** Computers perceive the world using sensors.
- Representation and Reasoning:** Agents maintain representations of the world and use them for reasoning.
- Learning:** Computers can learn from data.
- Natural Interaction:** Intelligent agents require many kinds of knowledge to interact naturally with humans.
- Societal Impact:** AI can have both positive and negative impacts on society.





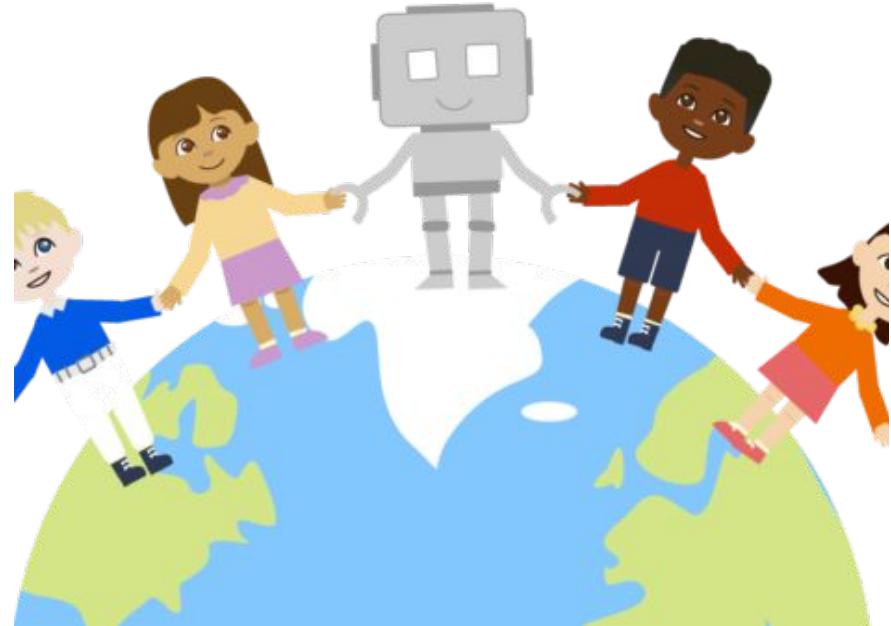
- ReadyAI is a Pittsburgh-based company that is developing AI curriculum and professional development materials for K-12 teachers.
- Makers of AI-IN-A-BOX: a turn-key solution that gives students a hands-on introduction to AI-enabled robotics using Calypso and the Cozmo robot.
- AI+ME: free online intro to AI for K-5 students.
- Sponsors of WAICY: the World Artificial Intelligence Competition for Youth.
- Visit <https://ReadyAI.org> to learn more.

AI + ME (AI and Me)

edu.readyai.org/courses/aime/

“AI+ME” is an online experience intended to provide young learners with the basics of AI. The lesson takes about one hour to complete. This is the first publicly available course introducing students to the “Five Big Ideas in AI” as defined by the AI4K12 Initiative.

Target Audience: Elementary School





Curriculum



AI-IN-A-BOX™

- 3 sets serving 3-15 students at a time.
- Color-labeled devices for best student experience.
- Fun and innovative teaching tools for your classroom.

K-12 Curriculum

- 8-12 hours of instruction
- Introductory, project-based courses
- Lesson plans, handouts, online courses.
- Suitable for every skill level
- No STEM teaching experience required
- PD available



First WAICY - July 2018 at CMU

5

Time zones

200+

Students

50+

Teams

(20+ remote participation)

"S.T.E.A.M.-Powered A.I."

- 50/50 Rubric
- Winning Project





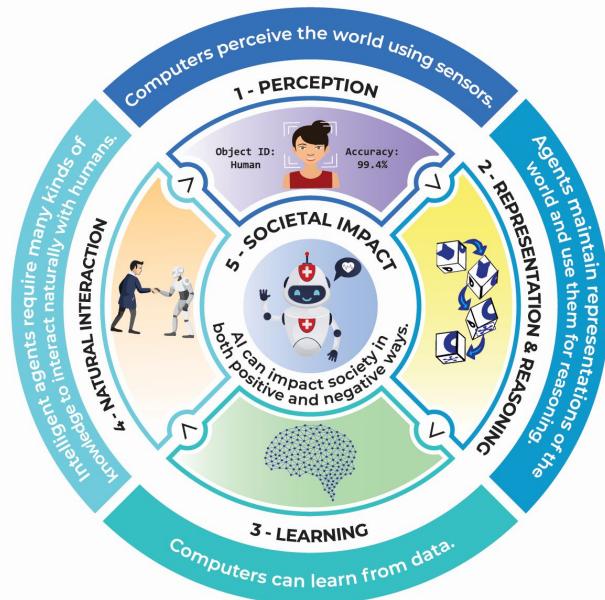
**STUDENTS DEVELOPED DEMONSTRATIONS OF
HOW THEY THINK AI CAN CHANGE THE WORLD**



WAICY 2019: S.T.E.A.M-Powered A.I.

July 27. 2019 at Nova Place in Pittsburgh, PA

AI Criteria



Multimedia Criteria

- Story line
- Visual Design
- Audio Design
- Theme/Message
- Preparation/Delivery
-
-
-





Part I: **Interactive Browser-Based AI demos**

Speech Recognition: Speech to Text Demos



1. Demo shows alternative parses
<https://www.cs.cmu.edu/~dst/SpeechDemo/>

The screenshot shows a web browser window with the URL <https://www.cs.cmu.edu/~dst/SpeechDemo/>. The page title is "Speech Recognition Demo". Below it, a instruction says "Speak into your microphone; see the results below.". A blue speech-to-text interface displays five green rectangular boxes containing the same text: "mine does the mine is because I would do all my print screens are backed up in my Dropbox". At the bottom is a green button labeled "MIC ON".



2. Demo speaks back what it heard
<https://speechnotes.co/>

The screenshot shows a web browser window with the URL <https://speechnotes.co/>. The page title is "SpeechNotes". It features a sidebar with icons for Apps, STEMinity, and STEM Ed. The main content area shows a note titled "Note_1, 16 Jan 2019" with the text "Click the mic to start dictating (note: the dictation feature works only on a Chrome browser.)". Below it, instructions say "New: Click the speaker icon (at the bottom-right) to read out loud your text for proof reading." and "For punctuation, you can use any of the 3 methods: (1) Dictation of the punctuation marks, (2) Key-typing the marks using your keyboard. This is usually the fastest, as it appends the mark to the speech, even before it finalized." On the right side, there's a vertical toolbar with icons for "Say", "Period", "Comma", "Question mark", "Colon", "Semi Colon", "Exclamation mark", and "Exclamation point". A tip on the right says: "Tip: While dictating, press Enter (on keyboard) to move results from buffer to editor."

AI & Creativity



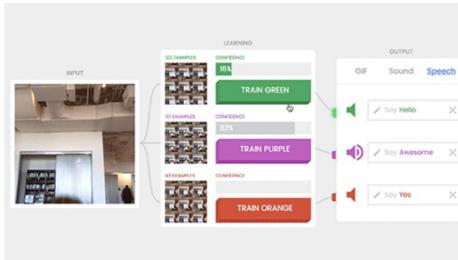
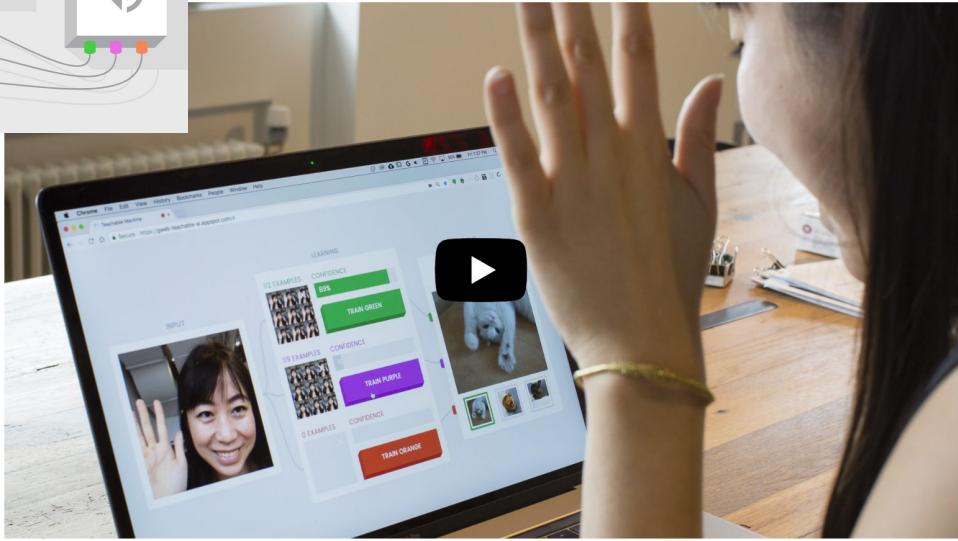
<https://experiments.withgoogle.com/mixlab>

Input & Output



Built with TensorFlow

- Teach a machine using your camera,
- live in the browser
- no coding required.

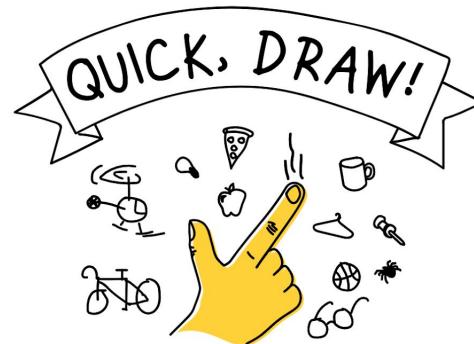


<https://experiments.withgoogle.com/teachable-machine>



Google's Quick, Draw!

<https://quickdraw.withgoogle.com/>



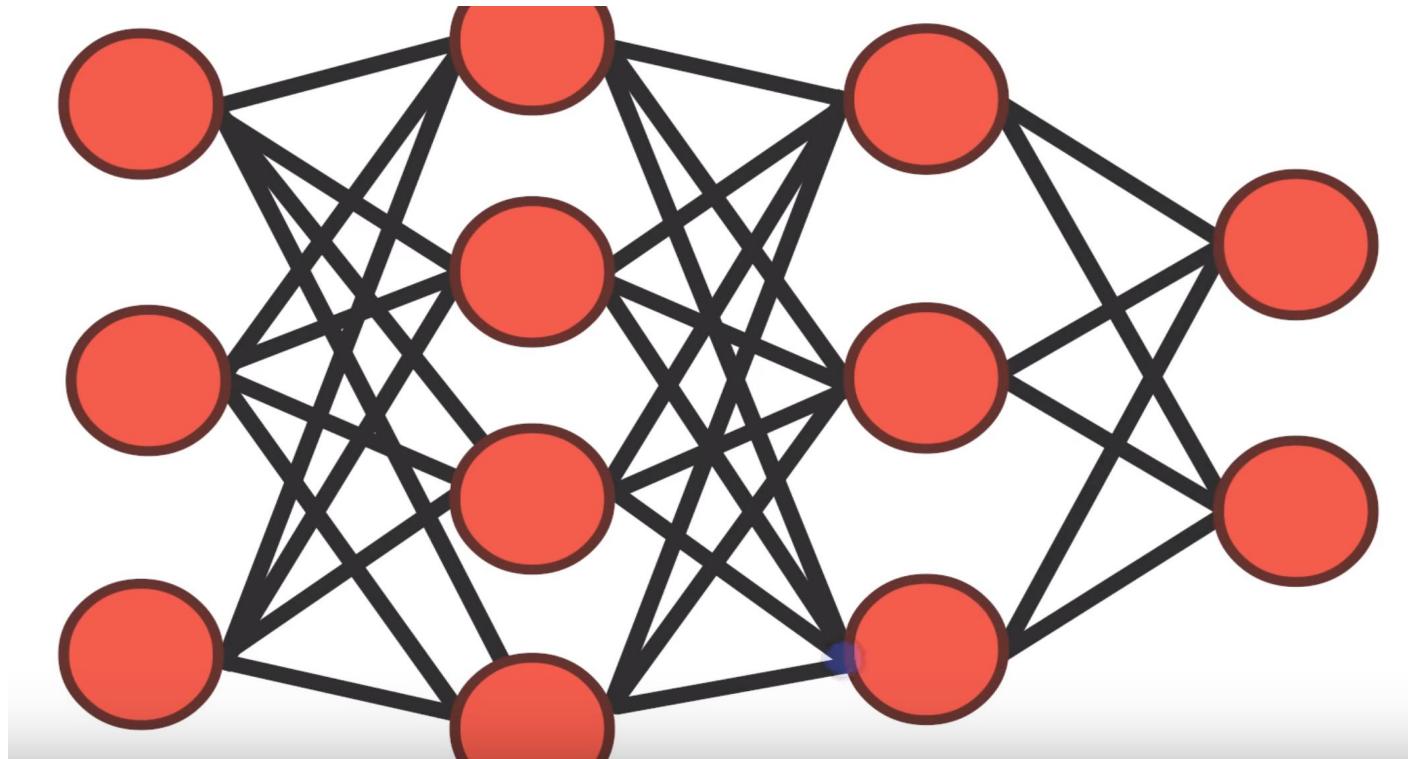
Can a neural network learn to recognize doodling?

Help teach it by adding your drawings to the [world's largest doodling data set](#), shared publicly to help with machine learning research.

Let's Draw!



Neural Networks in a minute



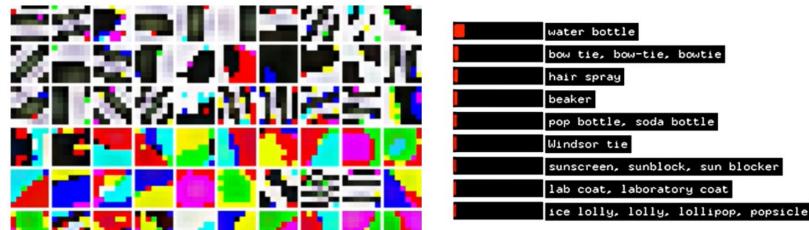
Deep Convolutional Neural Nets

Web Camera Input



Activation Map

Feature detection -
geometric shapes



Guesses of what is in
the video feed

<https://experiments.withgoogle.com/what-neural-nets-see>

TensorFlow Playground

<https://playground.tensorflow.org>

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

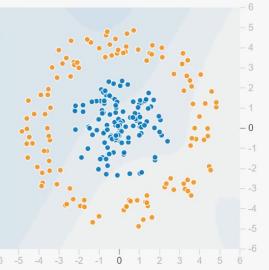
Epoch 000,000 Learning rate 0.03 Activation Tanh Regularization None Regularization rate 0 Problem type Classification

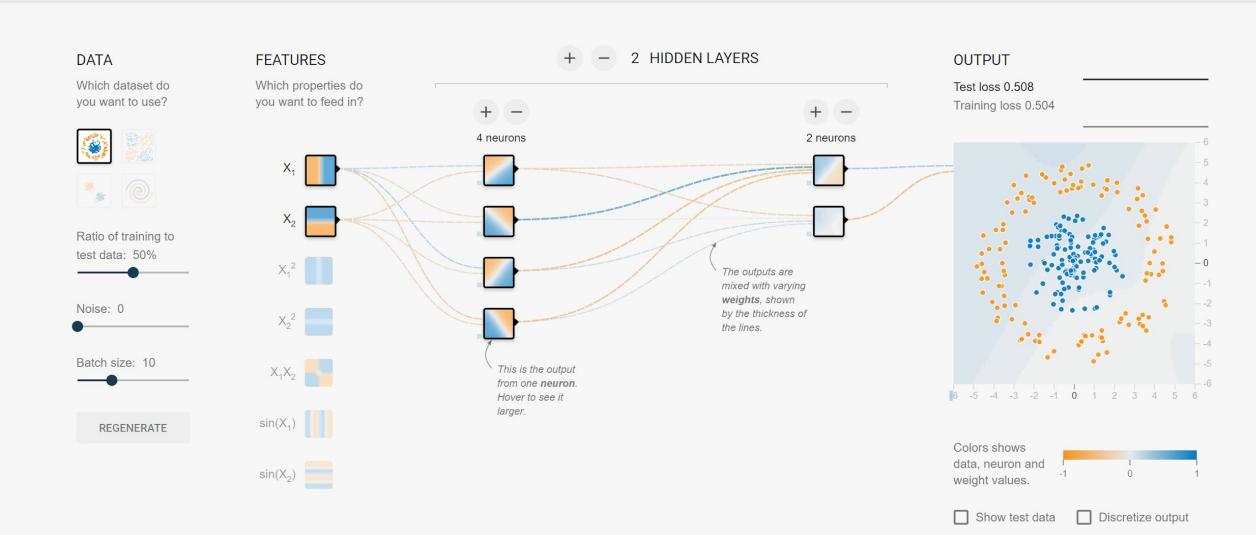
DATA
Which dataset do you want to use?

Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE

FEATURES
Which properties do you want to feed in?
 x_1 x_2 x_1^2 x_2^2 $x_1 x_2$ $\sin(x_1)$ $\sin(x_2)$

HIDDEN LAYERS
+ - 2 HIDDEN LAYERS
+ - 4 neurons + - 2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.

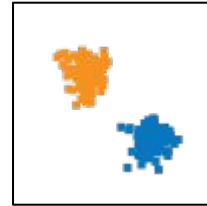
OUTPUT
Test loss 0.508 Training loss 0.504

Colors show data, neuron and weight values.
 Show test data Discretize output



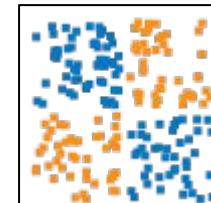
Tutorial: <https://cloud.google.com/blog/products/gcp/understanding-neural-networks-with-tensorflow-playground>

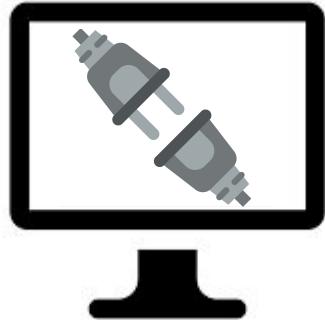
TensorFlow Exercise

1. Set problem to two gaussians:
2. Set # hidden layers to zero.
3. Turn on “discretize output”.
4. Train.



5. Set problem to four-region XOR:
6. Train with 0 hidden layers.
7. Train with 1 hidden layer.
8. Train with 2 hidden layers. Try several times.
9. Increase hidden layer 1 to 4 units and train again.





Part II: **AI Unplugged Activities**

How do you help your students get a feel for representation and reasoning?

- Unplugged exercises for students
 - Build your own taxonomy
 - Generate a search tree
- Experiments with smartphones, Alexa, Cozmo, etc.
- Online demos that display representations (e.g., sample parses)
- Animations of search algorithms

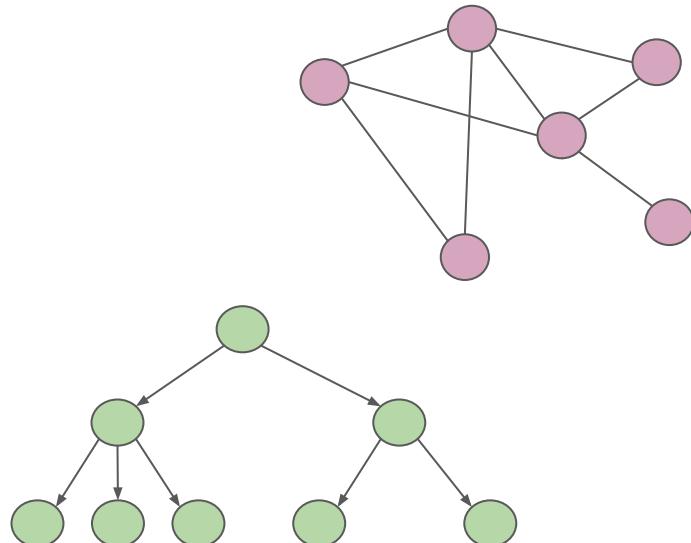
Graphs and Trees:Unplugged activities

Graphs and Tree are the Heart of Symbolic AI Representations

Cross-cutting connections:

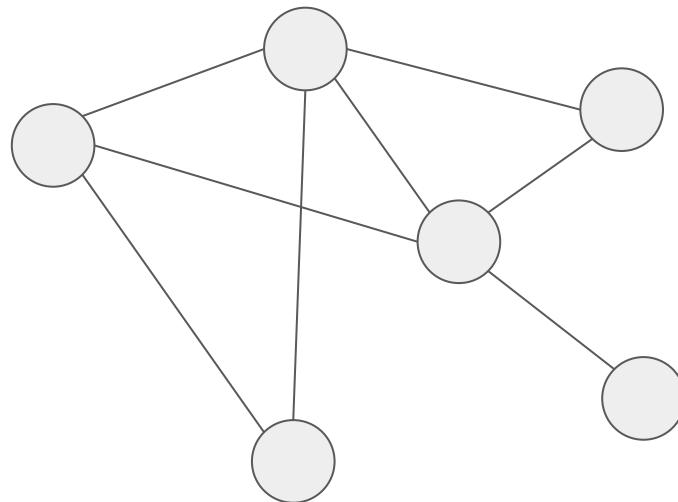
- (1) Computer science: data structures
- (2) Mathematics: graph theory

Graphs and trees are implemented in programming languages using lists, arrays, or dictionaries.



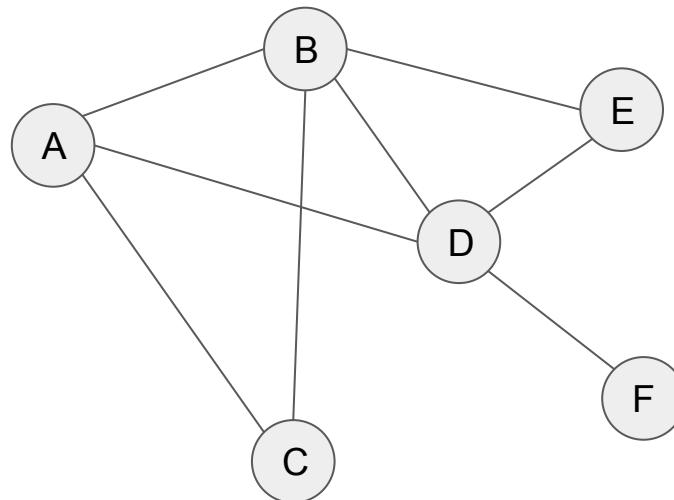
Graphs are Central to Symbolic AI Representations

In symbolic AI, the most common type of representation is the **graph**. (Trees are a special kind of graph.) Graphs are composed of **nodes** and **links**.



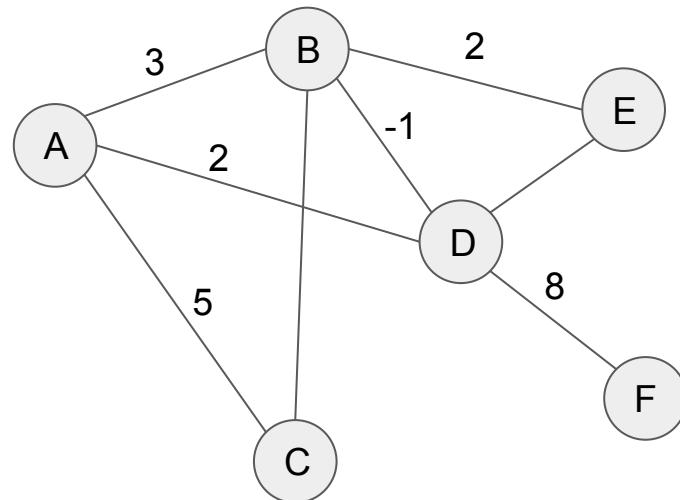
Labeled Graphs

The nodes of a graph may be “labeled”, i.e., they hold information.



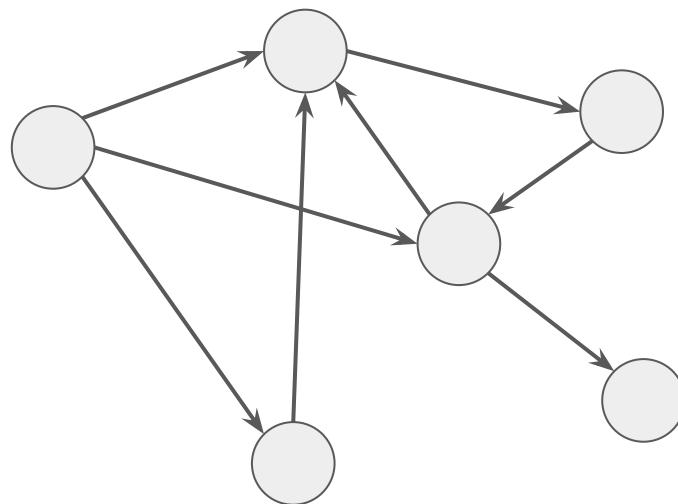
Labeled Graphs

The links may also be labeled.



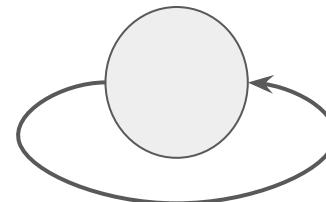
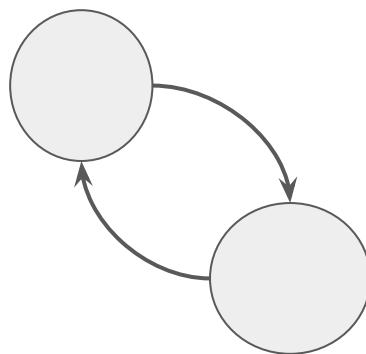
Directed Graphs

In a **directed** graph, each link has a direction, shown by an arrowhead:



Directed Graphs

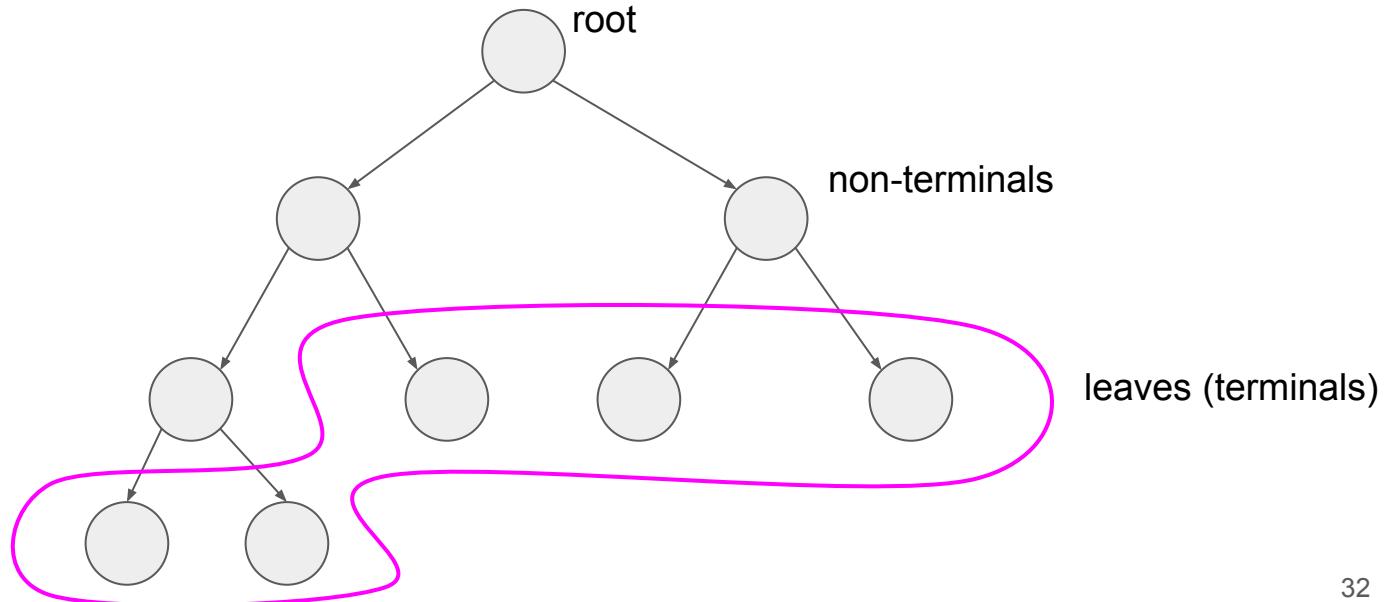
Cycles and self-links may be permitted.



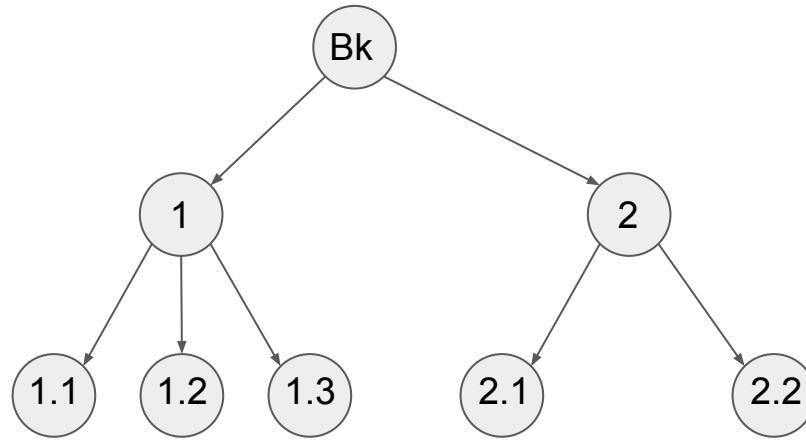
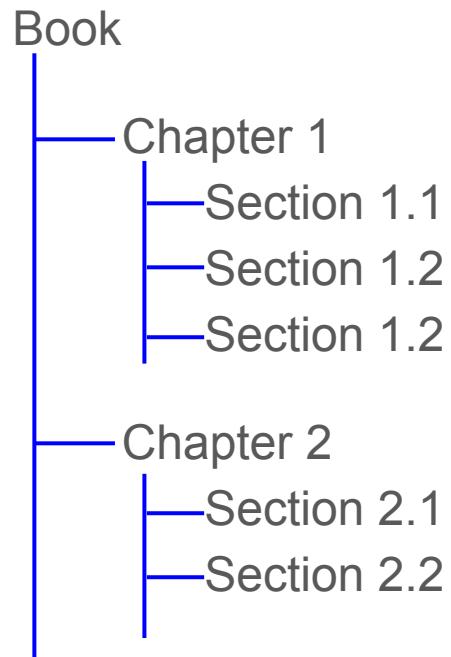
Trees

Trees are a special kind of graph.

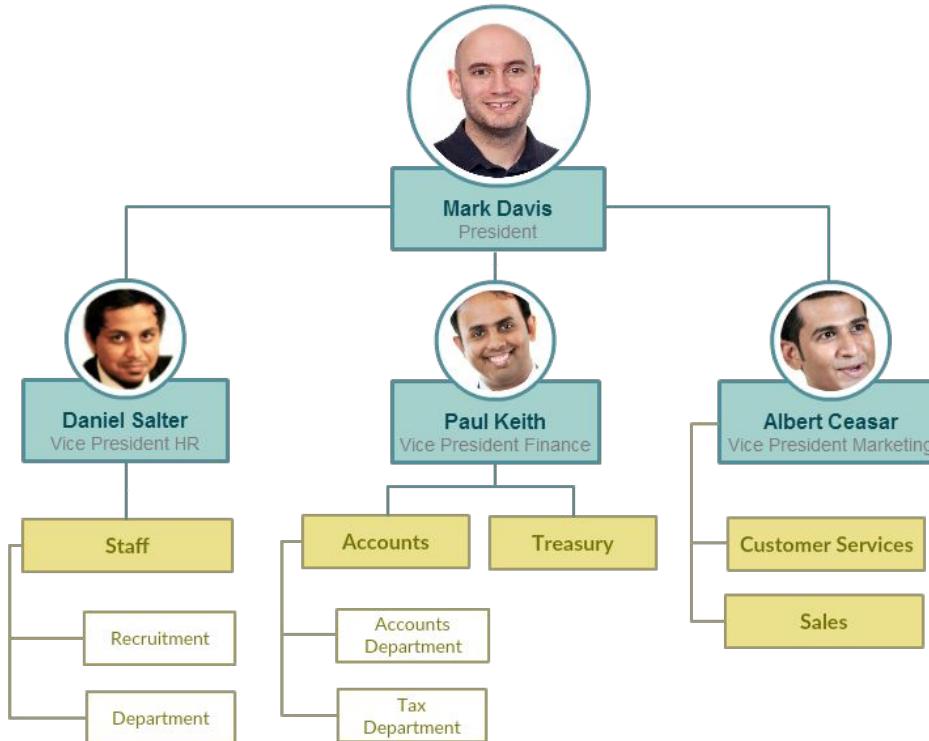
Each node has exactly one **parent**, except for the **root node**, which has none.
Nodes without **children** are called **leaves**, or **terminal nodes**.



Trees in real life: a book's table of contents



Trees in real life: an organizational chart



Trees in real life: taxonomic hierarchy

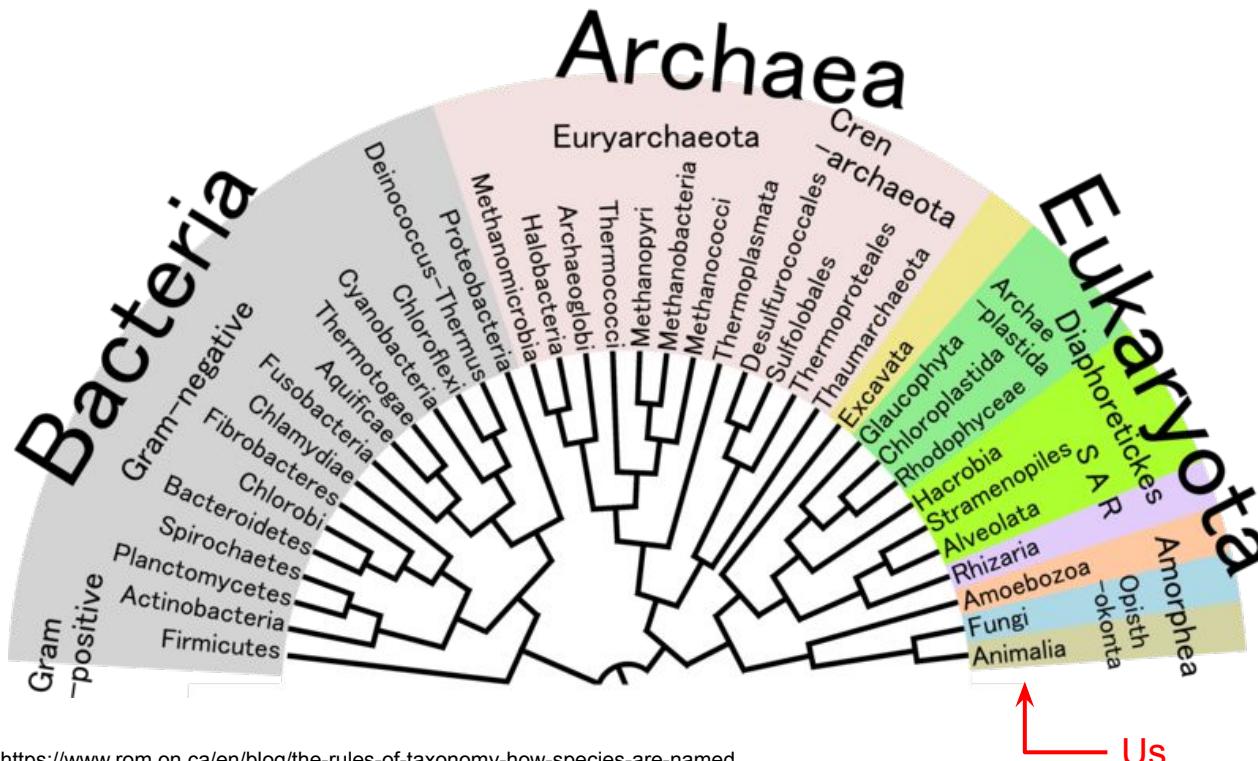
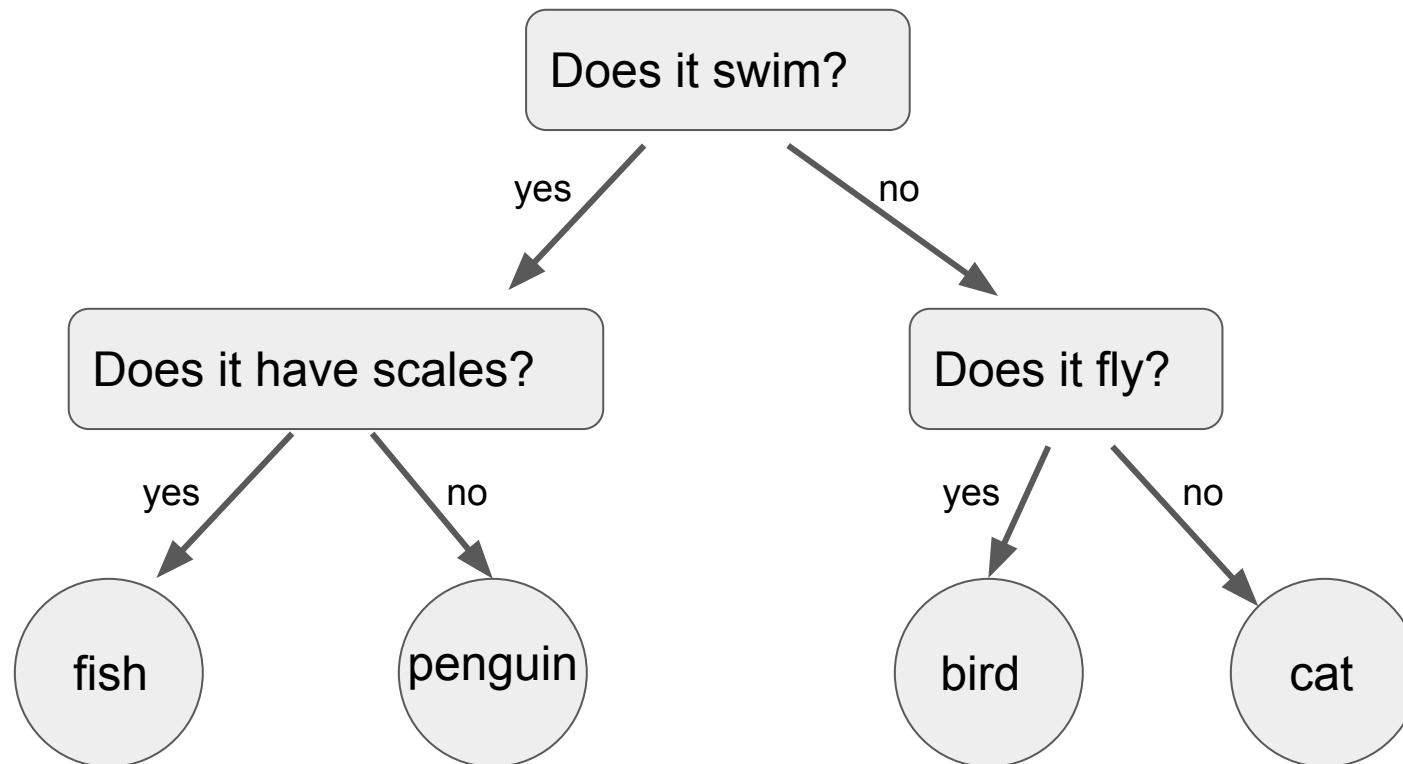


Image source: <https://www.rom.on.ca/en/blog/the-rules-of-taxonomy-how-species-are-named>

Unplugged Reasoning Exercise: Discrimination Net

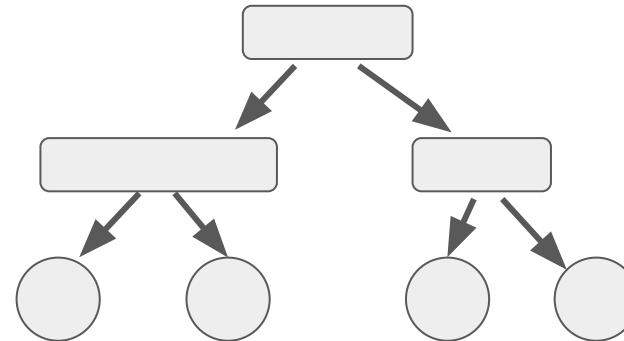
Discrimination Net: Guess the Animal



Discrimination Net: Guess the Animal

What is the **representation**?

- Tree structure
- Non-terminal nodes are questions
- Terminal nodes are animals



What is the **reasoning** algorithm?

- Start at the root node
- At a non-terminal node, ask the question.
- Use the answer to choose which node to visit next.
- At a terminal node, report that animal as your guess.

Growing the Discrimination Net

Computer: Does it have scales?

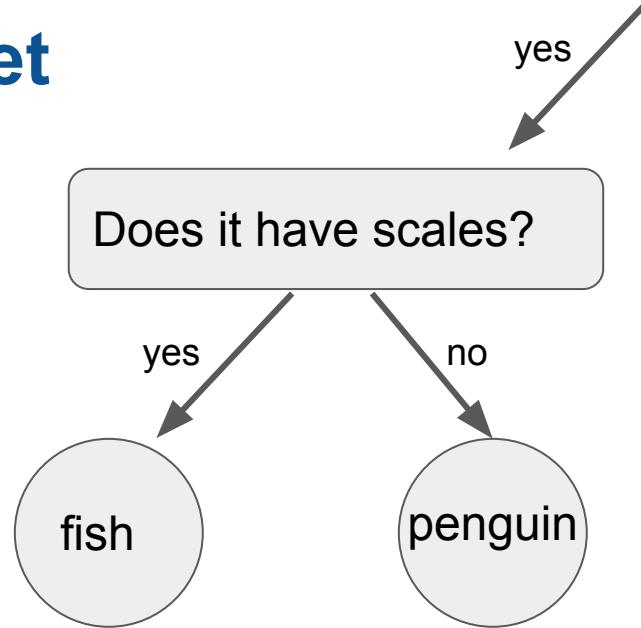
Human: no

Computer: I guess “penguin”. Is this correct?

Human: no

Computer: What was your animal?

Human: octopus



Growing the Discrimination Net

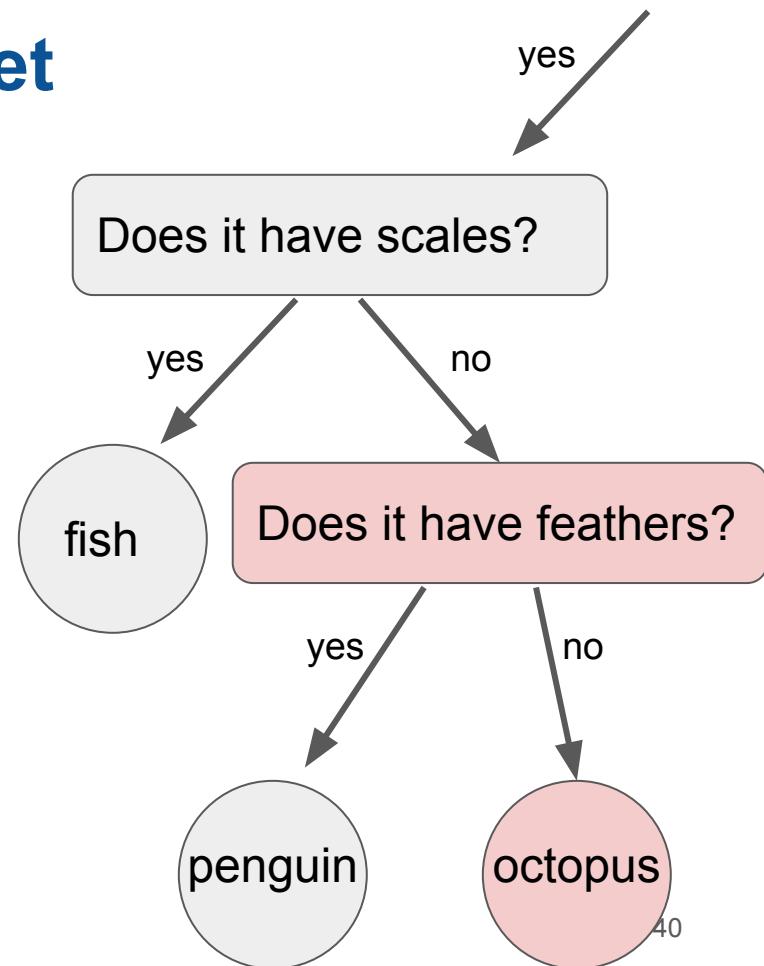
Computer: what question distinguishes between a penguin and an octopus?

Human: “Does it have feathers?”

Computer: What is the answer for octopus?

Human: “no”.

Computer: I'll remember that.



Unplugged Reasoning Exercise: Route Finding

Finding a route from Arad to Bucharest

Romania with step costs in km

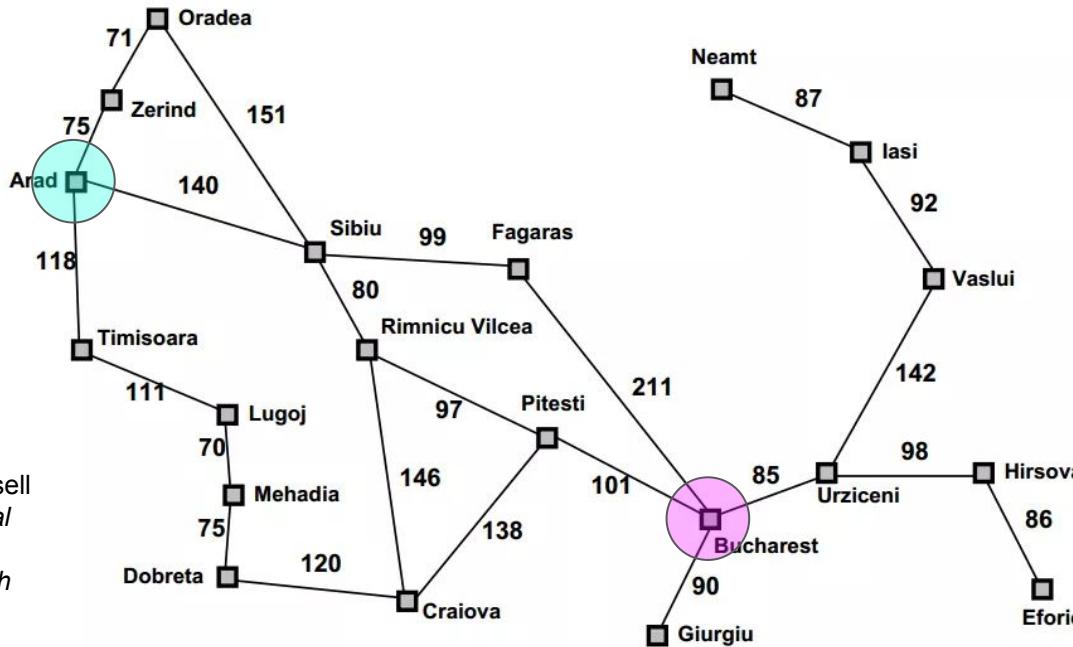
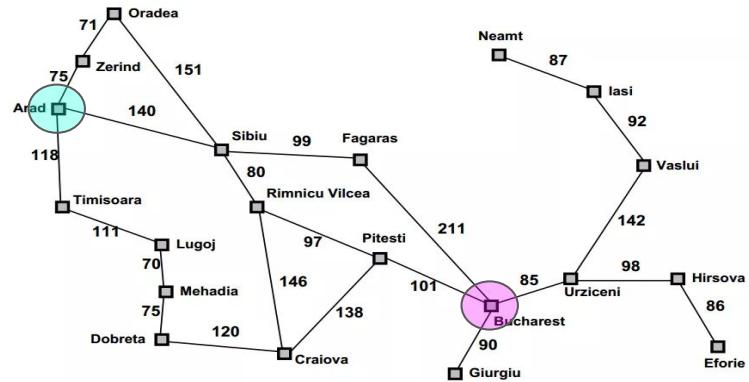


Figure from Russell & Norvig, *Artificial Intelligence: A Modern Approach*

Continue on
the next
slide

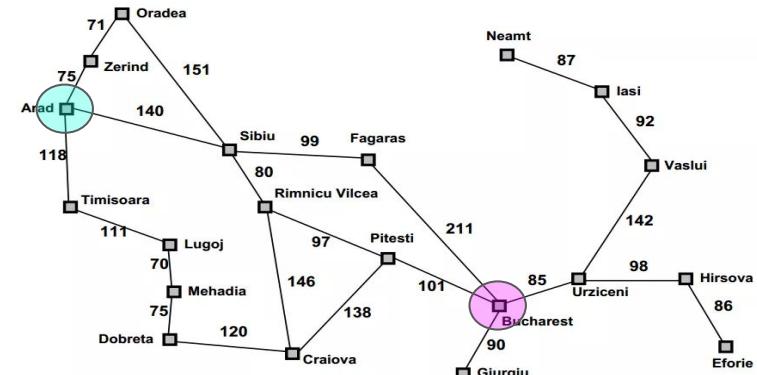
Simple route-finding algorithm

1. Start at Arad on line 1.
2. Write down every city you can reach directly from Arad on line 2.
3. Write down every city you can reach from there on line 3. Skip any city you've reached on any previous line.
4. Repeat until you reach Bucharest.
5. Trace backward from Bucharest to find the route to Arad.



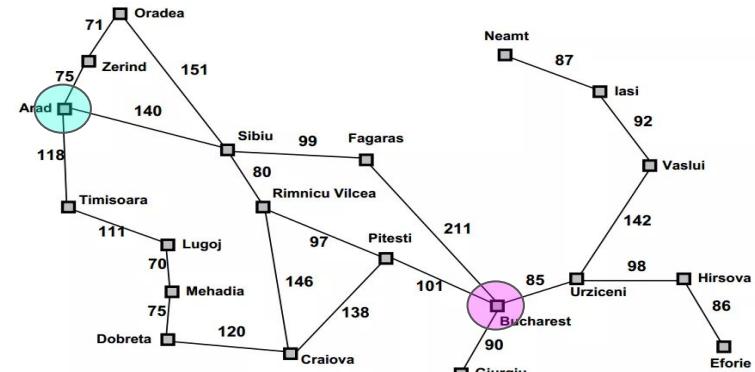
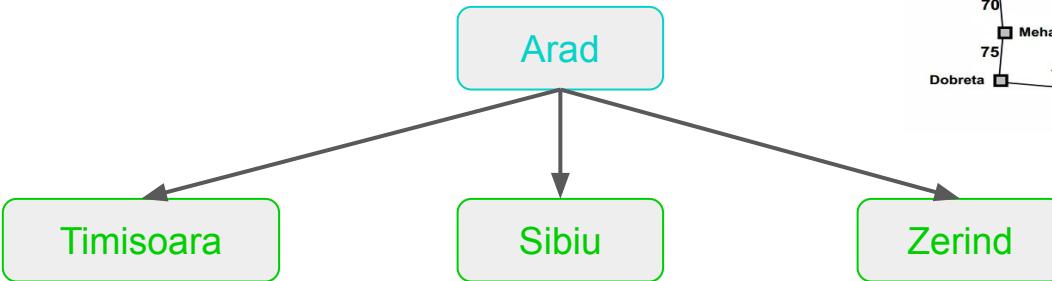
Drawing the search tree

Arad

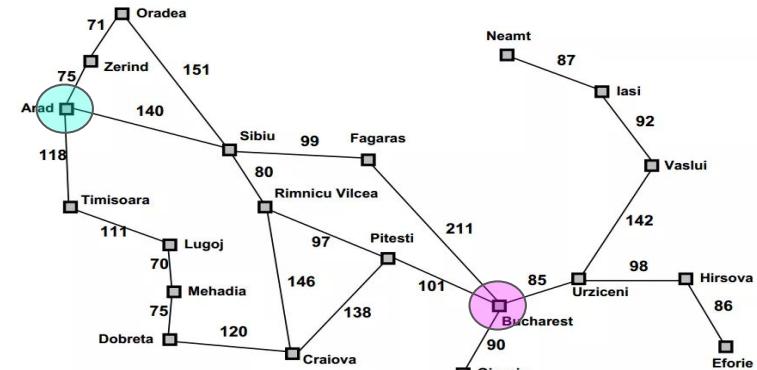
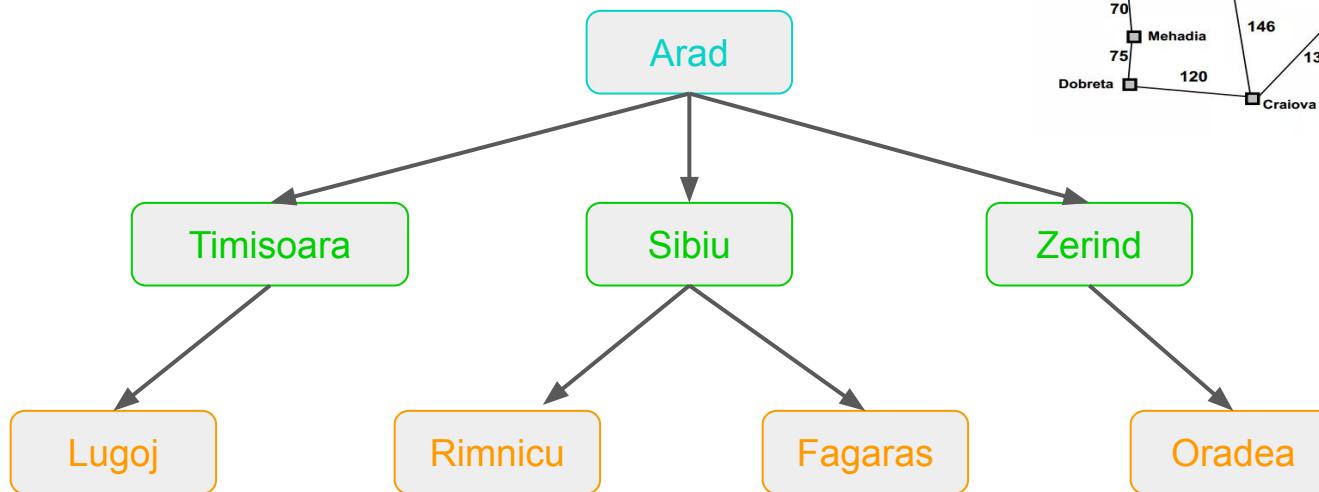


Draw this
on your
paper.

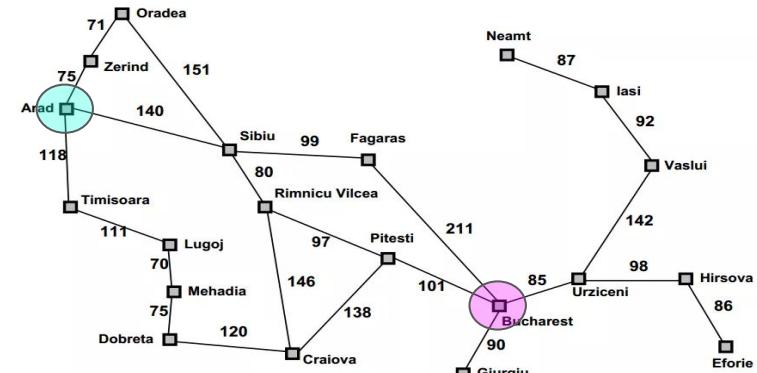
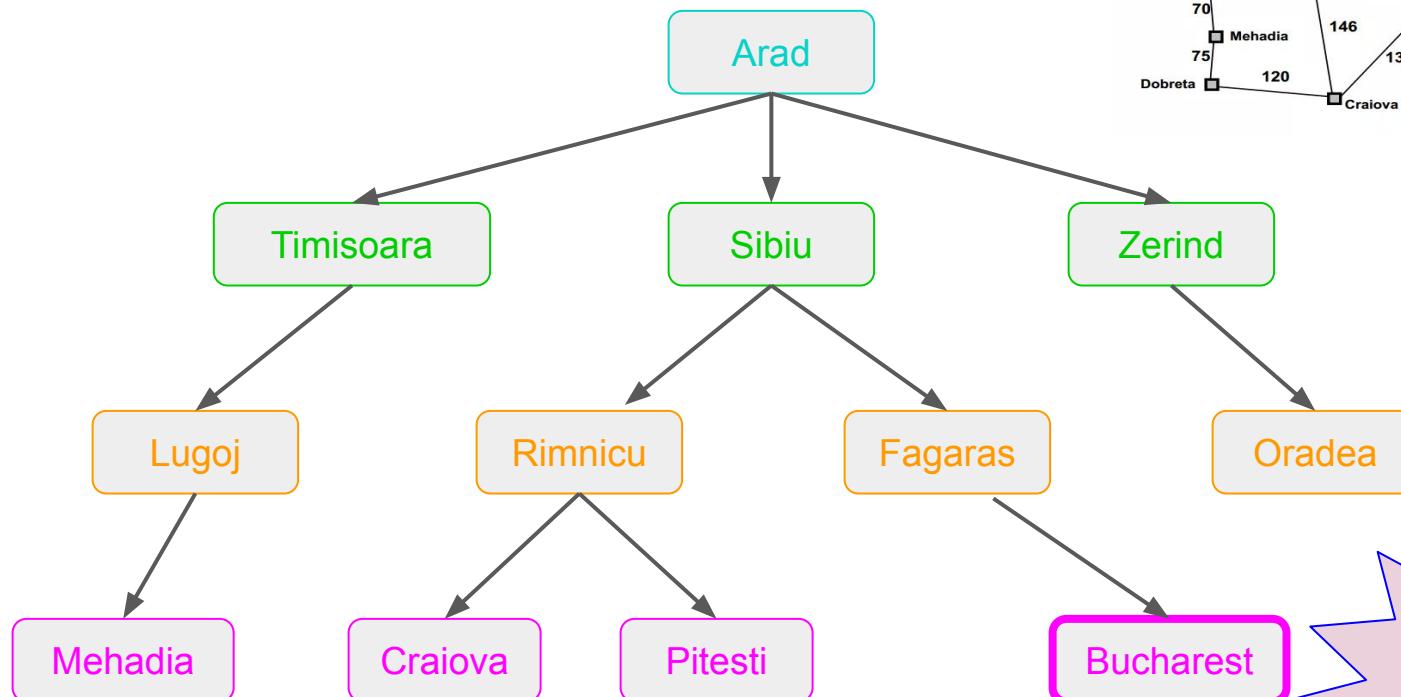
Drawing the search tree



Drawing the search tree

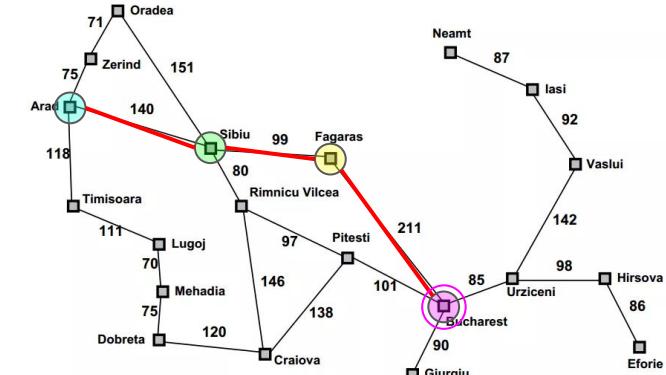
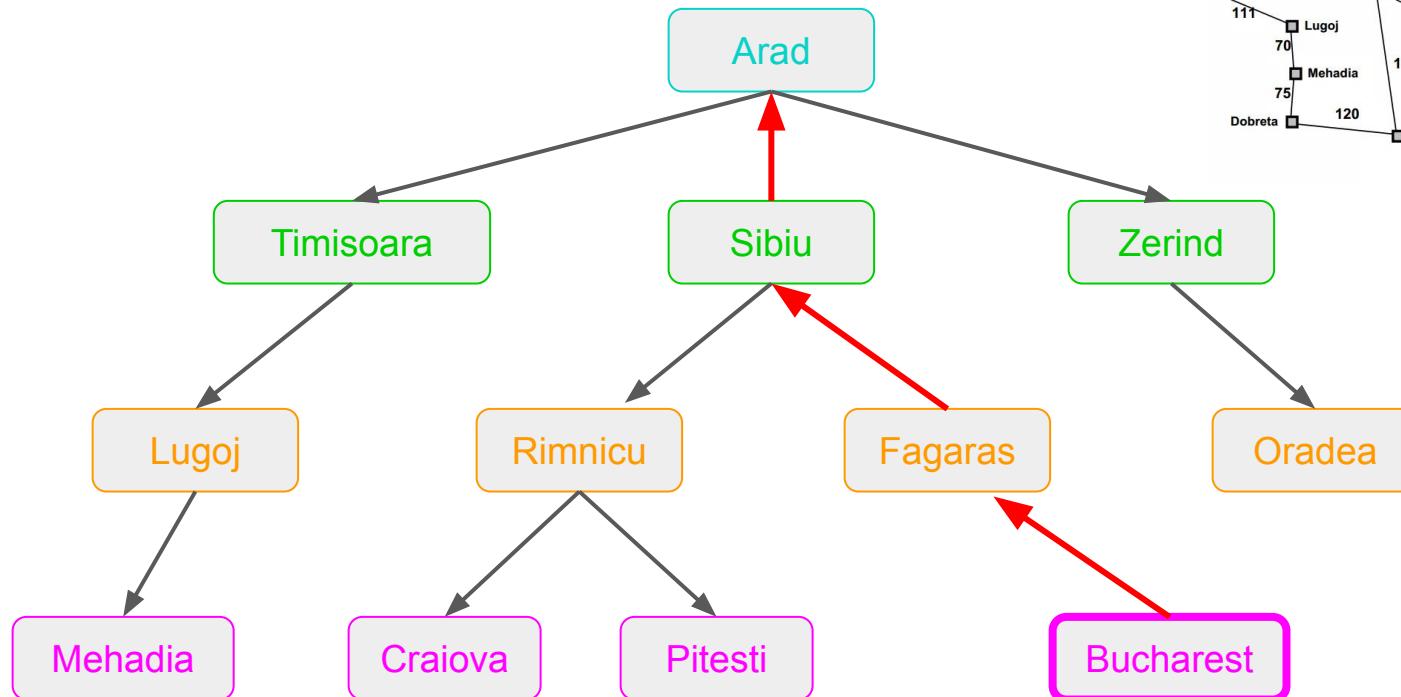


Drawing the search tree



How do we extract the solution?

To extract the solution path,
follow the parent links to the root.



Solving Route Finding Problems

Data structure/representation:

- The input was a labeled, undirected **graph** (like a map for a self-driving car)
- The reasoning algorithm constructed a **tree** as it searched for the answer.
- The output was a **path** (a sequence of nodes) through the graph

Algorithm/reasoning process:

- Find a route as quickly as possible: Breadth First Search (what we used)
- Find the shortest route? Best First Search (a little trickier)



Part IV:

AI Programming Frameworks for Kids



Machine Learning for Kids



Calypso



Machine Learning for Kids

<https://machinelearningforkids.co.uk>

Organize the School Library

Get started

Learn more

- 1** Collect examples of things you want to be able to recognise
- 2** Use the examples to train a computer to be able to recognise them
- 3** Make a game in Scratch that uses the computer's ability to recognise them

Classifier



A **classifier's** job is to sort things into categories



Spam Filters:
Spam, or
not-spam?



Loan Applications:
Low risk, Mid-risk,
High-risk

The screenshot shows a Facebook news feed for a user named Ofra Lior. The feed includes a post from Wix advertising website creation services. Other posts visible include one from Natural Relief Slippers about orthotic support slippers, and another from onlinemeetingnow.com about live or auto webinars.

Ad or movie recommender:
Sports fan? Romantic?
Horror fan? History buff⁵²

Training a Classifier



A **classifier's** job is to sort things into categories

How it Works:

Our Role:

We choose the categories & provide examples of each one.

Algorithms:

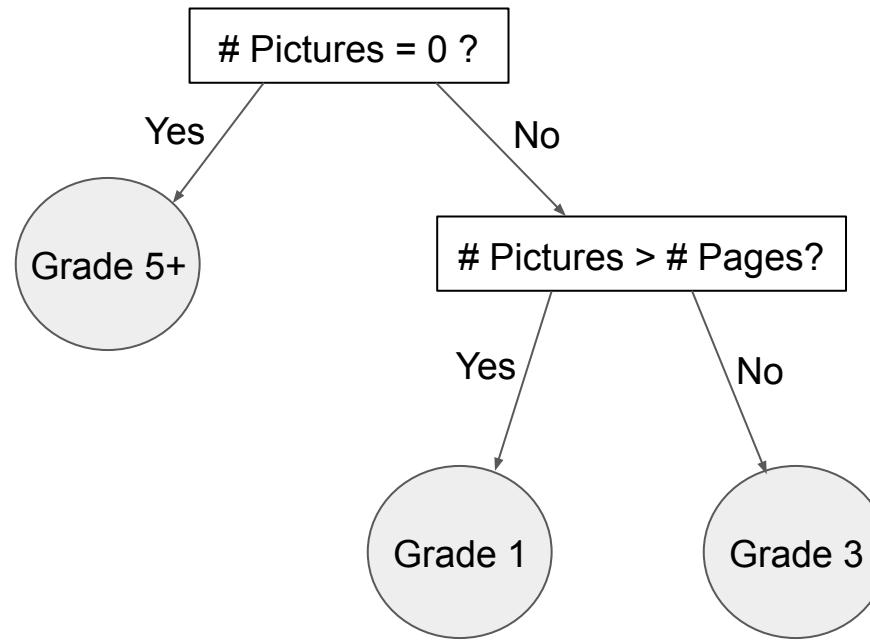
We let the machine learning algorithm figure out how to classify the examples the way we wanted them classified. Then we can test it on new data.

One machine learning method that can be used to do the classification is called a **decision tree**.



Decision tree - a collection of nested if-then statements that classify data.

Decision tree classifier: book reading level



We won't see the decision tree. But if we did it might look like this.

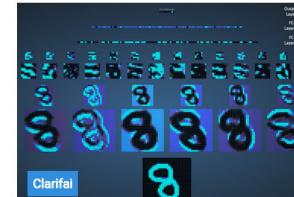
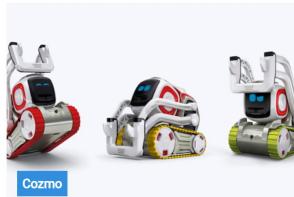
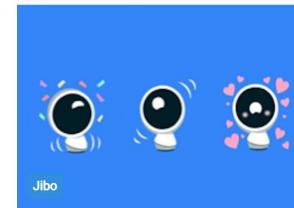
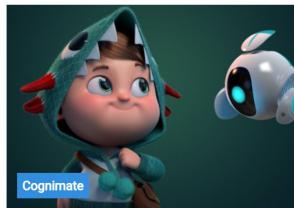


<http://cognimates.me>

AI extensions for Scratch.

Project creator [Stefania Druga](#)

The project was supported by the [Personal Robots Group](#) at MIT Media Lab, directed by [Cynthia Breazeal](#), Associate Professor of Media Arts and Sciences.



[Link to Supplemental Slides](#)



Calypso for Cozmo



- A robot intelligence framework that combines multiple types of AI:
 - Computer vision
 - Speech recognition
 - Landmark-based navigation
 - Path planning
 - Object manipulation
- Rule-based language inspired by Microsoft's Kodu Game Lab
- Teaches computational thinking
- Web site: <https://Calypso.software>

Calypso 0.9.04 - Google Chrome

15-294-A3 Rapid Prototype | 15-494/694 (5 unread) | WAICY 2019 Rubric - Read | Calypso 0.9.04 | +

127.0.0.1:43125/Calypso/index.html

Apps West Mifflin, PA Browse K-12 STE Seniors For Safe CMU Oracle Web S3 Admin Console Home - Workday The Best of the F 15-294-A3 Rapid Cognitive Robot

Stop program State machine view Switch characters Map editor view Stop program Scroll up/down

1

hello cozmo

speech recognition

rules

perception

world map

Cozmo's battery 3.9 volts Cube3 batt 1.24V (48%)

The screenshot shows the Calypso 0.9.04 software interface for programming a robot. On the left, a sidebar lists keyboard shortcuts for various functions like stopping the program, switching characters, and map editing. Below the sidebar are three stacked rule definitions:

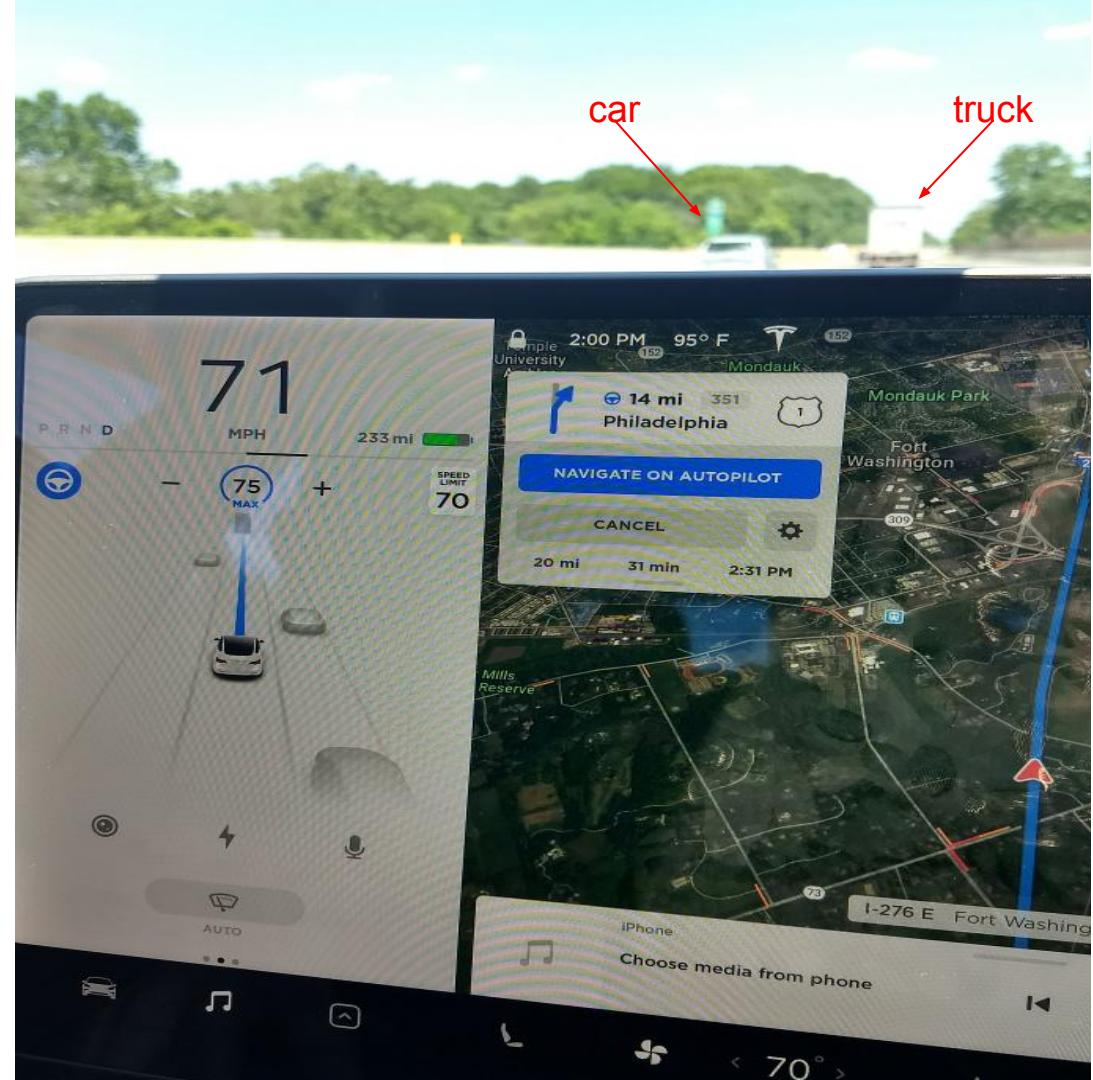
- Rule 1: WHEN "see cube 1" DO "move toward it".
- Rule 2: WHEN "bumped" DO "grab it".
- Rule 3: WHEN "+" DO "switch to page 2".

A pink callout box labeled "rules" points to the rule definitions. On the right side of the interface, there is a graphical representation of a robot arm pointing towards a cube, with the text "hello cozmo" displayed in a large orange box above it. A pink callout box labeled "speech recognition" points to this area. At the bottom left, there is a camera feed showing three cubes labeled "Lightcube 1 id=44", "Lightcube 2 id=45", and "Lightcube 3 id=1". A pink callout box labeled "perception" points to this camera feed. At the bottom right, there is a small world map icon and a pink callout box labeled "world map". The status bar at the bottom shows "Cozmo's battery 3.9 volts" and "Cube3 batt 1.24V (48%)".

Tesla's World Map

Cozmo can be considered “a self-driving car in the palm of your hand.”

At right is an image from a real self-driving car, a Tesla, showing the road and other nearby vehicles on its world map.



Laws of Calypso

First Law of *Calypso*

Each rule picks the closest matching object.

```
when see [cube] [?]
  move toward it
```

A robot on a grassy field sees a cube and moves toward it.

Second Law of *Calypso*

Any rule that can run, will run.

```
when bumped [red cube]
  grab it
when see [red cube]
  move toward it
```

A robot sees a red cube and moves toward it, even though it has already been grabbed. A second script shows the robot bumping into the cube and grabbing it.

Third Law of *Calypso*

When actions conflict, the earliest wins.

```
when see [red cube]
  move toward it
when see [blue cube]
  move toward it
```

A robot sees a red cube and moves toward it, even though it has also seen a blue cube. A second script shows the robot seeing a blue cube and moving toward it.

Fourth Law of *Calypso*

An indented rule can run only if its parent's action succeeds.

```
when bumped [green cube]
  do [grab it +]
  when scored [yellow score] > [0 points]
    do [play beep probe +]
```

A robot bumps into a green cube, which triggers a script to grab it. This script then checks if the yellow score is greater than 0. If it is, the robot plays a sound. The robot ends up with a score of 5.

Fifth Law of *Calypso*

On every cycle, earlier actions affect later rules.

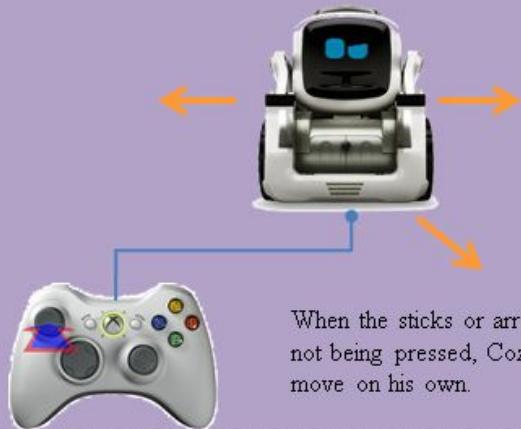
```
when bumped cube
  glow it blue
when see blue
  grab it
when got blue
  switch to page 2
```

A robot triggers three overlapping scripts. The first script makes the cube glow blue. The second script grabs the cube. The third script switches the page. The robot ends up with a score of 2.

Calypso Idiom Catalog

Let Me Drive

Use the sticks or arrow keys to drive Cozmo.

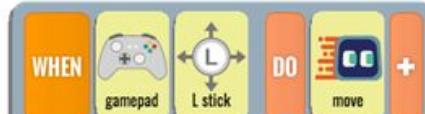


When the sticks or arrow keys are not being pressed, Cozmo is free to move on his own.

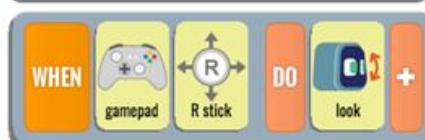
Let Me Drive



Use the sticks or arrow keys to drive Cozmo. The shoulder buttons control the lift.



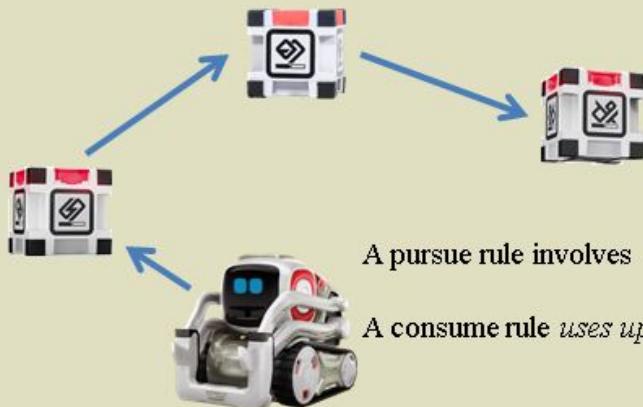
Allow control using only left stick (drive) or right stick (head angle).



Calypso Idiom Catalog

Pursue and Consume

Make Cozmo extinguish all the red cubes.



A pursue rule involves *motion*.

A consume rule *uses up* the object.

Pursue and Consume

Pursue rule



Consume rule



General Form:

WHEN see *thing* DO move toward it

WHEN bumped *thing* DO consume it

Calypso Curriculum

1. Carnegie Mellon Curriculum
 - Click on the “Curriculum” link at <https://Calypso.software> to visit the main curriculum page
 - 12 modules explain the features of Calypso and how to use them
 - Designed primarily to help teachers learn Calypso
2. New Brighton Curriculum
 - Click on the “New Brighton” link from the main curriculum page
 - Project-based approach designed for classroom implementation
 - Work in progress; more material will be available in July 2019
3. ReadyAI AI-IN-A-BOX Tutorials
 - Instructional videos available on ReadyAI’s YouTube channel
 - Click on the “ReadyAI” link on the main curriculum page

Other Resources & Next Steps

- AI4K12.org Resources Directory
- Join the AI4K12 mailing list: send mail to ai4k12@aaai.org and ask to be placed on the list.
- “Teaching AI in K-12” Symposium, Nov. 8-9, 2019 in Arlington, VA (AAAI Fall Symposium Series).
- [Hour of Code Build an Alexa Skill](#) (1hr 5 mins)
<https://learn.amazonfutureengineer.com/alexafact-skill/>

Let's continue this conversation
in our **session tomorrow, July 9 @ 10am**



How to Teach AI Across K-12

Location: West 212C

Questions?

Thank
You!

