Introduction
to
Theory of Computing

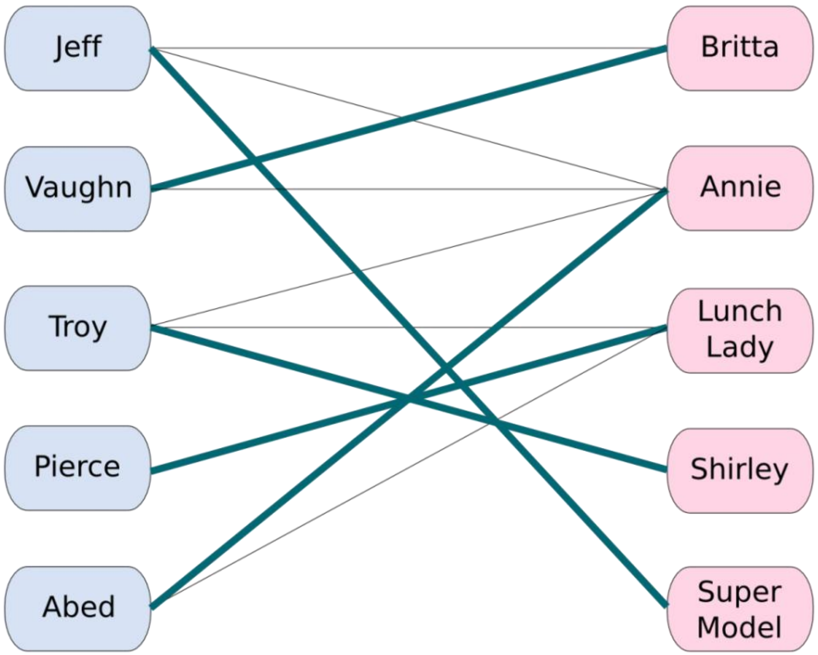
Peter Kogge

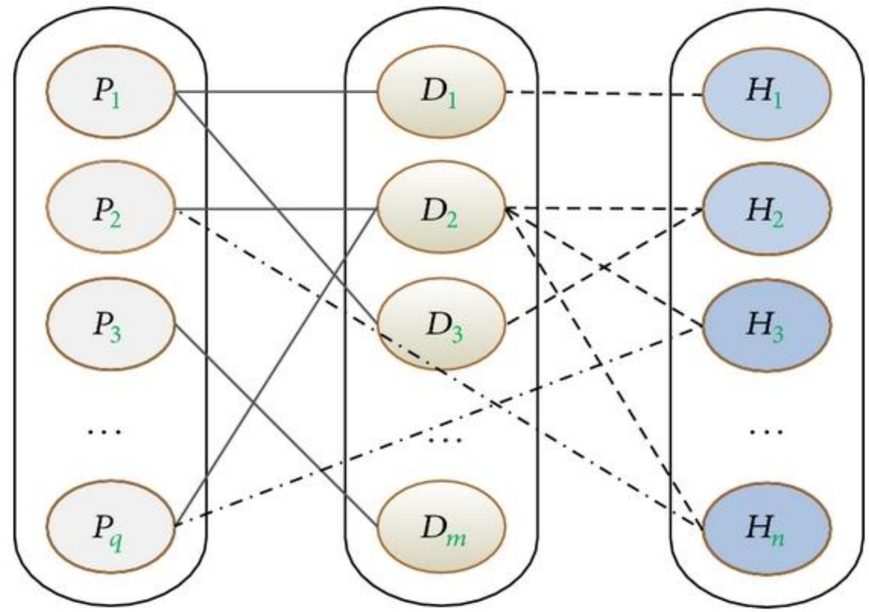
From The Dictionary

- **Theory**: system of ideas intended to explain something
  - Especially based on general principles independent of thing being explained
- **Computing**: (from Latin) to come together (to settle an account)
- **Philosophy**: study of the fundamental nature of knowledge, reality, and existence

This Class: a study of the fundamental nature and limits of computation

The Matching Problem

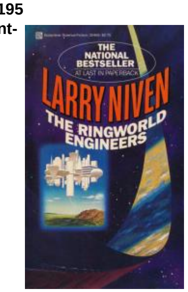




https://www.researchgate.net/profile/Simon_Poon/publication/279225880/figure/fig1/AS:301952400412694@1449002470456/Tripartite-graph-structure-of-TCM-Here-instances-of-different-objects-are-represented.png

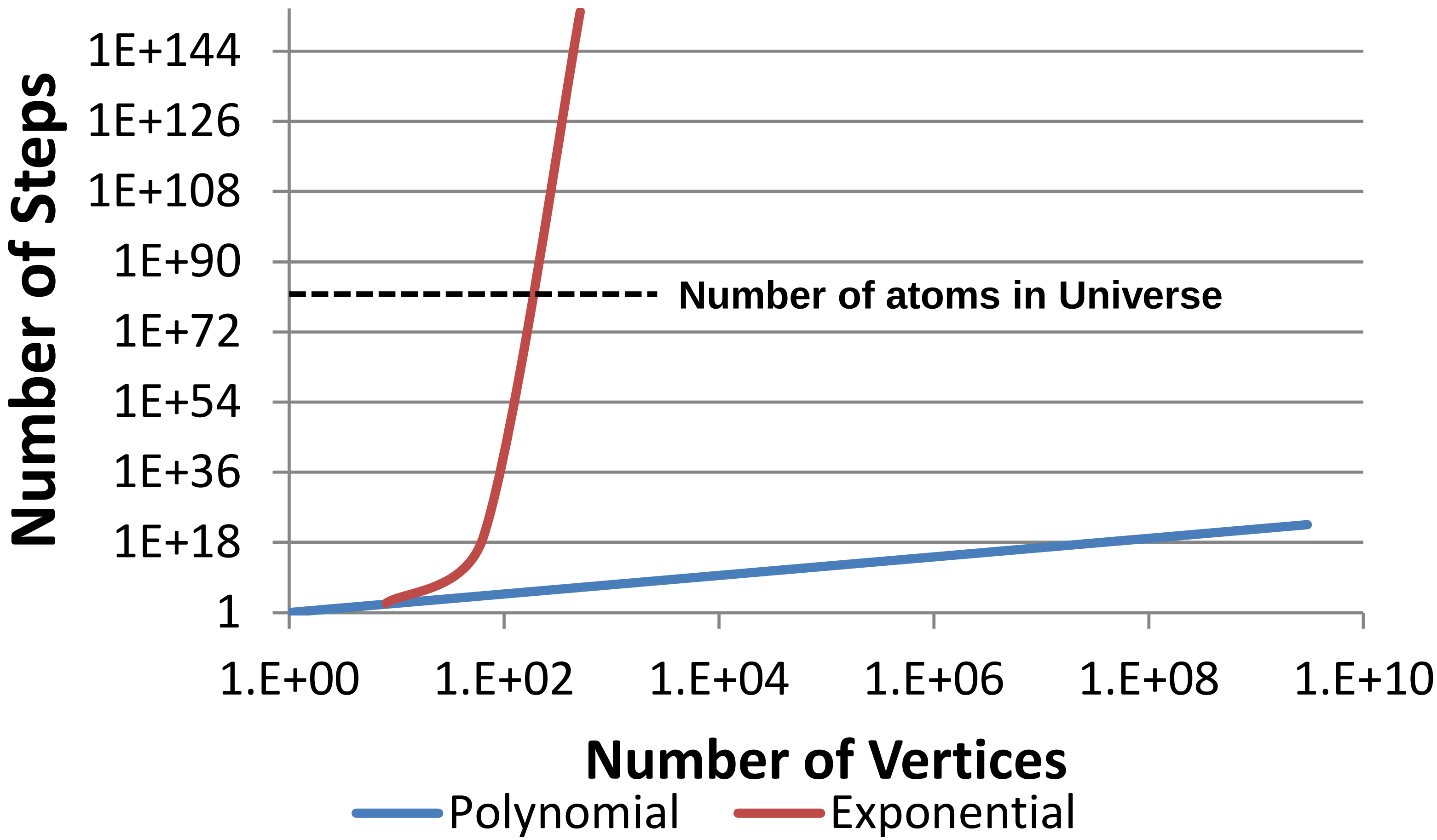
Bipartite Matching: a.k.a
2-Gender marriage problem:
Solvable in "polynomial" time
$O(V^{2.4})$ or $O(E^{10/7})$

3-Gender marriage problem:
Known to be "NP-Hard"
Probably exponential: $O(2^V)$



It continues to be $O(2^V)$ for more than 3 genders

Does It Matter to a Match-Making Computer?



Problems requiring exponential # of steps are *HARD*!!

Another Example: Boolean Satisfiability (SAT)

- **SAT: Is there a assignment of values to variables in a Boolean expression making it true**
- **Example: (~x v y) & (x v y) & (x v ~y)**
  - **x=1, y=1 makes expression true**
- **Example: (~x v y) & (x v y) & (x v ~y) & (~x v ~y)**
  - **No assignment of values make expression true**
- **Trivial algorithm: create truth table to test all possible cases ($2^{|V|}$)**
- **Can we do better?**

SAT is perhaps *THE* fundamental problem in computing!

Key Questions for Such Problems

- What is it that we “count” when discussing “how hard” a problem is?
- Are there variations in our basic model of computing?
- What classes of problems are solvable by each model?
- Why are some problems “hard”
- Is there a “universal problem”?
- Is there a “universal computing model”?
- Are there problems that are intrinsically hard even on a universal computer?

Computing Theory In Perspective

- **Architecture**: Design of inhabitable structures
- **Organization**: Functional interaction of Key Subsystems
- **Design**: Implementation in a real technology
- **Execution Model**: How a computer executes a program
- **Algorithm**: Step-by-step description of a computation to solve some problem
- **Programming Model**: Expression of Algorithm in form that executes on a real computer

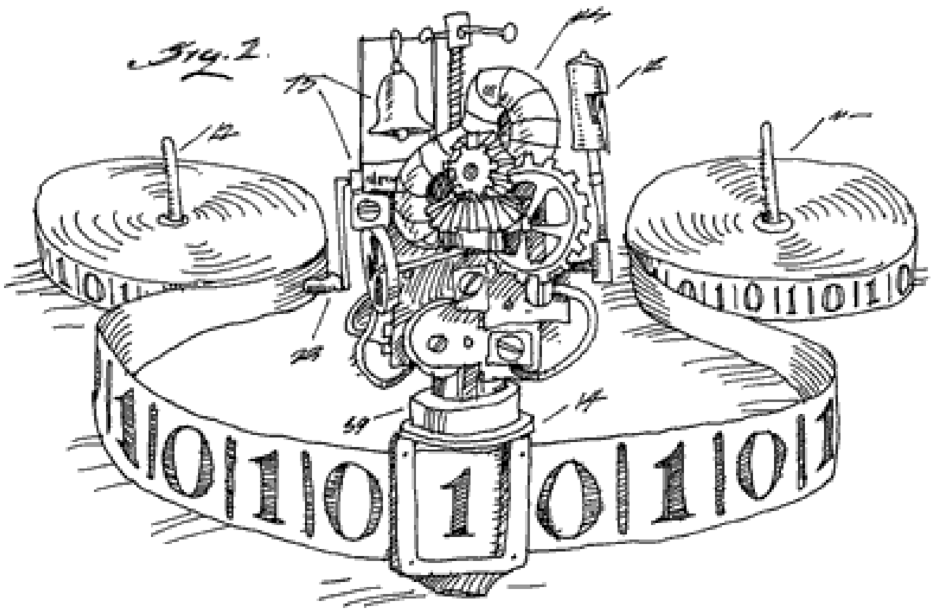
Particularly Relevant

- **Abstract Machine: simplified model of a class of computer systems**
  - **Today's computers are all von Neumann**
- **Automata Theory: formal definitions of 3 basic classes of abstract machines**
- **Complexity Theory: what makes some problems intrinsically hard and others simple?**
- **Computability Theory: what problems can be solved by algorithms executable on what classes of automata**

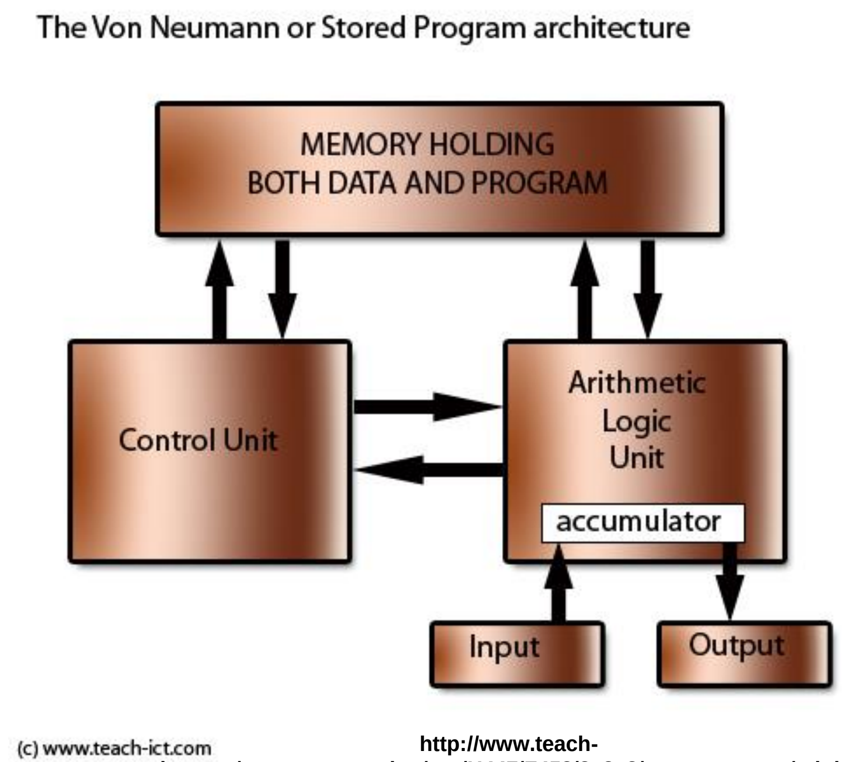
Classes of Automata

- **Automata**: (Greek for "self-acting") Device that
  - Accepts strings of input data one character at a time
  - Generates an output (at some point)
  - Fixed set of **states** it can be in
  - Follows a stored set of **transition rules**
    - For each input & current state, what is new state
- **Finite Automata**: No memory other than state
  - **Deterministic** (**DFA**): transition rules id at most only 1 new state
  - **NonDeterministic** (**NFA**): multiple transitions possible
- **Push Down Automata (PDA)**: Stack available of intermediate results
- **Turing Machines** (**TM**): Infinite tape available for intermediate results

Today: Turing Machines Rule!



http://www.worldofcomputing.net/theory/turing-machine.html



http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_3/vonn_neuman/miniweb/images/von_newmann_architecture.jpg

Turing Machine:

1. Read a character from a tape
2. Get operation from table lookup
3. Write a character to tape
4. Move tape left or right
5. Repeat

Von Neumann Architecture:

1. Read instruction from memory
2. Read a datum from memory
3. Do an operation
4. Determine next instruction
5. Repeat

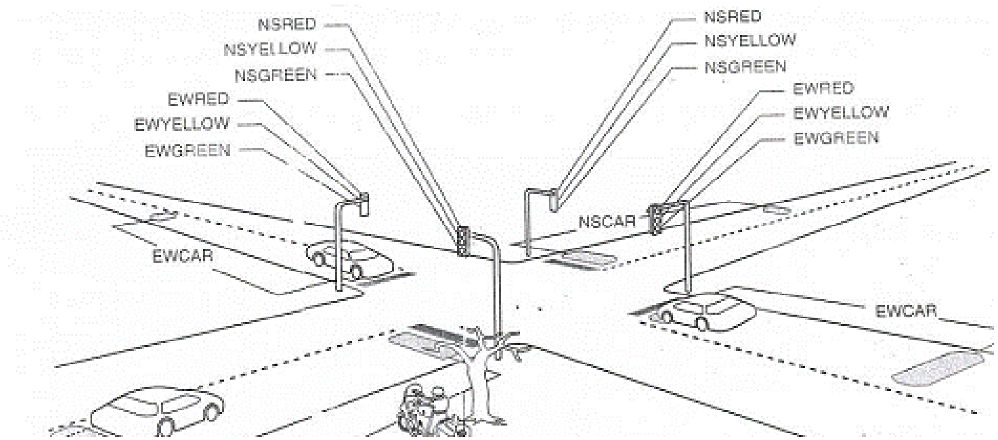
More on Language Definitions

- **Alphabet: set of characters that can be used in a program**
- **Symbol: member of an alphabet**
- **Syntax: formal rules for valid substrings**
- **Grammar: expression of syntax rules**
- **Semantics: formal description of what valid strings mean in terms of algorithm execution**

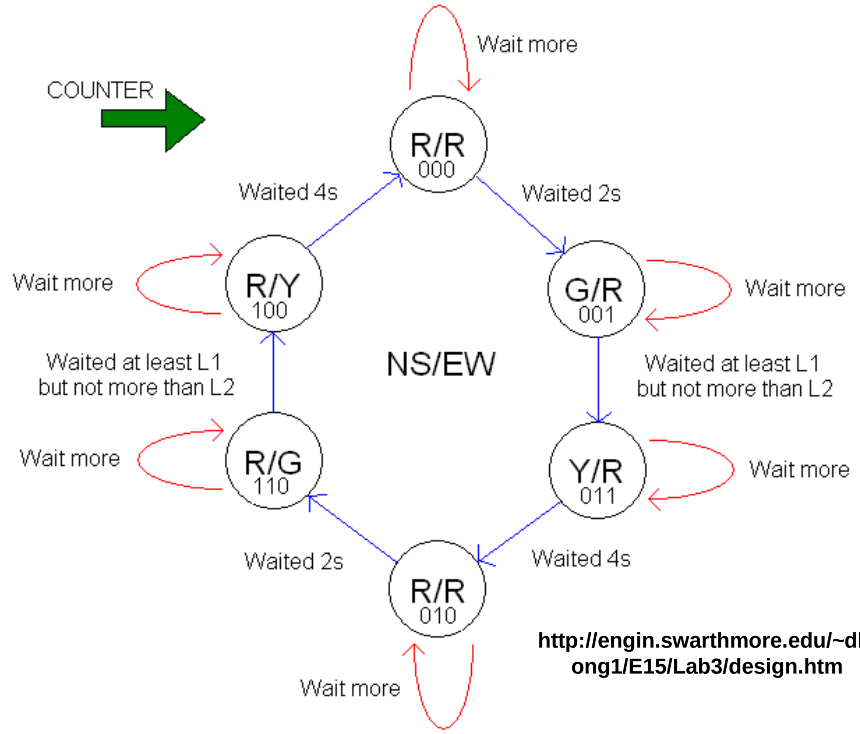
Classes of Languages

- **Language Recognition: transition rules can be generated to**
  - **Say YES for any input string if in that language**
  - **Say NO for any input string not in that language**
- **Regular Expressions: can be recognized by FA**
- **Context Free: can be recognized by PDA**
- **Context Sensitive and Unrestricted: can be recognized by a Turing Machine**

A Simple Finite Automata



State Diagram Representation



http://engin.swarthmore.edu/~dluong1/E15/Lab3/design.htm

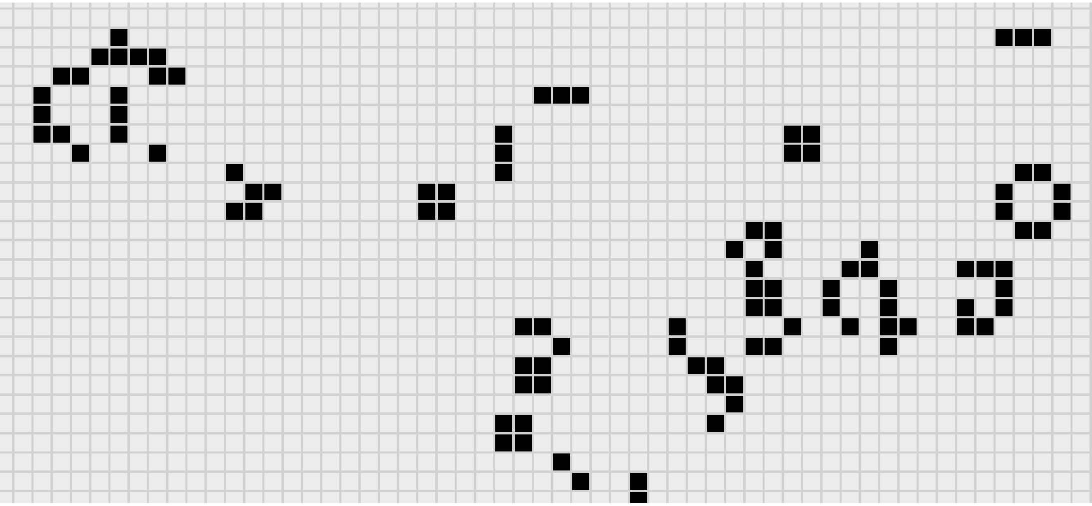
Transition Function Representation

δ(current_state, input) = new_state

e.g.

δ(G/R, ewcar) = Y/R

A “Multi-dimensional” Automata
Conway’s “Game of Life”



http://www.math.cornell.edu/~lipa/mec/lesson6.html

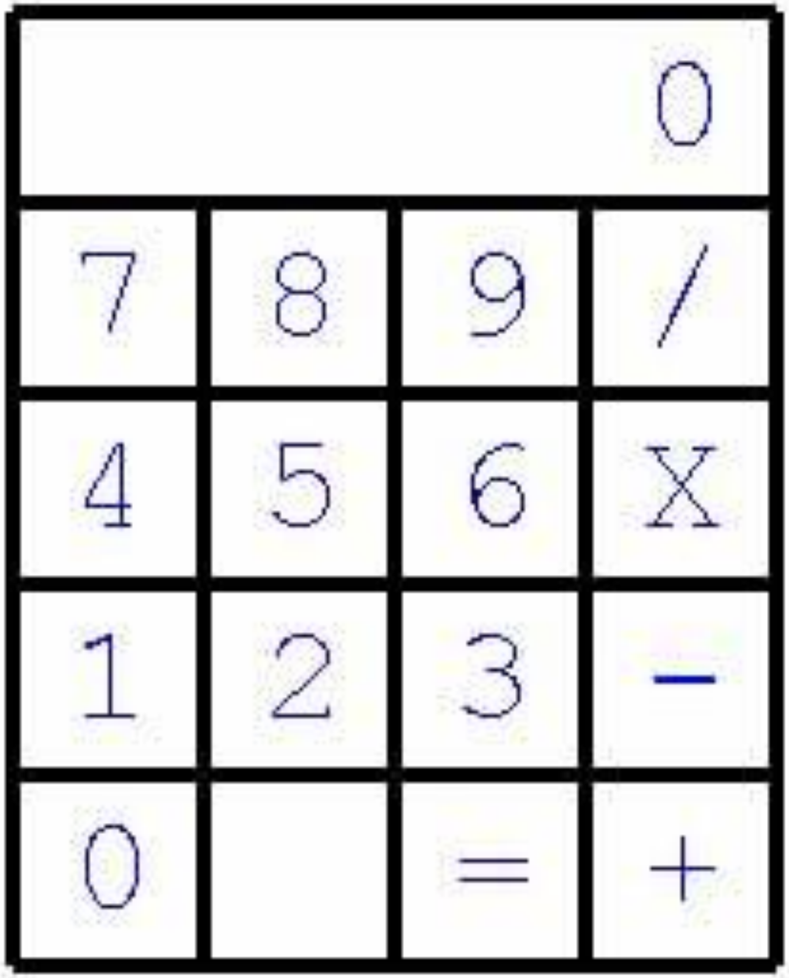
A “live” cell dies if
- Less than 2 live neighbors
- Greater than 3 live neighbors

A dead cell becomes “alive” if
- exactly 3 live neighbors

Rationale for these choices:

- No initial pattern for which there is a simple proof that the population can grow without limit.
- There should be initial patterns that apparently do grow without limit.
- There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in the following possible ways:

  Fading away completely (from overcrowding or too sparse)
  Settling into a stable or oscillating pattern.



- **What are valid inputs?**
- **What is output?**
- **What is operation?**
- **What is “language” that is accepted?**
- **How much memory is here?**

Describe the Calculator's Operation

If you press	The calculator does the following
0,1,2,3,4,5, 6,7,8,9	
+,-,*,/	
=	

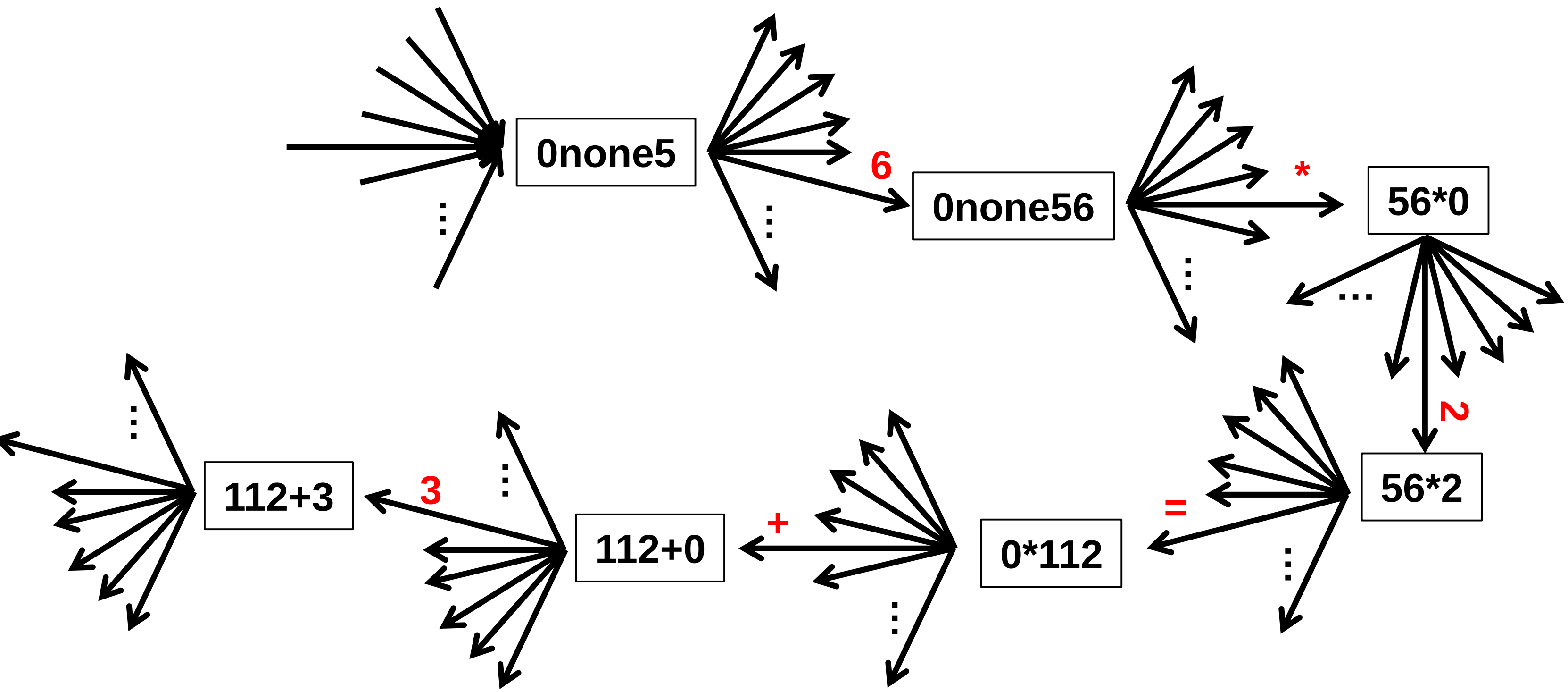
Food for Thought: What happens if you press a digit after "="?

Describe the Calculator's Operation

Notional "Transition Table"	
If you press	The calculator does the following
0,1,2,3,4,5, 6,7,8,9	Shift Display left and insert digit
+,-,*,/	• Remember the operation • Remember current displayed # • Reset display # to 0
=	Compute: Remembered # "operation" Displayed #, and display result

Food for Thought: What happens if you press a digit after "="?

A Subset of a “State Diagram”



ixj is “state” where “i” is saved #, j is displayed #, and “x” is last operator

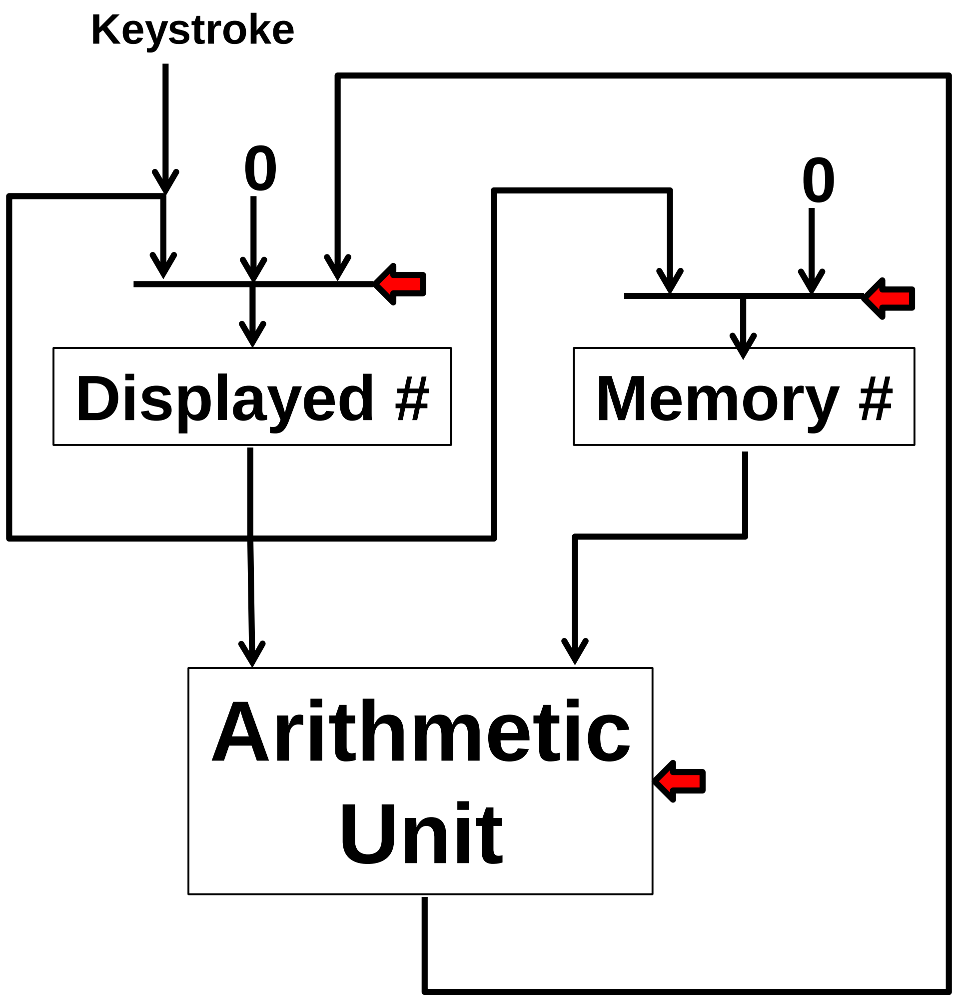
“none” represent no operation pushed since power on

Symbol on edge is a button push

Alternative: Transition Function

- δ is denoted as the **Transition Function**
- δ(current_state, input) = new_state
- If state represented as "ixj" where
  - "i" is last # saved in calculator
  - "j" is number currently being displayed
  - "x" is last operation button pushed
- Then some sample entries for δ include
  - δ(2+3, 5) = 2+35 (push 5 onto right of displayed #)
  - δ(2+35, "="") = 0=70
  - Many, many more, but finite # of them

What's the "Abstract Machine?"



SavedOperation

Keystroke	State 1		State 2		...		State N	
	Operation	Next State	Operation	Next State			Operation	Next State
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
+								
-								
*								
/								
=								

Control Signals

State = concatenate(Displayed#,memory#,SavedOperation)

Can We Compute ALL Expressions

- ❑ Can we compute **12*34 + 56 / 78 =** ?
- ❑ Can we compute **12*34 + 45*67 =** ?
- ❑ What is the computable language?

<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 Symbol

<number> → <digit> | <number> <digit>

<op> → + | - | * | /

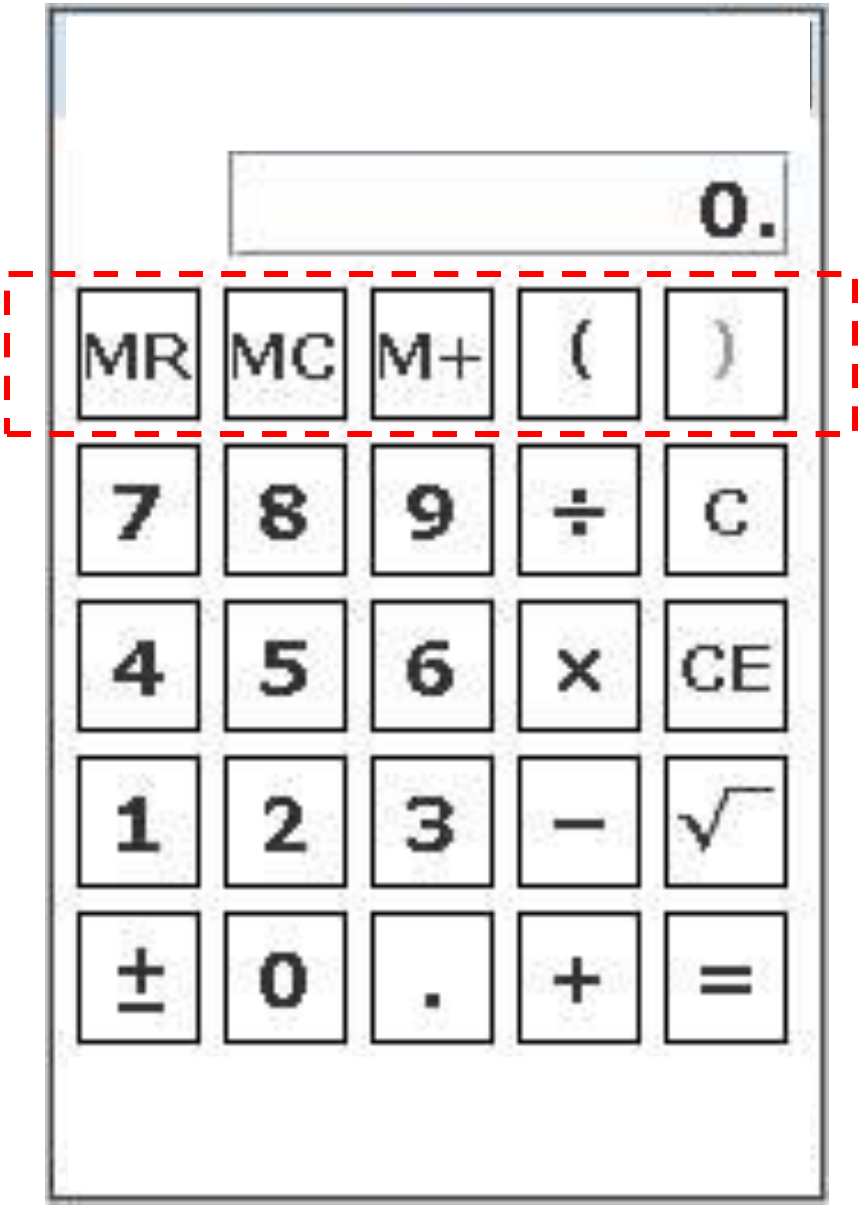
<expression> → <number> | Meta-symbol

<expression> <op> <number> =

Non-terminal

This kind of grammar notation is often called “BNF”

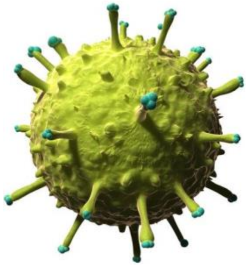
Does This Change Anything?



- What are valid inputs?
- What is output?
- What is “language” that is accepted?
- How much memory is here?

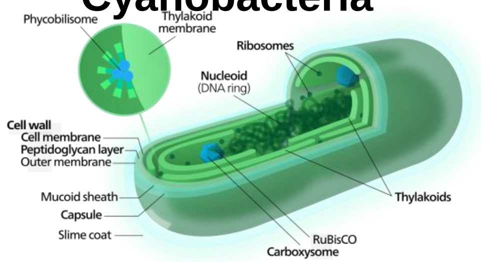
More Food for Thought?

Virus



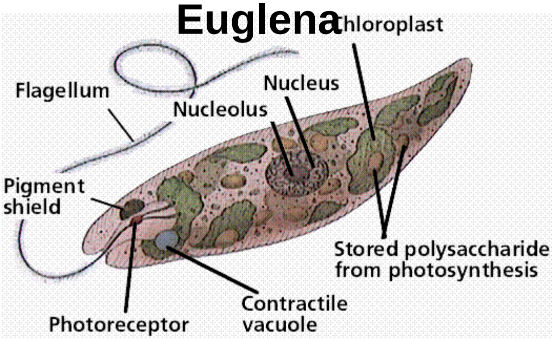
http://pop.h-cdn.co/assets/15/23/980x490/landscape-1433433160-virus-swineflu.jpg

Cyanobacteria



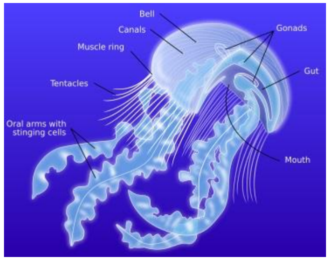
https://upload.wikimedia.org/wikipedia/commons/thumb/5/58/Cyanobacterium-inline.svg/2000px-Cyanobacterium-inline.svg.png

Euglena



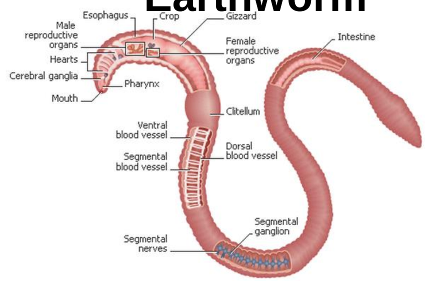
http://www.schursastrophotography.com/roboimages/visonlogic/onepixeleye/euglena.gif

Jellyfish



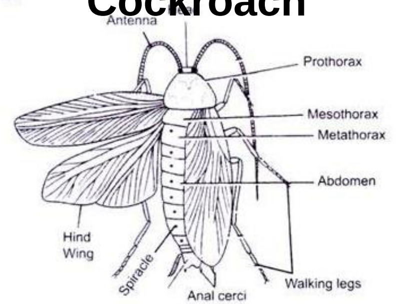
https://upload.wikimedia.org/wikipedia/commons/thumb/6/6b/Anatomy_of_a_jellyfish-en.svg/2000px-Anatomy_of_a_jellyfish-en.svg.png

Earthworm



https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiOhcrfncnOAhVITSYKHSFAAA0QjRwIBw&url=http%3A%2F%2Fearthwormresources.weebly.com%2Freproduction-and-development.html&psig=AFQjCNGKeHGRIYTCQIh2BSAKXPjAIs-PLQ&ust=1471550174895324

Cockroach



http://www.biologydiscussion.com/wp-content/uploads/2014/09/clip_image002_thumb18.jpg

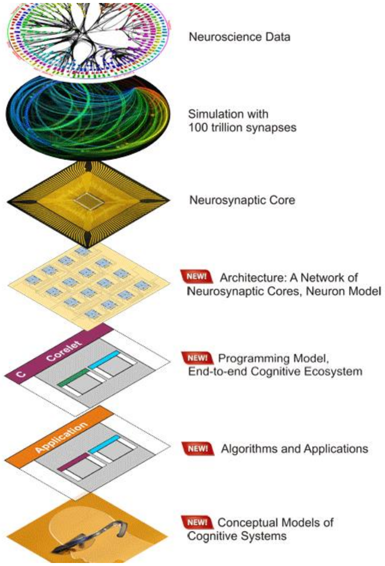
https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjp2dbFn8nOAhVDMSYKHTY5AuQQjRwIBw&url=http%3A%2F%2Fscicurious.scientopia.org%2F2013%2F07%2F19%2Ffriday-wierd-science-mopey-mice-pee-their-feelings%2F&psig=AFQjCNGvZ6Ld_N19E2vUByxJe6OxL2HkRQ&ust=1471550607879756

https://images-na.ssl-images-amazon.com/images/G/01/img15/pet-products/small-tiles/23695_pets_vertical_store_dogs_small_tile_8._CB312176604_.jpg

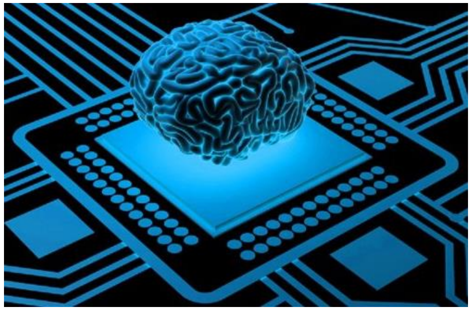
http://i.dailymail.co.uk/i/pix/2011/11/01/article-0-0C1D461000000578-689_468x482.jpg

http://wiinnebago.com/wp-content/uploads/2013/06/HR-People.jpg

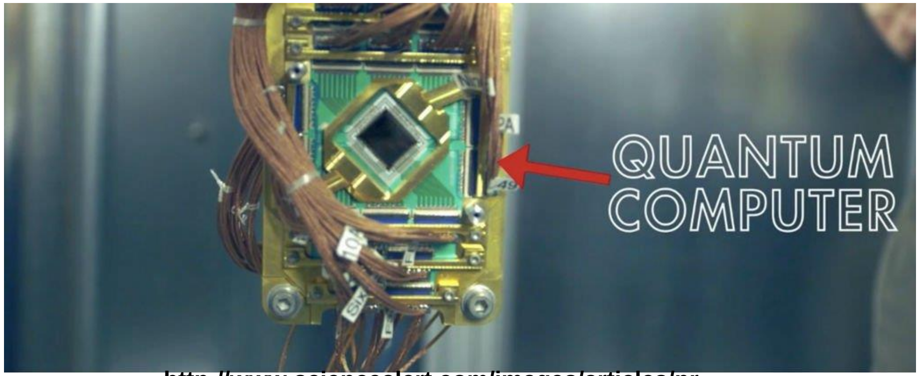
What's Next?



http://3.bp.blogspot.com/-y4yiZzcyyDk/UgO_gBocm-I/AAAAAAAAAnmA/ISvU3OkdgEo/s1600/synapsestack.png



https://media.licdn.com/mpr/mpr/AAEAAQAAAAAAAAXLAAAAJDI5N2M1ZmJmLWJiODQtNDFkZC1iZTRmLWY4N2ViNDA1MmE1ZQ.jpg



http://www.sciencealert.com/images/articles/processed/quantum-computer_1024.jpg

https://www.skotcher.com/wall/12ea2fc3e99298fdd4d12a13c69a2c56/2001-a-space-odyssey-hurricane-monolith-planets-starlight.jpg