Cellular Automata

and

Turing Universality in the Game of Life

by Paul Rendell

presented by

David Thue

Outline

- Cellular Automata
- The Game of Life
- Turing Machines
- A Turing Machine in the Game of Life

Interactive Motivation

- Every third person:
  - ***Please stand up***

Interactive Motivation

- Everyone:
  - ***Count the <u>total number of people standing</u> in:***
    - ***<u>your row</u>*** (*side to side of the room*), *and*
    - ***<u>your column</u>*** (*front to back of the room*)

      ... <u>including yourself</u>

Interactive Motivation

- Everyone:
  - ***Count the <u>total number of people standing</u> in:***
    - ***<u>your row</u>*** (*side to side of the room*), *and*
    - ***<u>your column</u>*** (*front to back of the room*)

      ... <u>including yourself</u>
- When I say “Go”...
  - ***if the number you counted was:***
    - **Odd:** ***Sit down* (*or stay seated*)**
    - **Even:** ***Stand up* (*or stay standing*)**
    - **Zero:** ***Stand up***

Cellular Automata

- used as a framework for investigating the behavior of complex, extended systems

Cellular Automata

- used as a framework for investigating the behavior of complex, extended systems
- Given:
  - an initial configuration of the parts of the system

Cellular Automata

- used as a framework for investigating the behavior of complex, extended systems
- Given:
  - an initial configuration of the parts of the system
  - a set of rules describing how each part changes over time, based on parts nearby

Cellular Automata

- used as a framework for investigating the behavior of complex, extended systems
- Given:
  - an initial configuration of the parts of the system
  - a set of rules describing how each part changes over time, based on parts nearby
- Observe the behaviour of the system over time.

Cellular Automata

- used as a framework for investigating the behavior of complex, extended systems
- Given:
  - an initial configuration of the parts of the system
  - a set of rules describing how each part changes over time, based on parts nearby
- Observe the behaviour of the system over time.
  - Conceived by John von Neumann in the 1940s, toward designing a theoretical self-replicating machine.

Cellular Automata

- In general, Cellular Automata have:
  - a *k*-dimensional set of cells
    - *k* is a positive integer
  - a positive number of states that can describe each cell
  - a set of rules describing how the state of each cell changes over time, based on the states of nearby cells

Collision-based Computing

http://www.safetycenter.navy.mil/photo/archive/photo113p.htm

http://www.shirleyannparker.com/enewsletters/tcc_mar02.htm

Cellular Automata

- Uses:
  - Modelling Nature
    - crystals, fluid flow, insect colonies, ecosystems, traffic, city growth
  - Making Machine Makers
    - build a machine (*i.e.*, a configuration of states) where:
      - Input: A configuration of states that *specify* another machine
      - Output: A configuration of states that *is* the specified machine

Demo

von Neumann's Universal Constructor

Cellular Automata

- von Neumann's constructor *works*, but:
  - 29 possible states per cell
  - complicated state transitions
- *Surely there's an easier solution...*
- In 1970, John H. Conway found one.

The Game of Life

John H. Conway

- Born on Dec. 26, 1937 in Liverpool, England
- From BA to Professor at Cambridge, 1959-1983
- Became the Jon von Neumann Chair of Mathematics at Princeton, 1986.

http://www.adeptis.ru/vinci/m_part3_3.html

John H. Conway

- Research:
  - Group Theory
  - Surreal Numbers
  - Knot Theory
  - Game Theory
  - Quadratic Forms
  - Cellular Automata

http://www.adeptis.ru/vinci/m_part3_3.html

John H. Conway

- Awards and Honours
  - Fellow of the Royal Society of London
  - Polya Prize of the London Mathematical Society
  - Frederic Esser Nemmers Prize in Mathematics
  - Leroy P Steele Prize for Mathematical Exposition
  - and more...

http://www.adeptis.ru/vinci/m_part3_3.html

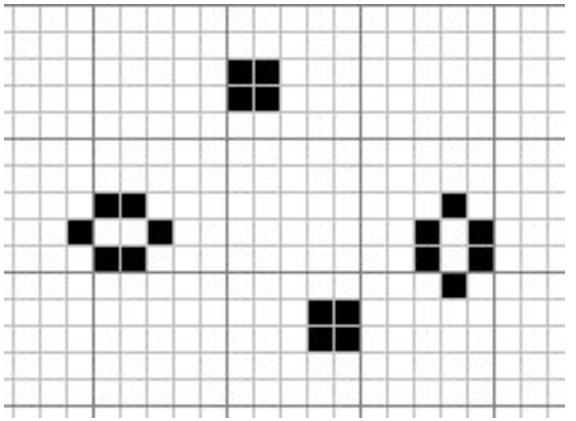
The Game of Life

"[*A*]*fter the rejection of many patterns, triangular and hexagonal lattices as well as square ones, and of many other laws of birth and death, including the introduction of two and even three sexes*[*, a*]*cres of squared paper were covered, and he and his admiring entourage of graduate students shuffled poker chips, foreign coins, cowrie shells, Go stones or whatever came to hand, until there was a viable balance between life and death."*

- R. K. Guy

The Game of Life

- The World:
  - an (optionally) infinite 2D grid of square cells
- The States:
  - Alive: the cell is filled
  - Dead: the cell is empty

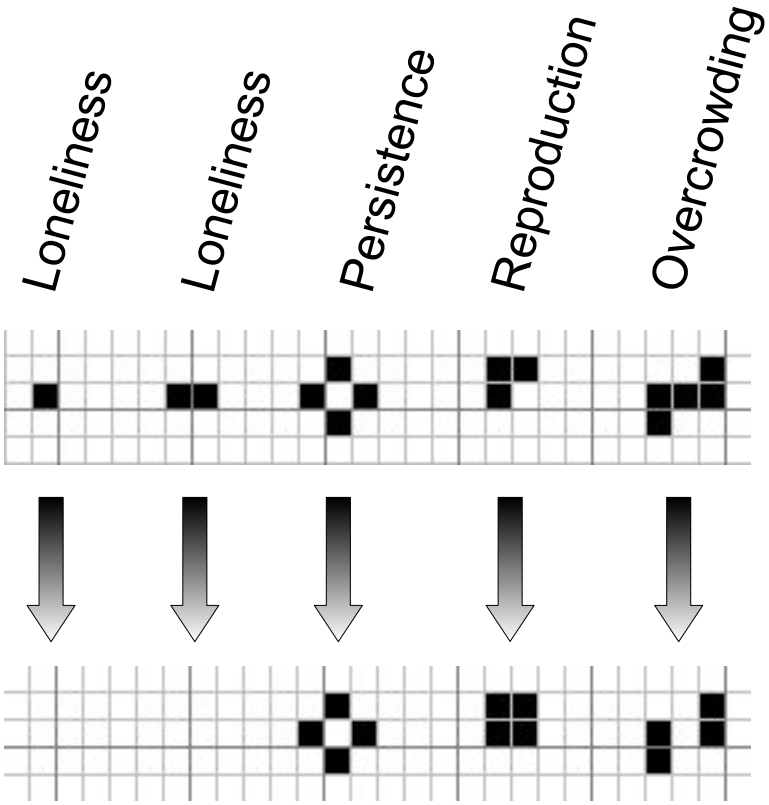


The Game of Life

- The Rules:
  - ***at every time step, for every cell,***
    - ***consider the 8 neighbours around it***
  - ***if the number of living neighbours is:***
    - **< 2:** ***The cell dies* (*or stays dead*)** **- loneliness**
    - **= 2:** ***The cell stays the same.*** **- persistence**
    - **= 3:** ***The cell is born* (*or stays alive*)** **- reproduction**
    - **> 3:** ***The cell dies* (*or stays dead*)** **- overcrowding**

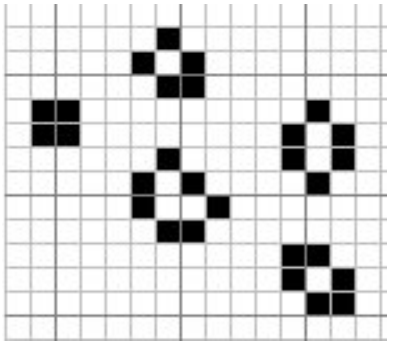
The Game of Life

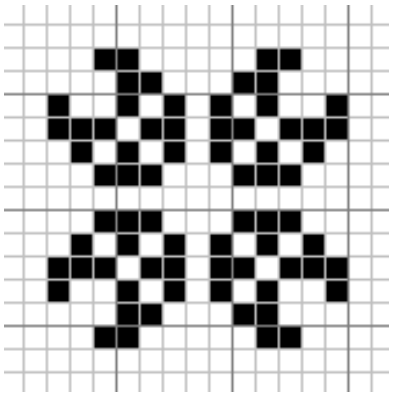
- n: number of live neighbours
  - **n < 2:** ***Die by Loneliness***
  - **n = 2:** ***Stay by Persistence***
  - **n = 3:** ***Live by Reproduction***
  - **n > 3:** ***Die by Overcrowding***



The Game of Life

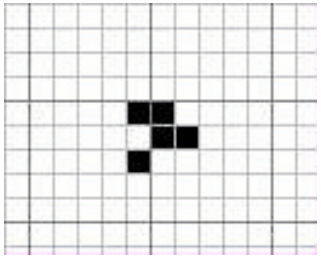
- Pattern:
  - a unique arrangement of living cells
- Still Life:
  - a pattern that does not change in time
- Oscillator:
  - a pattern that continuously alternates between two or more unique arrangements



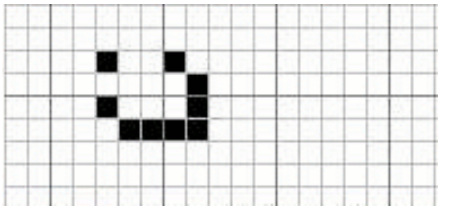


The Game of Life

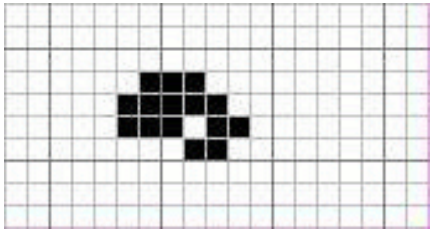
- Space Ship:
  - an oscillator that moves across space over time
- Kinds of Space Ships:
  - Gliders:



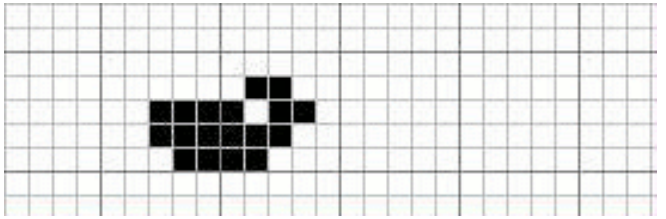
  - Light Weight Space Ship (LWSS)



  - Medium Weight Space Ship (MWSS)

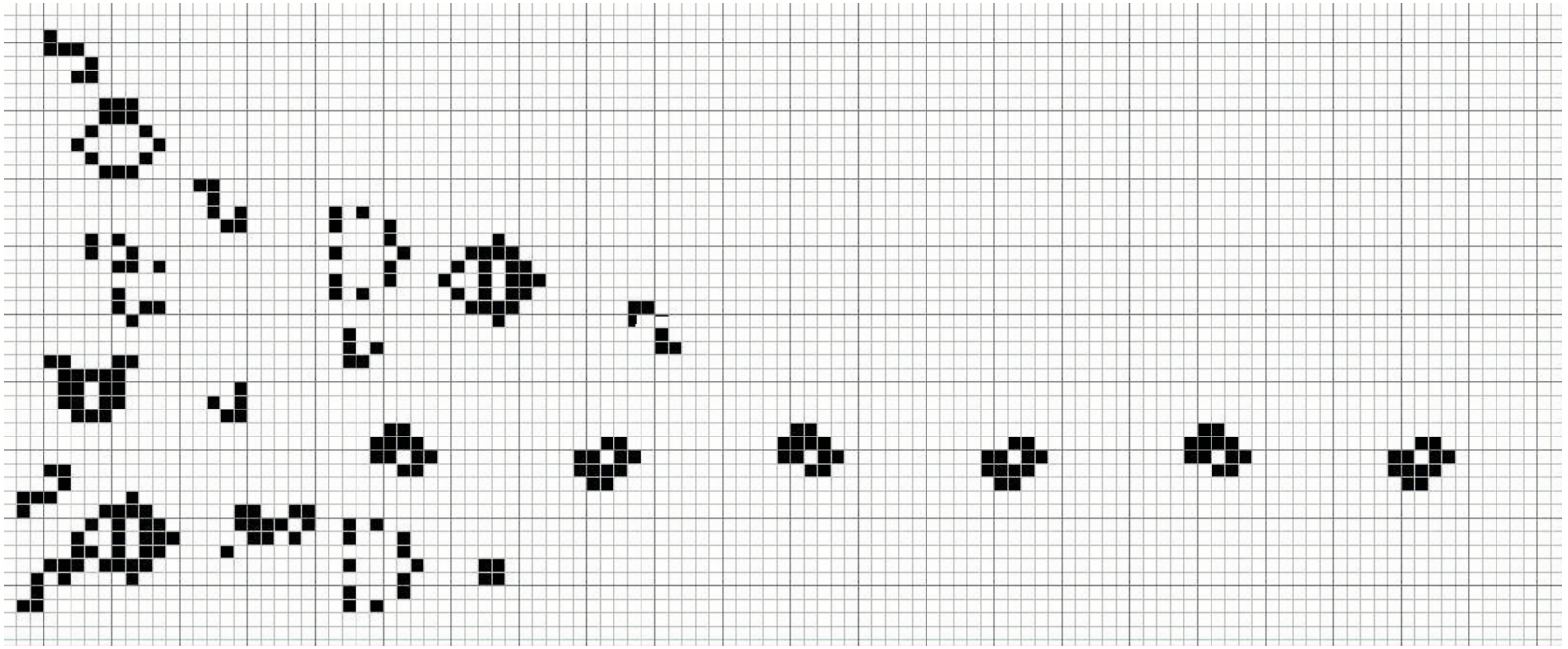


  - Heavy Weight Space Ship (HWSS)



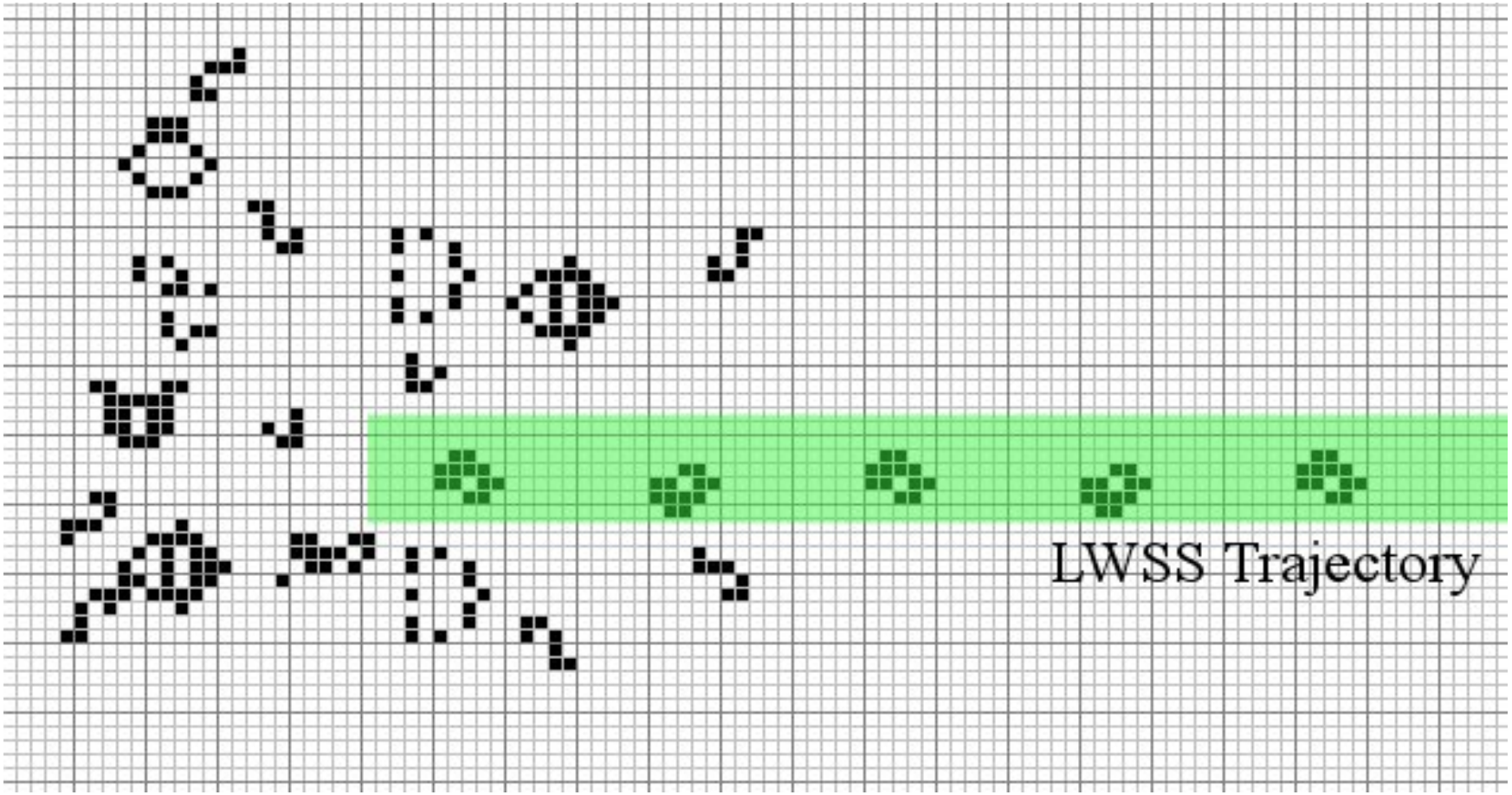
The Game of Life

- Fleet:
  - a collection of space ships travelling along the same trajectory
- Gun:
  - a pattern that periodically generates a space ship along a fixed trajectory

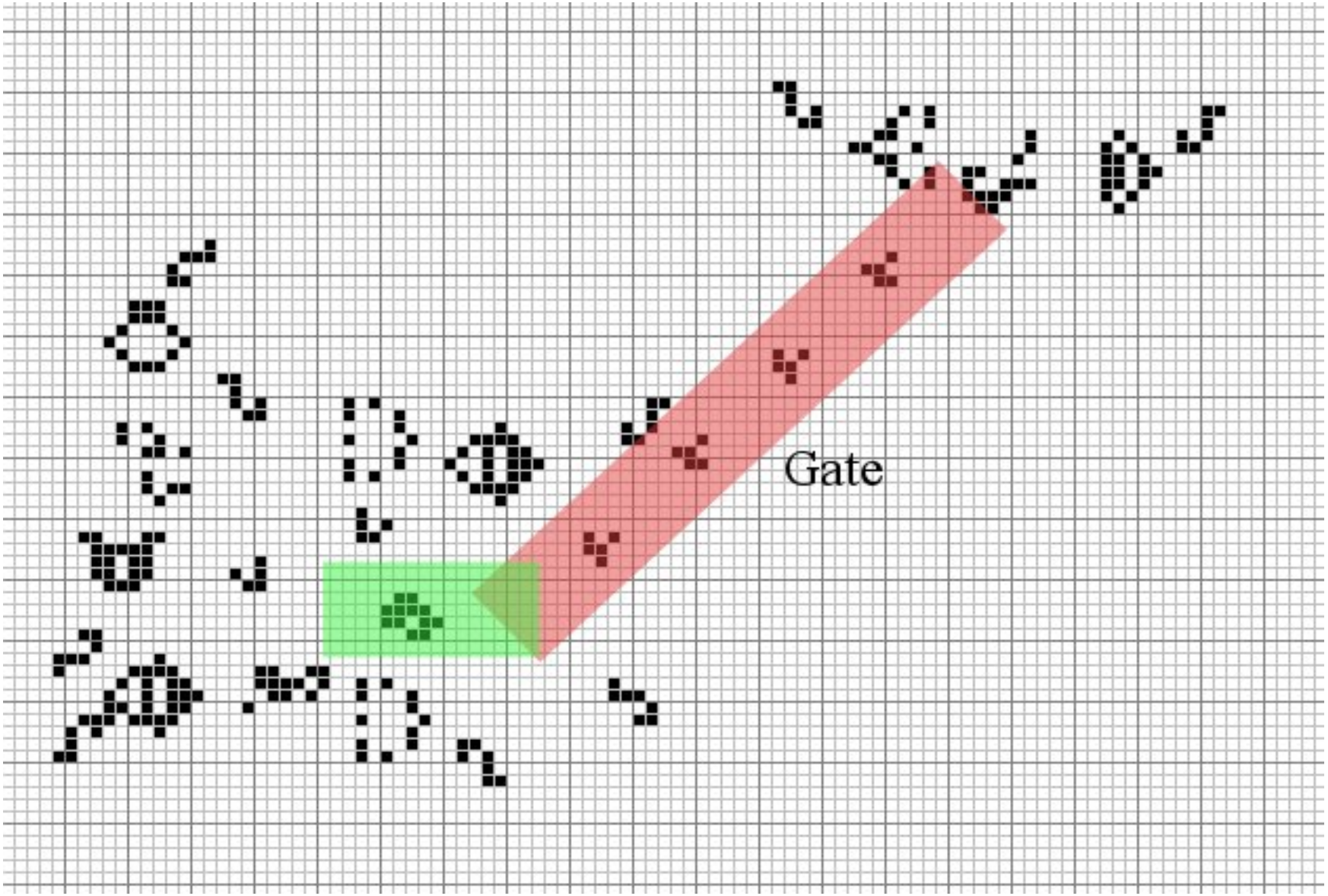


The Game of Life

- Controlling what happens: Gates



The Game of Life



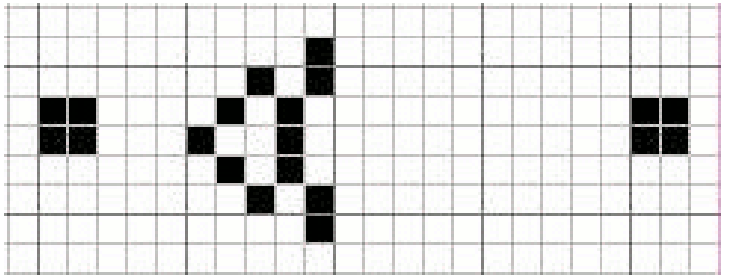
The Game of Life

- Stability:
  - when building complex machines, it would be nice if they stayed together
  - certain patterns must be *stabilized* to be useful
  - ex: the Queen Bee

Demo: Queen Bee Instability

The Game of Life

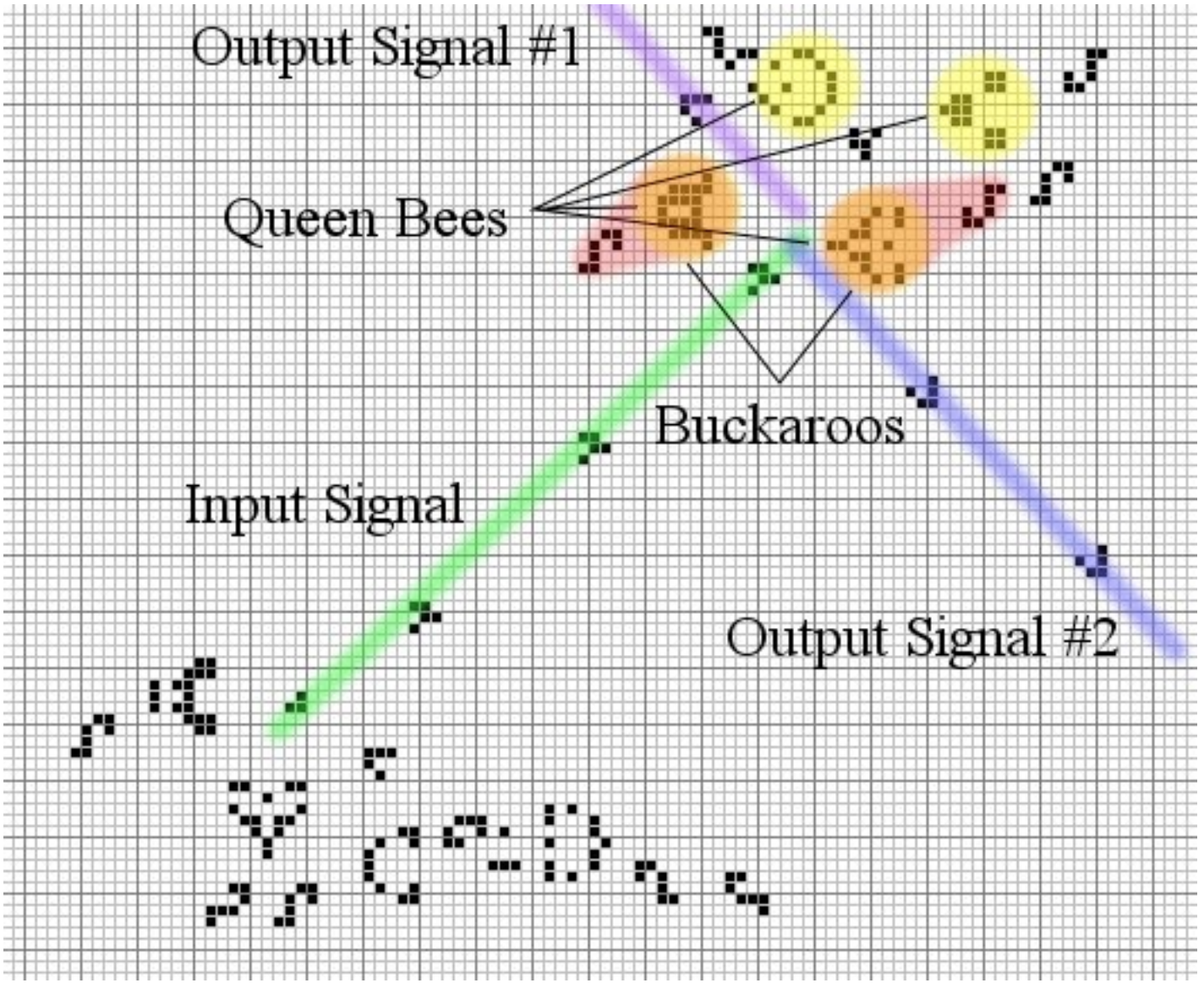
- Stability:
  - when building complex machines, it would be nice if they stayed together
  - certain patterns must be *stabilized* to be useful
  - ex: the Queen Bee



Stabilized Queen Bee

The Game of Life

- Duplicating a Signal:
  - Fanouts



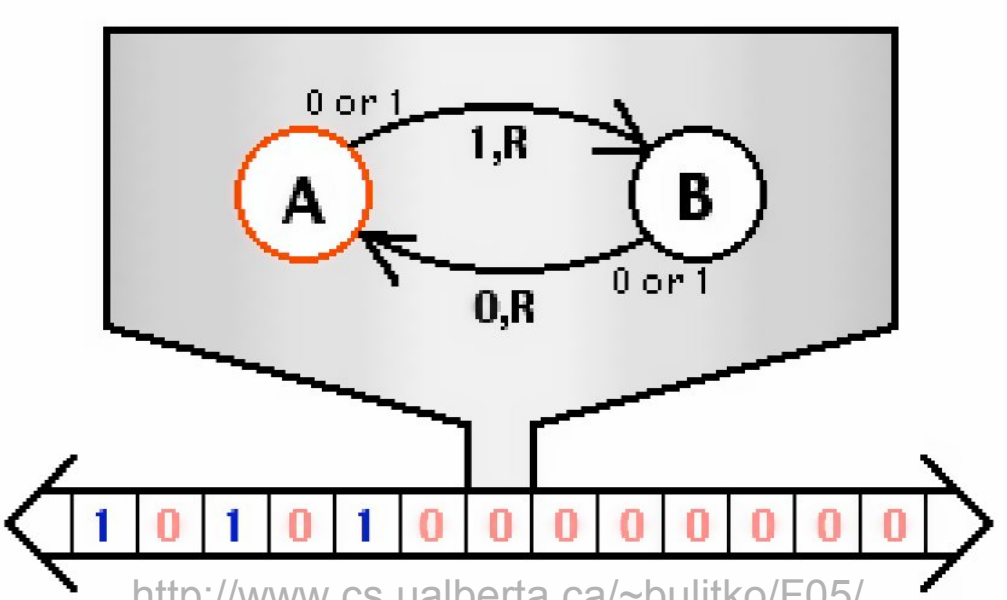
The Game of Life

Questions?

Brief Digression:
Turing Machines

Turing Machines

- A Turing Machine has:
  - a finite set of states
  - rules for transitioning between states
  - an infinite sequence of cells (called a "tape")

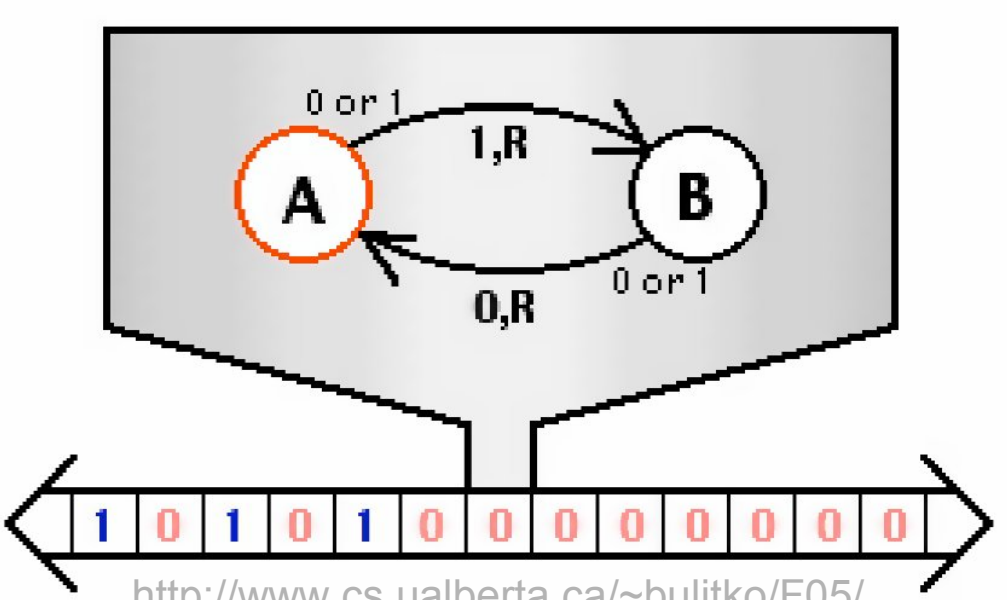


http://www.cs.ualberta.ca/~bulitko/F05/CMPUT651/slides/2005-11-07-VB.pdf

  - a set of symbols describing the possible contents of each cell in the tape
  - the ability to read and write the symbol in a single cell
  - the ability to move along the tape to access different cells

Turing Machines

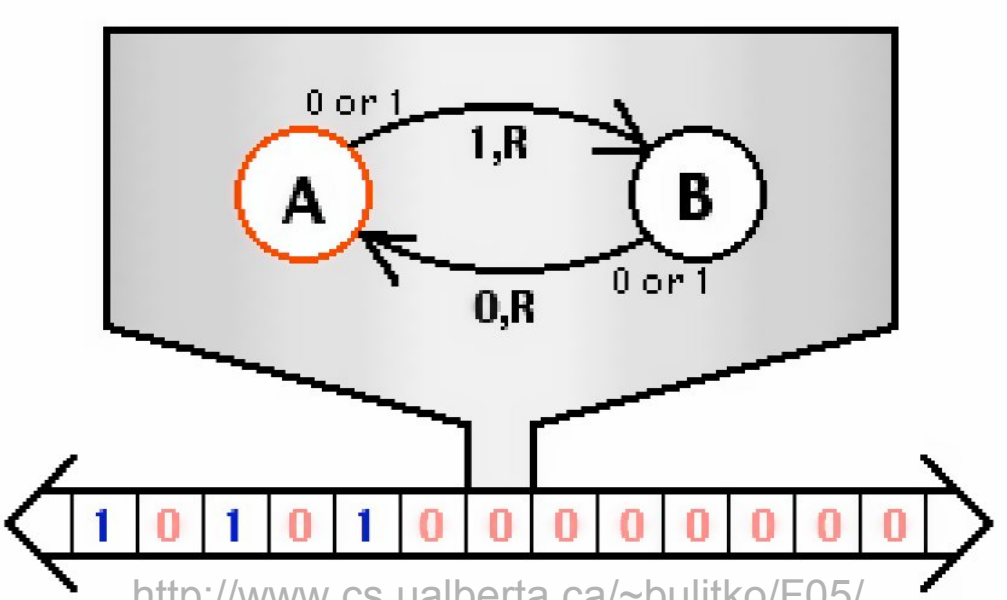
- Each state transition rule specifies:
  - the current state
  - the symbol read from the current cell
  - the state to move to
  - the symbol to write to the current cell
  - the direction to move the machine along the tape



http://www.cs.ualberta.ca/~bulitko/F05/CMPUT651/slides/2005-11-07-VB.pdf

Turing Machines

- For every time step,
  - read the symbol from the current cell
  - find the transition rule that matches both the current state and the symbol read
  - change to the specified new state
  - write the specified symbol
  - move the machine in the specified direction



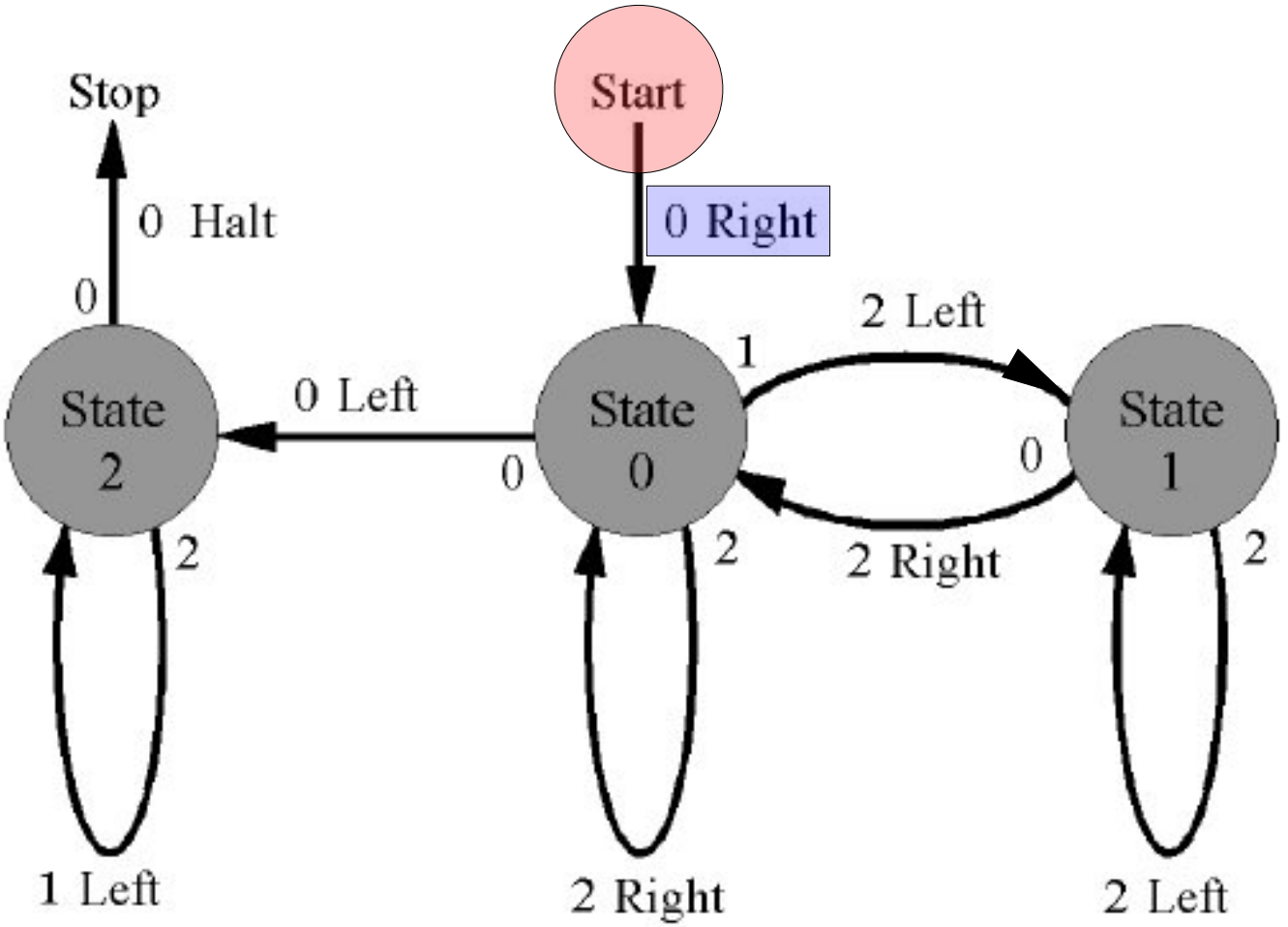
http://www.cs.ualberta.ca/~bulitko/F05/CMPUT651/slides/2005-11-07-VB.pdf

Universal Turing Machines

- Input:
  - the specification of a Turing Machine
    i.e., a set of states and state transitions
  - the input to the specified Turing Machine
- Output:
  - the result of having run the specified Turing Machine on the given input
- A Universal Turing Machine can simulate any Turing Machine, including itself.

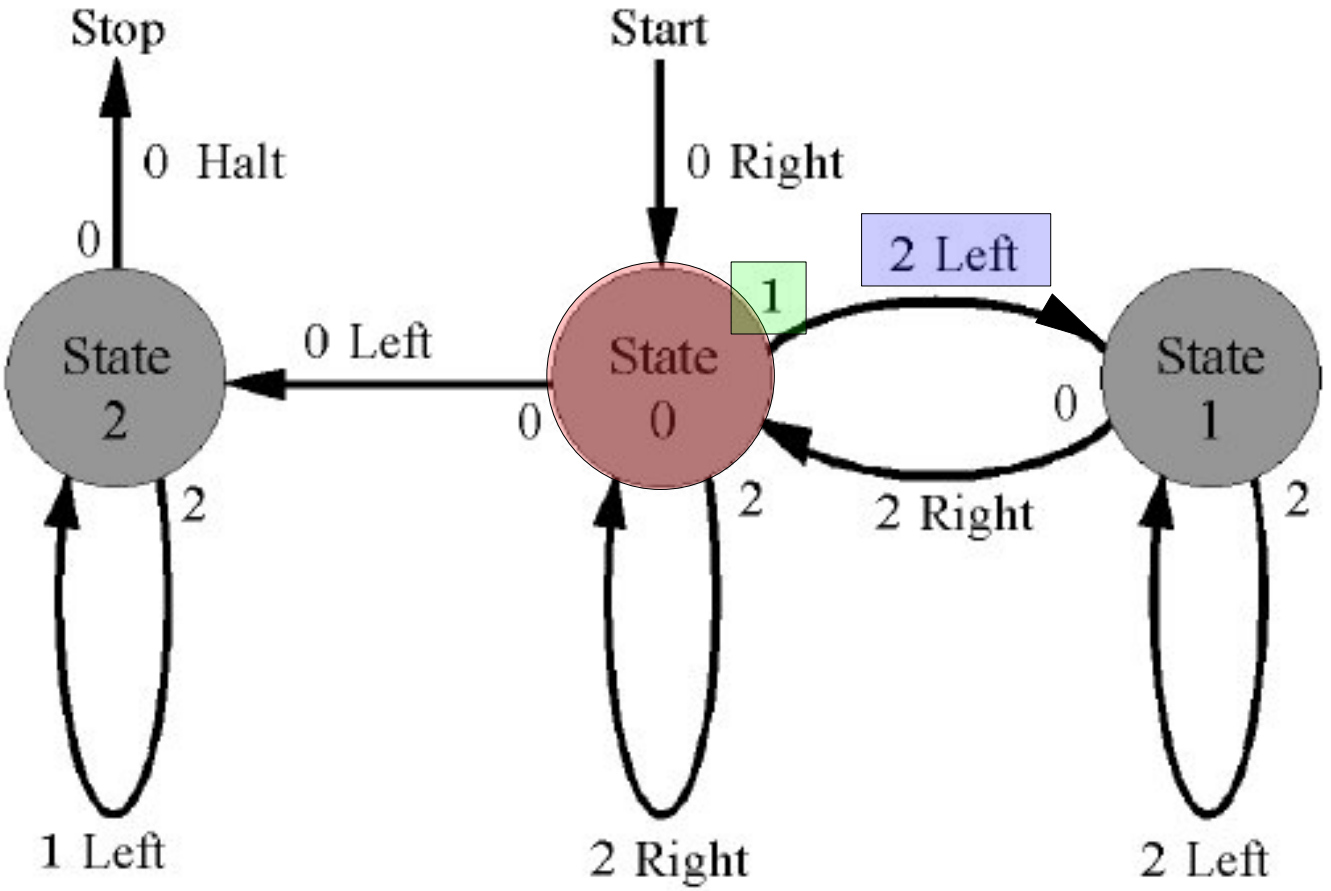
Turing Machines

0000<u>0</u>11000



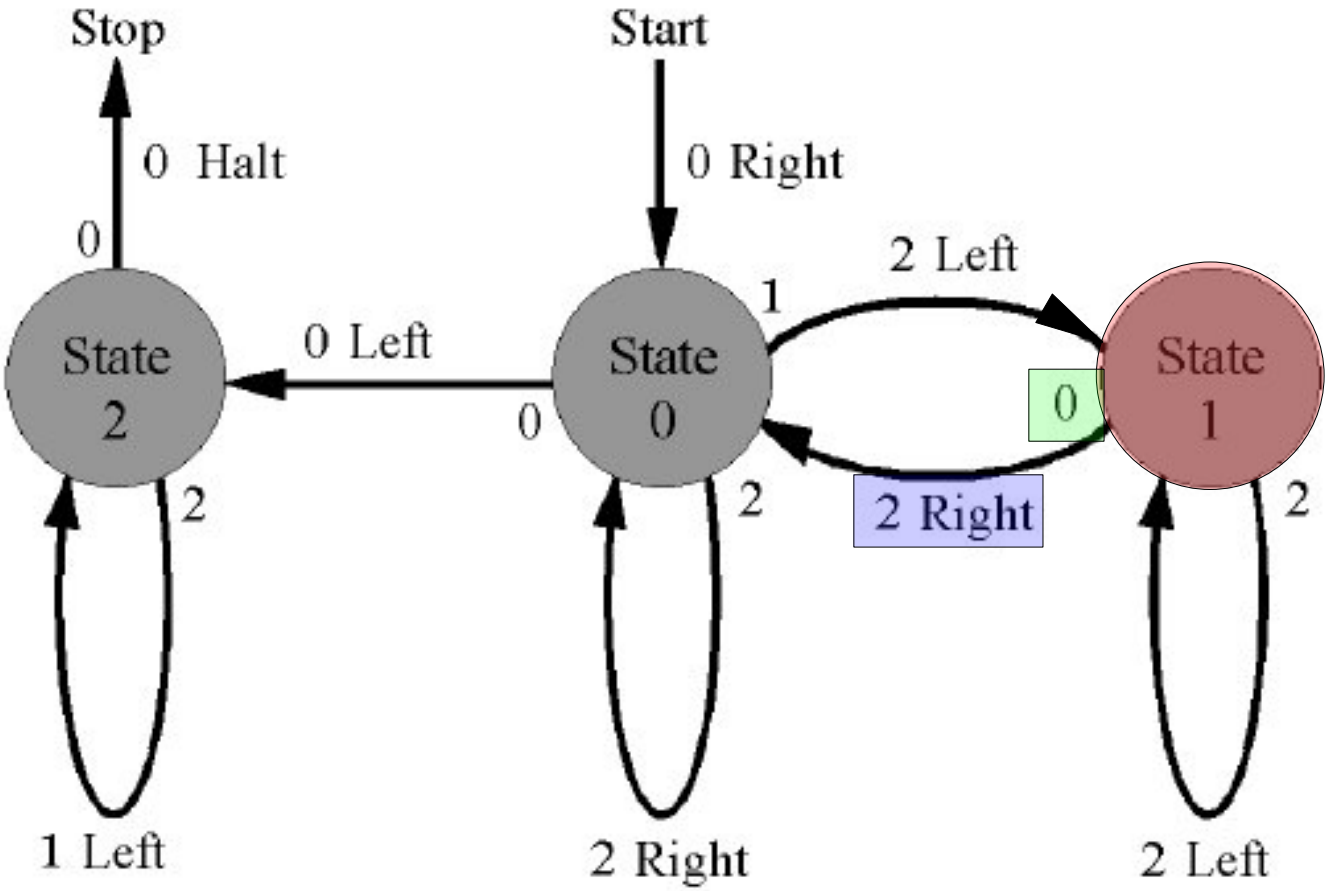
Turing Machines

0 0 0 0 0 1 1 0 0 0



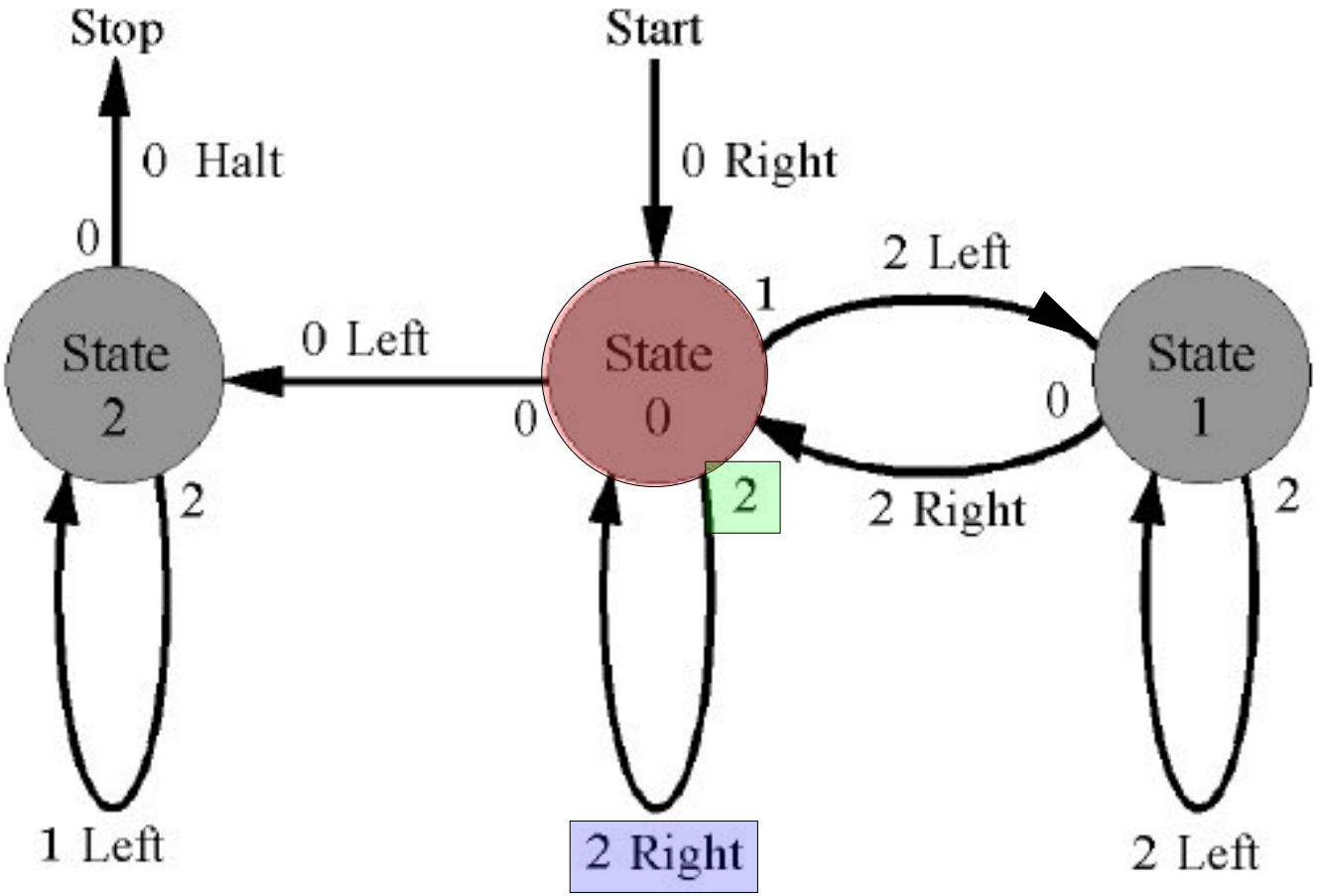
Turing Machines

0 0 0 0 0 2 1 0 0 0



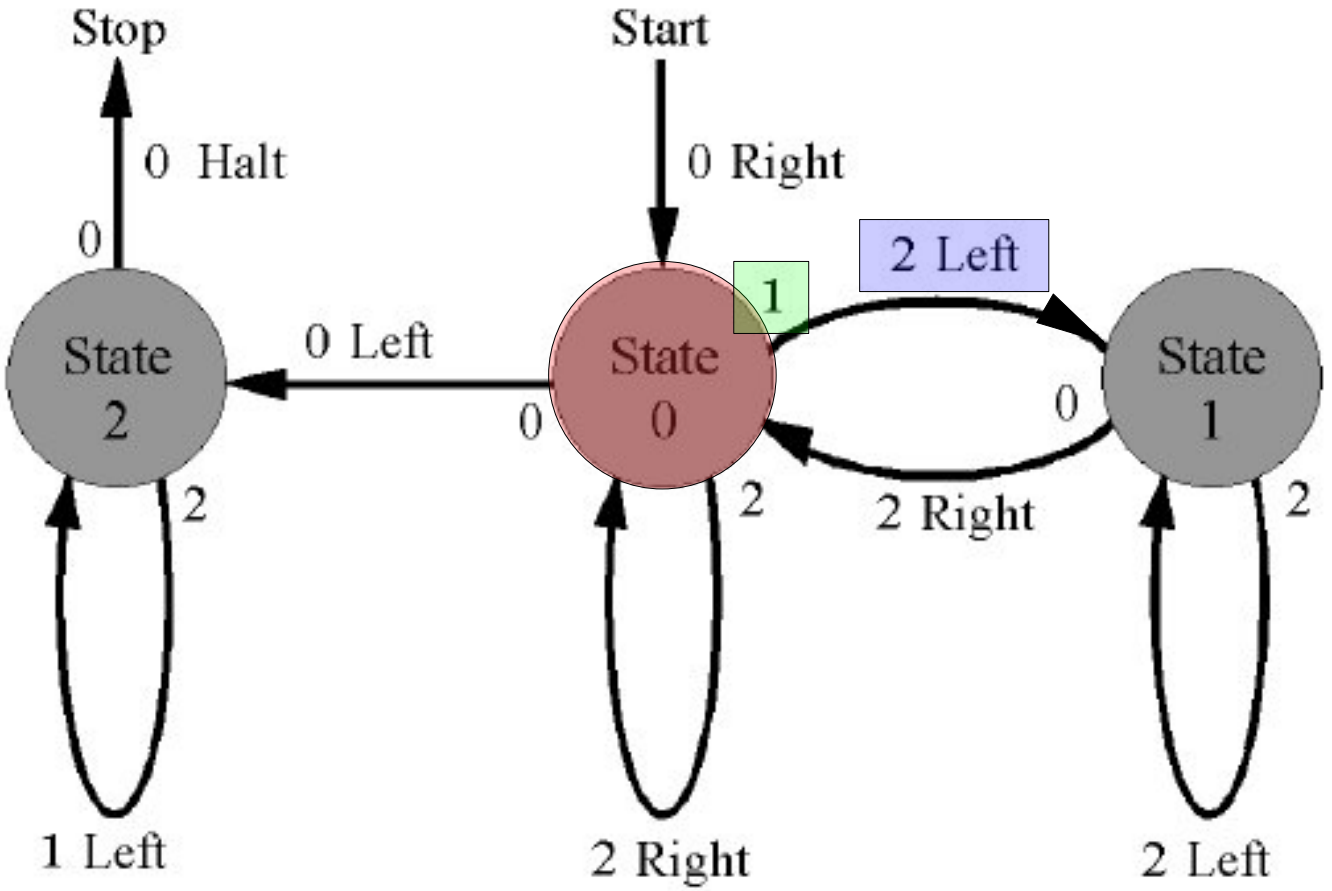
Turing Machines

0 0 0 0 2 2 1 0 0 0



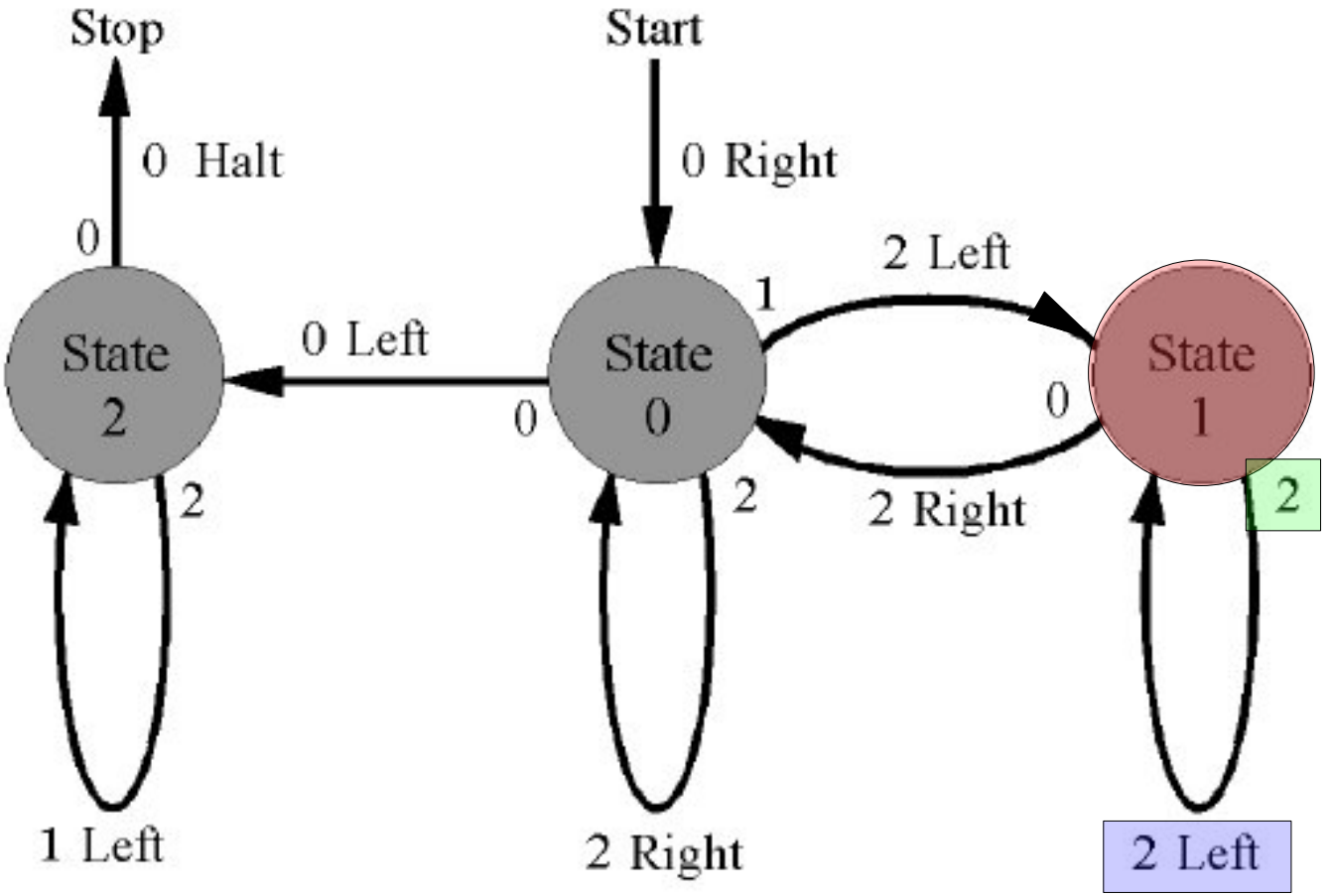
Turing Machines

0 0 0 0 2 2 1 0 0 0



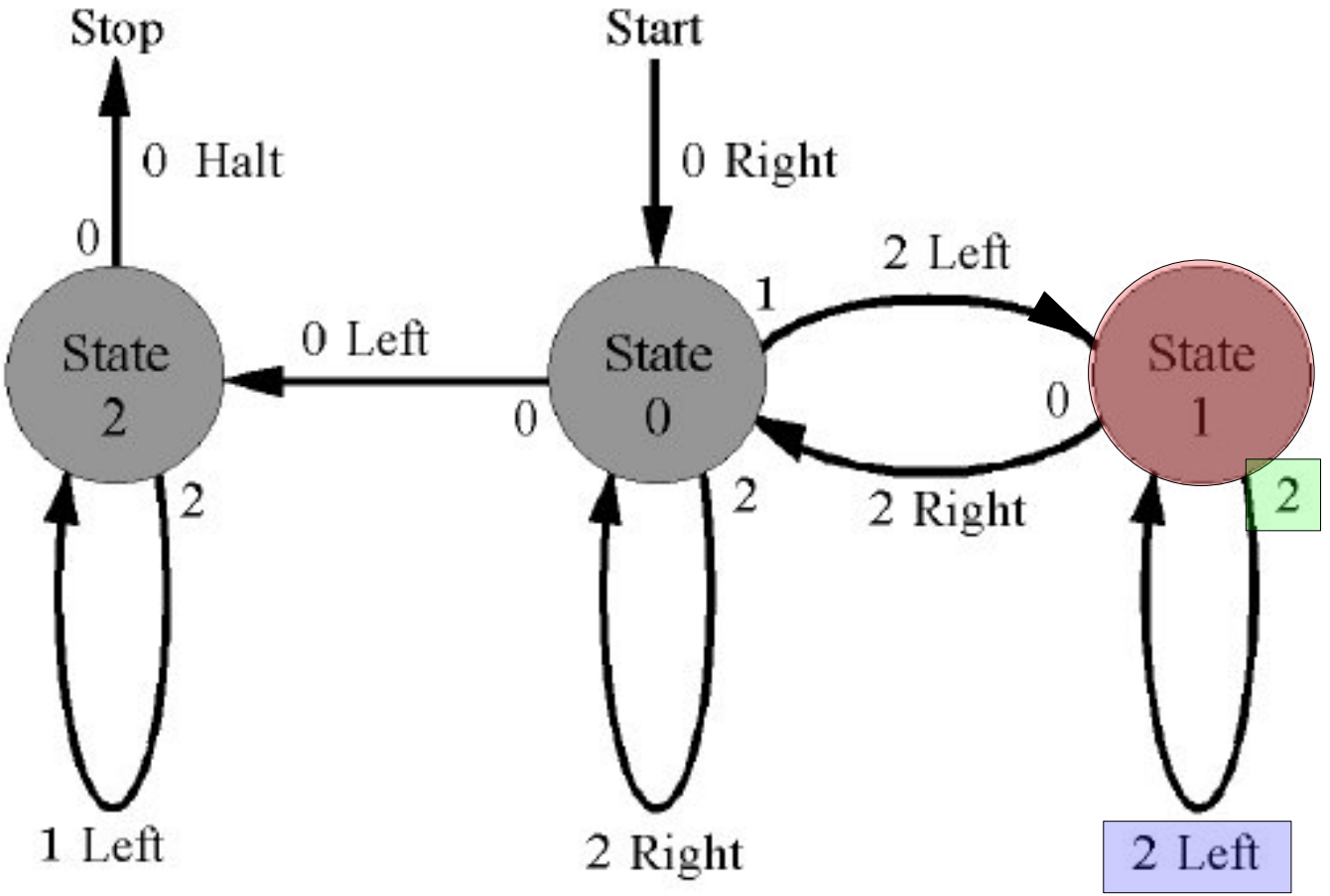
Turing Machines

0 0 0 0 2 2 2 0 0 0



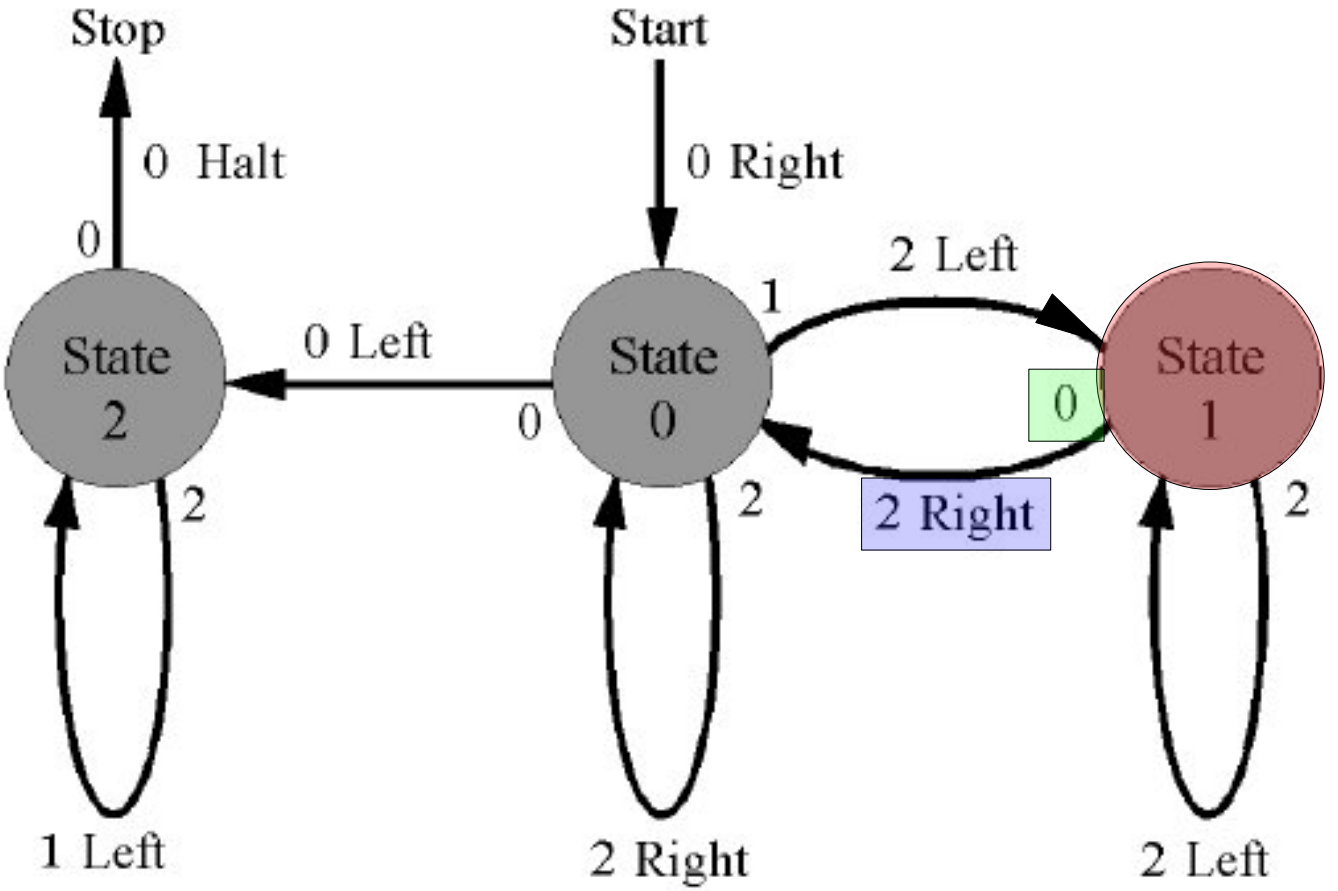
Turing Machines

0 0 0 0 2 2 2 0 0 0



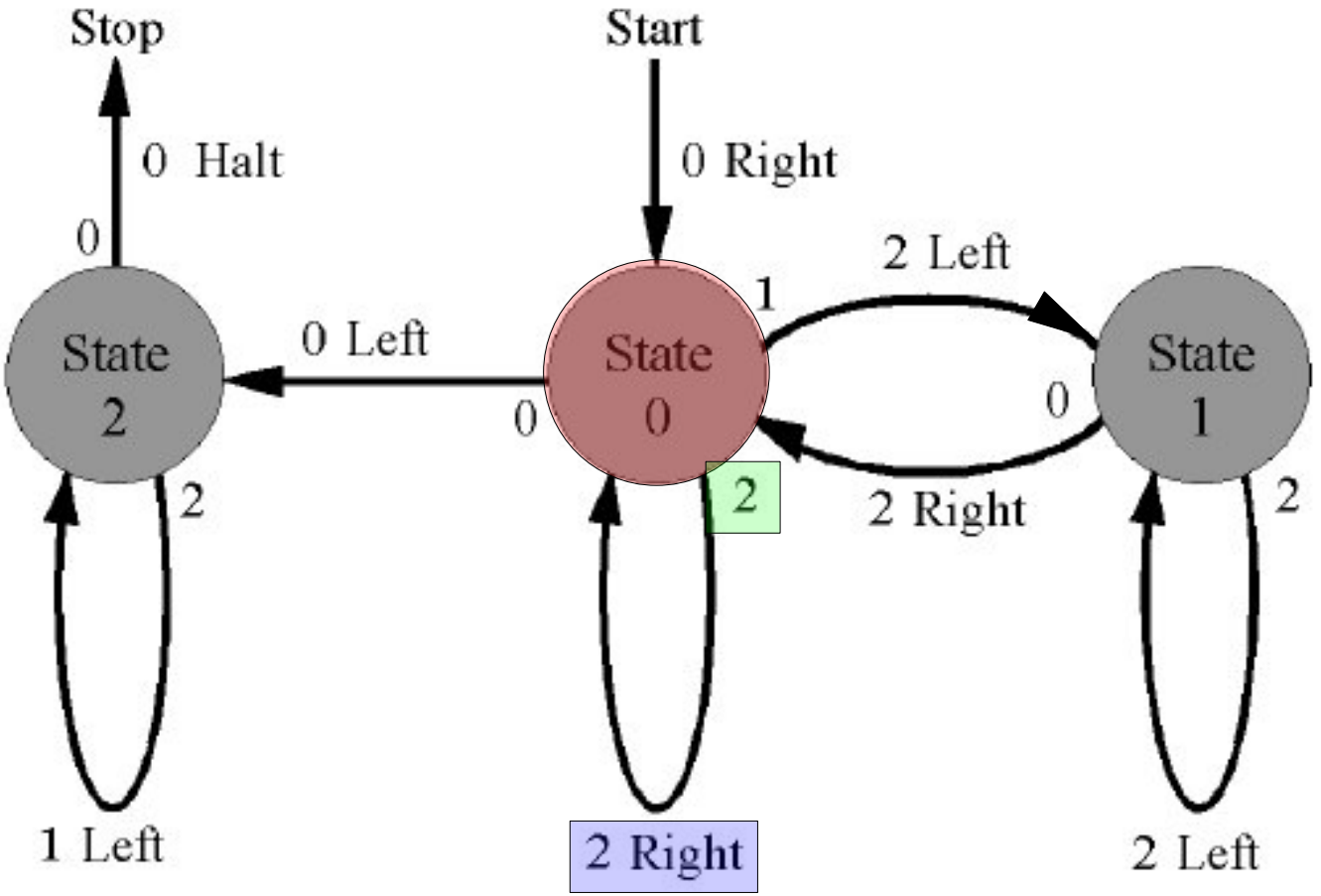
Turing Machines

0 0 0 0 2 2 2 0 0 0



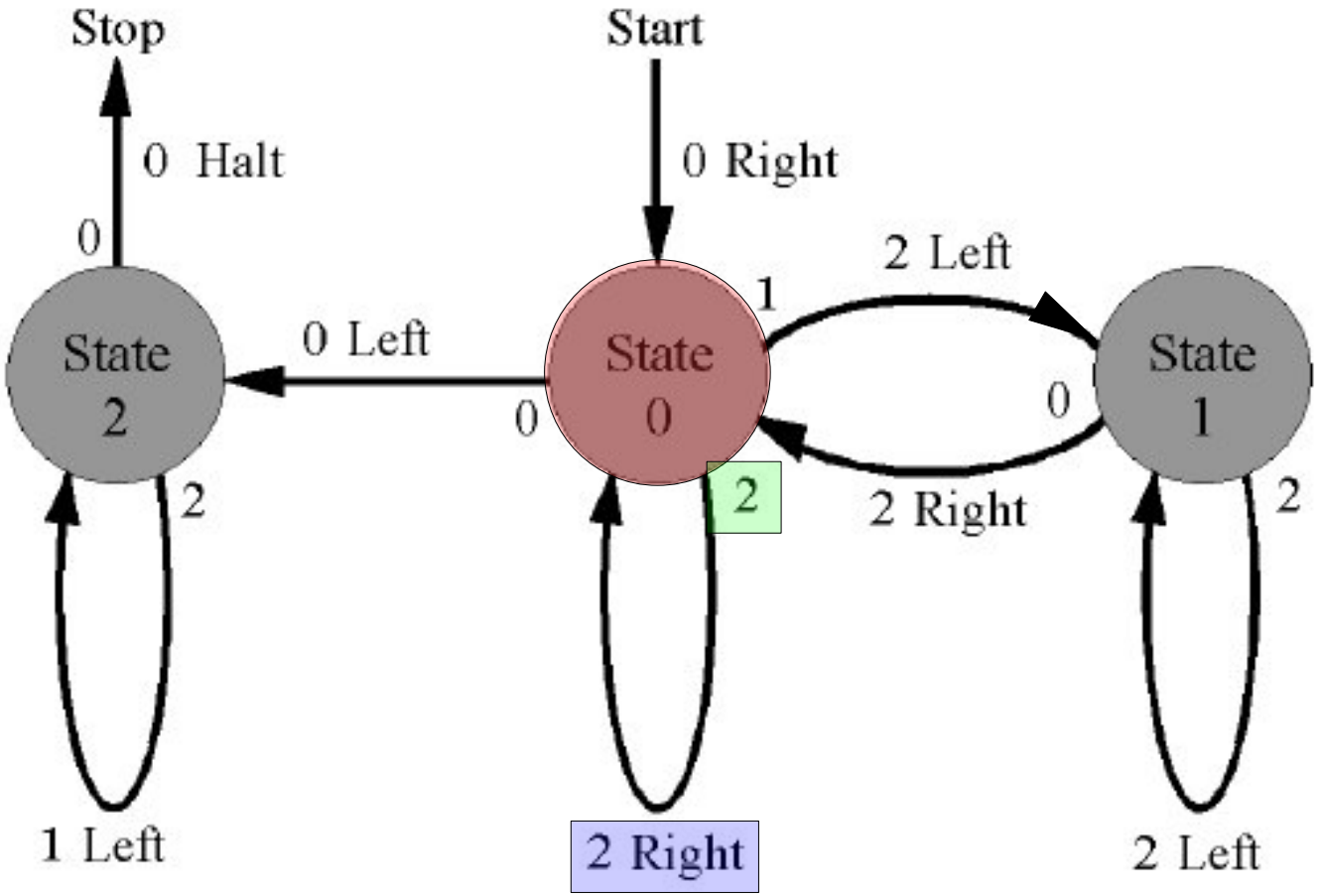
Turing Machines

0002222000



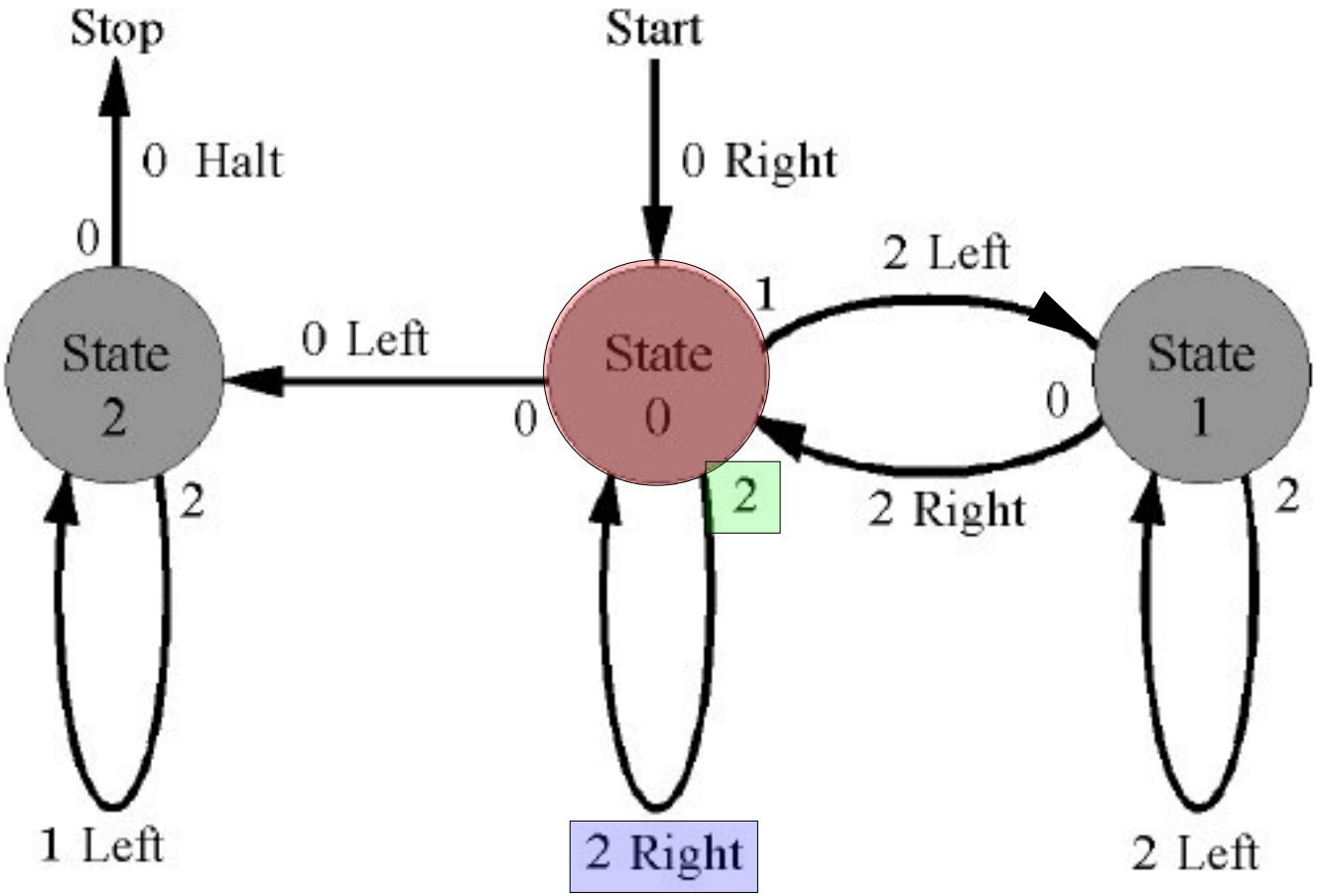
Turing Machines

0 0 0 2 2 2 2 0 0 0



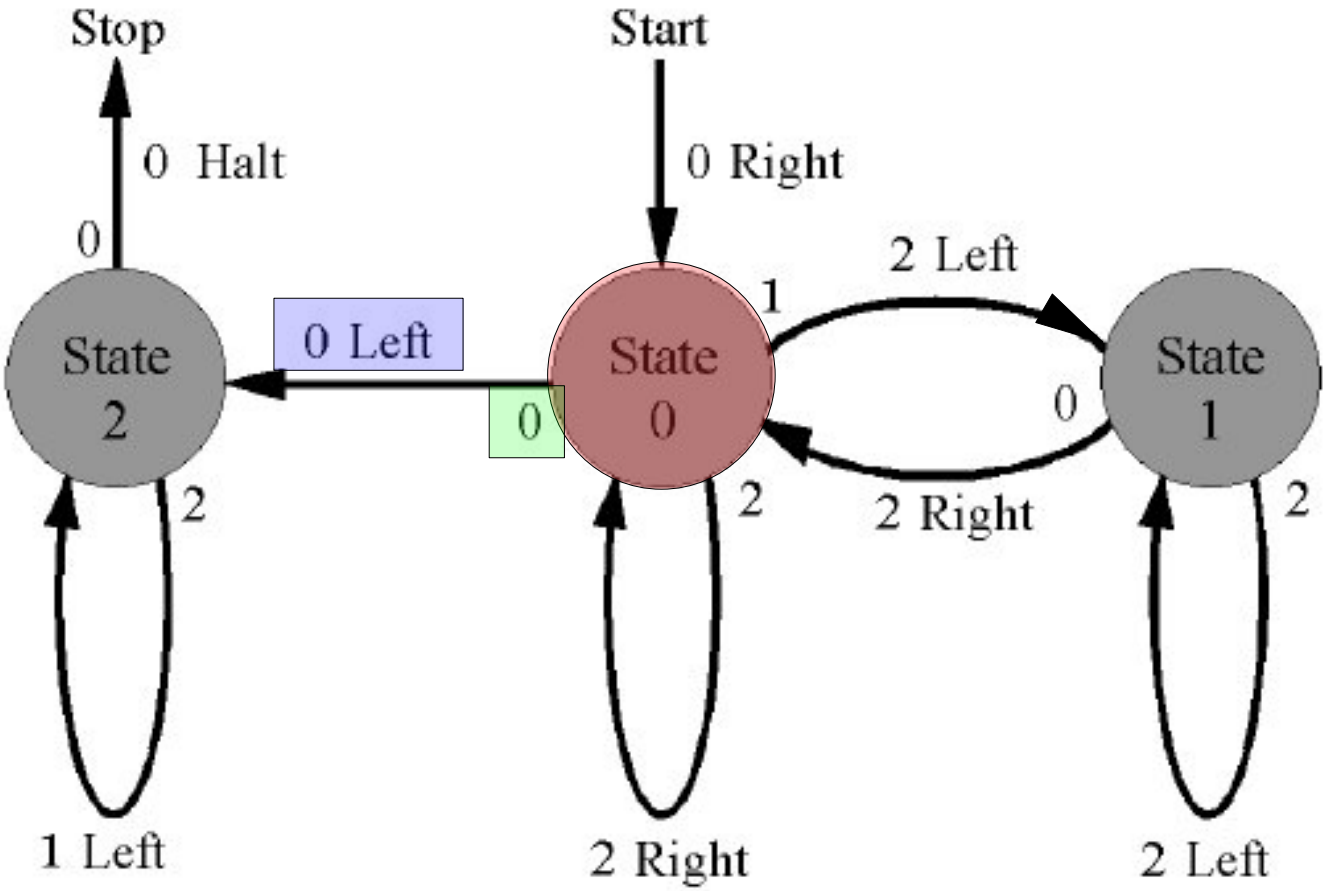
Turing Machines

0 0 0 2 2 2 2 0 0 0



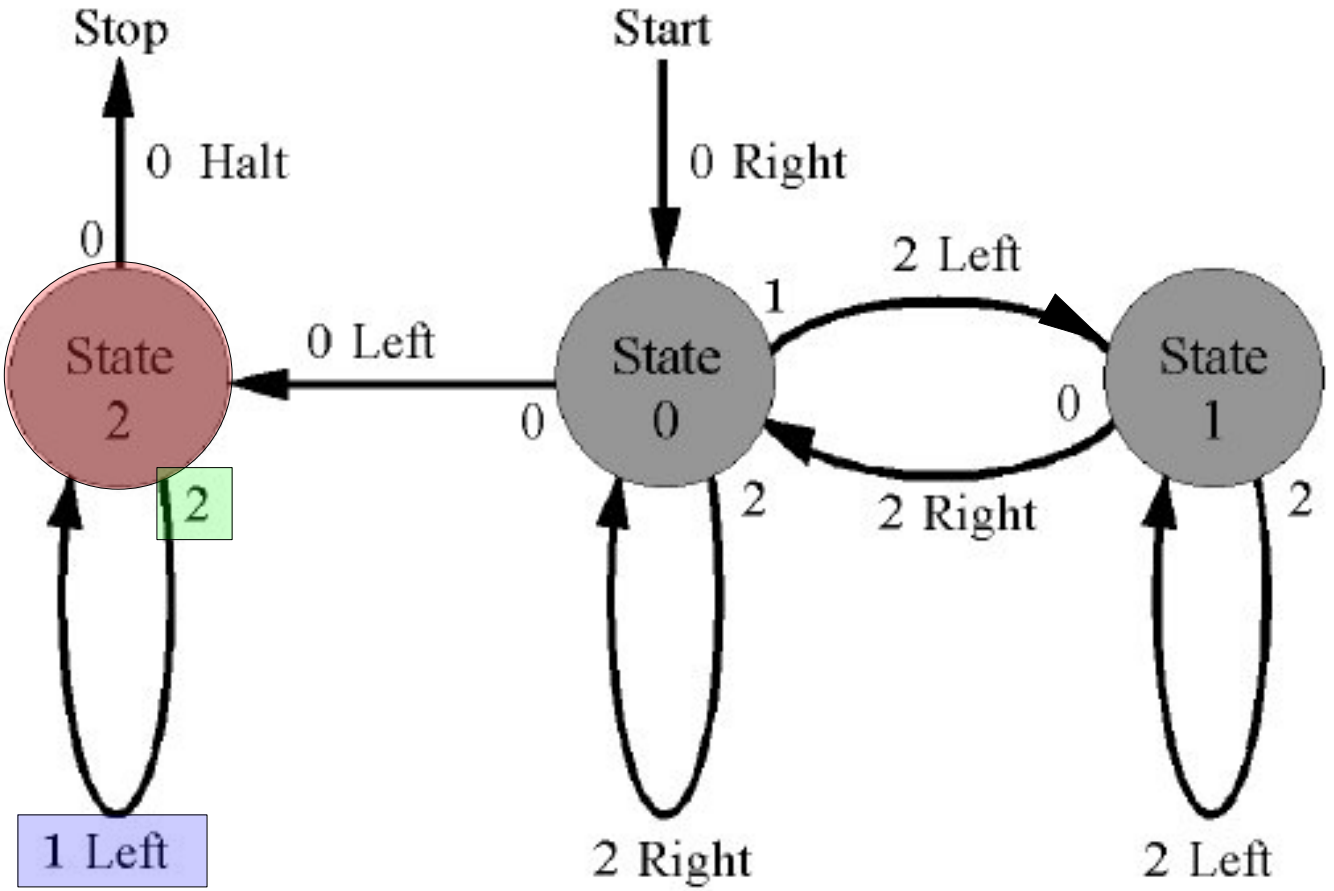
Turing Machines

0 0 0 2 2 2 2 0 0 0



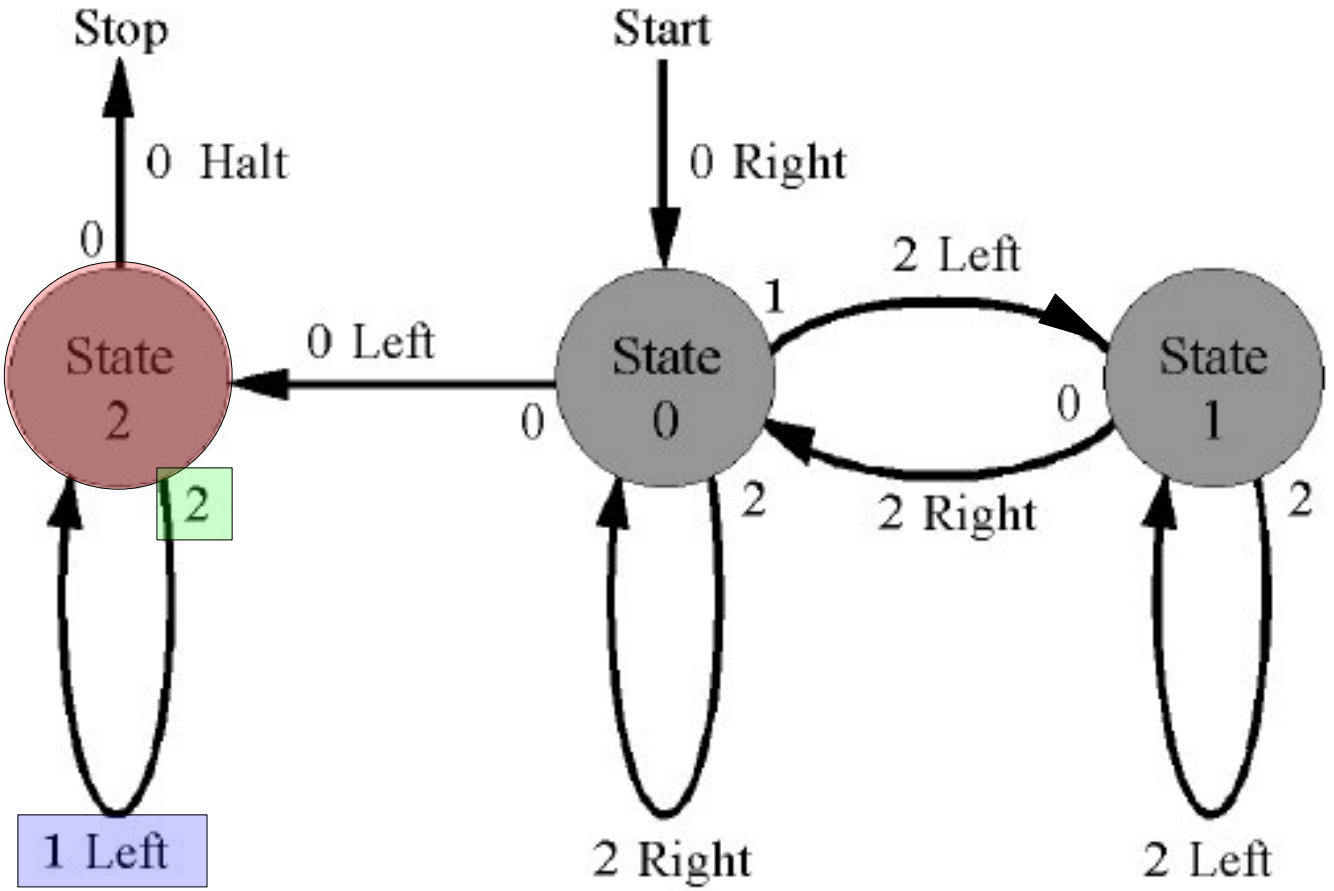
Turing Machines

0 0 0 2 2 2 2 0 0 0



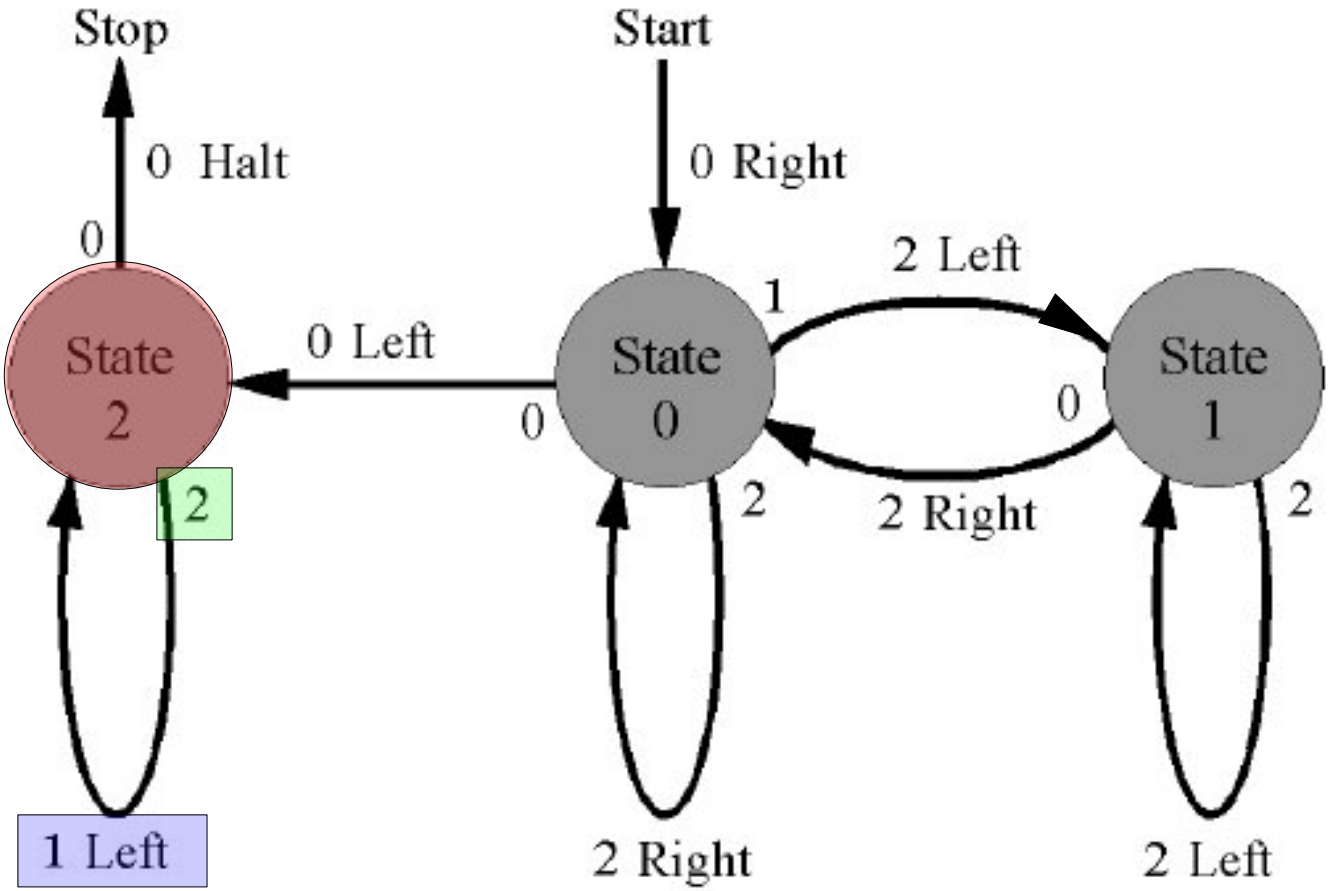
Turing Machines

0 0 0 2 2 2 1 0 0 0



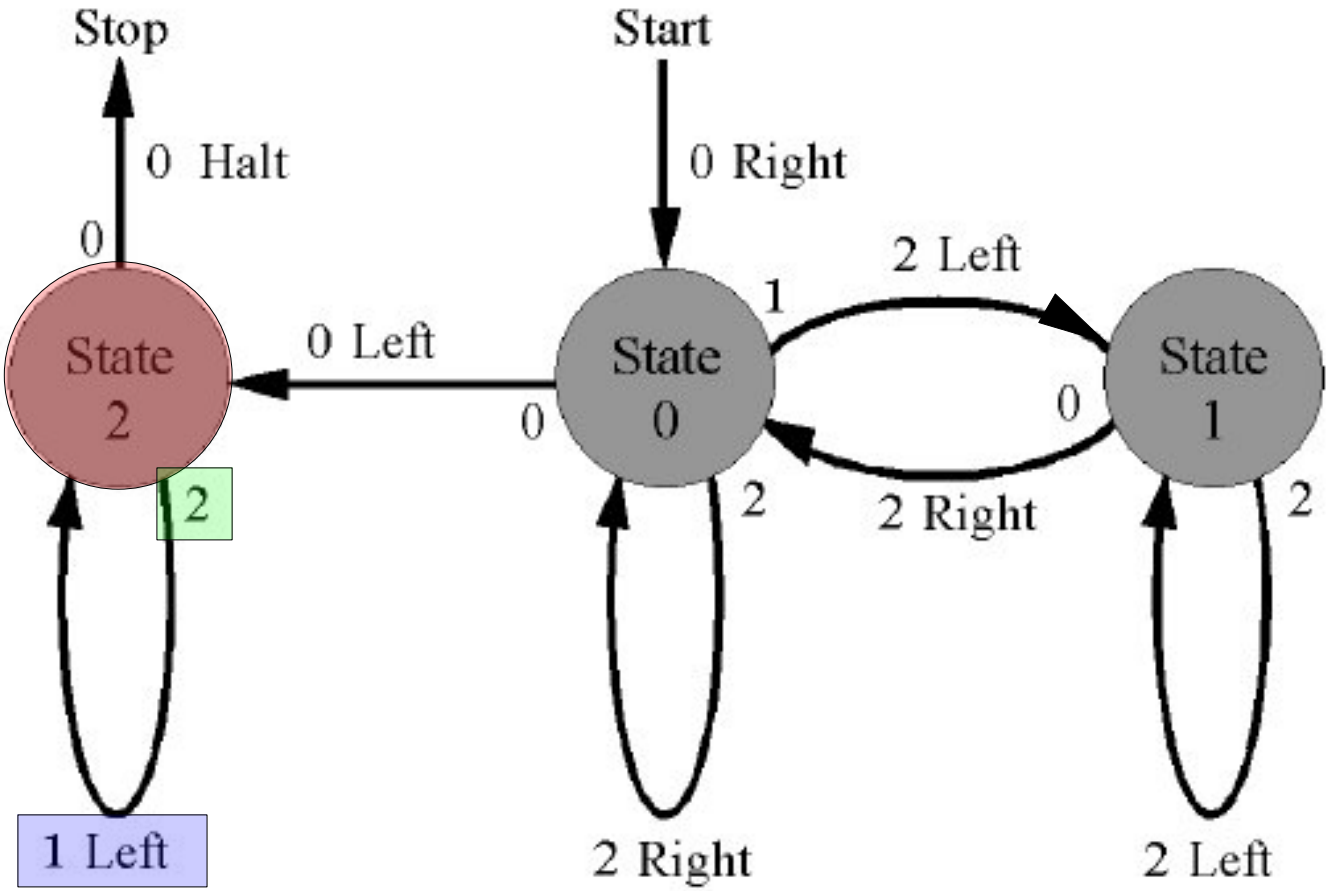
Turing Machines

0002211000



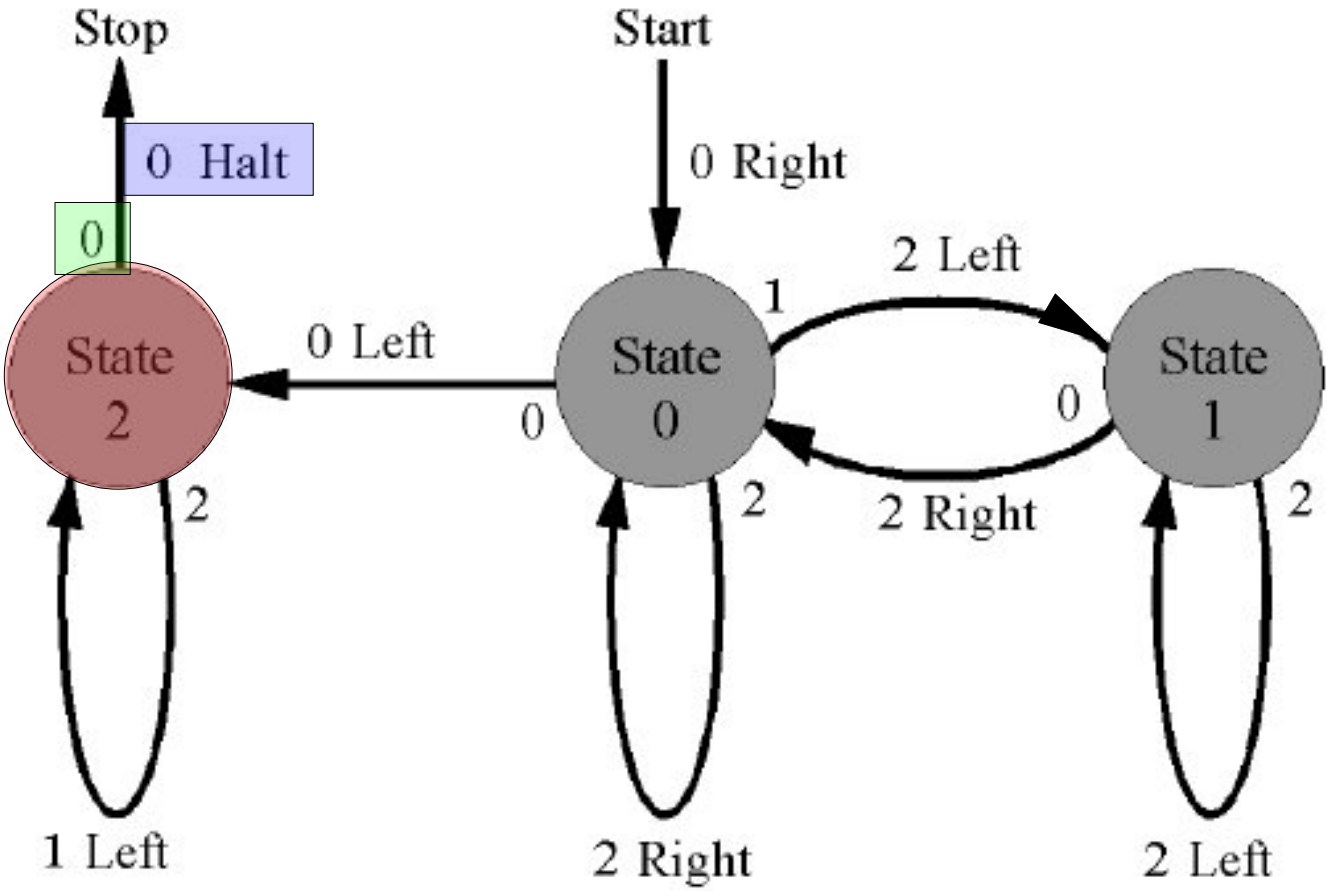
Turing Machines

0002111000



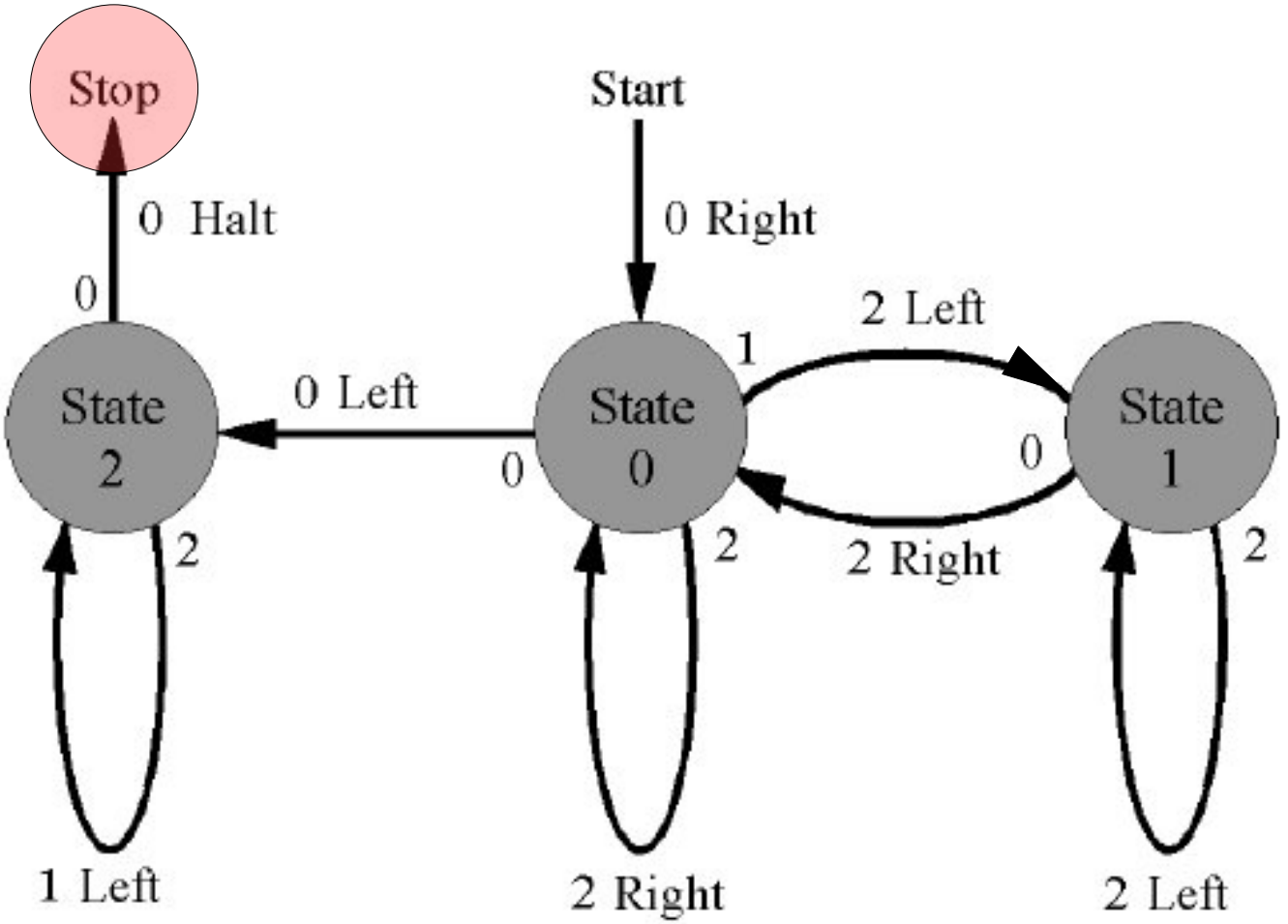
Turing Machines

0 0 0 1 1 1 1 0 0 0



Turing Machines

0 0 0 1 1 1 1 0 0 0



A Turing Machine in the Game of Life

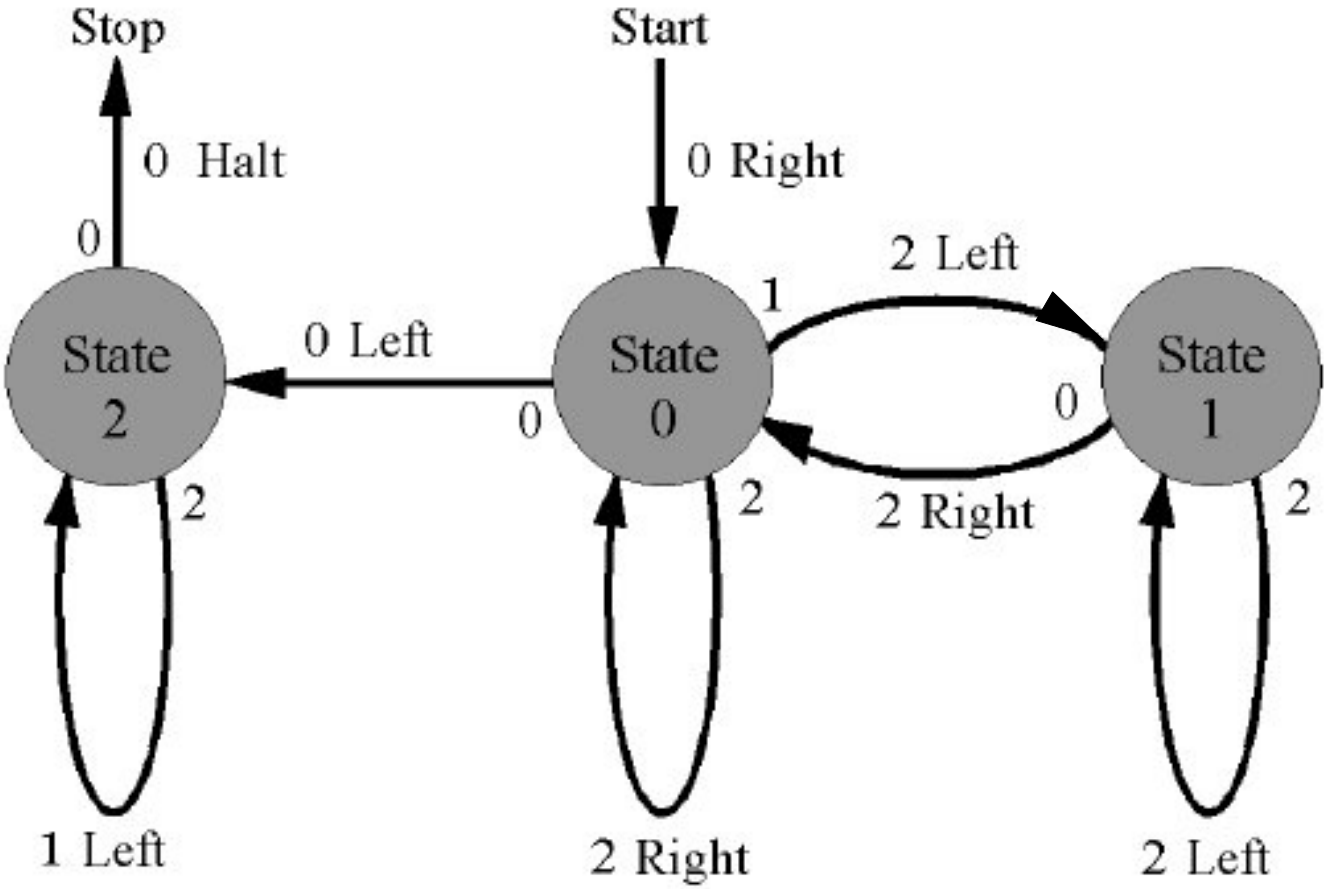
A Turing Machine in the Game of Life

- What do we need?
  - states
  - transition rules
  - a tape
  - possible symbols for each cell in the tape
  - cell read/write access
  - multiple-cell access

A Turing Machine in the Game of Life

- What do we need?
  - **states**
  - **transition rules**
  - a tape
  - **possible symbols for each cell in the tape**
  - cell read/write access
  - multiple-cell access

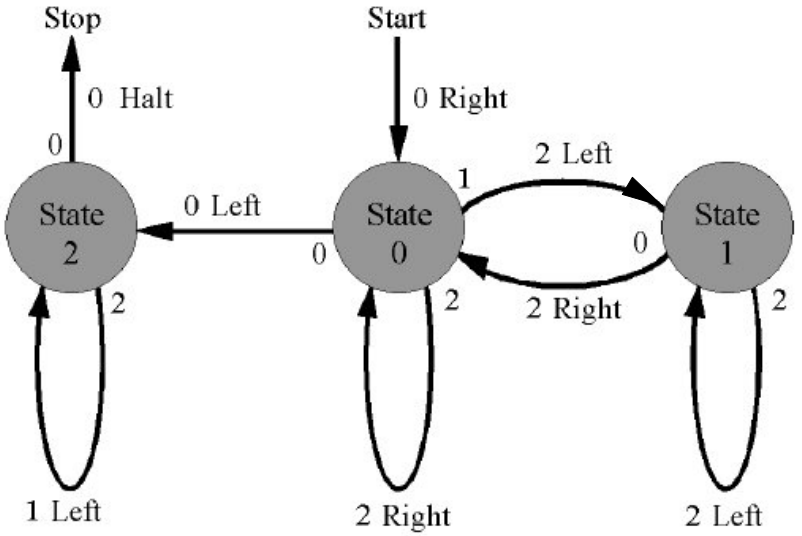
A Turing Machine in the Game of Life



Why not use these?

A Turing Machine in the Game of Life

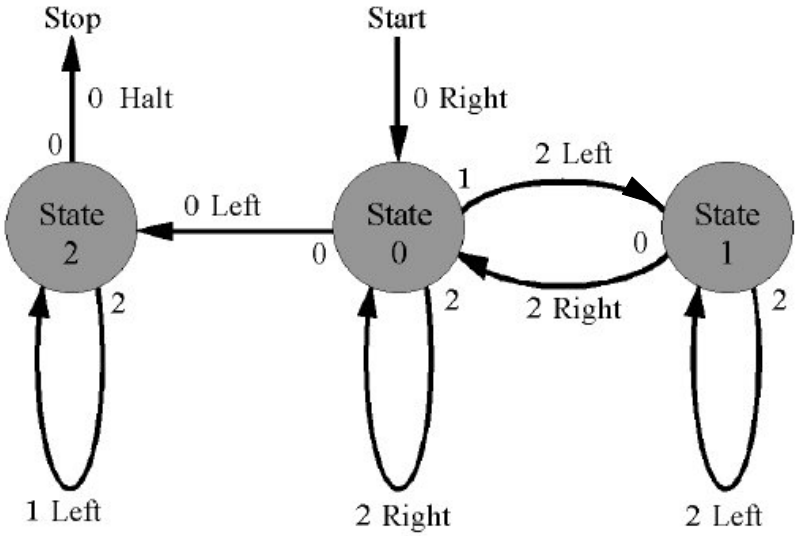
- States: 0, 1, 2
- Symbols: 0, 1, 2
- Transition Rules:
  - states to move to (0, 1, 2)
  - symbols to write (0, 1, 2)
  - directions to move in (Left, Right)



- What about the current state and symbol read?
  - *We have the information, but we'll deal with it later.*

A Turing Machine in the Game of Life

- We need to represent:
  - 3 States: 2 bytes
  - 3 Symbols: 2 bytes
  - 2 Directions: 1 byte

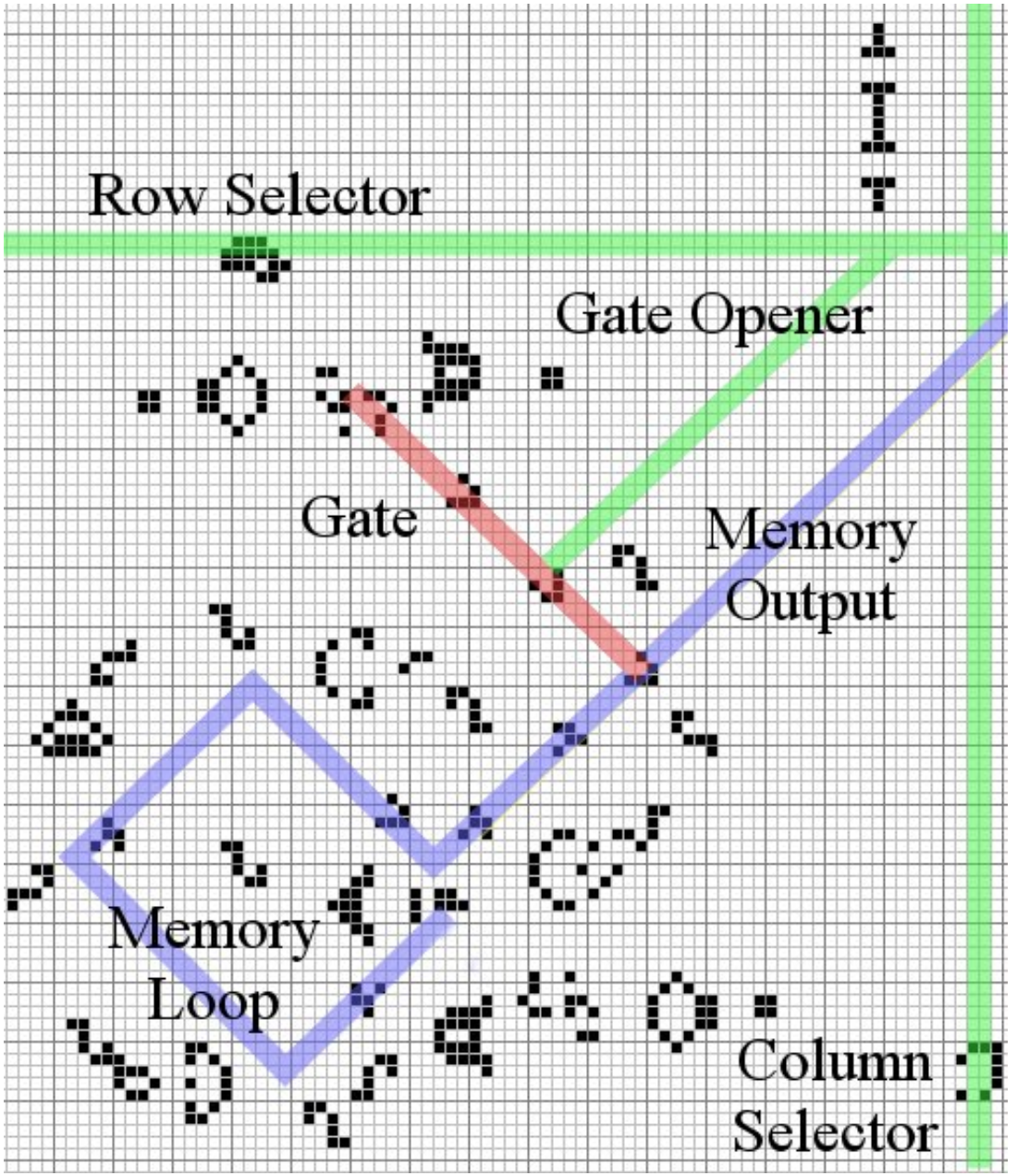


- A Universal Turing Machine would need:
  - 16 States: 4 bytes
  - 8 Symbols: 3 bytes
  - 2 Directions: 1 byte

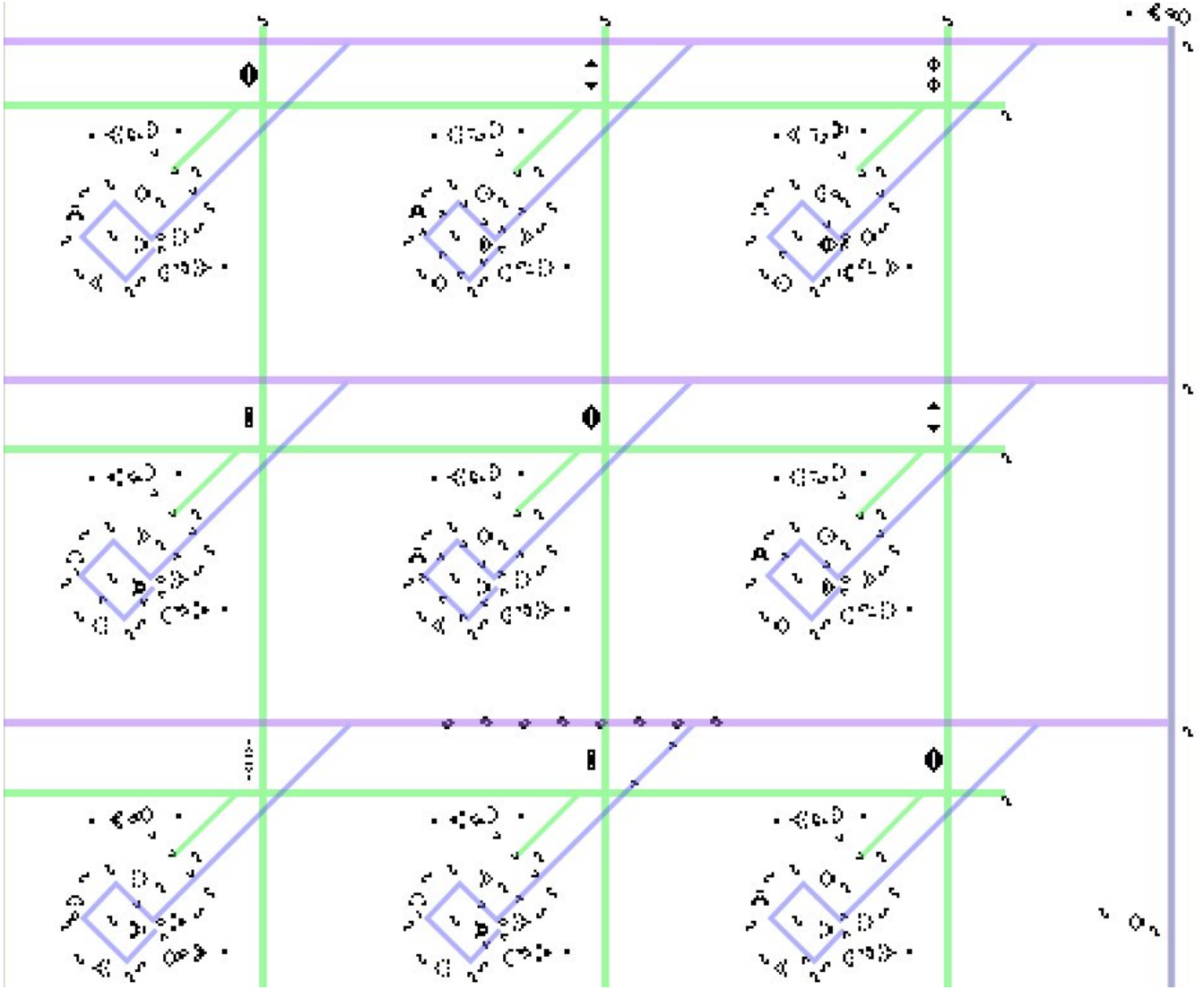
Use these, to keep the design extensible.

A Turing Machine in the Game of Life

- How do we store transition rules?
  - we need a memory
- Rendell's Memory Cell
  - loop one output of a fanout to form an 8-glider memory
  - control output with a 1GAP8 gate

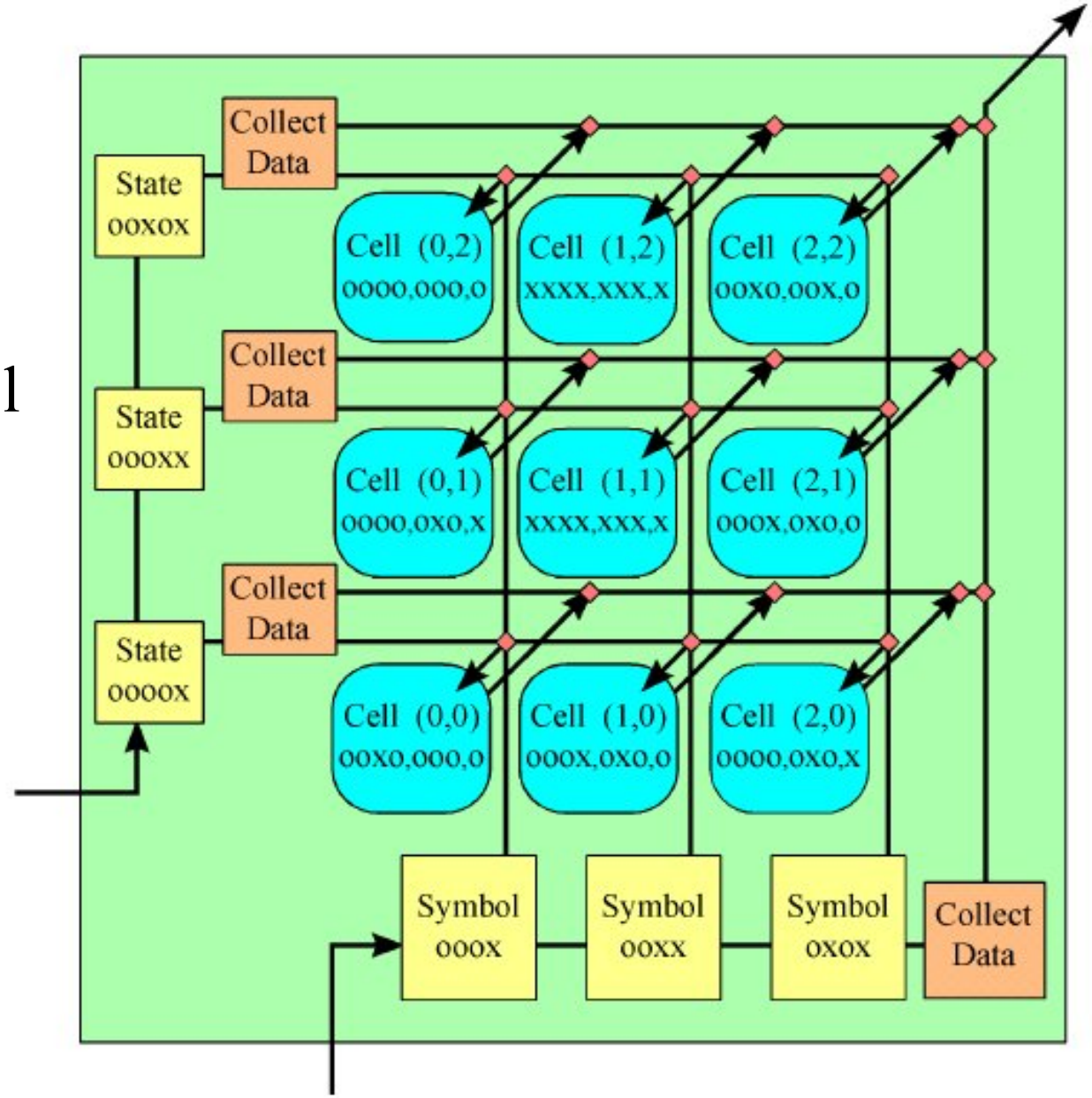


A Turing Machine in the Game of Life

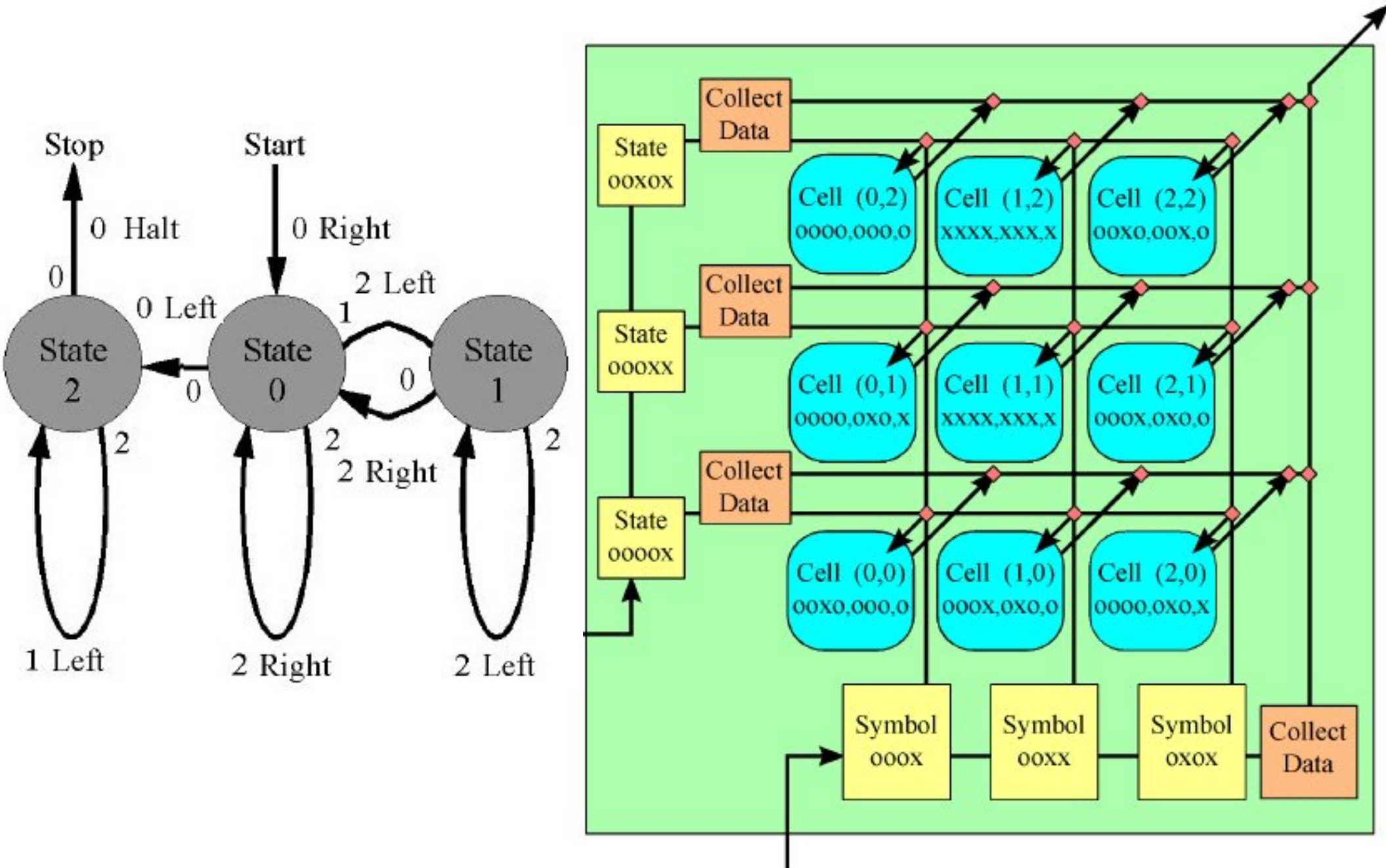


A Turing Machine in the Game of Life

- Indexing the Arrays:
  - use the current state and symbol to index the appropriate cell
  - trailing X's indicate the presence of a signal



A Turing Machine in the Game of Life



A Turing Machine in the Game of Life

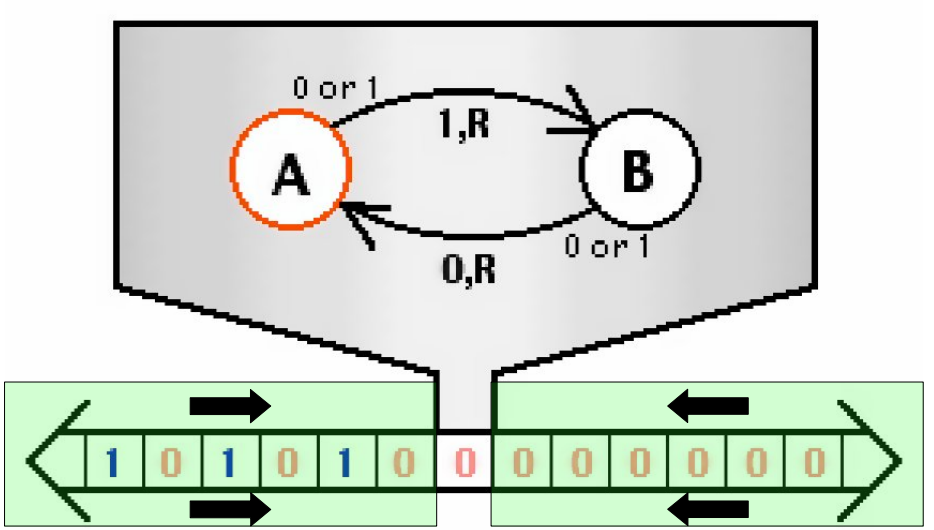
- What do we need?
  - states ✓
  - transition rules ✓
  - a tape
  - possible symbols for each cell in the tape ✓
  - cell read/write access
  - multiple-cell access

A Turing Machine in the Game of Life

- What do we need?
  - states ✓
  - transition rules ✓
  - **a tape**
  - possible symbols for each cell in the tape ✓
  - **cell read/write access**
  - **multiple-cell access**

A Turing Machine in the Game of Life

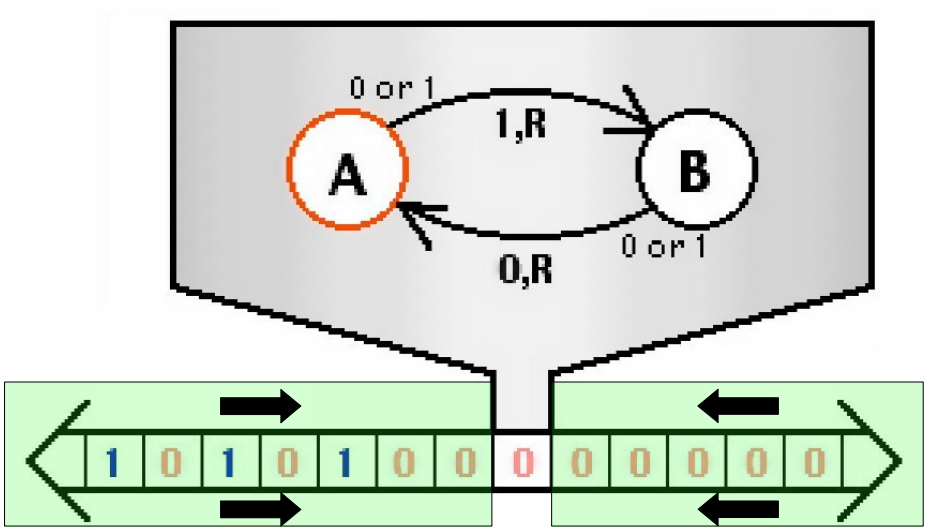
- Two Stacks ≡ One Tape
  - arrange two stacks (with tops facing) on either side of the read/write head
  - to move the machine along the tape, pop one stack and push the other



Stack Op.	Machine Motion	Stack Op.
Push	Right	Pop
Pop	Left	Push

A Turing Machine in the Game of Life

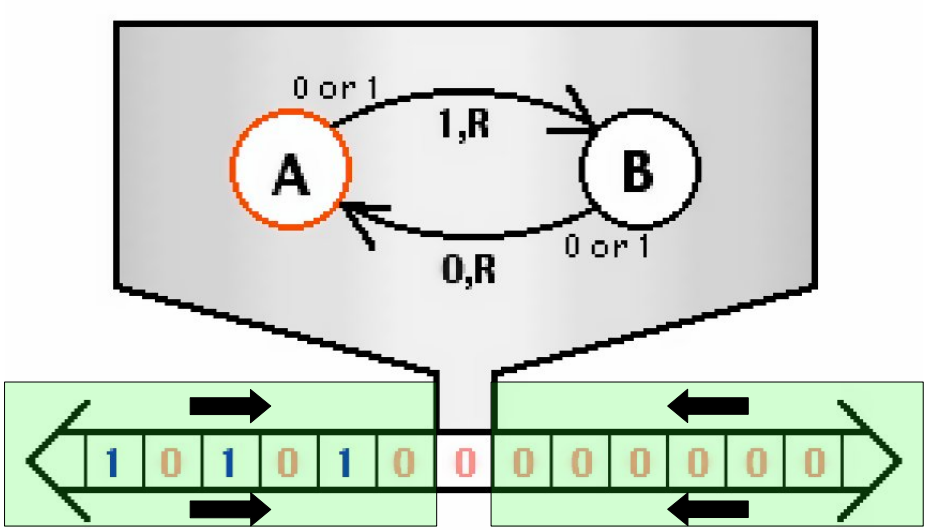
- Two Stacks ≡ One Tape
  - arrange two stacks (with tops facing) on either side of the read/write head
  - to move the machine along the tape, pop one stack and push the other



Stack Op.	Machine Motion	Stack Op.
Push	**Right**	**Pop**
Pop	Left	Push

A Turing Machine in the Game of Life

- Two Stacks ≡ One Tape
  - arrange two stacks (with tops facing) on either side of the read/write head
  - to move the machine along the tape, pop one stack and push the other

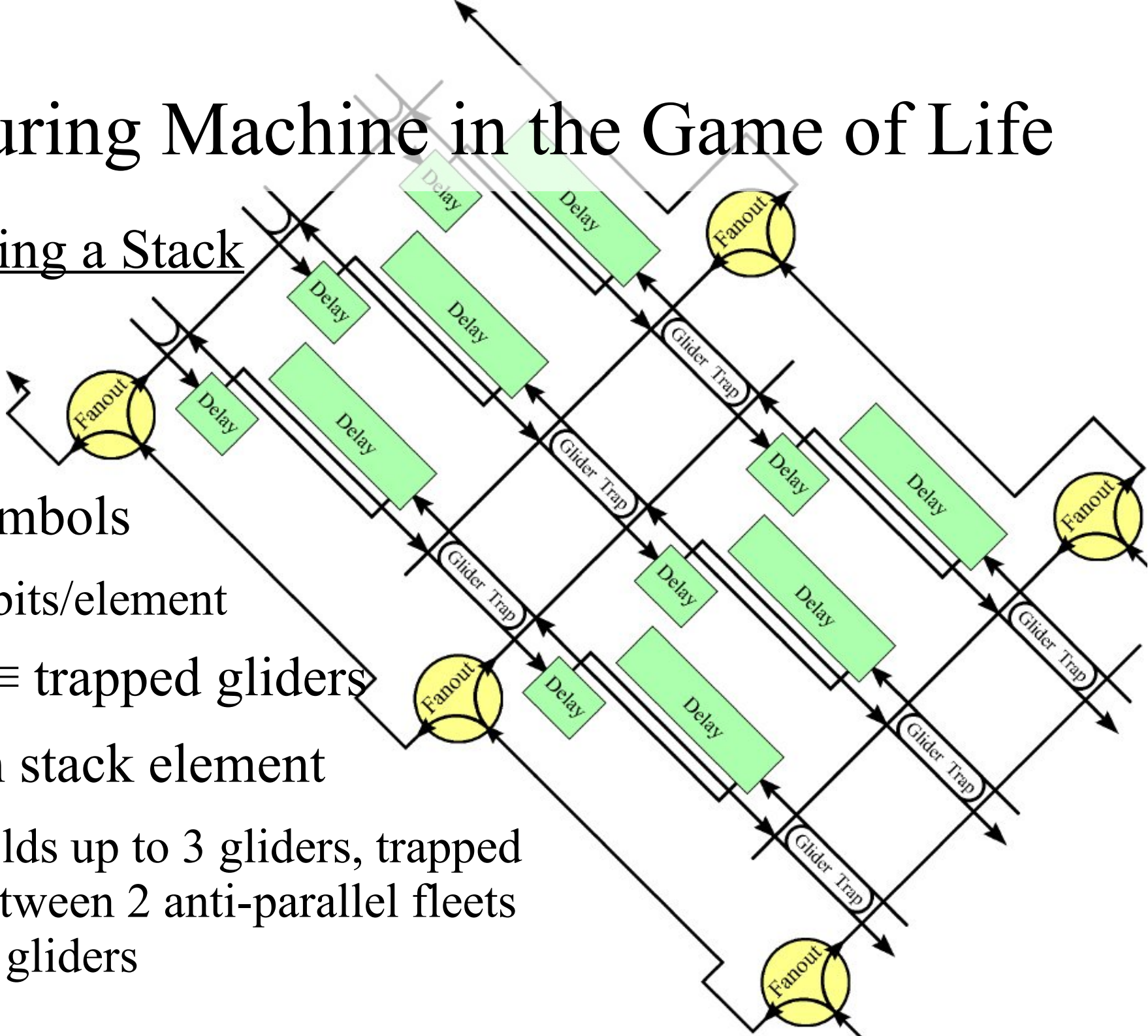


Stack Op.	Machine Motion	Stack Op.
Push	Right	Pop
Pop	**Left**	**Push**

A Turing Machine in the Game of Life

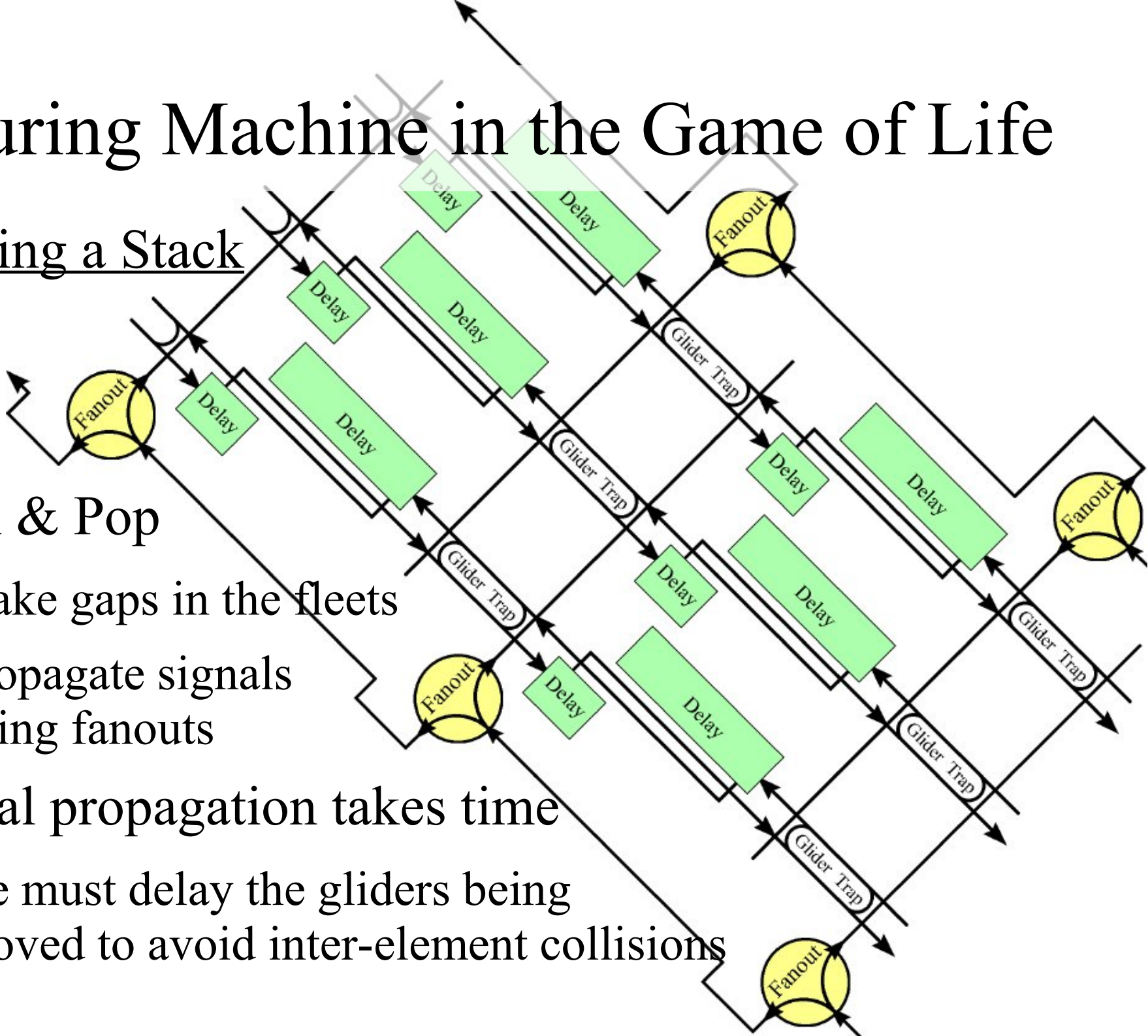
- Making a Stack

- 8 Symbols
  - 3 bits/element
- bits ≡ trapped gliders
- Each stack element
  - holds up to 3 gliders, trapped between 2 anti-parallel fleets of gliders



A Turing Machine in the Game of Life

- Making a Stack
- Push & Pop
  - make gaps in the fleets
  - propagate signals using fanouts
- Signal propagation takes time
  - we must delay the gliders being moved to avoid inter-element collisions



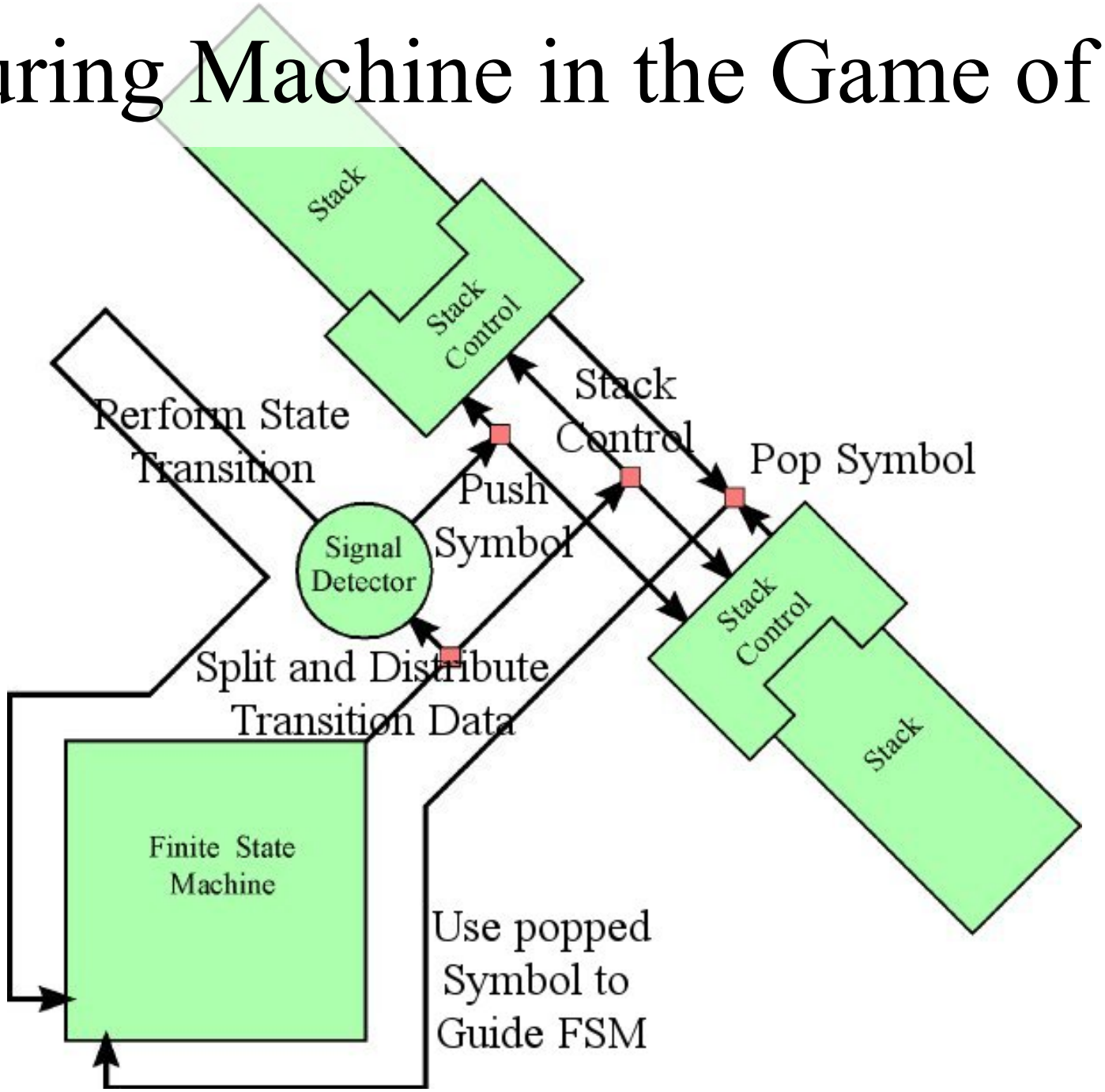
A Turing Machine in the Game of Life

- What do we need?
  - states ✓
  - transition rules ✓
  - a tape ✓
  - possible symbols for each cell in the tape ✓
  - cell read/write access ✓
  - multiple-cell access ✓

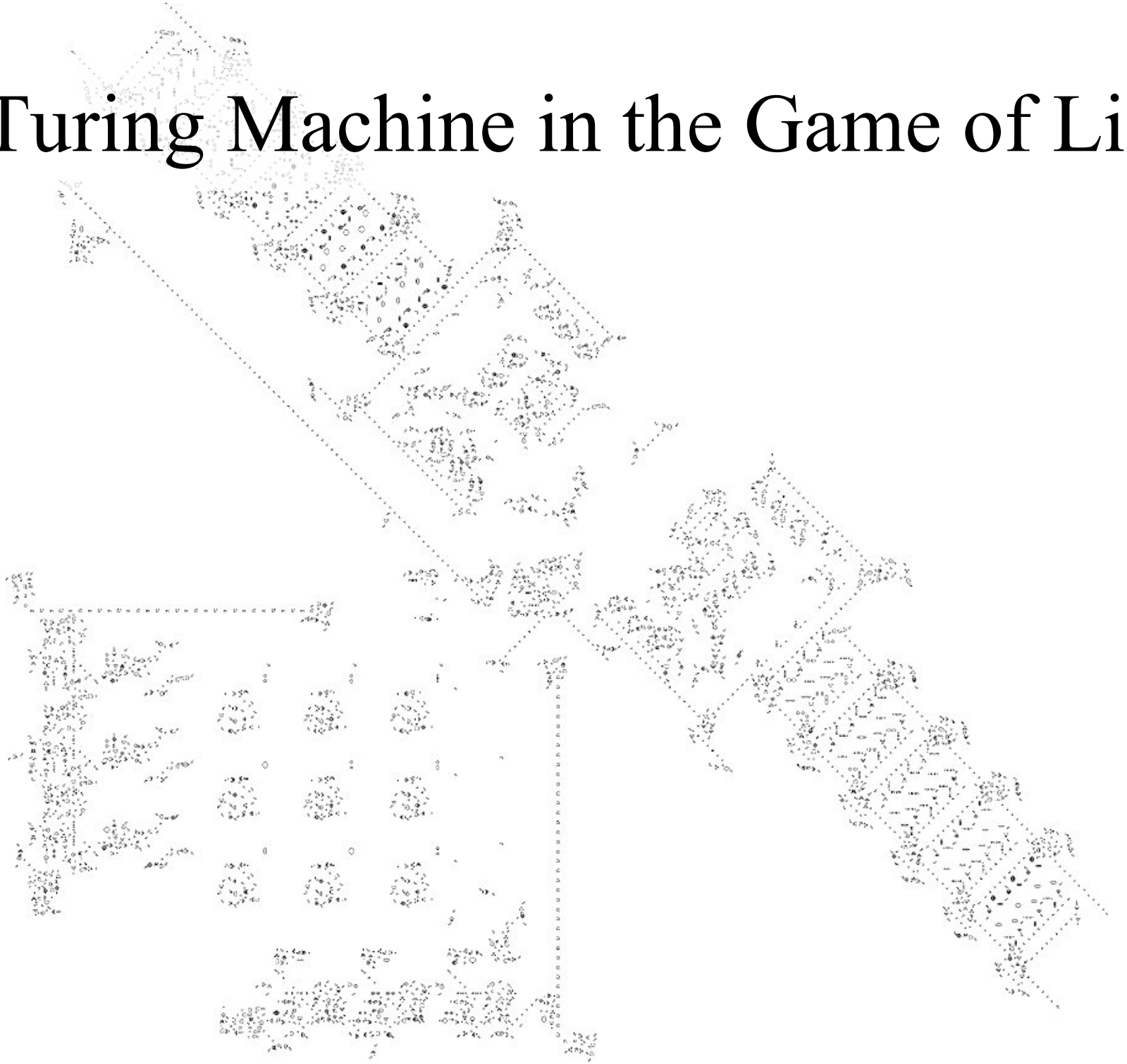
Is that really everything?

How does it all connect?

A Turing Machine in the Game of Life



A Turing Machine in the Game of Life



Decision Making

- At a basic level:
  - if your neighbours change, follow the rules of the world
    - the rules decide what happens
- At a higher level:
  - if a Turing Machine can make a decision, then so can a Cellular Automata
    - we build a Universal Turing Machine in the Cellular Automata, and feed it our deciding Turing Machine plus its input

Strengths & Weaknesses

- Weaknesses of Rendell's Design:
  - we already knew that it could be done
  - non-infinite tape
  - “simple”, but still quite complicated
- Strengths of Rendell's Design:
  - for many, seeing is believing – confirms the proof
  - easy extensions (# of cells and length of tape)
  - could be extended to a Universal Turing Machine

Strengths & Weaknesses

- Weaknesses of Cellular Automata:
  - choosing a useful set of rules can be hard
  - simulating enough cells to be interesting can be very computationally intensive
- Strengths of Cellular Automata:
  - massive parallelism
  - helps us see “the big picture”

References

- Rendell, Paul. “Turing Universality in the Game of Life”. Chapter Draft for Collision-based Computing. On the Internet: http://www.cs.ualberta.ca/~bulitko/F05/CMPUT651/papers/tm.pdf
- Details of a Turing Machine in Conway's Game of Life
  - http://seashell-bb.co.uk/life/gol/tmdetails.htm

References

- John von Neumann's Universal Constructor
  - http://www.sq3.org.uk/Evolution/JvN/
- Moshe Sipper, A Brief Introduction To Cellular Automata
  - http://www.cs.bgu.ac.il/~sipper/ca.html
- Cellular Automata
  - http://cscs.umich.edu/~crshalizi/notebooks/cellular-automata.html"
- John Conway
  - http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Conway.html"
- Igblan - Life Universal Computer
  - http://www.igblan.free-online.co.uk/igblan/ca/"

Questions?