بسم الله الرحمن الرحیم

نظریه علوم کامپیوتر

نظریه علوم کامپیوتر - بهار ۱۴۰۰-۱۴۰۱ - جلسه بیست و یکم: محاسبات نامتداول

Theory of computation - 002 - S21 - Unconventional Computing

What is computation?

- A computer is a physical system whose:
  - physical states can be seen as representing elements of another system of interest
  - transitions between states can be seen as operations on these elements

- Three basic steps:
  1. Input data is coded into a form appropriate for physical system
  2. Physical system shifts into new states and finally, to an output state
  3. Output state of system is decoded to extract results of computation

- We can now look back and see how these 3 steps are instantiated in silicon, DNA, neural, and quantum computers.

Digital Computing: Rapid serial computing in silicon

- Basic Mechanism: Silicon switches implement Boolean logic circuits and manipulate binary variables with near-zero error
- Main Features: Hierarchical approach allows extremely fast general-purpose sequential computing:
  - Transistors → switches → gates → combinational and sequential logic → finite-state behavior → … → sequential algorithm
- Moore’s law of exponential technology scaling: Chip complexity (transistor density) has doubled every 1.5 years, as “feature” sizes on a chip keep decreasing; Clock frequencies have doubled every ~3 years

Digital Computing: Problems and Projections

- Problems: Approaching physical, practical, and economic limits.
  - *Photolithography*: Component sizes (~ 0.1 µm) getting close to the wavelength of light used for etching
  - *Tunneling and other quantum effects*: atomic scale of components cause current leakage, corrupting the circuit…
  - *Clock speed too high*: signals can only travel a fraction of a mm in one cycle – can’t reach all components…
  - *Economics*: Chip fabrication is becoming too expensive, while transistors are becoming too cheap…

- Reasonable projections: Moore’s law may continue for the next 10-15 years (at most, until 2020):
  - Minimum predicted feature size: 0.03µm, to yield 1 billion transistors on a standard 15mm×15mm silicon die
  - Projected clock rate at 0.03µm: 40GHz

DNA Computing: Parallel computing by molecules

- Basic Mechanism: Biochemical properties and microscopic scales of organic molecules allow massively parallel solutions to hard search problems

- Main Features: Basic steps in DNA computation:
  - *Encode*: Map problem onto DNA strands using the alphabet (A,C,T,G)
  - *Exhaustive Search*:
    - Generate all possible solutions by subjecting strands simultaneously to biochemical reactions
    - Use molecular techniques to eliminate invalid solutions
  - *The result: Turing Universal DNA computing*
  - *Application:* Can solve NP-complete problems (e.g. TSP) for problem sizes that are too large to solve on current digital computers

DNA Computing: Problems and Future Directions

- Problems:
  - Scaling: Standard exhaustive search approach does not scale well
    - Polynomial time solutions but exponential volume of DNA
      - 2^{70} DNA strands of length 1000 = 8 kilograms
  - DNA processing is slow, cumbersome, and error prone
    - Few seconds to 1 hour or more per reaction
    - Approximate matches and mutations may give incorrect results
- Future Directions:
  - Directed self-assembly of solutions rather than exhaustive search
    - Cuts down on volume (Winfree, Seeman, and others)
  - Surface-based DNA computing: Allows more control of individual strands and reactions, and facilitates automation, at the cost of few total number of DNA strands for problem solving.

Neural Computing: Emulating the brain

- Basic Mechanism: Distributed networks of neuron-like units compute parallel, adaptive, and fault-tolerant solutions to hard pattern recognition and control problems

- Main Features: Non-linear mappings between inputs and outputs are learned from examples by adjusting connection weights; network generalizes and can compute outputs for novel inputs.

- Problems:
  - *Scaling*: Simulating large networks is still computationally infeasible
  - *Picking parameters* (e.g. no. of units, learning rate) is still an art

- Future Directions:
  - Hardware implementations in VLSI: may allow scaling to large sizes
  - Probabilistic methods (e.g. Bayesian techniques) provide a principled approach to picking network parameters and to learning & inference.

Quantum Computing: Parallel computation in quantum systems

- Basic Mechanism: Parallel computation along all possible computational paths, with appropriate interference of probability amplitudes, allows exponential speedup of solutions to some search problems

- Main Features: Problem instances encoded as states of a quantum system (e.g. spins of *n* electrons, polarization values of *n* photons etc.)

  E.g. 2 bit states of 2 electrons = |00>, |01>, |10>, or |11>

  1. The system is put into a superposition of all possible states, each weighted by its probability amplitude (= a complex number c_i)

     E.g. *Qubits* for 2 electrons = c_1 |00> + c_2 |01> + c_3 |10> + c_4 |11>

  2. The system evolves according to quantum principles:
     1. *Unitary matrix operation*: describes how superposition of states evolves over time when no measurement is made
     2. *Measurement operation*: maps current superposition of states to one state based on probability = square of amplitudes c_i

     E.g. probability of seeing output bits (00) is $| c_1|^2$

Quantum Computing: Problems and Future Directions

- Problems:
  - *Decoherence*: Environmental noise may inadvertently "measure" the system, thereby disturbing the computation
    - Error correcting codes may help ([Shor et al.])
  - *Scaling*: All physical implementations so far (NMR, Cavity QED, etc.) have failed to scale beyond a few qubits.

- Future Directions:
  - Hardware Implementations: New physical substrates are needed that allow manipulations of large numbers of qubits (superpositions of states) with little or no decoherence
  - New Algorithms: New ways of exploiting quantum parallelism are needed that allow solutions to NP-complete problems

The Future of Computing: Some Predictions

- From *Visions: How science will revolutionize the 21st century* (1997) by Michio Kaku, Henry Semat Professor of Theoretical Physics at City College of New York and co-founder of string theory.

- By 2020:
  - Microprocessors will become as cheap as scrap paper (< 1 cent/ processor)
  - Invisible/ubiquitous computing in all appliances: smart homes, smart clothes, smart jewelry, smart shoes etc.
  - Intelligent appliances that listen, sense, communicate, and act
  - Internet creates an “intelligent planet” akin to a “Magic Mirror” that stores the “wisdom of the human race.”
  - End of the silicon age: microchip components cannot be made smaller without taking into account quantum effects

The Future of Computing: Some Predictions

- By 2050:
  - Physical implementation of alternative computing models
    - Optical computers
    - Molecular, DNA, and Quantum computers
  - Holographic 3D monitors
  - Molecular machines and nanotechnology
  - Robotic automatons with common sense, human-like vocabulary and conversation skills, ability to learn from mistakes
- By 2100:
  - Robots achieve “self-awareness” and consciousness
    - Can work as secretaries and assistants
  - Quantum theory and nanotechnology allow duplication of neural patterns of the brain on a computer
  - Biotechnology and computer technology allow humans to “merge” with their computerized creations

“We are very near to the time when virtually no essential human function, physical or mental, will lack an artificial counterpart…machines (will) carry on our cultural evolution, including their own construction and increasingly rapid self-improvement…our DNA will find itself out of a job, having lost the evolutionary race to a new kind of competition.”

-- Hans Moravec (*Mind Children*, 1988)

"We are very near to the time when virtually no essential human function, physical or mental, will lack an artificial counterpart…machines (will) carry on our cultural evolution, including their own construction and increasingly rapid self-improvement…our DNA will find itself out of a job, having lost the evolutionary race to a new kind of competition."

-- Hans Moravec (*Mind Children*, 1988)

"Prediction is very hard, especially when it's about the future."

--Yogi Berra

Intrinsic Universal Family of Signal Machines

Mohammad Hadi Foroughmand Araabi

Sharif University of Technology

Joint work with:
Florent Becker, Jèrôme Durand-Lose, and Sama Goliaei

12 April 2017

Signal Machine: Definition and Examples

Idea of Signal in Computation via Cellular Automata

Signals in Cellular Automata ([Goto, 1996])

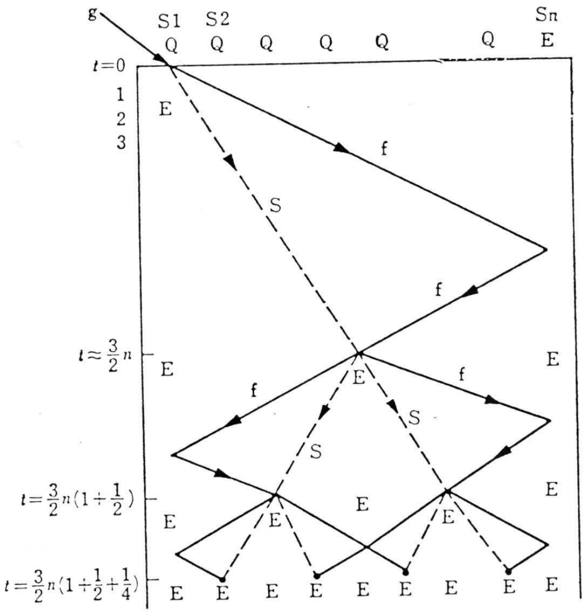
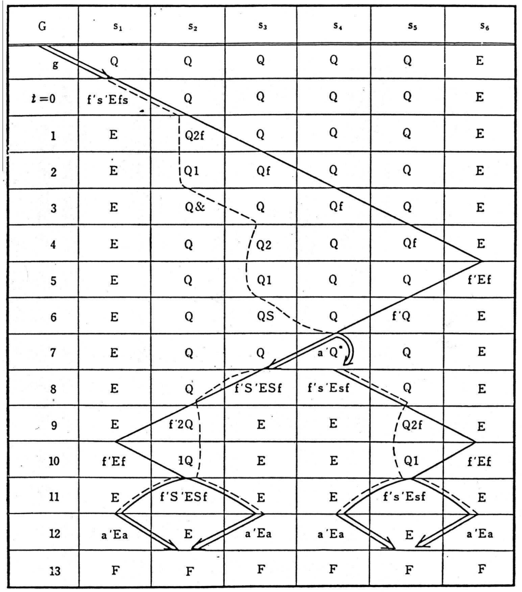

g
S1
S2
Sn
Q
Q
Q
Q
Q
Q
E
$t=0$
1
2
3
E
f
S
f
$t\approx\frac{3}{2}n$
E
E
f
f
E
S
S
$t=\frac{3}{2}n(1+\frac{1}{2})$
E
E
E
E
E
$t=\frac{3}{2}n(1+\frac{1}{2}+\frac{1}{4})$
E
E
E
E
E
E
E
E
E

図 3·5　一斉射撃の問題（連続近似）

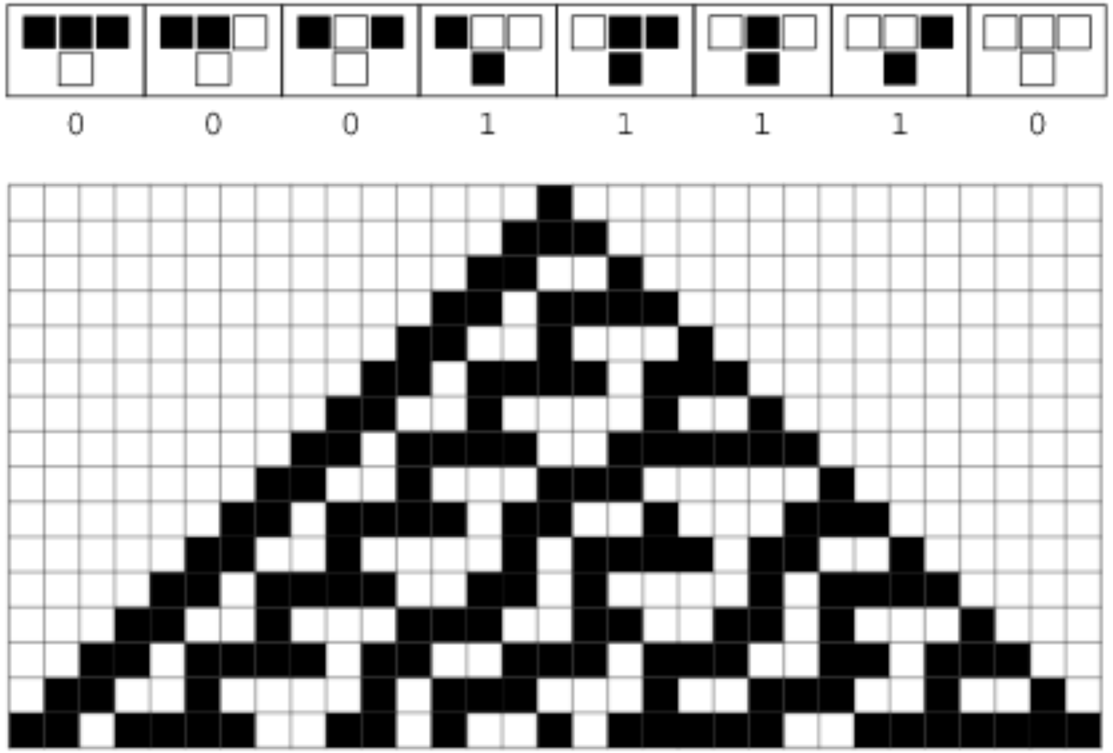


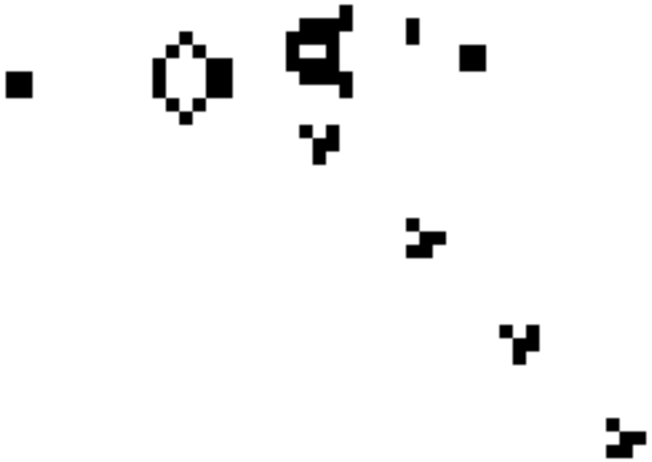
G	s_1	s_2	s_3	s_4	s_5	s_6
g	Q	Q	Q	Q	Q	E
$t=0$	f's'Efs	Q	Q	Q	Q	E
1	E	Q2f	Q	Q	Q	E
2	E	Q1	Qf	Q	Q	E
3	E	Q&	Q	Qf	Q	E
4	E	Q	Q2	Q	Qf	E
5	E	Q	Q1	Q	Q	f'Ef
6	E	Q	QS	Q	f'Q	E
7	E	Q	Q	a'Q*	Q	E
8	E	Q	f'S'ESf	f's'Esf	Q	E
9	E	f'2Q	E	E	Q2f	E
10	f'Ef	1Q	E	E	Q1	f'Ef
11	E	f'S'ESf	E	E	f's'Esf	E
12	a'Ea	E	a'Ea	a'Ea	E	a'Ea
13	F	F	F	F	F	F

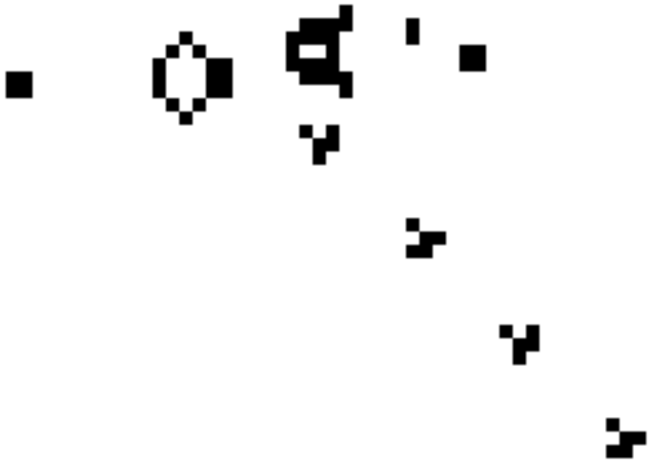
図 3·6　一斉射撃解（$n=6$）

Cellular automata

rule 30







Idea of Signal in Computation via Cellular Automata

Signals in Cellular Automata ([Mazoyer, 1996, Fig. 8 and 19] and [Mazoyer and Terrier, 1999, Fig. 18])

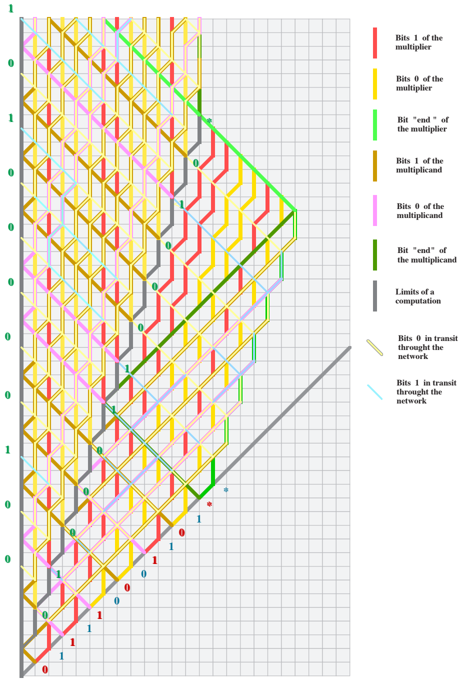

Bits 1 of the multiplier
Bits 0 of the multiplier
Bit "end " of the multiplier
Bits 1 of the multiplicand
Bits 0 of the multiplicand
Bit "end" of the multiplicand
Limits of a computation
Bits 0 in transit throught the network
Bits 1 in transit throught the network

Figure 8: Computing $(ab)^2$.

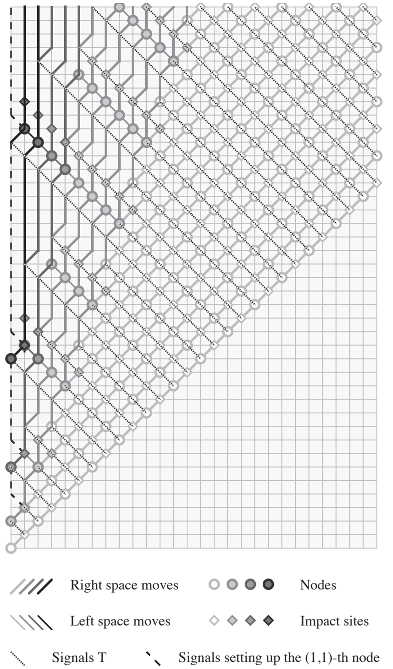


Figure 9: Setting up an infinite family of regular safe grids (the darkness of the grid indicates its rank).

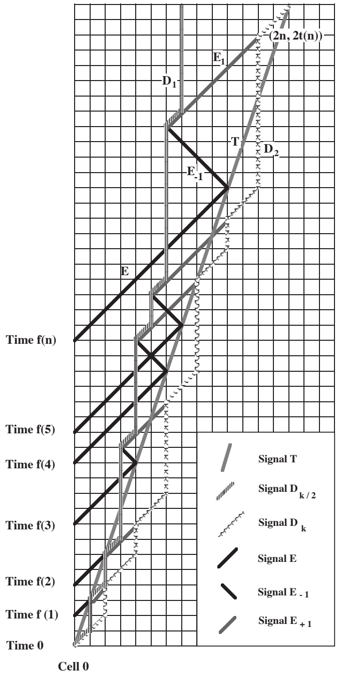
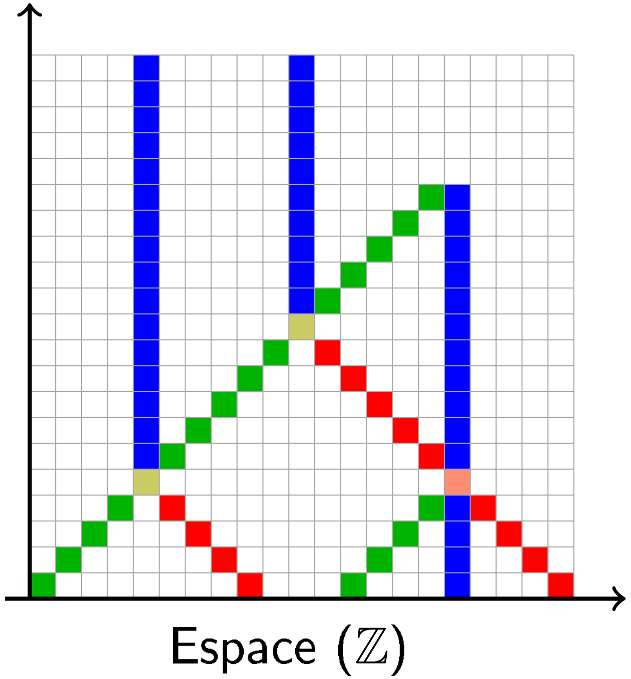
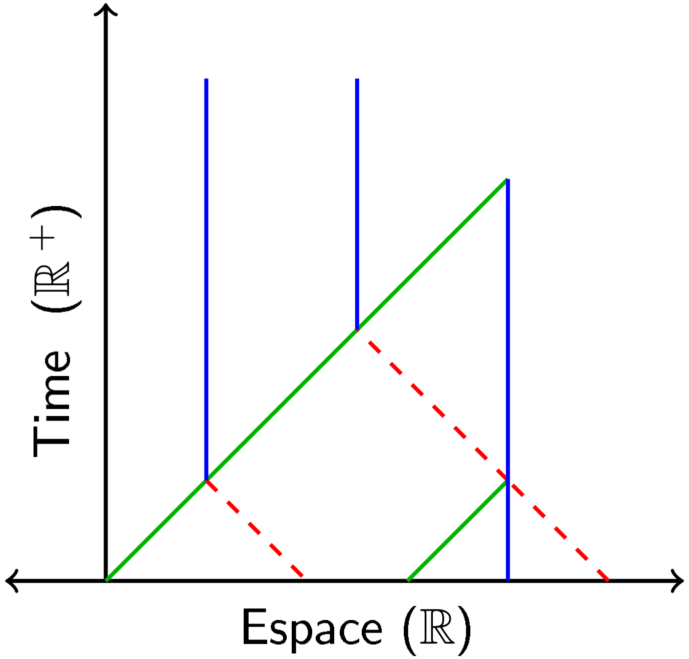


Figure 18: Characterization of the sites $(n, f(n))$.

Signal Machine Informal Definition

Signal Machine: A **Continuous** Generalization of Signals in Cellular Automata





Signal Machine Informal Definition, An Example

A signal machine is defined by:

- 1-Dimensional Space
- A set of meta-signals (with specified speeds)
- A set of collision rules

For computation:

- Initial configuration (Input)
- Space-Time diagram (Trace of computation)

Signal Machine Informal Definition, An Example

A signal machine is defined by:

- 1-Dimensional Space
- A set of meta-signals (with specified speeds)
- A set of collision rules

For computation:

- Initial configuration (Input)
- Space-Time diagram (Trace of computation)

Meta-signals:

M, $S(\text{M}) = 0$
div, $S(\text{div}) = 3$
hi, $S(\text{hi}) = 1$
lo, $S(\text{lo}) = 3$
back, $S(\text{back}) = -3$

Signal Machine Informal Definition, An Example

A signal machine is defined by:

- 1-Dimensional Space
- A set of meta-signals (with specified speeds)
- A set of collision rules

For computation:

- Initial configuration (Input)
- Space-Time diagram (Trace of computation)

Meta-signals:

M, $S(\text{M}) = 0$
div, $S(\text{div}) = 3$
hi, $S(\text{hi}) = 1$
lo, $S(\text{lo}) = 3$
back, $S(\text{back}) = -3$

Rules:

$\{ \text{ div, M } \} \rightarrow \{ \text{ M, hi, lo } \}$
$\{ \text{ lo, M } \} \rightarrow \{ \text{ back, M } \}$

Signal Machine Informal Definition, An Example

A signal machine is defined by:

- 1-Dimensional Space
- A set of meta-signals (with specified speeds)
- A set of collision rules

For computation:

- Initial configuration (Input)
- Space-Time diagram (Trace of computation)

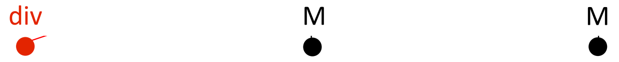
Meta-signals:

M, $S(\text{M}) = 0$
div, $S(\text{div}) = 3$
hi, $S(\text{hi}) = 1$
lo, $S(\text{lo}) = 3$
back, $S(\text{back}) = -3$

Rules:

$\{$ div, M $\} \rightarrow \{$ M, hi, lo $\}$
$\{$ lo, M $\} \rightarrow \{$ back, M $\}$

Initial Configuration:



Signal Machine Informal Definition, An Example

A signal machine is defined by:

- 1-Dimensional Space
- A set of meta-signals (with specified speeds)
- A set of collision rules

For computation:

- Initial configuration (Input)
- Space-Time diagram (Trace of computation)

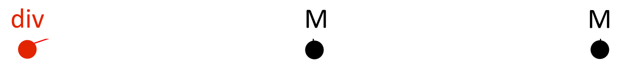
Meta-signals:

M, $S(\text{M}) = 0$
div, $S(\text{div}) = 3$
hi, $S(\text{hi}) = 1$
lo, $S(\text{lo}) = 3$
back, $S(\text{back}) = -3$

Rules:

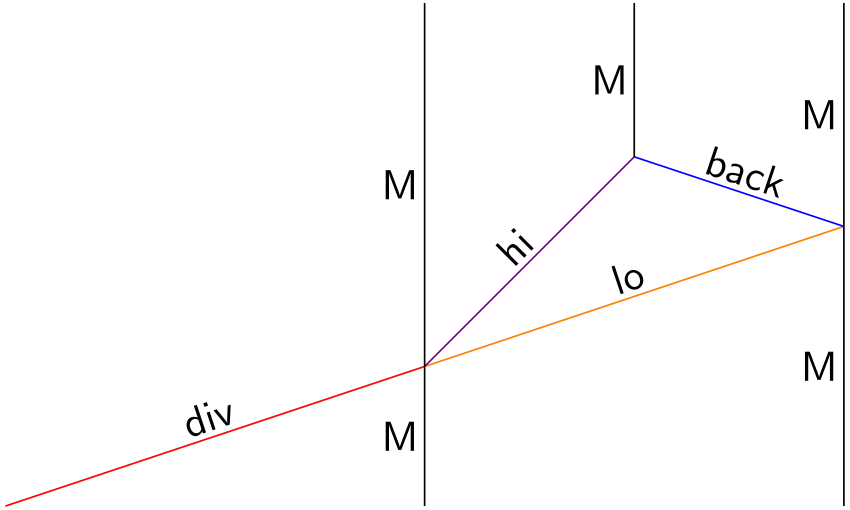
$\{\text{ div, M }\} \rightarrow \{\text{ M, hi, lo }\}$
$\{\text{ lo, M }\} \rightarrow \{\text{ back, M }\}$

Initial Configuration:



Signal Machine Informal Definition, An Example

Space-Time diagram:



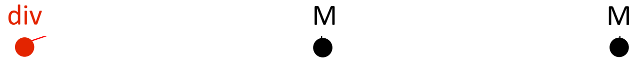
Meta-signals:

M, $S(\mathrm{M}) = 0$
div, $S(\mathrm{div}) = 3$
hi, $S(\mathrm{hi}) = 1$
lo, $S(\mathrm{lo}) = 3$
back, $S(\mathrm{back}) = -3$

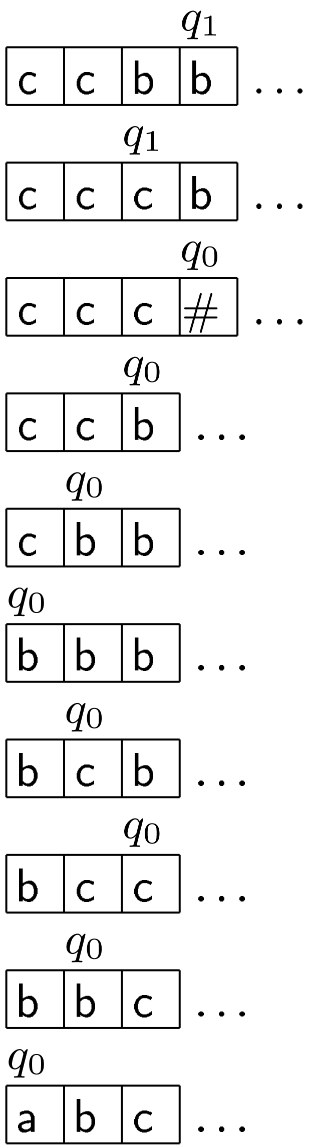
Rules:

$\{\ \mathrm{div},\ \mathrm{M}\ \} \rightarrow \{\ \mathrm{M},\ \mathrm{hi},\ \mathrm{lo}\ \}$
$\{\ \mathrm{lo},\ \mathrm{M}\ \} \rightarrow \{\ \mathrm{back},\ \mathrm{M}\ \}$

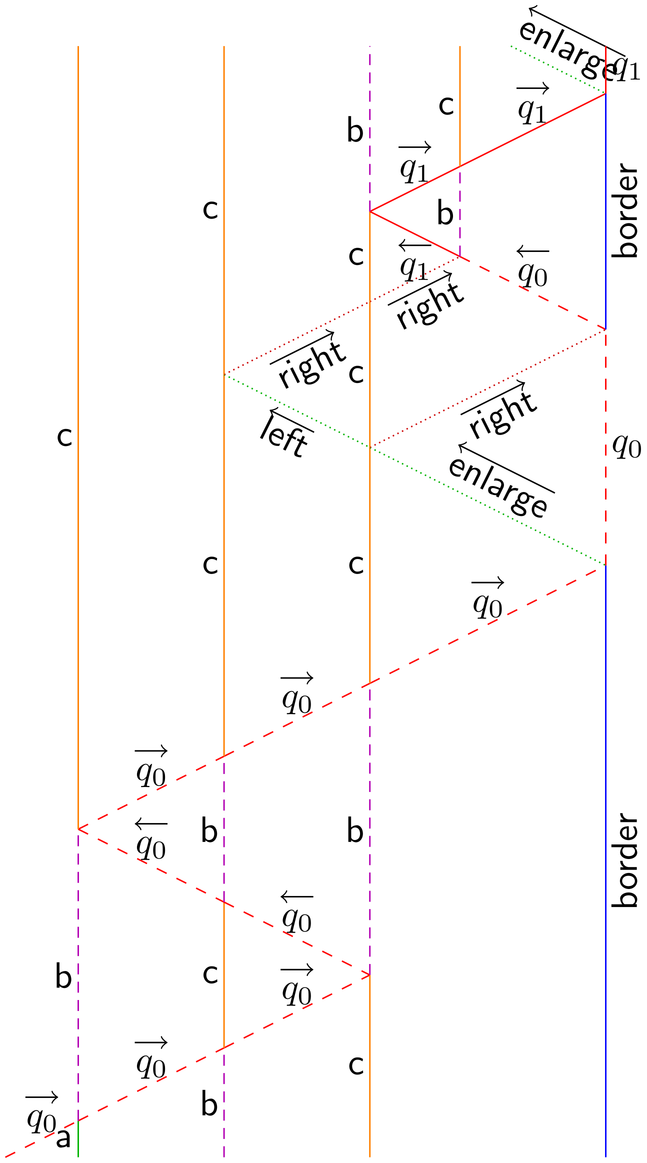
Initial Configuration:



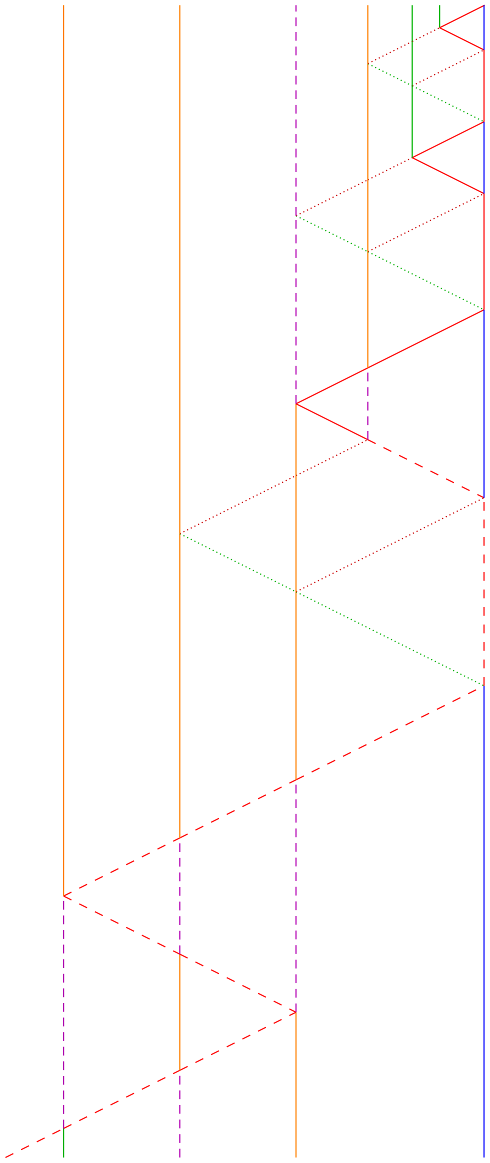
Simulating Turing machine



(a) Turing machine run



(b) Labeled simulation



(c) Longer simulation

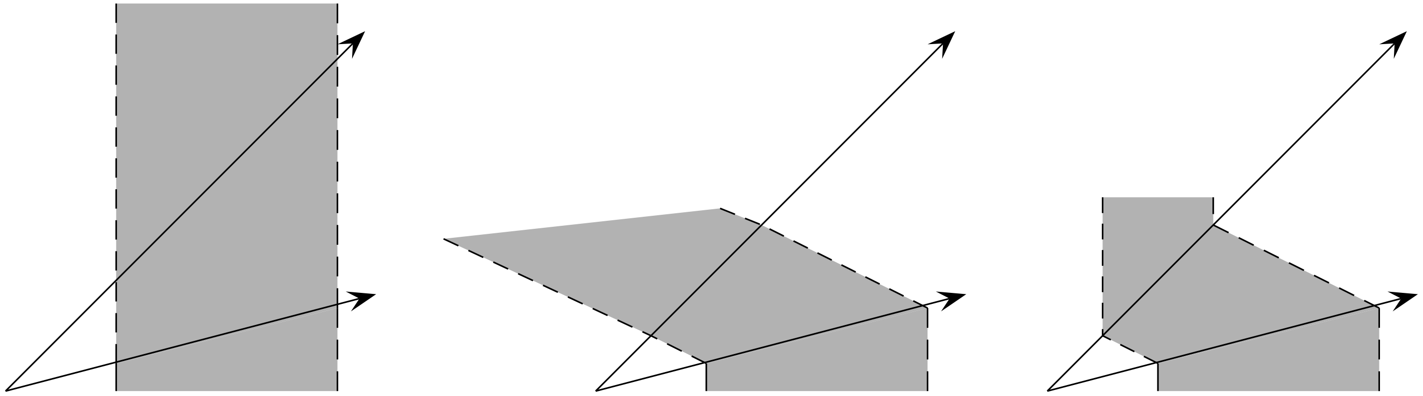
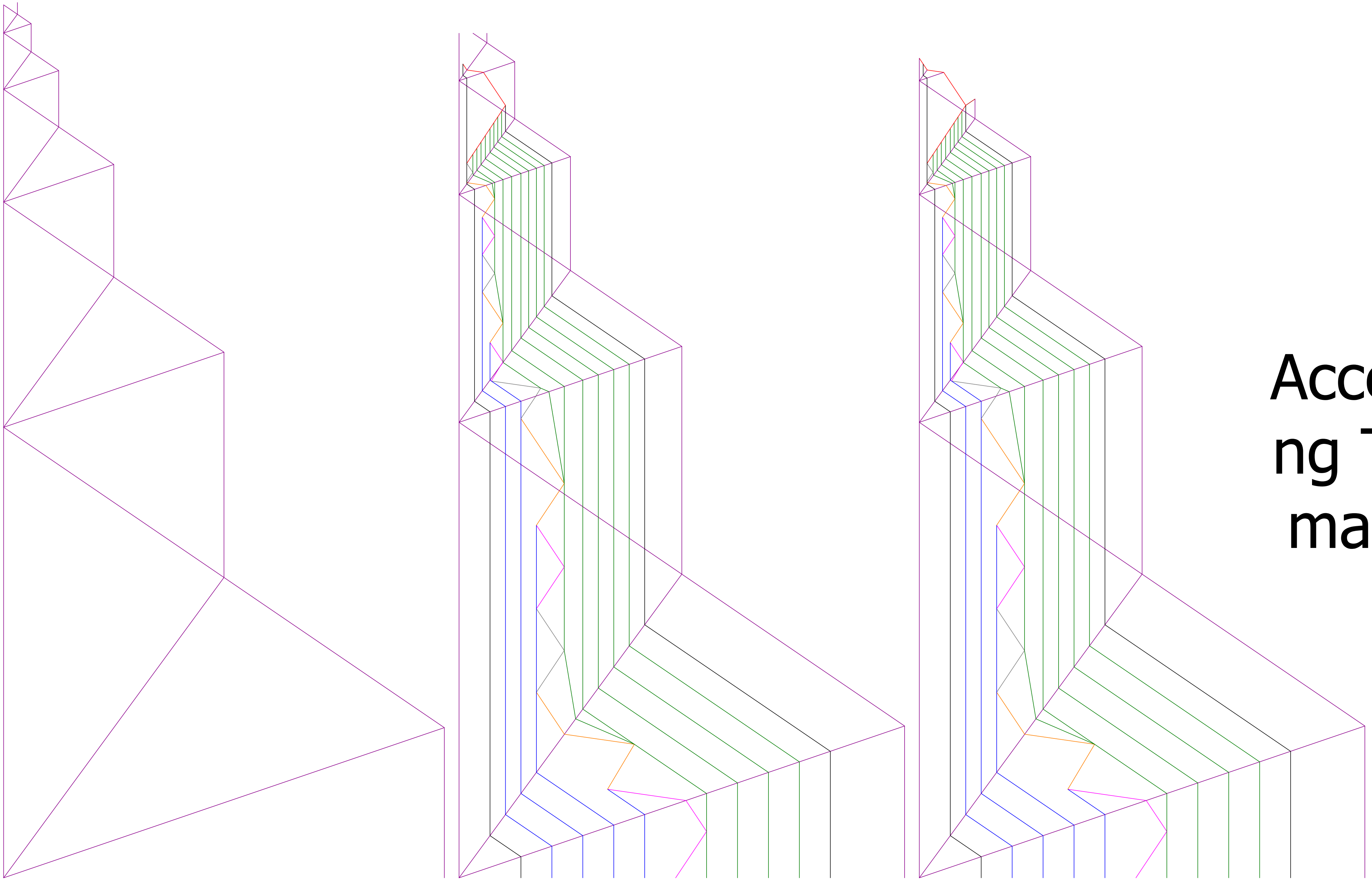


Figure 4: Shrinking principle.

Accelerating Turing machine



Accelerati
ng Turing
machine

Known Results

- Without accumulation (the case for this talk):
  - Rational signal machines are equivalent to Turing machines
  - (Irrational) Signal machines are equivalent to linear-BSS model [DL07]
- With accumulation
  - Signal machines can solve the halting problem of Turing machines [DL12]
  - Signal machines can simulate Type-2 Turing machines [DL11]
  - Signal machines can simulate BSS machines [DL08]