Self-Assembly of Decidable Sets

Matthew J. Patitz, and Scott M. Summers

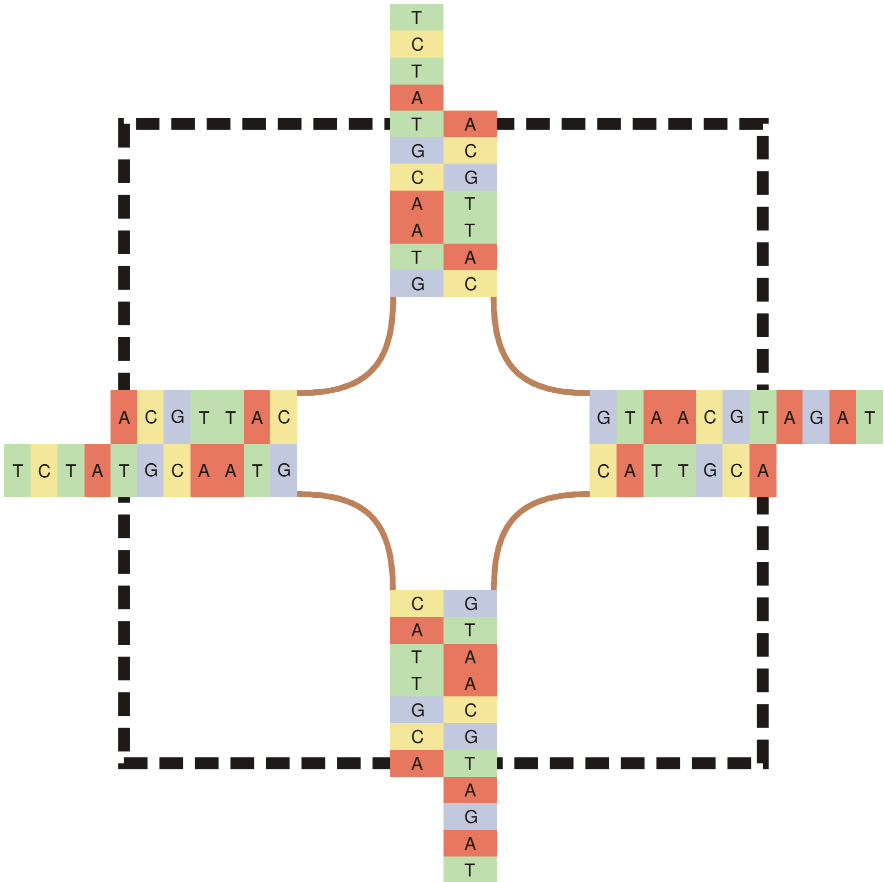
Iowa State University, USA

Outline of Presentation

- Overview of Tile Assembly Model
- The wedge construction
- A new characterization of decidable languages
- Analysis of space requirements

DNA Tile Self-Assembly
Seeman, starting in 1980s

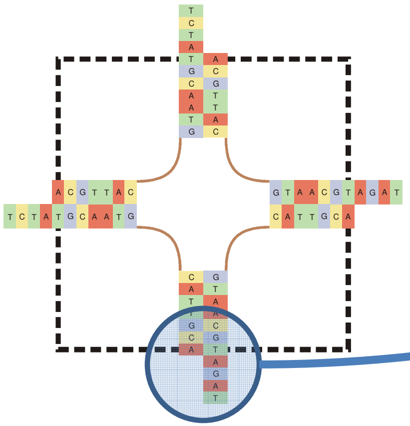
DNA tile, oversimplified:



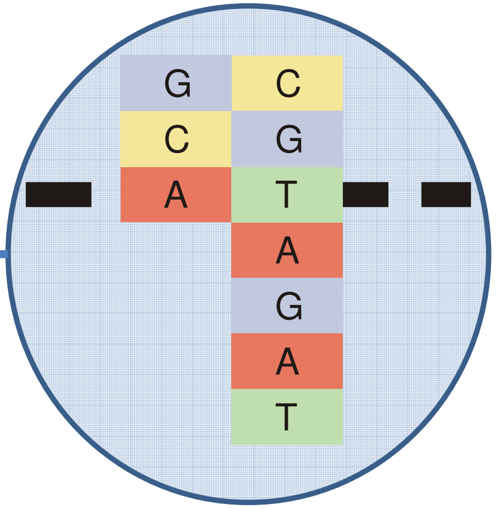
Four single DNA strands bound by Watson-Crick pairing (A-T, C-G).

DNA Tile Self-Assembly
Seeman, starting in 1980s

DNA tile, oversimplified:



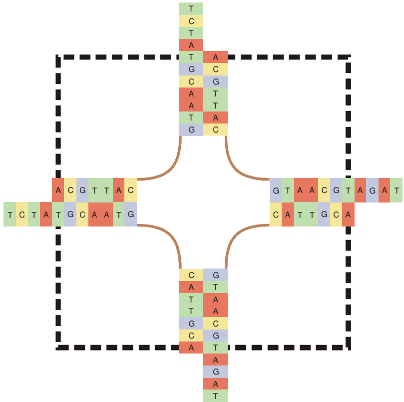
Four single DNA strands bound by Watson-Crick pairing (A-T, C-G).



“Sticky ends” bind with their Watson-Crick complements, so that a regular array self-assembles.

DNA Tile Self-Assembly
Seeman, starting in 1980s

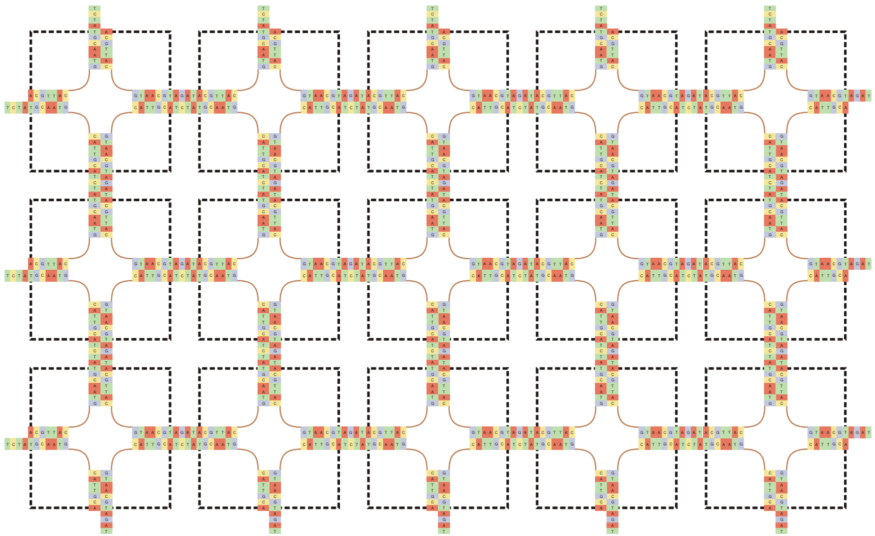
DNA tile, oversimplified:



Four single DNA strands bound by Watson-Crick pairing (A-T, C-G).

Choice of sticky ends allows one to *program* the pattern of the array.

“Sticky ends” bind with their Watson-Crick complements, so that a regular array self-assembles.



DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

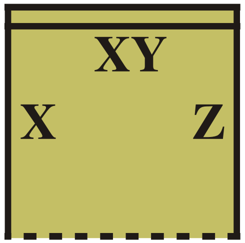
DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001

- Tile = unit square

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001



- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).

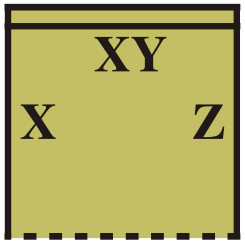
Strength 0

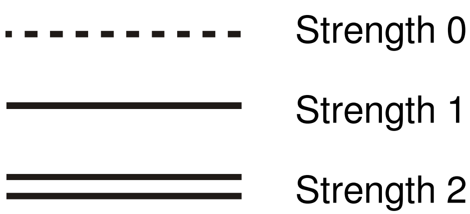
Strength 1

Strength 2

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001

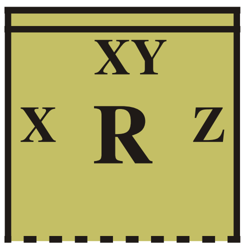




- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001



Strength 0

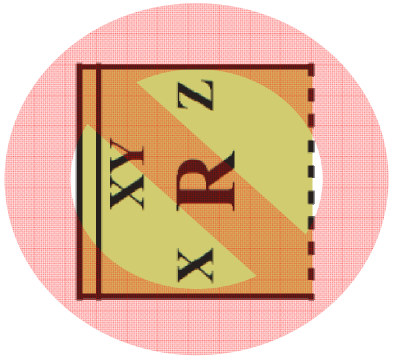
Strength 1

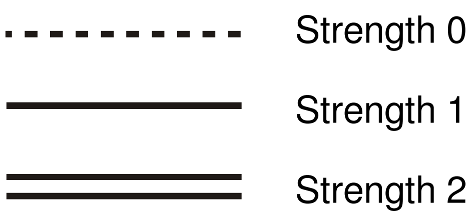
Strength 2

- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001

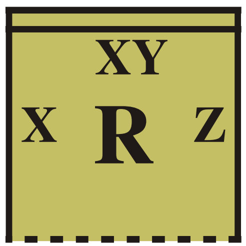


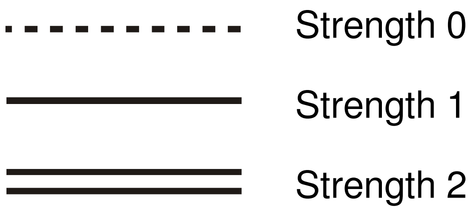


- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.
- Tiles cannot be rotated.

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001



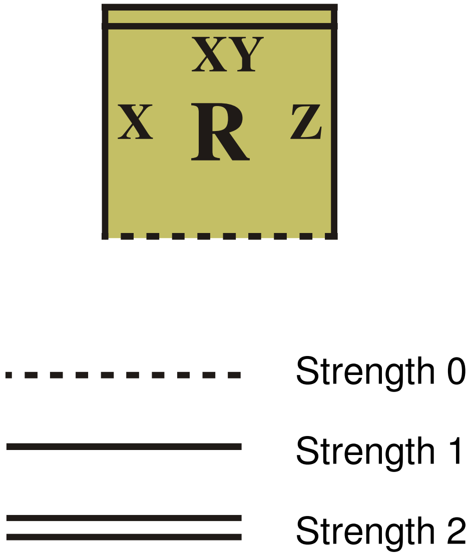


- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.
- Tiles cannot be rotated.

- Finitely many tile types

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001

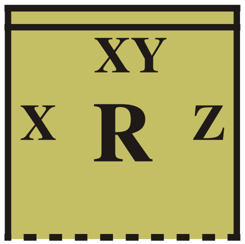


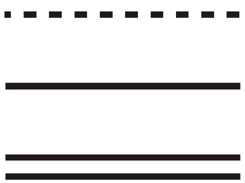
- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.
- Tiles cannot be rotated.

- Finitely many tile types
- Infinitely many tiles of each type available

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001





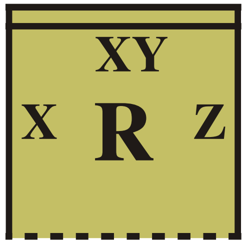
Strength 0
Strength 1
Strength 2

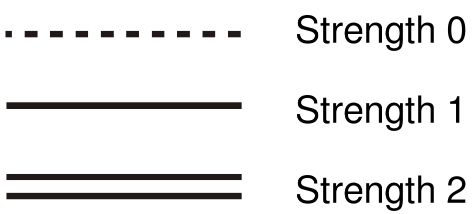
- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.
- Tiles cannot be rotated.

- Finitely many tile types
- Infinitely many tiles of each type available
- Assembly starts from a *seed tile* (or *seed* assembly).

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001



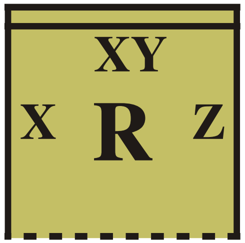


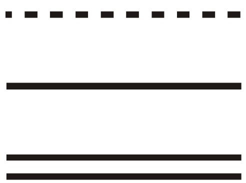
- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.
- Tiles cannot be rotated.

- Finitely many tile types
- Infinitely many tiles of each type available
- Assembly starts from a *seed tile* (or *seed* assembly).
- A tile can attach to the existing assembly if it binds with total strength at least 2 (the "temperature").

DNA Tile Self-Assembly
Winfree, Ph.D. thesis, 1998

Extension of Wang tiling, 1961
Refined in Paul Rothemund's Ph.D. thesis, 2001





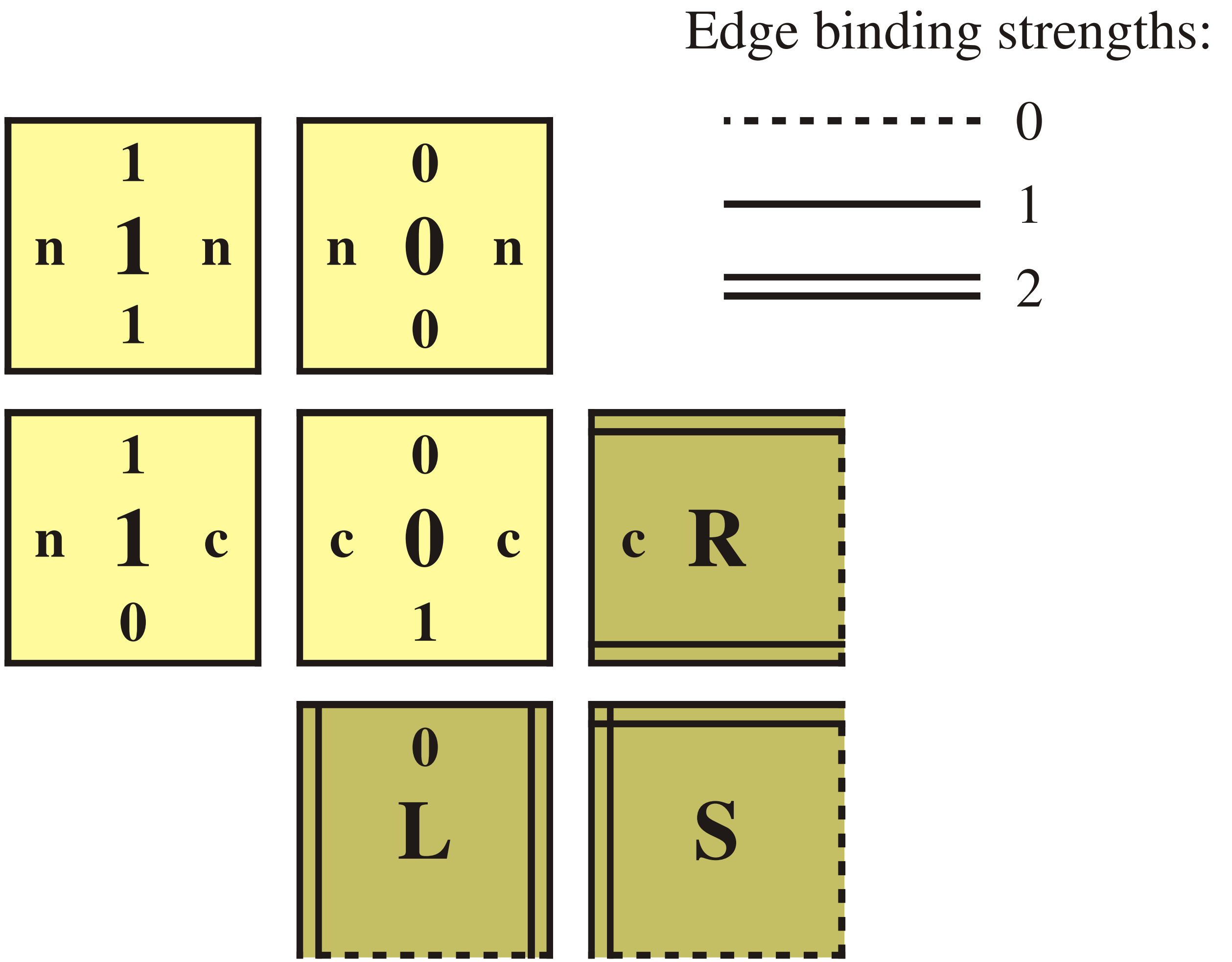
Strength 0
Strength 1
Strength 2

- Tile = unit square
- Each side has *glue* of certain *kind* and *strength* (0, 1, or 2).
- If tiles abut with matching kinds of glue, then they bind with this glue's strength.
- Tiles may have labels.
- Tiles cannot be rotated.

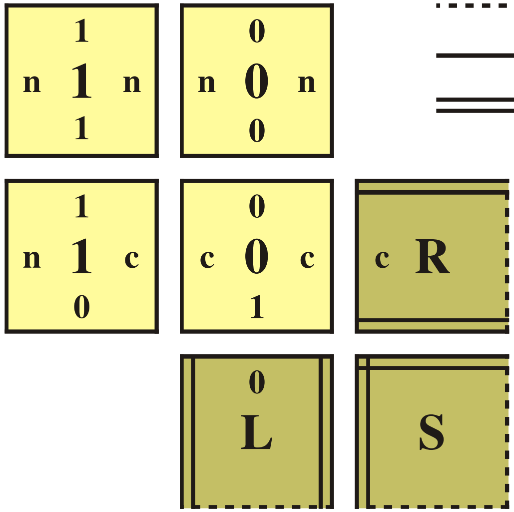
- Finitely many tile types
- Infinitely many tiles of each type available
- Assembly starts from a *seed tile* (or *seed* assembly).
- A tile can attach to the existing assembly if it binds with total strength at least 2 (the "temperature").

NEXT: An example…

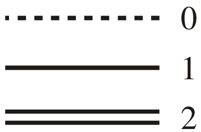
Tile Assembly Example

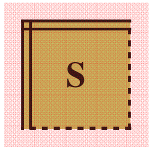


Tile Assembly Example



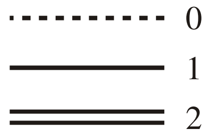
Edge binding strengths:

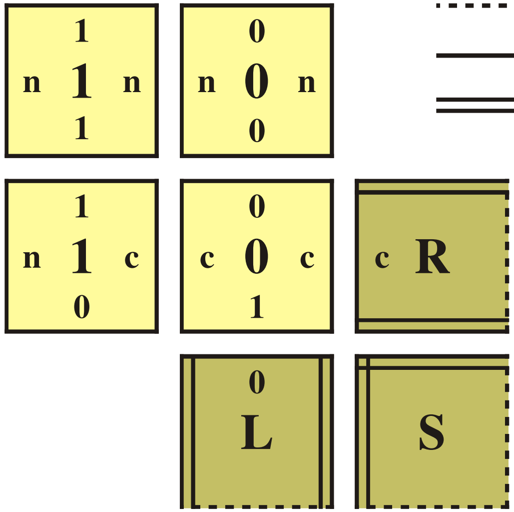


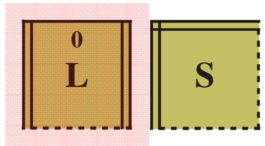


Tile Assembly Example

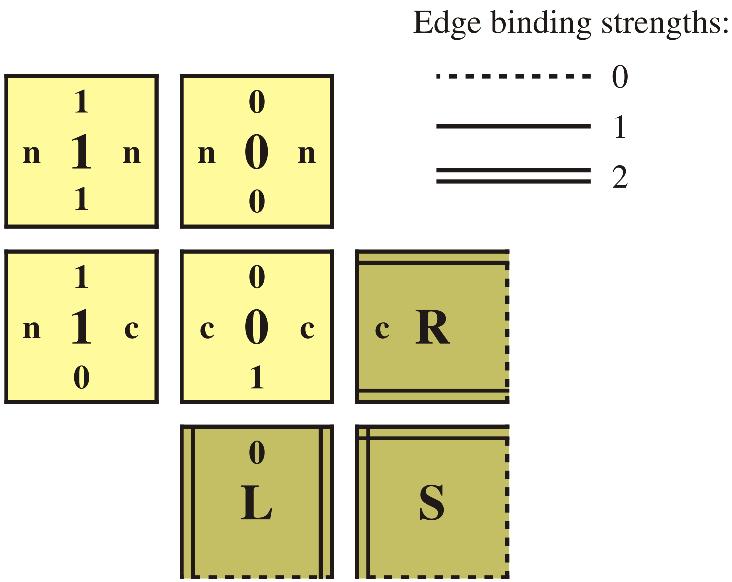
Edge binding strengths:

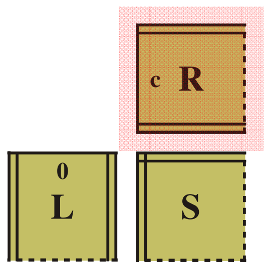




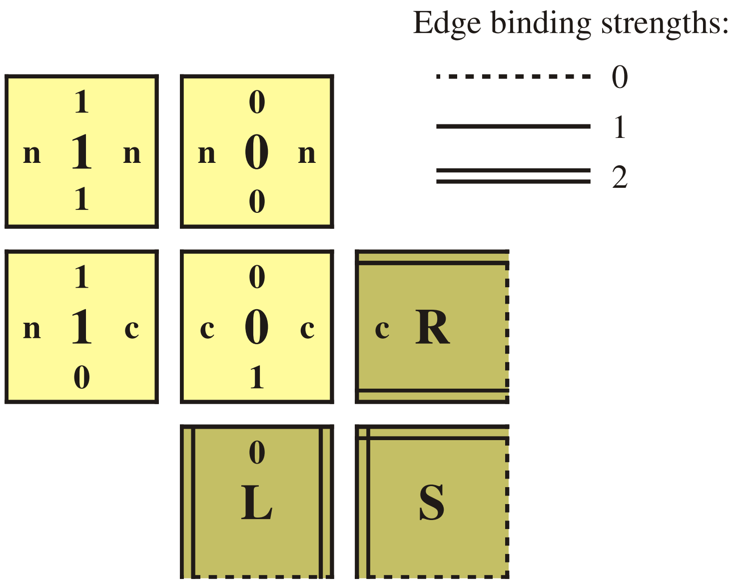


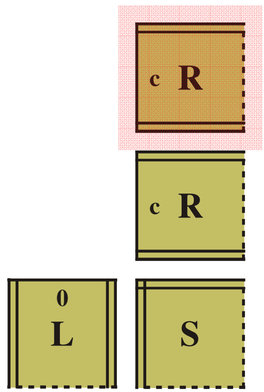
Tile Assembly Example



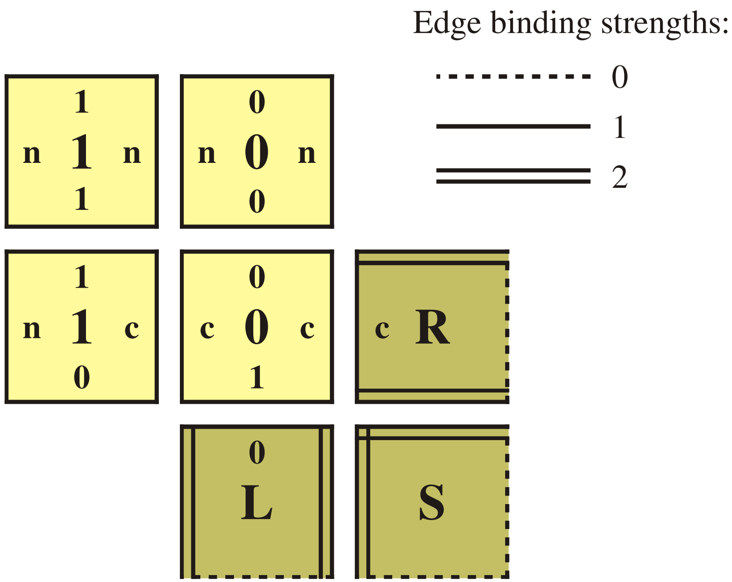


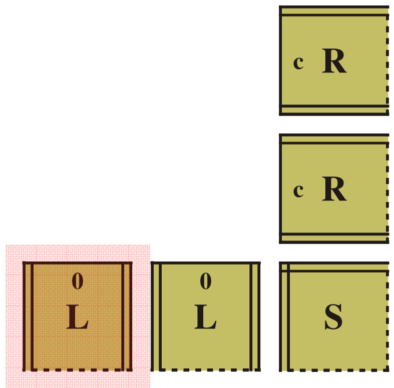
Tile Assembly Example



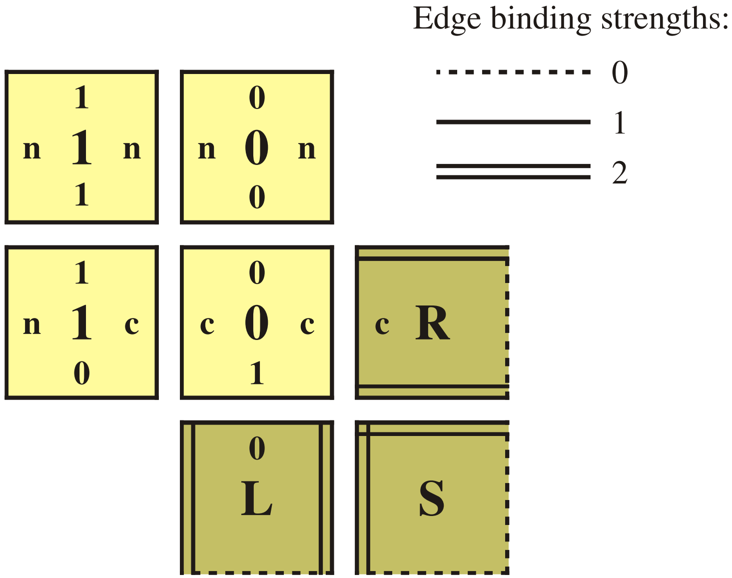


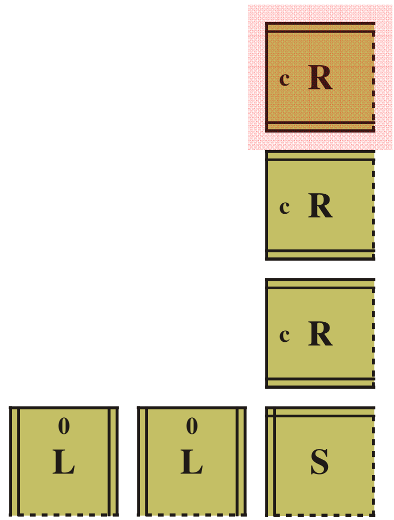
Tile Assembly Example



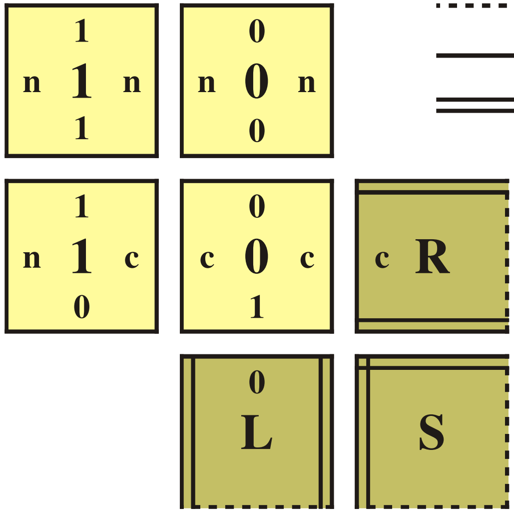


Tile Assembly Example

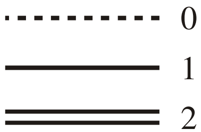


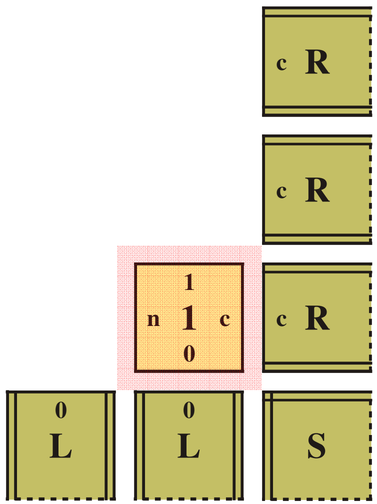


Tile Assembly Example

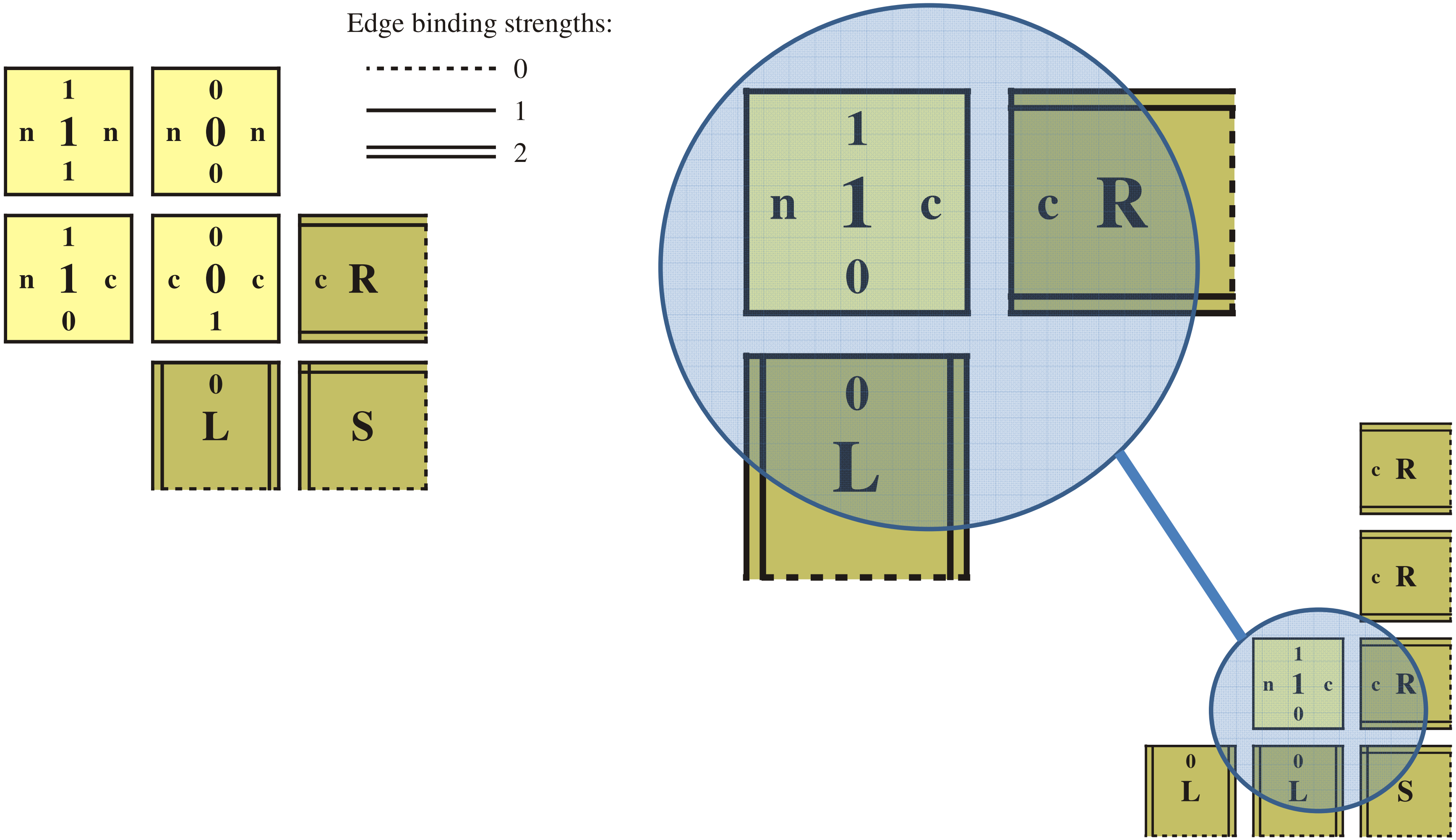


Edge binding strengths:

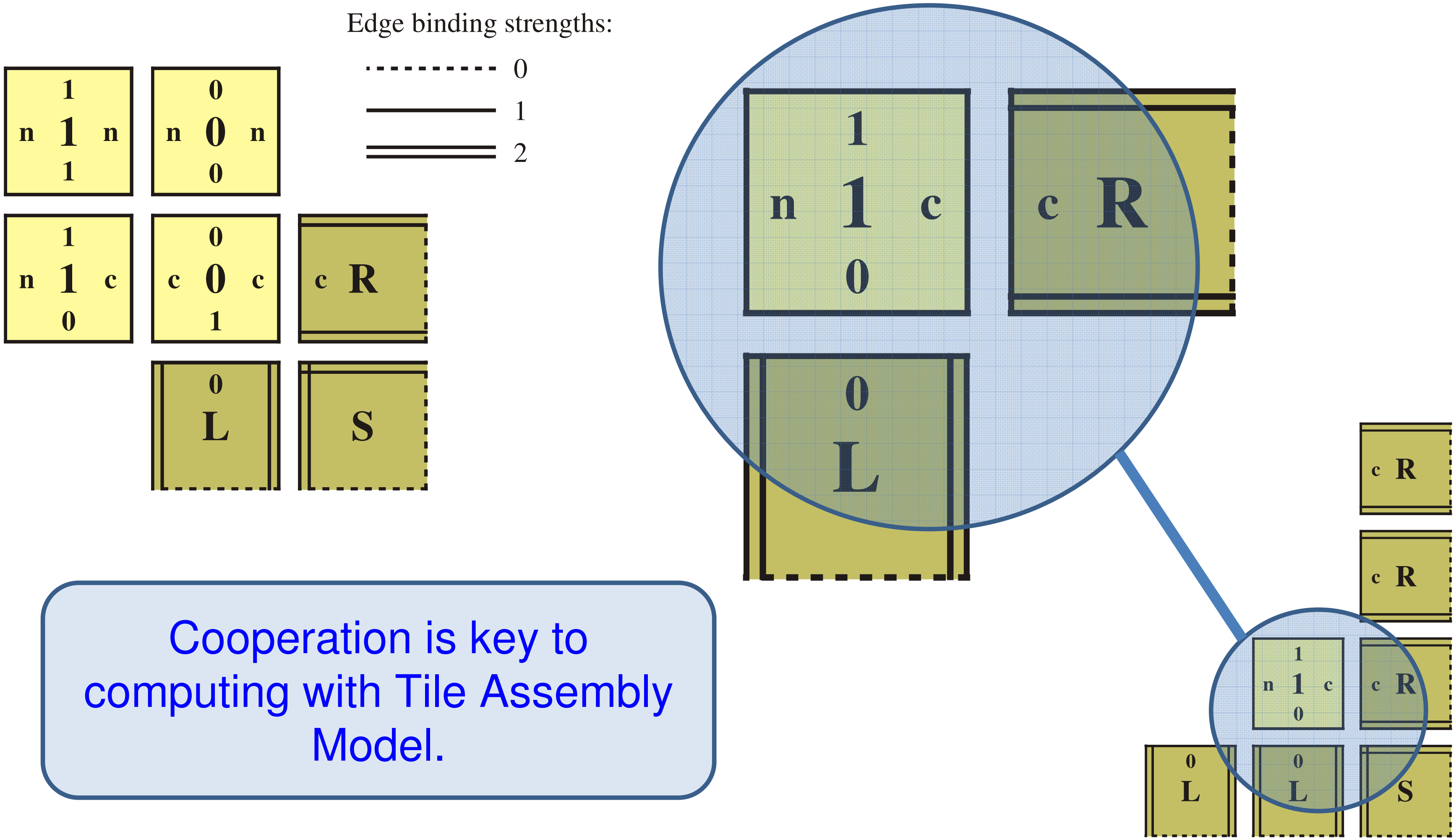




Tile Assembly Example

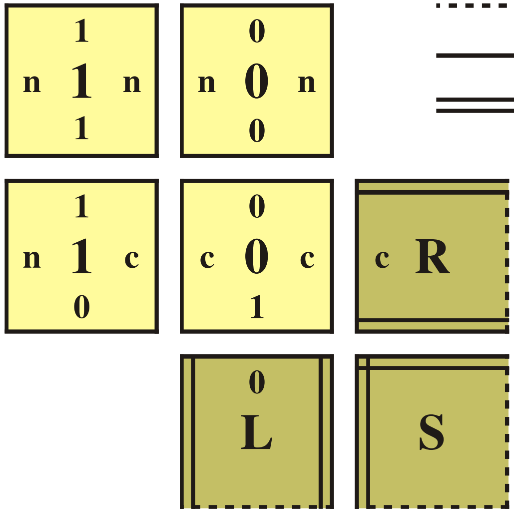


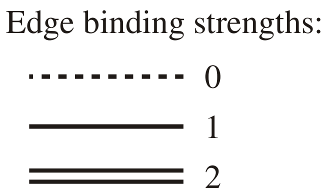
Tile Assembly Example

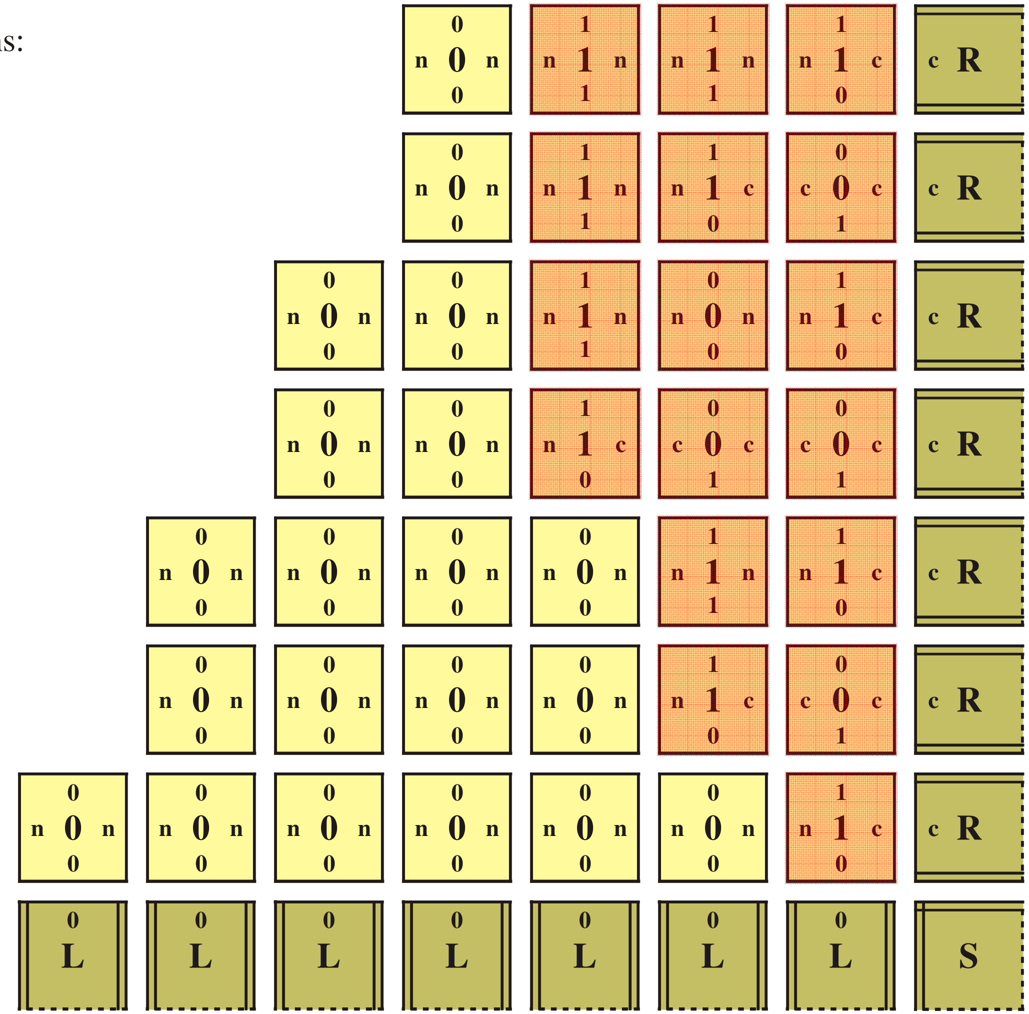


Cooperation is key to computing with Tile Assembly Model.

Tile Assembly Example







Our Results

Our Results

In this paper, we explore the connection between decidable sets and geometry in the Tile Assembly Model.

Our Results

In this paper, we explore the connection between decidable sets and geometry in the Tile Assembly Model.

We demonstrate a construction which tiles representations of decidable sets, then analyze the space requirements.

Our Results

In this paper, we explore the connection between decidable sets and geometry in the Tile Assembly Model.

We demonstrate a construction which tiles representations of decidable sets, then analyze the space requirements.

First we'll define two supplementary constructions…

The Wedge Construction

- Named for the shape of the assembly

The Wedge Construction

- Named for the shape of the assembly
- Utilizes well known technique of simulating a Turing machine with a tile assembly

The Wedge Construction

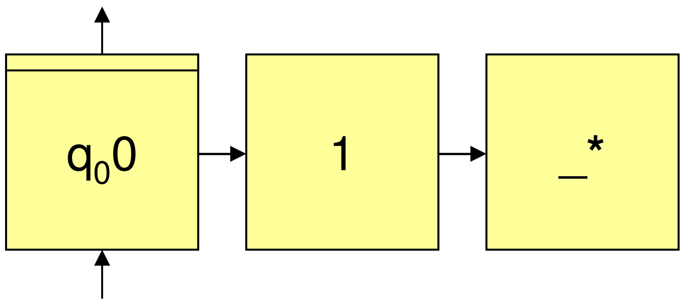
- Named for the shape of the assembly
- Utilizes well known technique of simulating a Turing machine with a tile assembly
  - Every row of the assembly encodes the entire configuration of the Turing machine (tape and state) at a particular step in the computation

The Wedge Construction

- Named for the shape of the assembly
- Utilizes well known technique of simulating a Turing machine with a tile assembly
  - Every row of the assembly encodes the entire configuration of the Turing machine (tape and state) at a particular step in the computation
  - The assembly simulates a ‘one-way infinite to the right’ tape by adding a tile on the right side of the row for each subsequent computation step

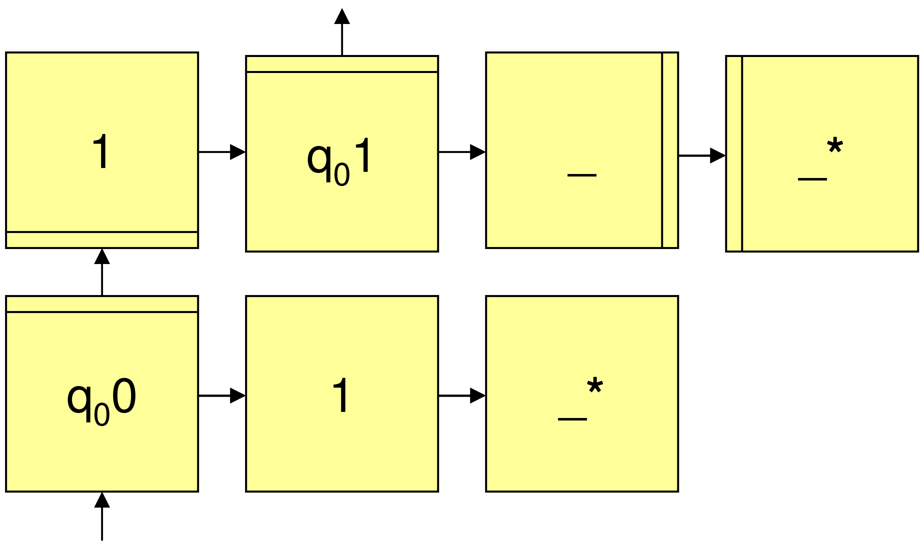
Wedge Construction Example

Wedge Construction Example



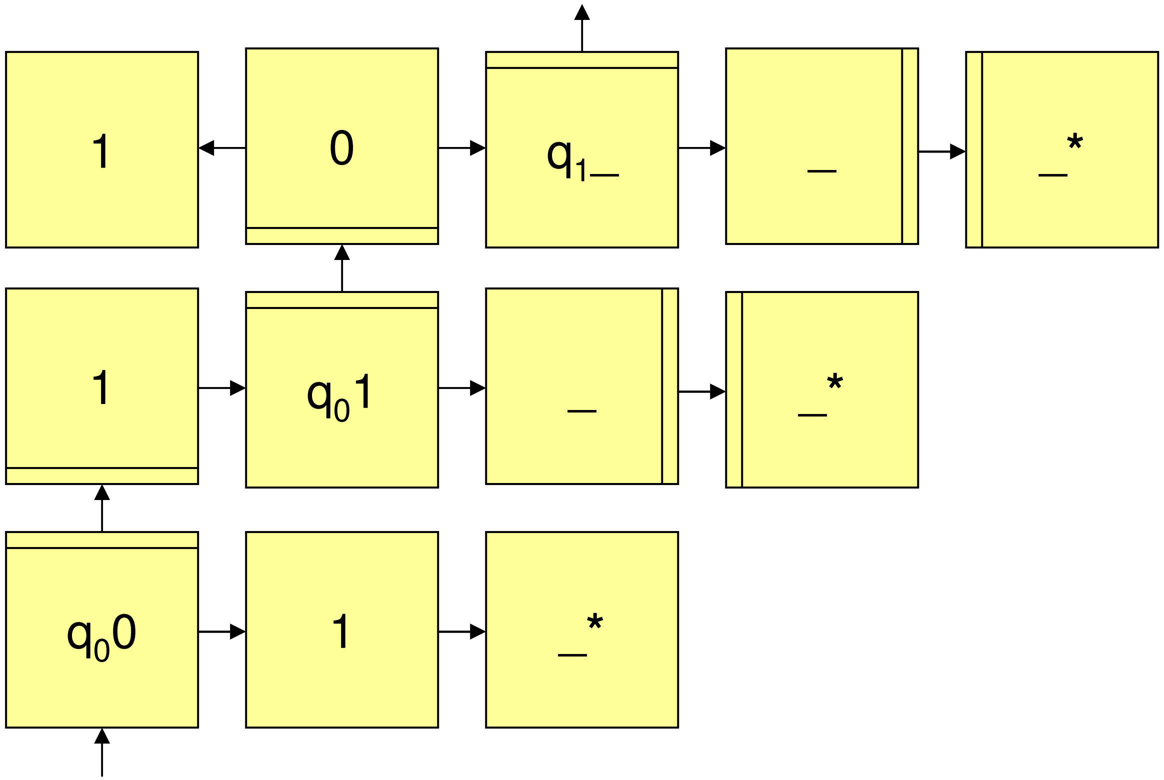
Initial configuration: TM on input ’01’

Wedge Construction Example



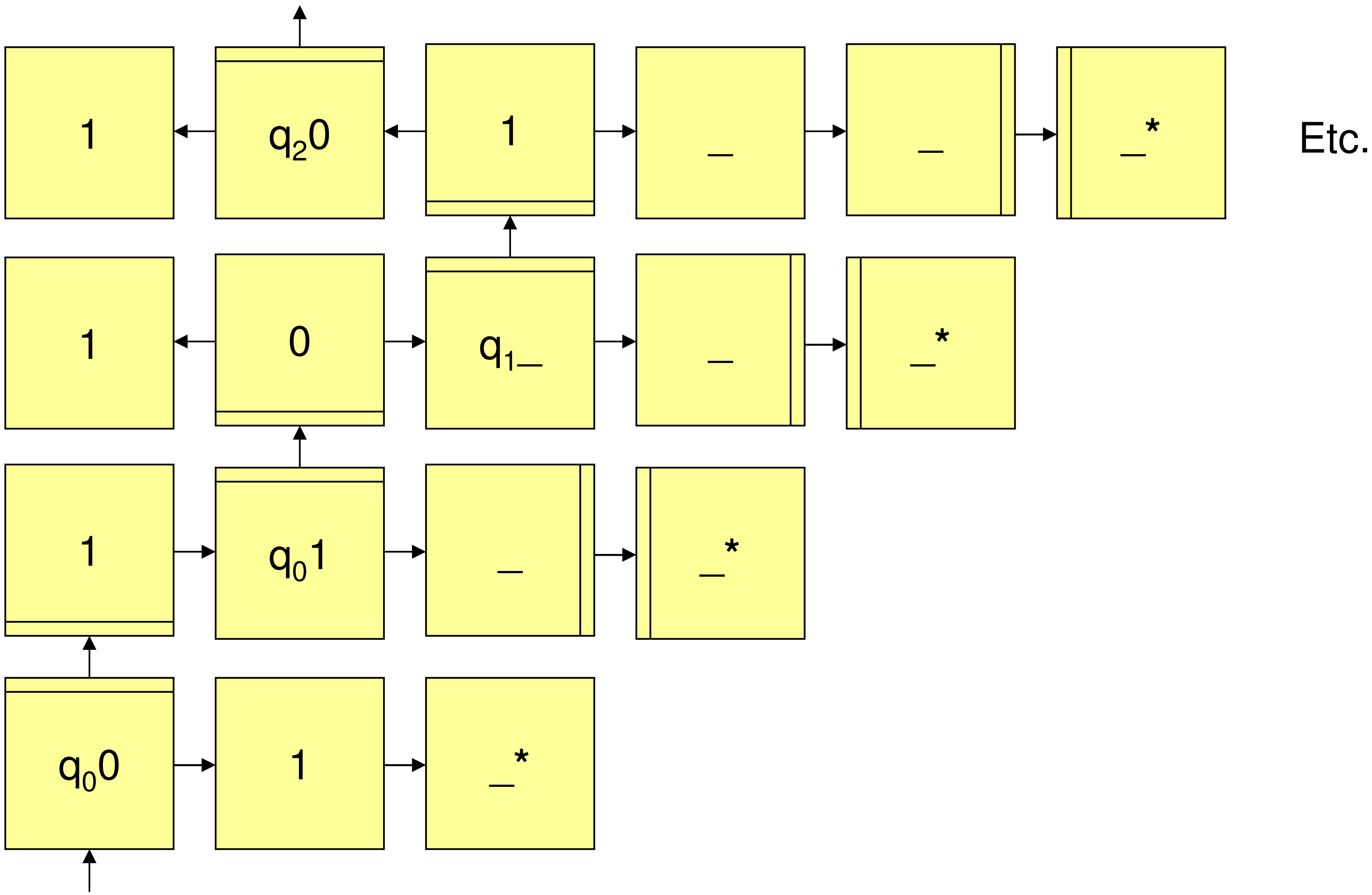
After 1 computation step

Wedge Construction Example



After 2 computation steps

Wedge Construction Example



Log-Width Binary Counter

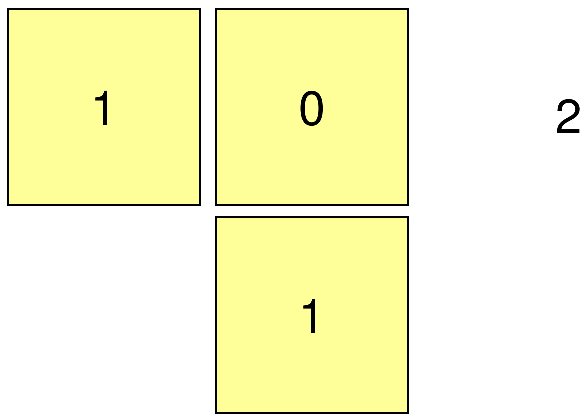
- An assembly which simulates a binary counter from 0 to infinity as it grows upward
- Each row represents a single value, which is one greater than the value of the row beneath it
- The width of each row is equal to the (floor of) the $\log_2$ of the counter's value in that row

Log-Width Binary Counter Example

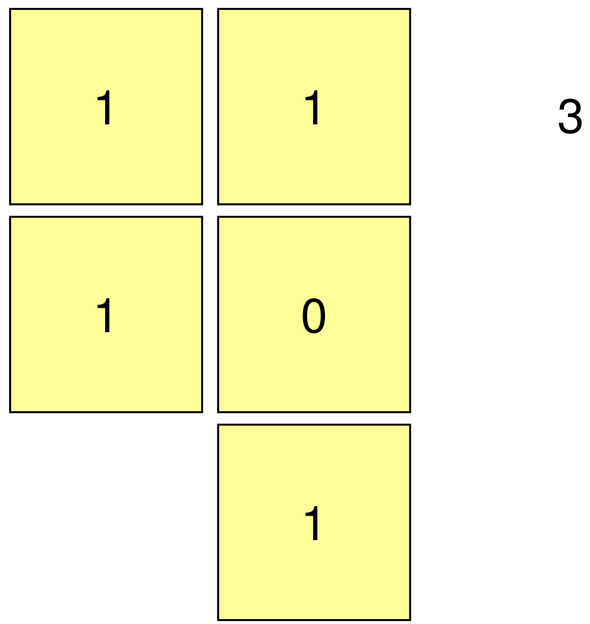
1

1

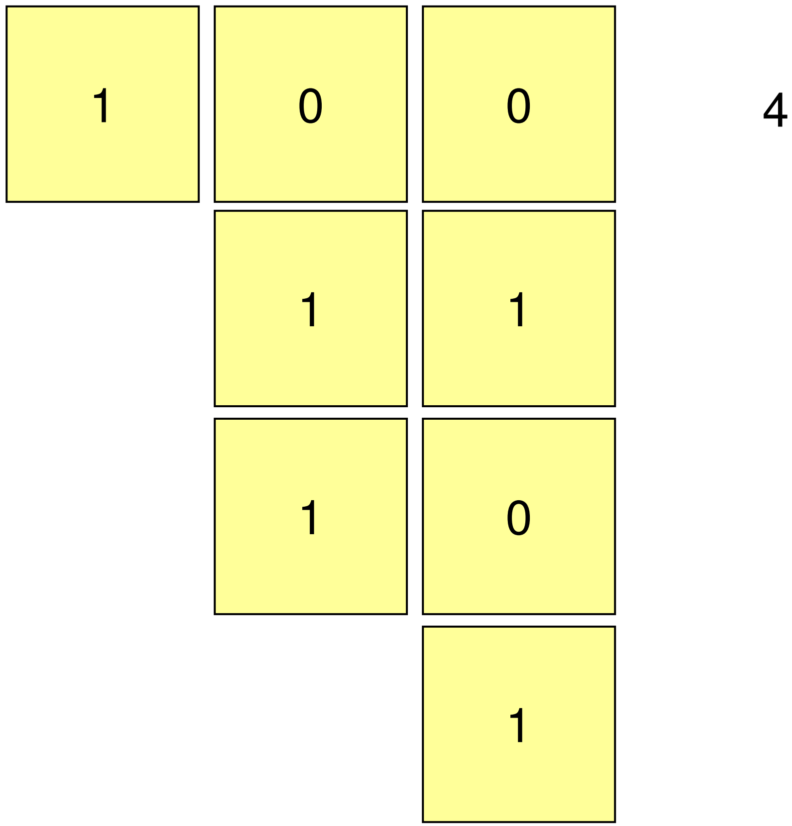
Log-Width Binary Counter Example



Log-Width Binary Counter Example



Log-Width Binary Counter Example



Log-Width Binary Counter Example

1	0	0	0	0
	1	1	1	1
	1	1	1	0
	1	1	0	1
	1	1	0	0
	1	0	1	1
	1	0	1	0
	1	0	0	1
	1	0	0	0
		1	1	1
		1	1	0
		1	0	1
		1	0	0
			1	1
			1	0
				1

A New Characterization of Decidable Languages

Theorem: Let $A \subseteq \mathbf{N}$. The set A is decidable if and only if $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble.

A New Characterization of Decidable Languages

Theorem: Let $A \subseteq \mathbf{N}$. The set A is decidable if and only if $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble.

Proof: ($\rightarrow$) This direction of our proof consists of a construction that demonstrates the claim.

Construction Overview

Fact: If *A* is decidable, then there exists a TM *M* which halts on every input and accepts exactly those that are in *A*

Construction Overview

while $0 \le n < \infty$ do

  simulate M on the binary representation of n

  if M accepts, then

    output 1

  else

    output 0

  end if

  $n := n + 1$

end while

Construction Overview

Our construction implements that algorithm by stacking wedge constructions on top of each other

Construction Overview

Our construction implements that algorithm by stacking wedge constructions on top of each other

Each wedge construction simulates *M* on an input value one greater than the wedge construction below it

Construction Overview

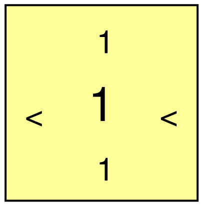
Our construction implements that algorithm by stacking wedge constructions on top of each other

Each wedge construction simulates *M* on an input value one greater than the wedge construction below it

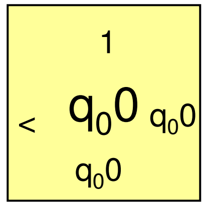
Incrementing the inputs is done by embedding a log-width binary counter within the wedge constructions

Construction Technique: Embedding Functionality

Log-width binary counter tile

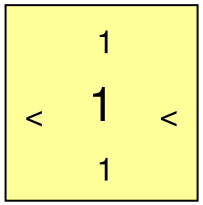


Turing machine tile

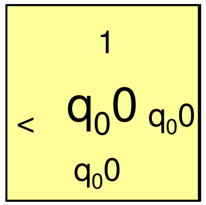


Construction Technique: Embedding Functionality

Log-width binary counter tile



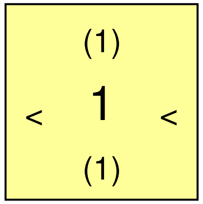
Turing machine tile



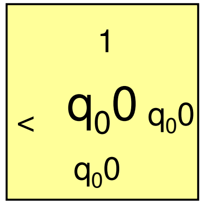
Turing machine tile passing binary counter value upward

Construction Technique: Embedding Functionality

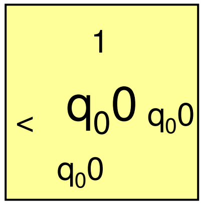
Log-width binary counter tile



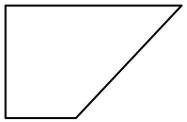
Turing machine tile



Turing machine tile passing binary counter value upward

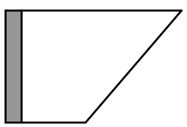


Construction Diagram



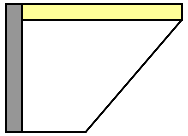
Start with a wedge construction
which simulates TM *M* on input 0

Construction Diagram



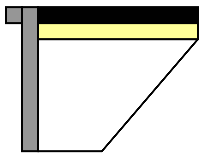
Embed a log-width binary
counter along the left side

Construction Diagram



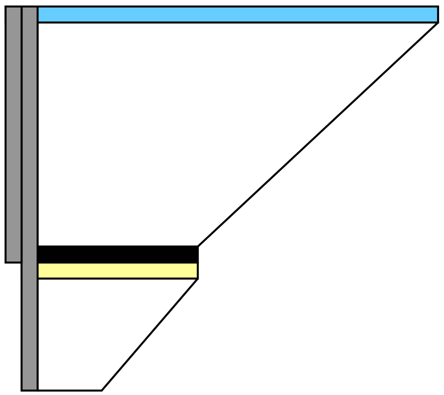
Once $M(0)$ halts and accepts or rejects,
make a row specifying the result

Construction Diagram



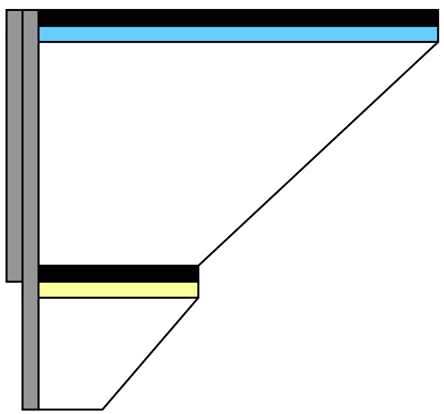
Next, add a row which increments the counter

Construction Diagram



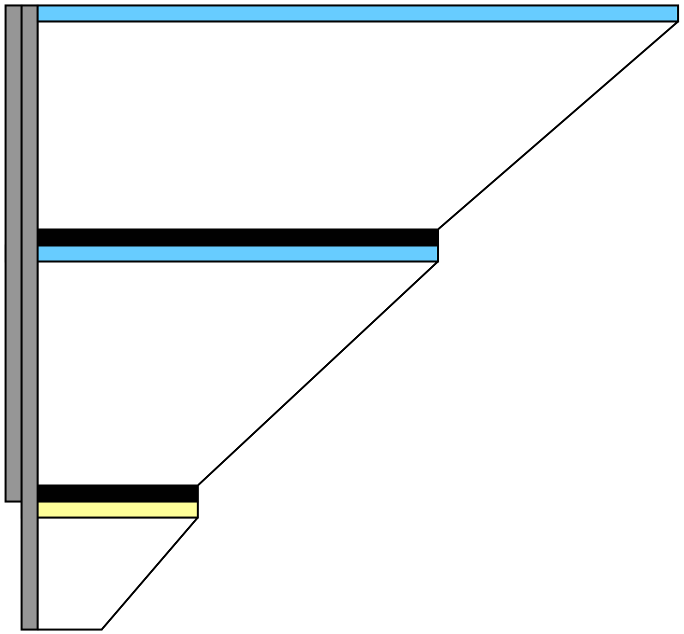
Use the new counter value
to begin the simulation
of *M*(1)

Construction Diagram



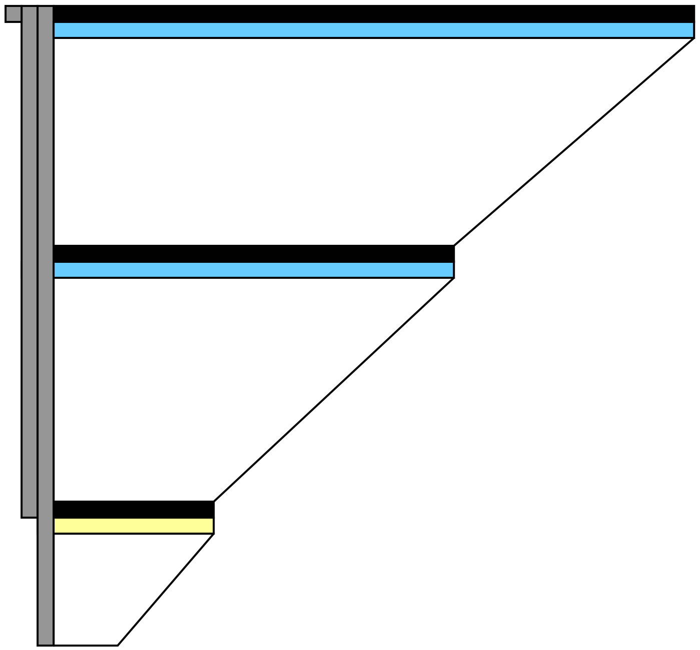
Increment the counter

Construction Diagram



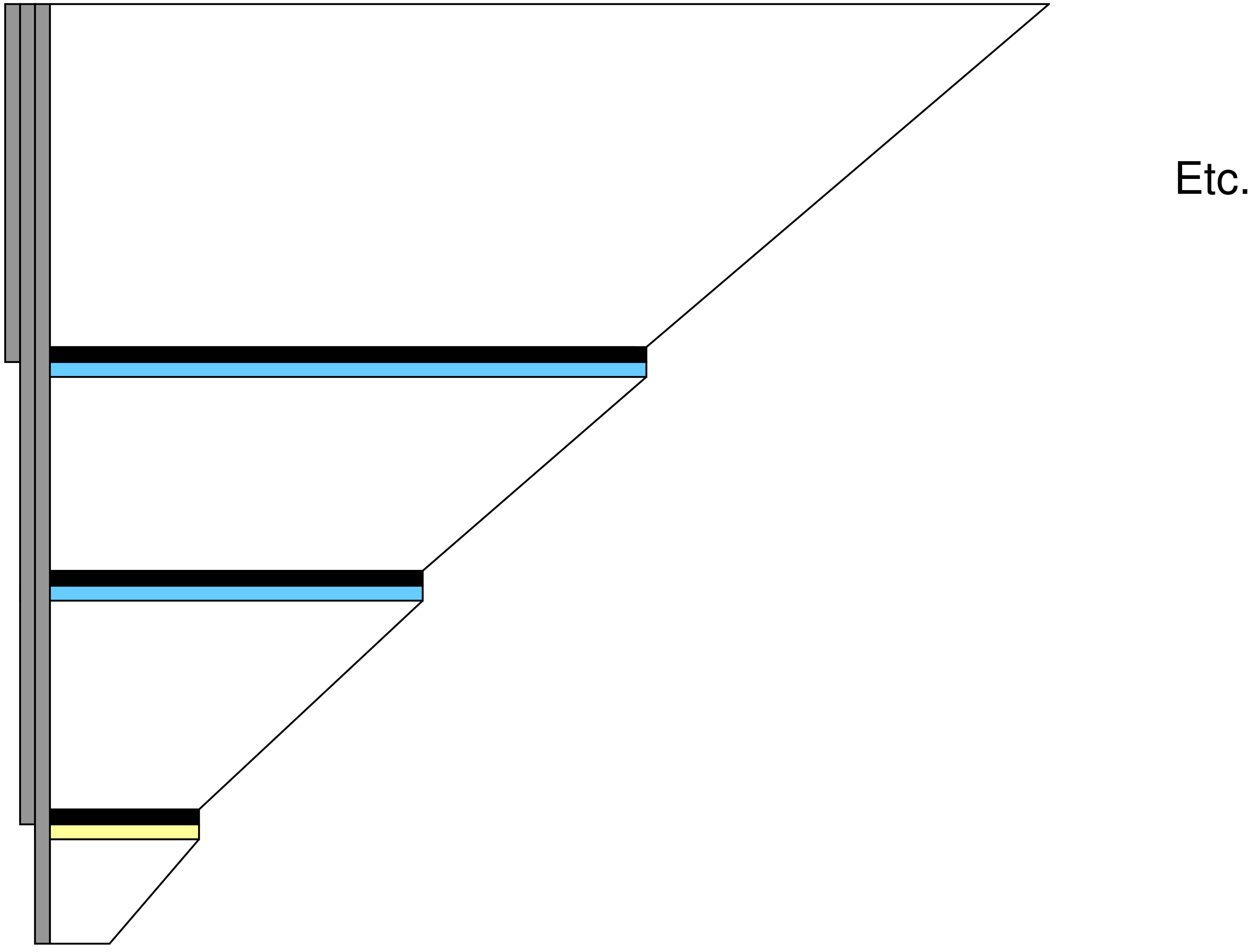
Simulate $M(2)$

Construction Diagram

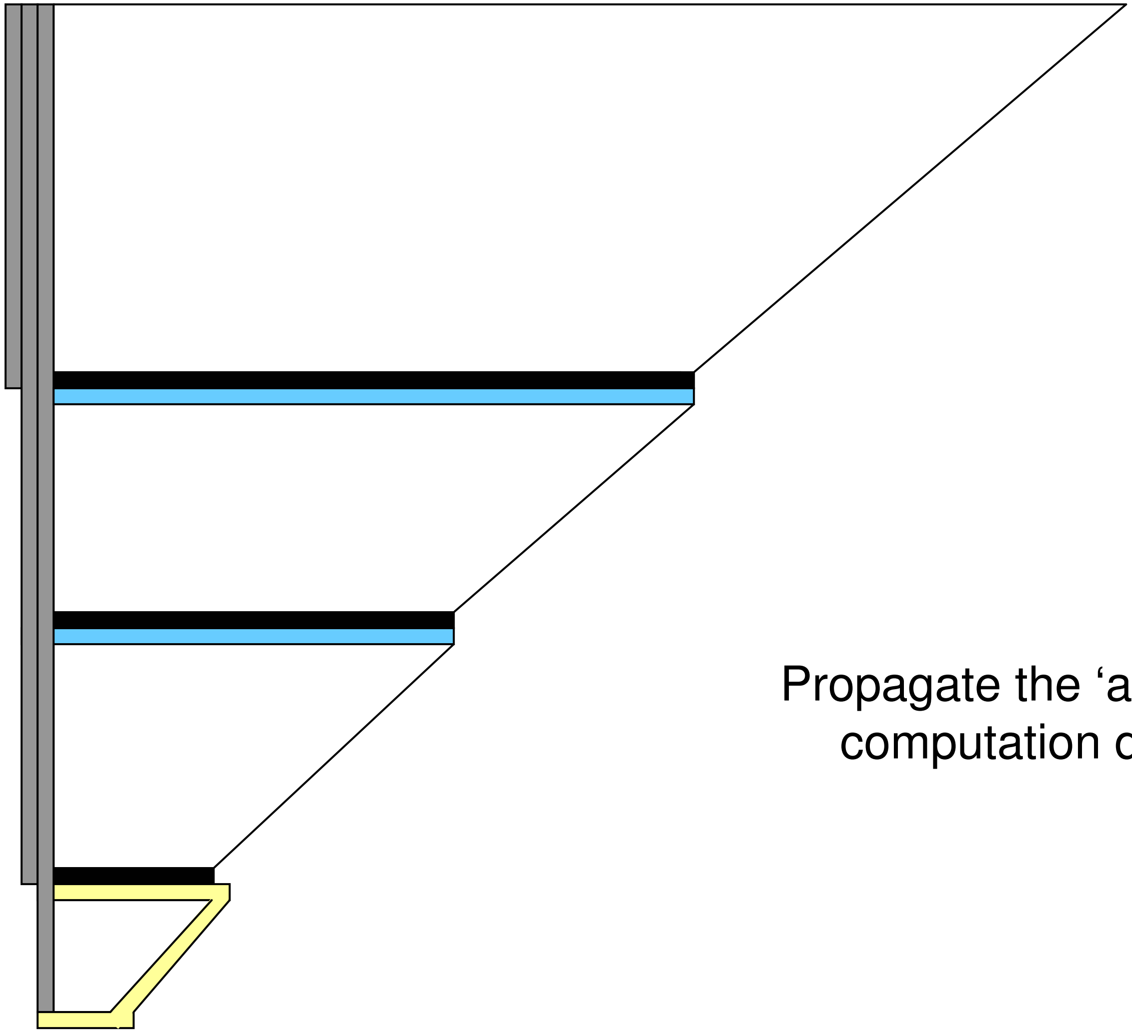


Increment the counter

Construction Diagram

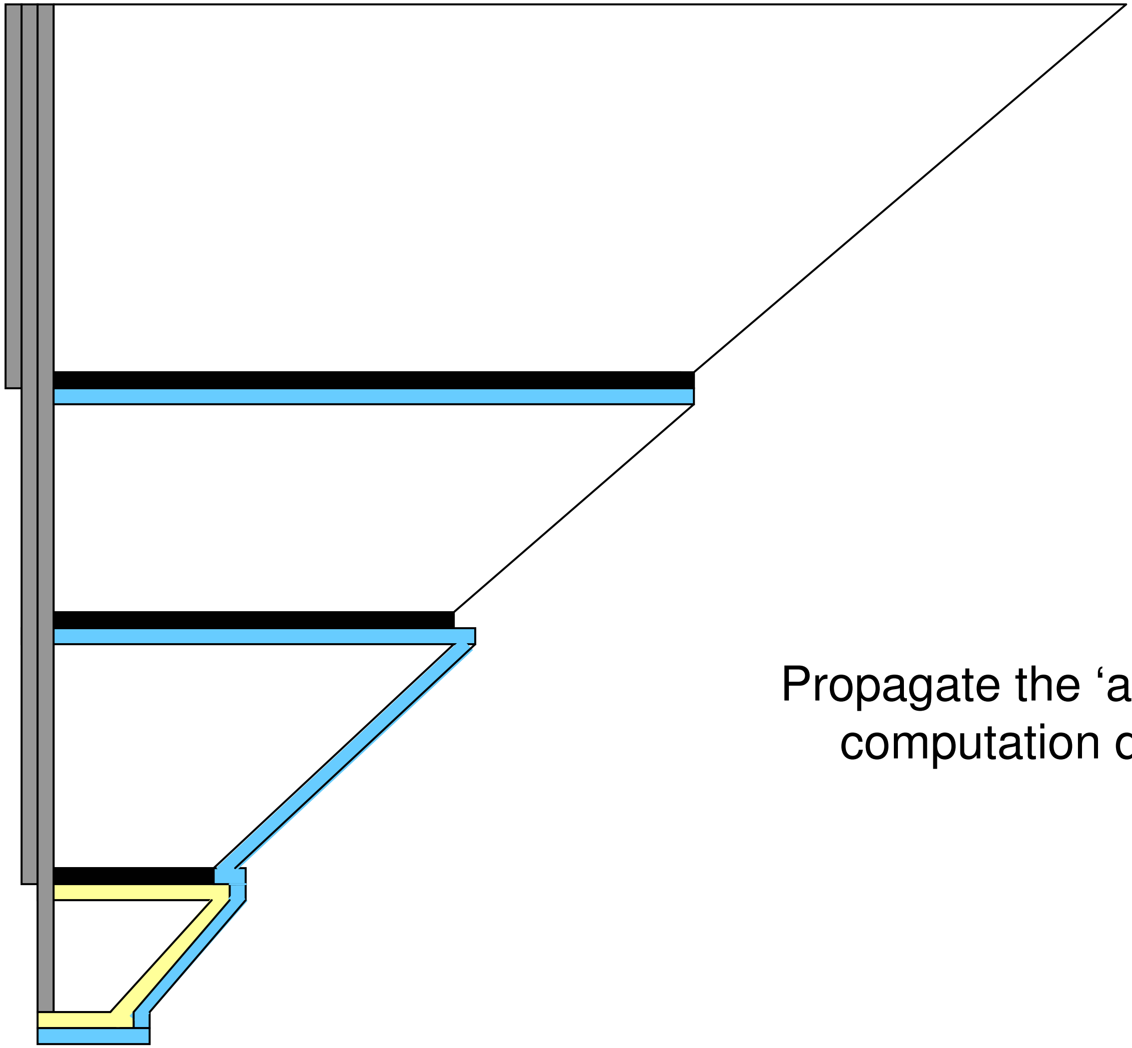


Construction Diagram



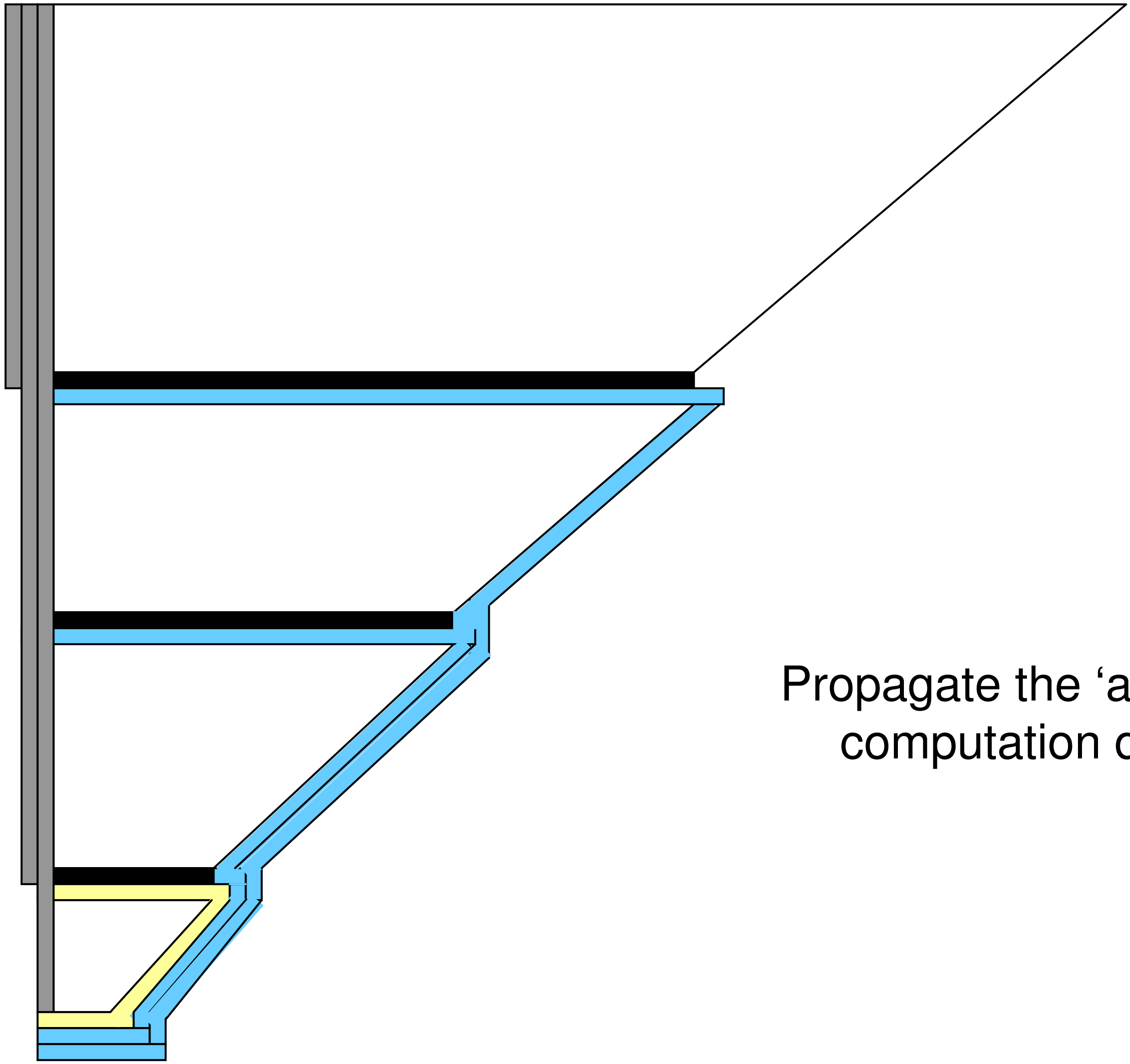
Propagate the 'answers' from each computation down to the axis

Construction Diagram



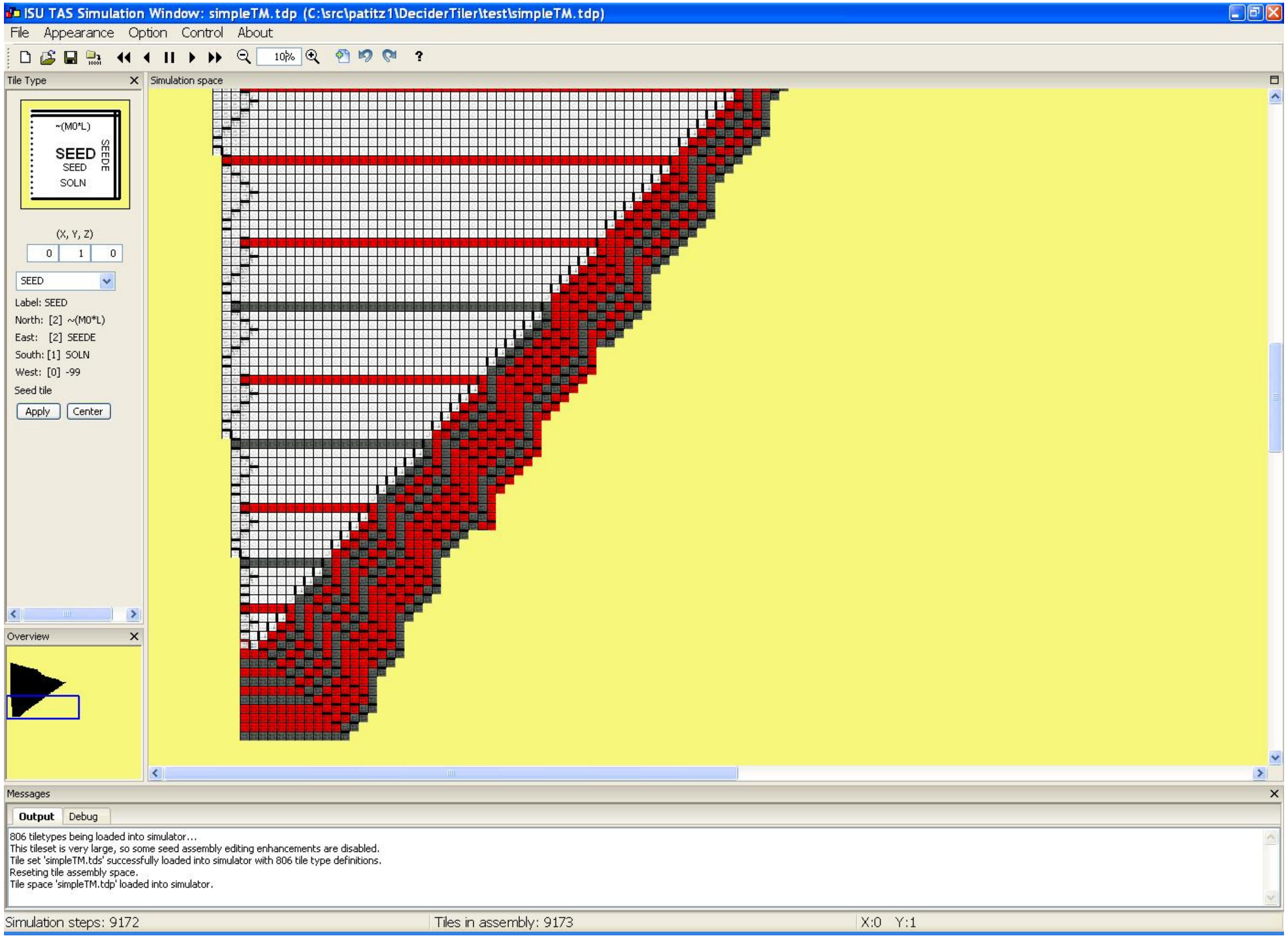
Propagate the ‘answers’ from each computation down to the axis

Construction Diagram



Propagate the ‘answers’ from each computation down to the axis

Construction Example



A New Characterization of Decidable Languages

This construction proves that if a set is decidable, then $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble.

A New Characterization of Decidable Languages

Proof: (⟵) This direction of the proof uses the existence of self-assembly simulators to prove that if $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble, then the set A is decidable.

Proof Sketch (←)

- Assume A and A^c both weakly self-assemble

Proof Sketch (←)

- Assume A and A^c both weakly self-assemble
- Then there exist tile assembly systems $\mathcal{T}_{A\times\{0\}}$ and $\mathcal{T}_{A^c\times\{0\}}$ in which they weakly self-assemble

Proof Sketch (←)

- Assume A and A^c both weakly self-assemble
- Then there exist tile assembly systems $\mathcal{T}_{A\times\{0\}}$ and $\mathcal{T}_{A^c\times\{0\}}$ in which they weakly self-assemble
- For an input n, simulate both tile assembly systems in parallel

Proof Sketch ($\leftarrow$)

- Assume A and A^c both weakly self-assemble
- Then there exist tile assembly systems $\mathcal{T}_{A \times \{0\}}$ and $\mathcal{T}_{A^c \times \{0\}}$ in which they weakly self-assemble
- For an input n, simulate both tile assembly systems in parallel
    - Accept if $\mathcal{T}_{A \times \{0\}}$ puts a black tile at $(n,0)$

Proof Sketch (←)

- Assume A and A^c both weakly self-assemble
- Then there exist tile assembly systems $\mathcal{T}_{A\times\{0\}}$ and $\mathcal{T}_{A^c\times\{0\}}$ in which they weakly self-assemble
- For an input n, simulate both tile assembly systems in parallel
  - Accept if $\mathcal{T}_{A\times\{0\}}$ puts a black tile at $(n,0)$
  - Reject if $\mathcal{T}_{A^c\times\{0\}}$ puts a black tile at $(n,0)$

A New Characterization of Decidable Languages

This completes the proof that a set A is decidable if and only if $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble.

Second Main Result

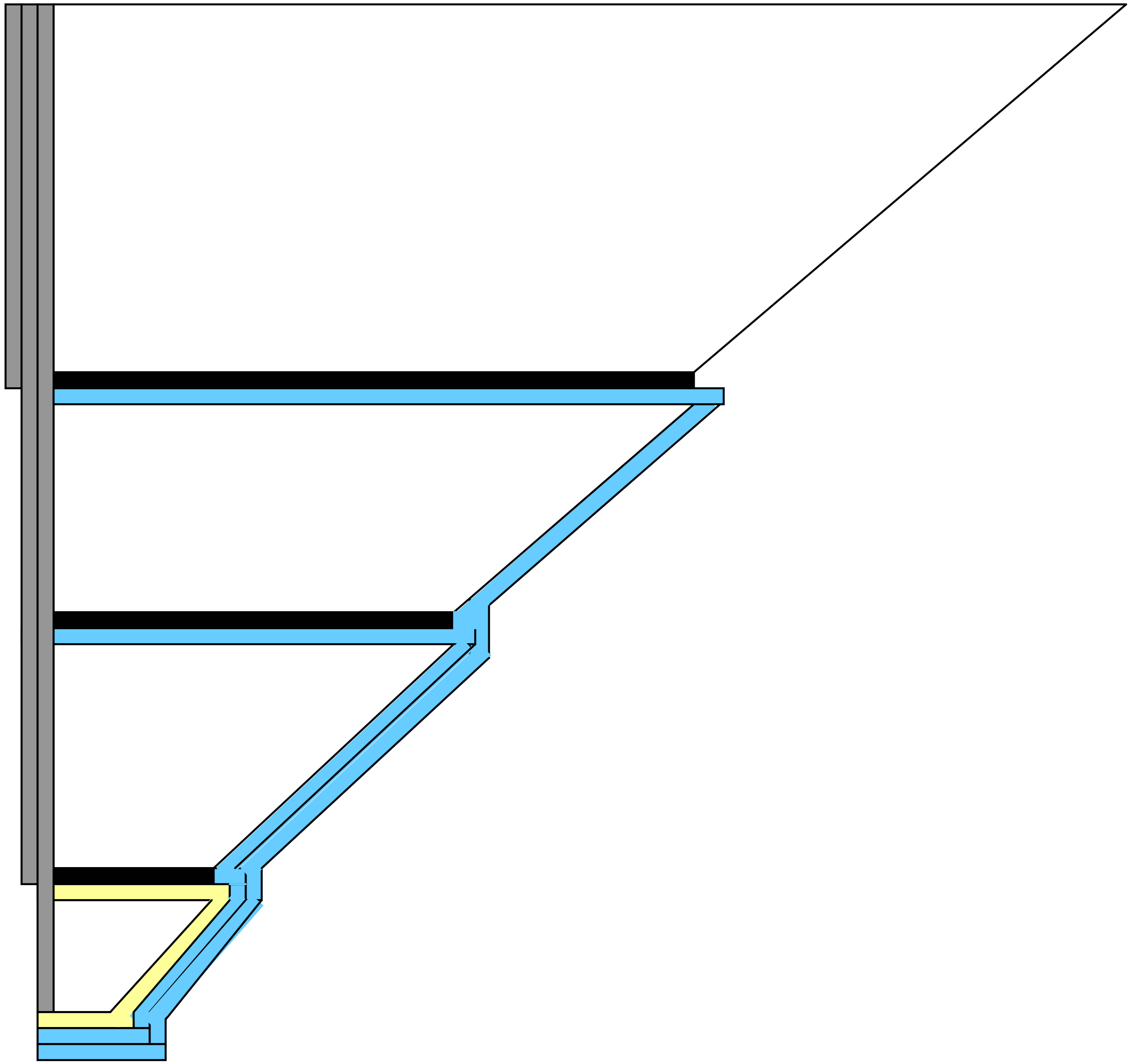
To prove our first main result, we constructed a tile assembly system that placed at least one tile in three different quadrants.

Second Main Result

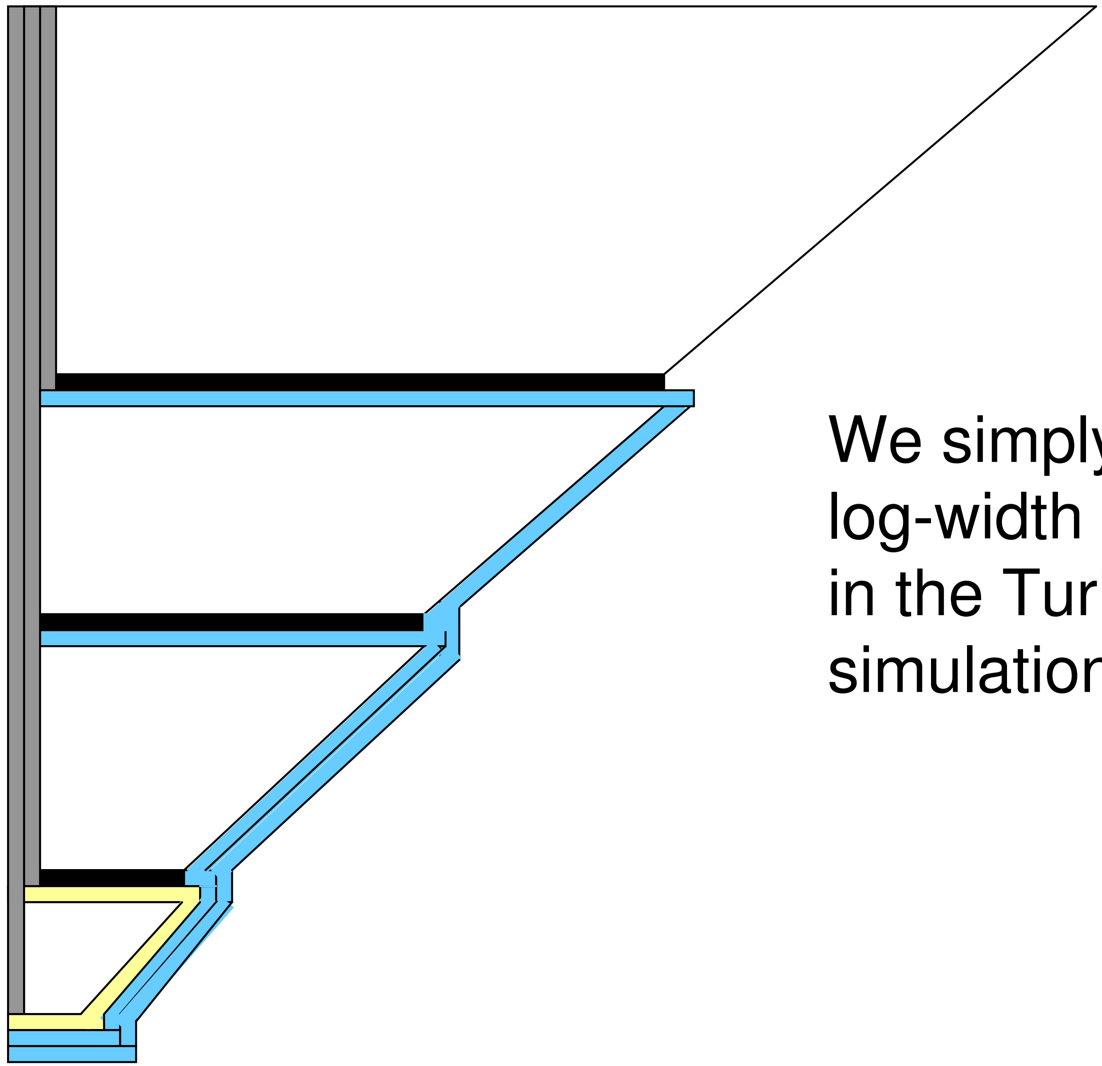
To prove our first main result, we constructed a tile assembly system that placed at least one tile in three different quadrants.

Note that it *is* possible to prove our first main result while placing tiles in only **two quadrants**.

Two quadrants



Two quadrants



We simply embed the log-width binary counter in the Turing machine simulation.

Second Main Result

However, if the language *A* has sufficient space complexity, **AND** your tile assembly system *resembles* our construction in the sense that the TM is simulated “row by row,” then two quadrants of space are **necessary**.

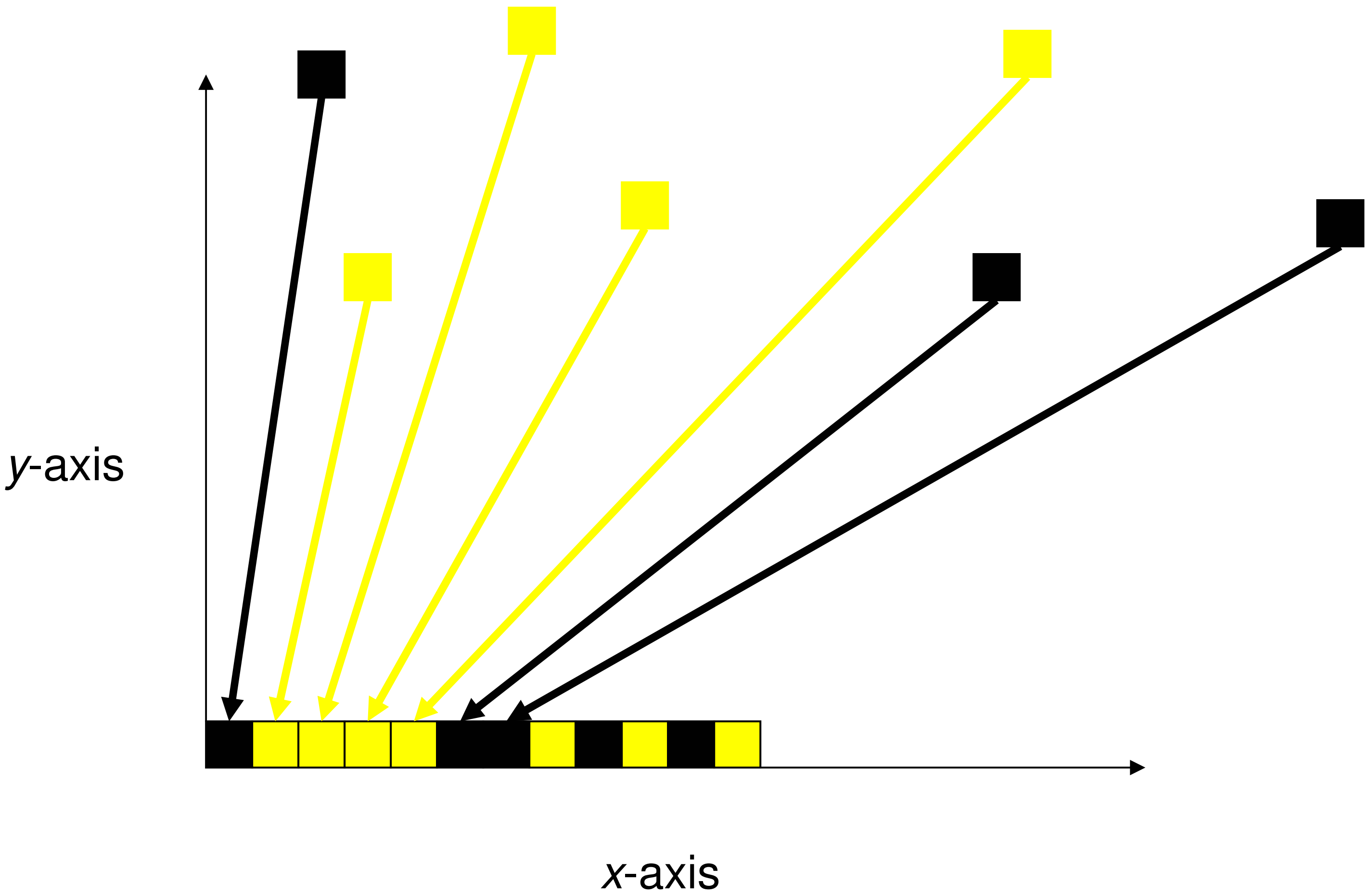
Second Main Result (proof idea)

Assumption #1: connected to each point along the *x*-axis is a *unique* longest path originating from some unique point in the first quadrant that carries the answer to the question:

Second Main Result (proof idea)

Assumption #1: connected to each point along the *x*-axis is a *unique* longest path originating from some unique point in the first quadrant that carries the answer to the question: *does this TM accept/reject this input?*

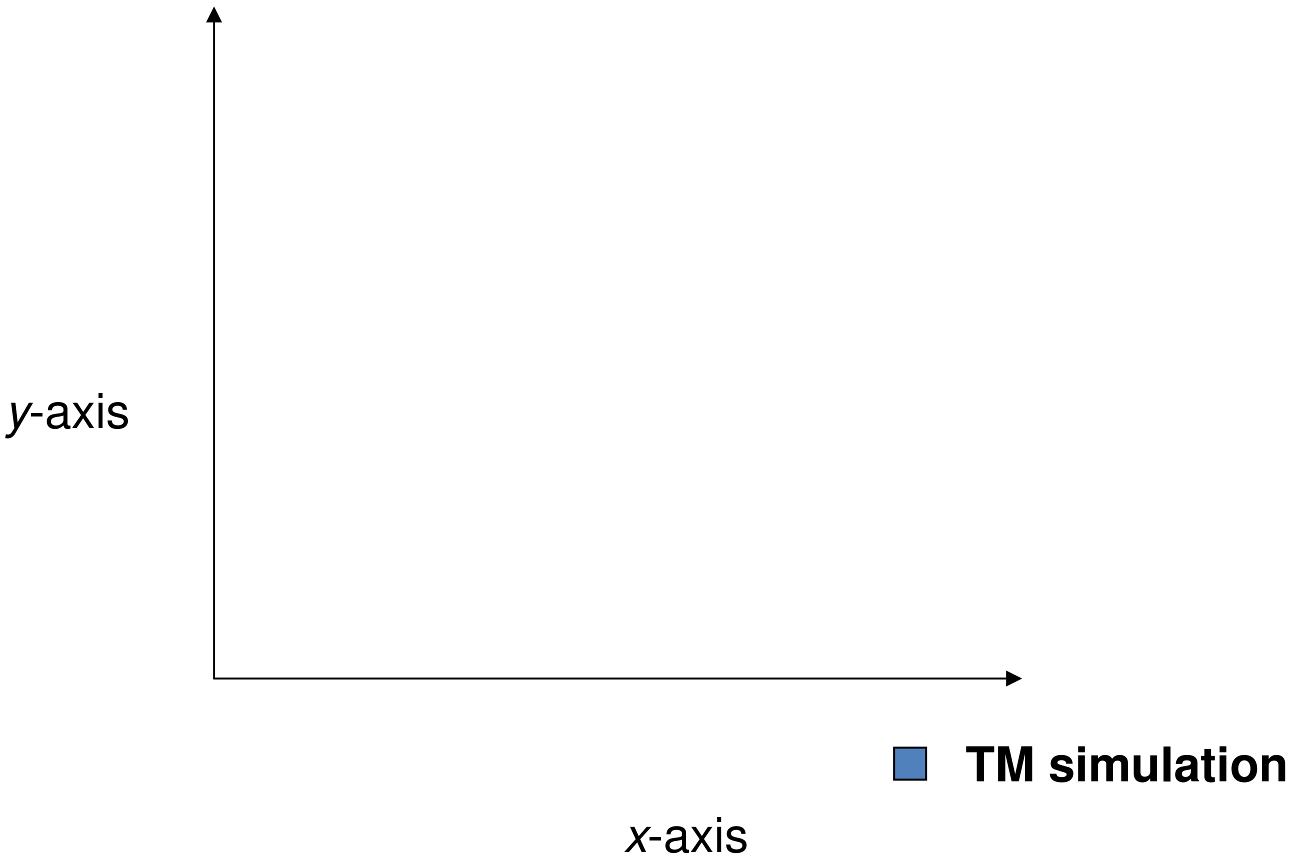
Second Main Result (proof idea)



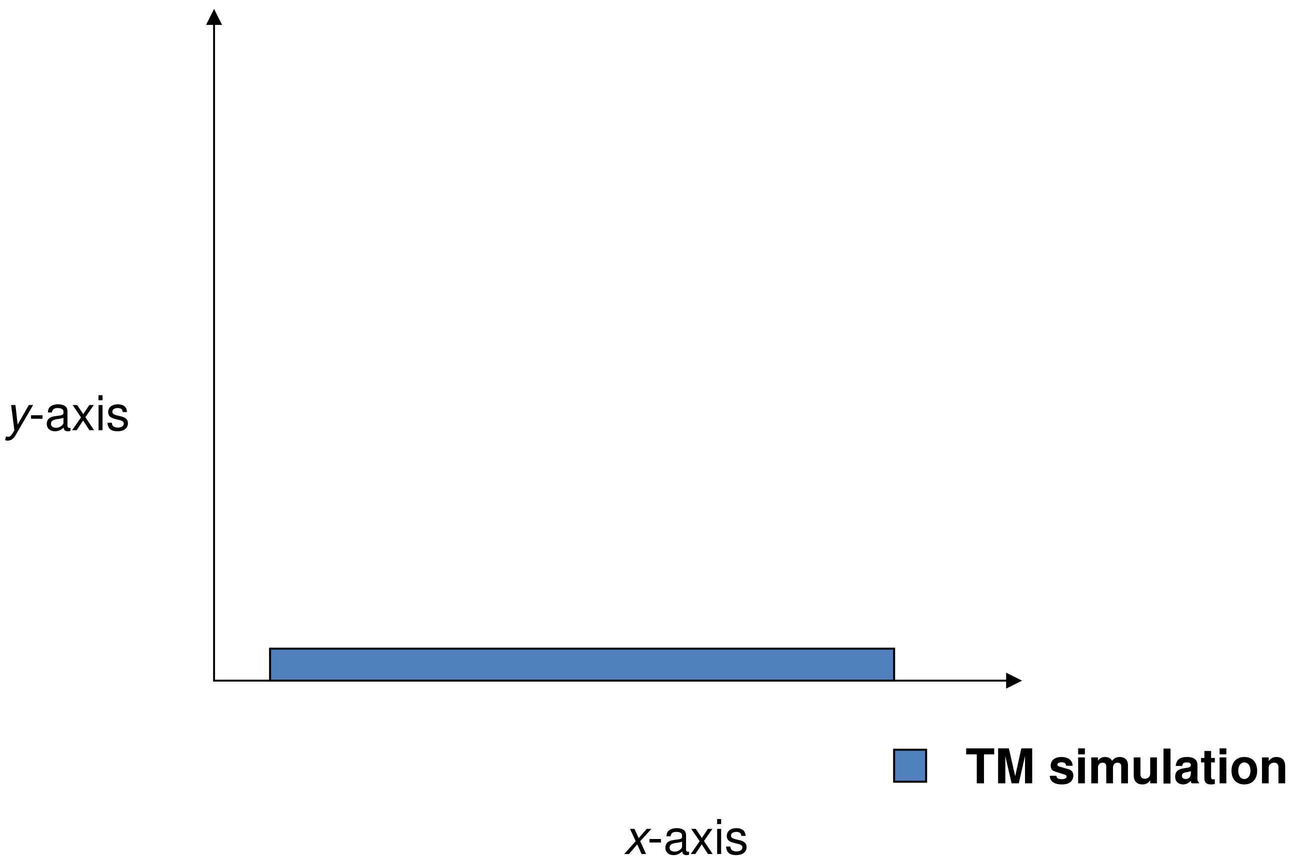
Second Main Result (proof idea)

Assumption #2: aside from all of the paths mentioned on the previous slide, the rest of the assembly can be self-assembled (entirely) one row at a time

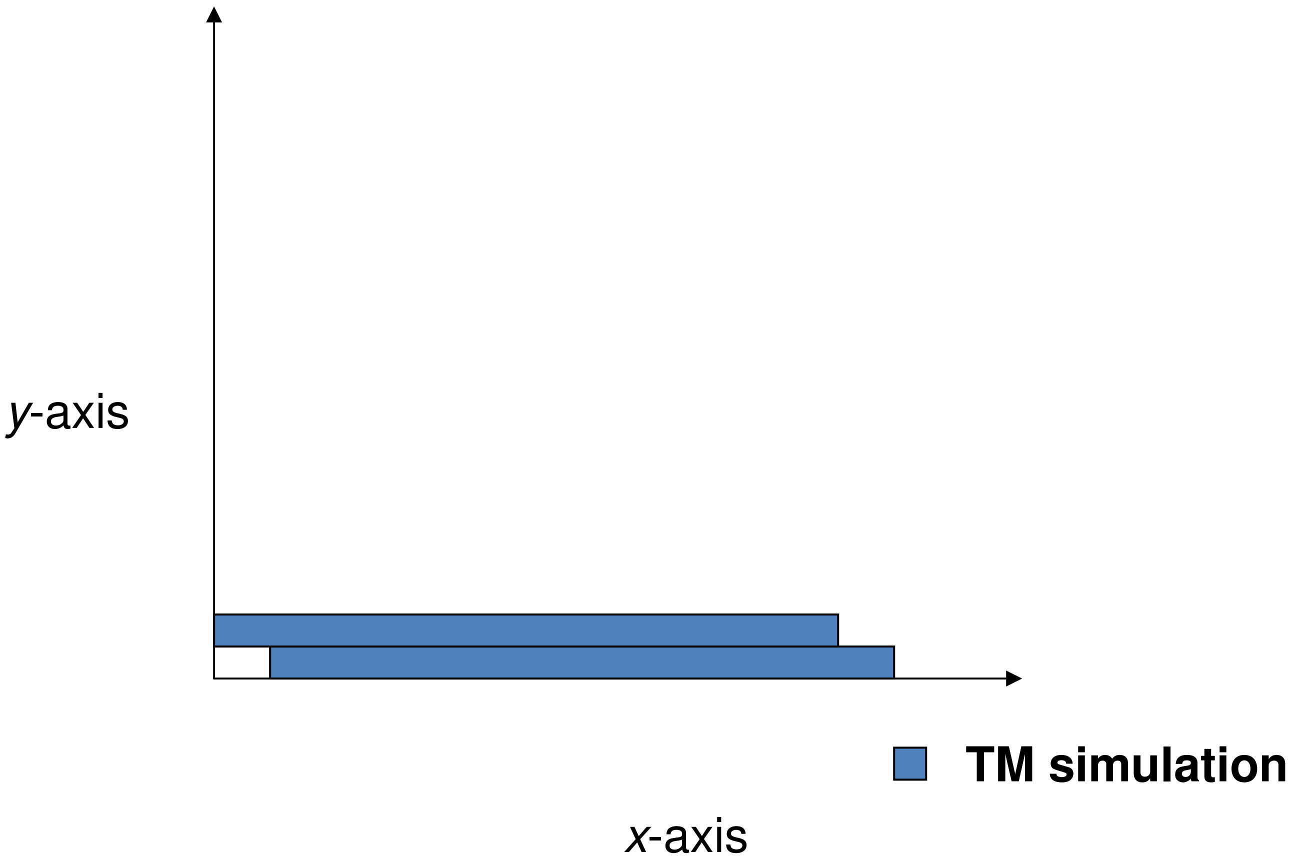
Second Main Result (proof idea)



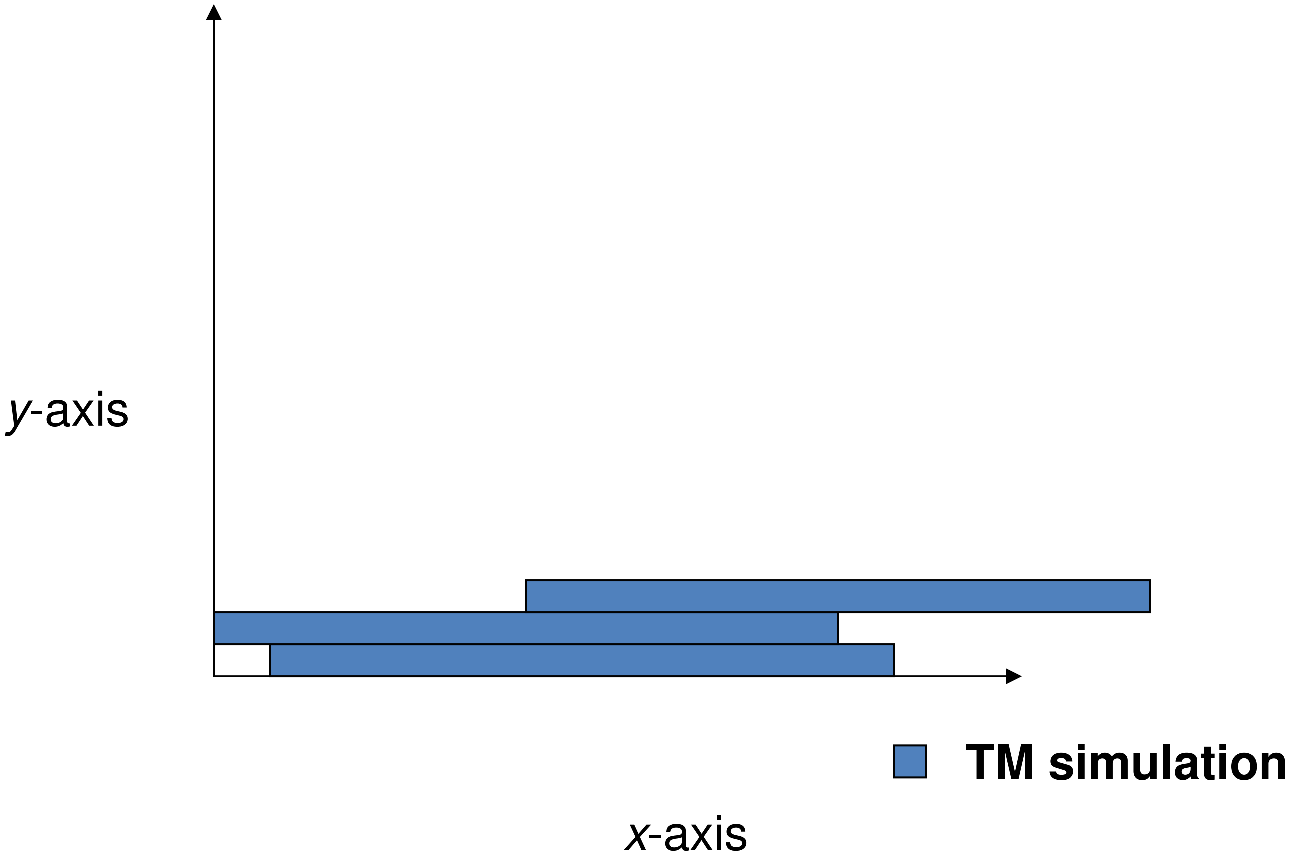
Second Main Result (proof idea)



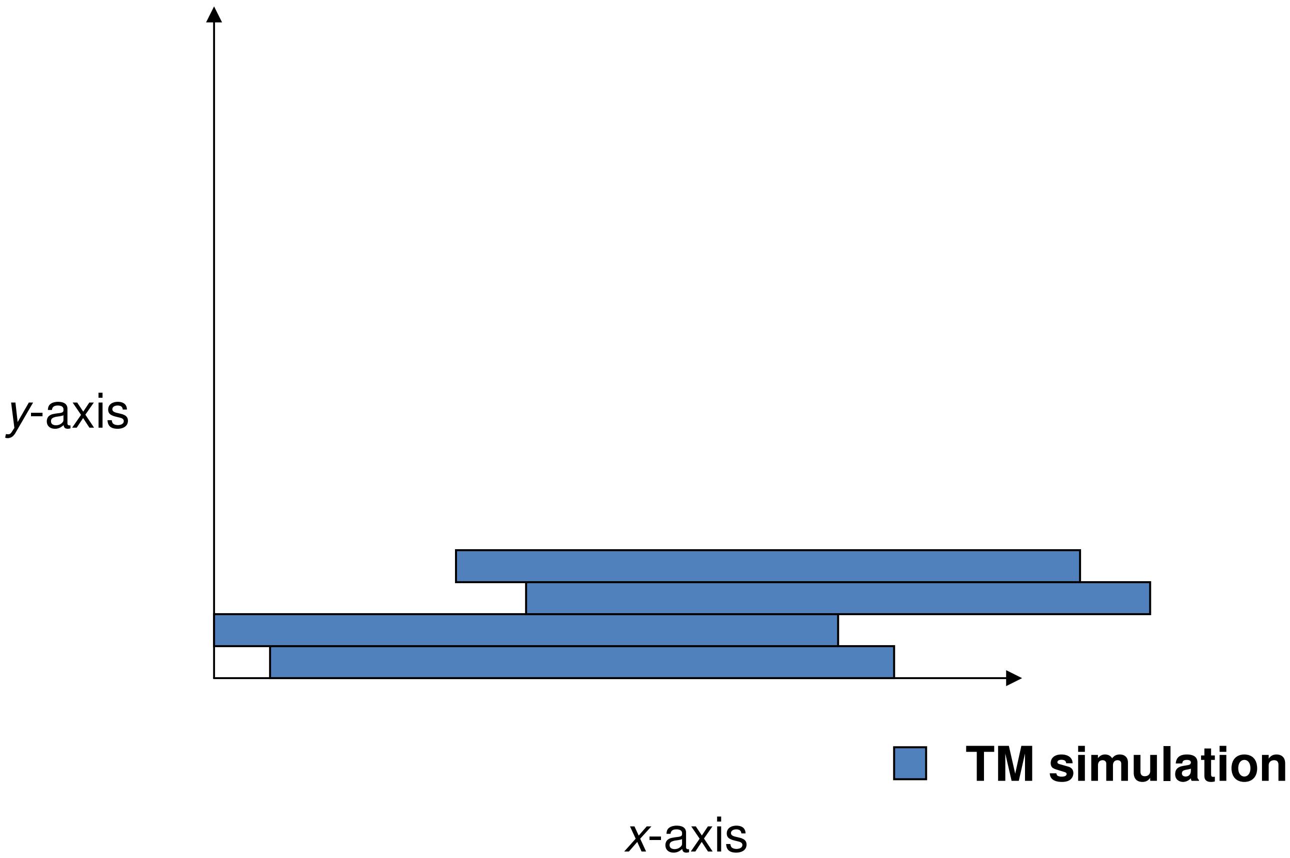
Second Main Result (proof idea)



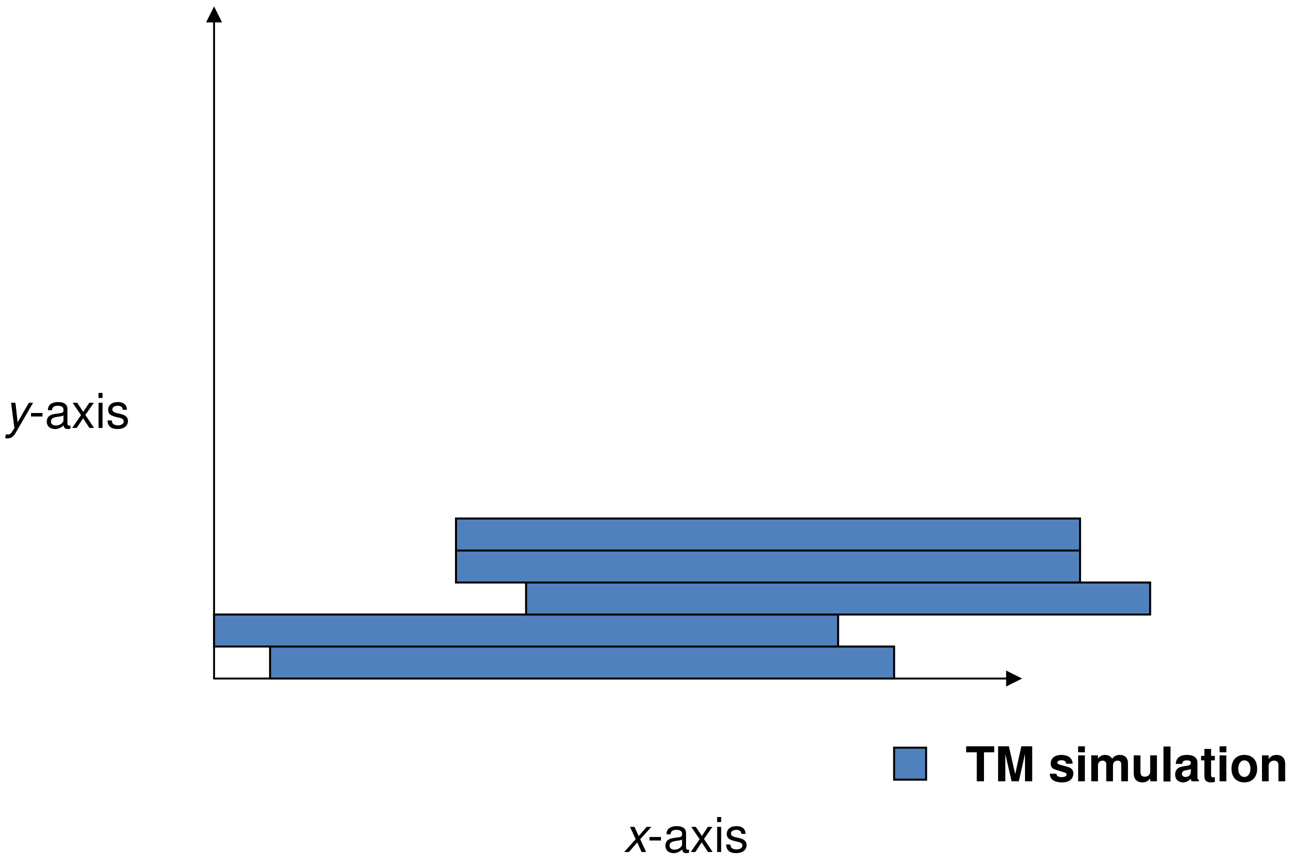
Second Main Result (proof idea)



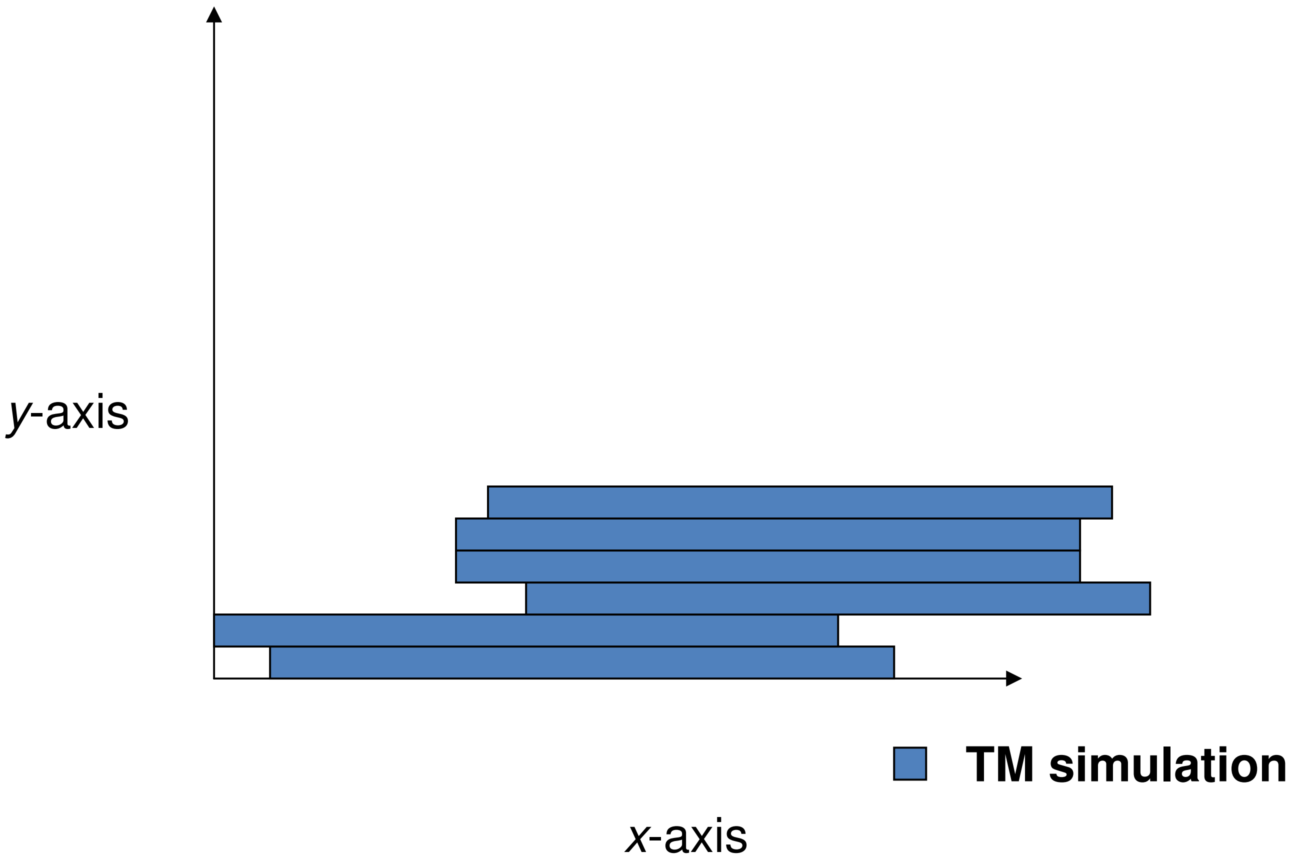
Second Main Result (proof idea)



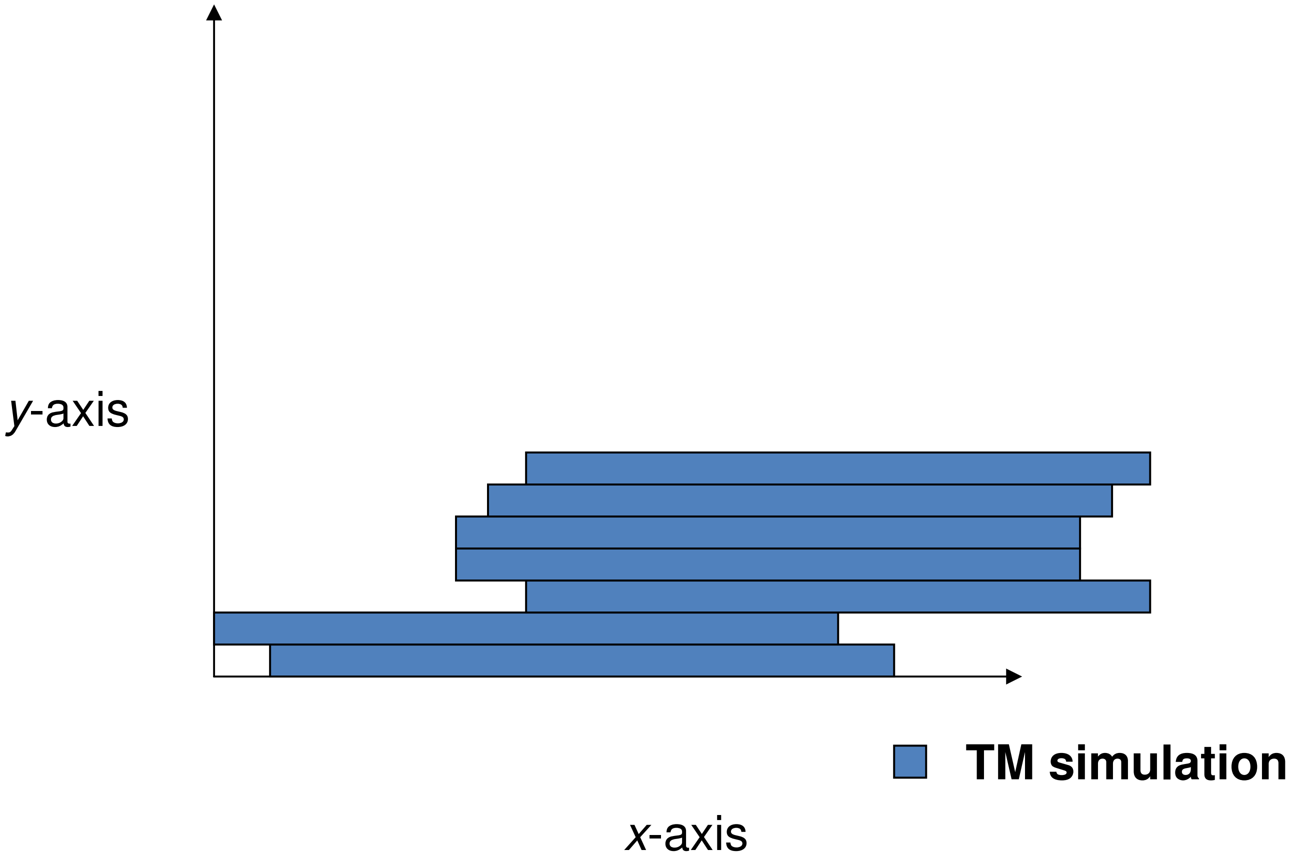
Second Main Result (proof idea)



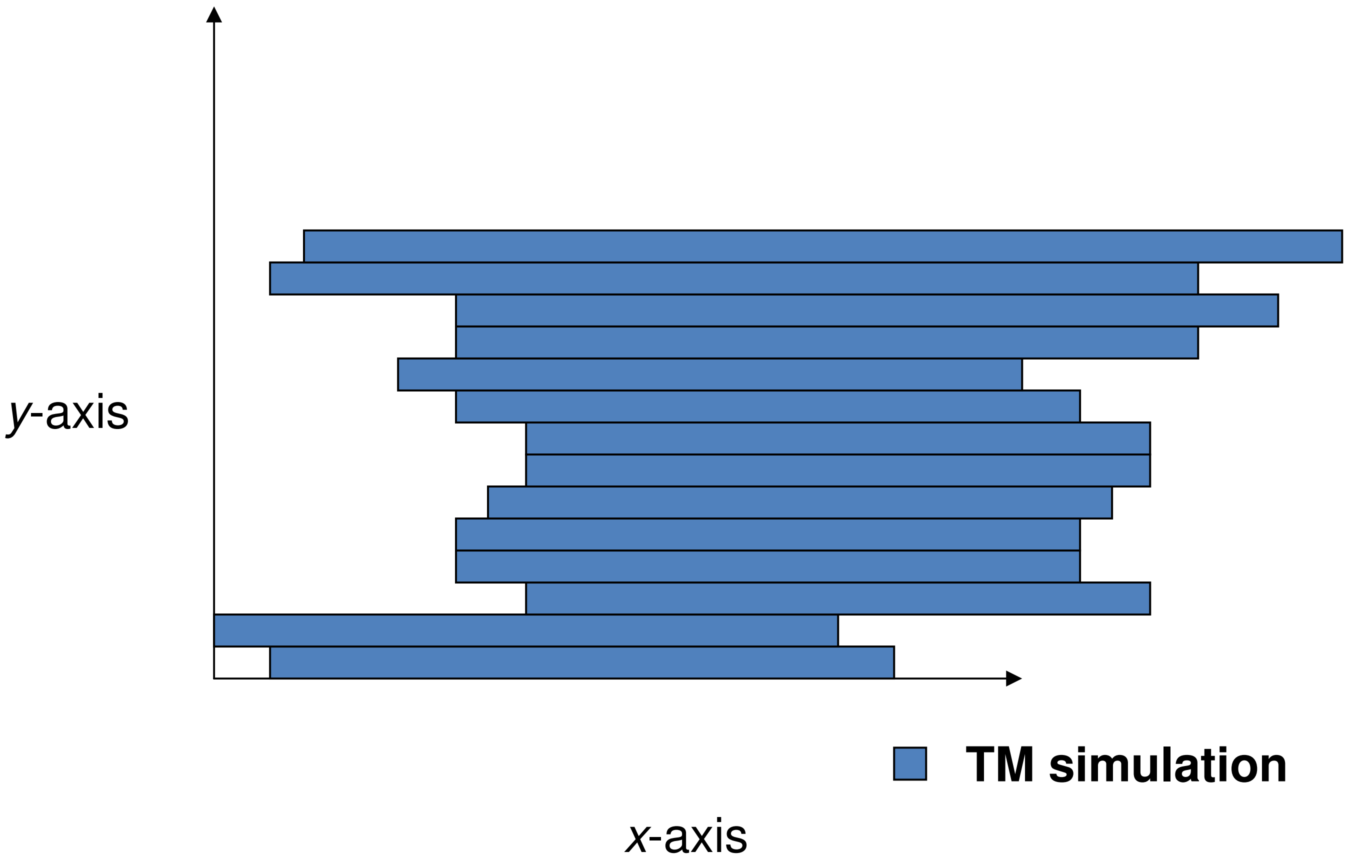
Second Main Result (proof idea)



Second Main Result (proof idea)



Second Main Result (proof idea)



Second Main Result (proof idea)

- Proof by contradiction

Second Main Result (proof idea)

- Proof by contradiction
- At least one of the yellow or black paths must turn “left” at some point (otherwise the language has small space complexity)

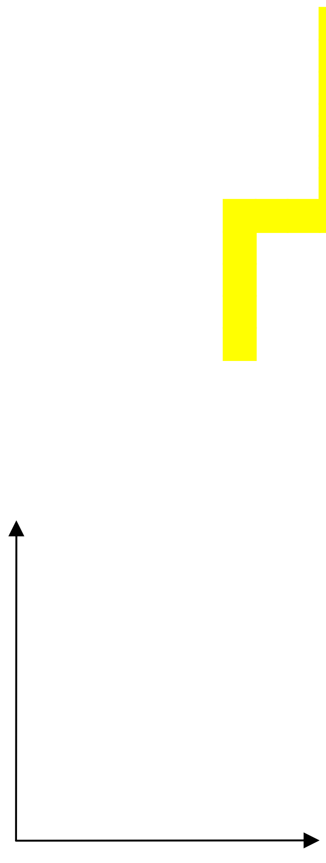
Second Main Result (proof idea)

- Proof by contradiction
- At least one of the yellow or black paths must turn “left” at some point (otherwise the language has small space complexity)
- Once a single path turns left, all must do so.

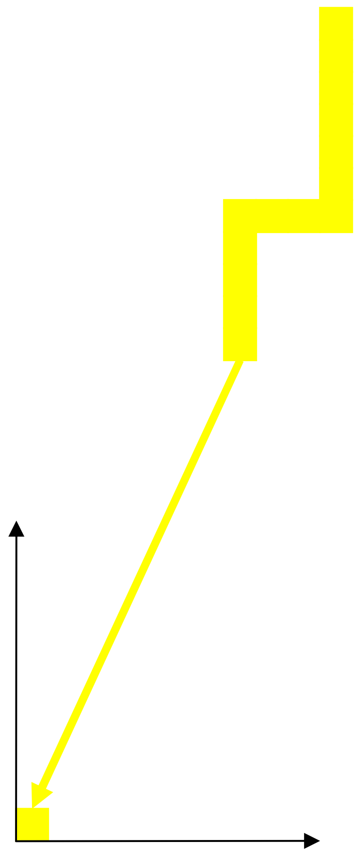
Second Main Result (proof idea)

- Proof by contradiction
- At least one of the yellow or black paths must turn “left” at some point (otherwise the language has small space complexity)
- Once a single path turns left, all must do so.
- Infinitely many paths turning left must eventually creep into another (adjacent) quadrant.

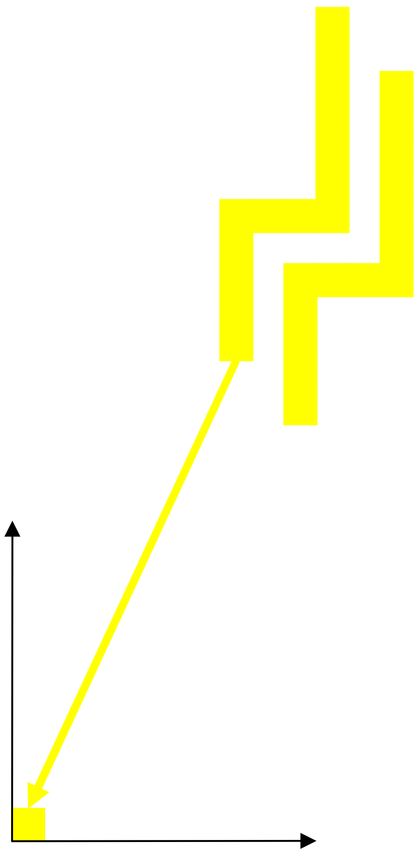
Second Main Result (proof idea)



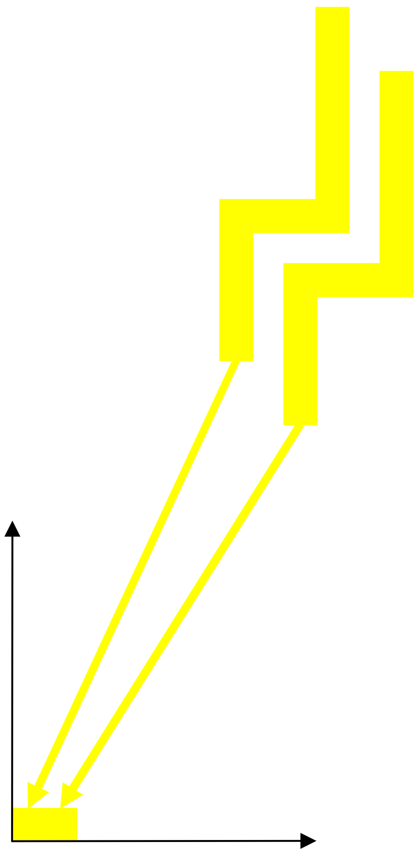
Second Main Result (proof idea)



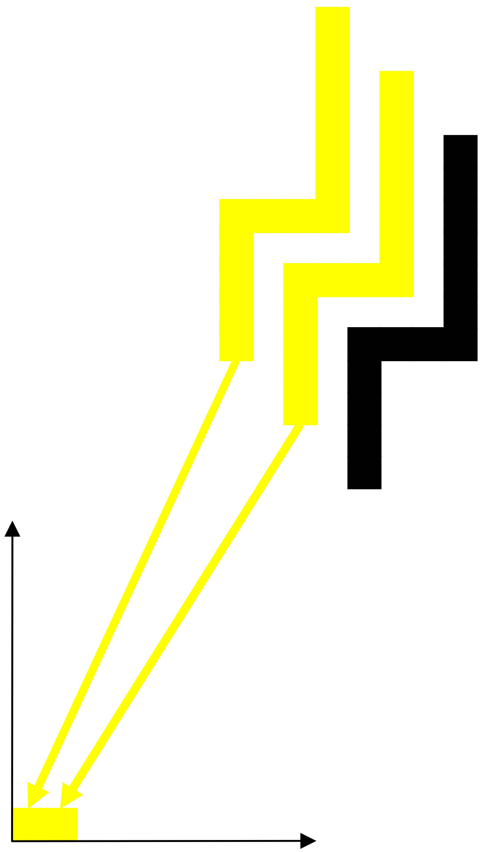
Second Main Result (proof idea)



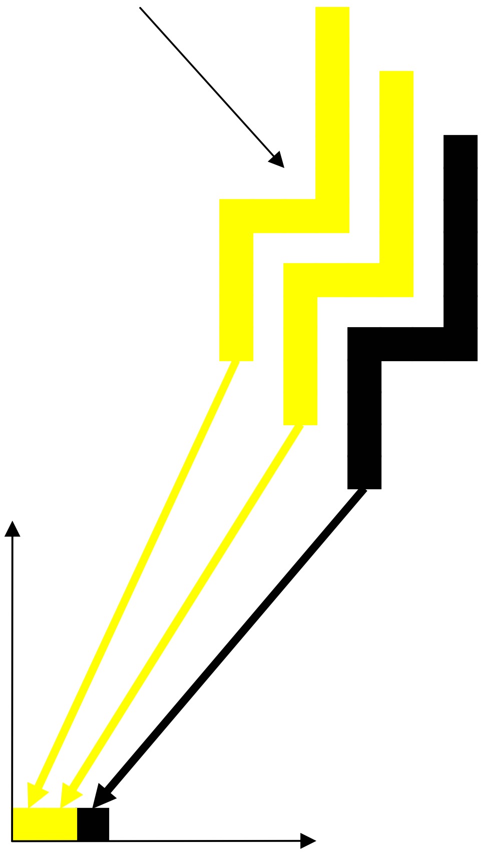
Second Main Result (proof idea)



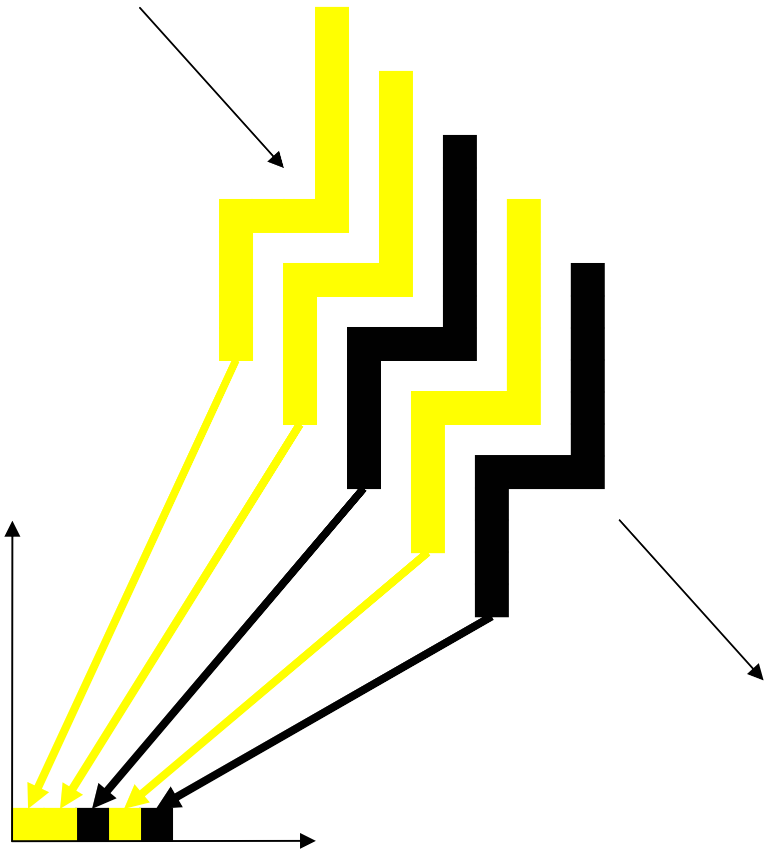
Second Main Result (proof idea)



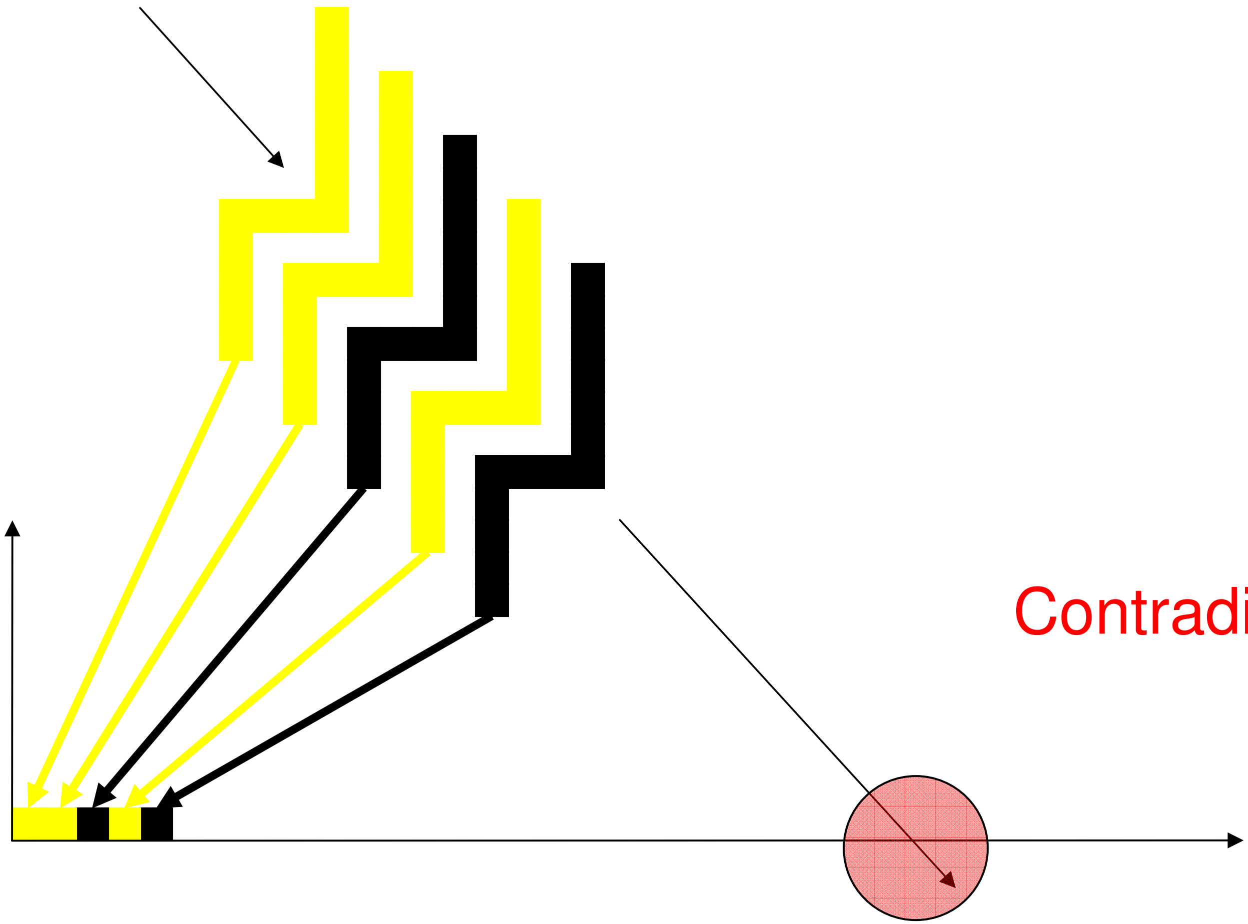
Second Main Result (proof idea)



Second Main Result (proof idea)



Second Main Result (proof idea)



Contradiction!

Web Site

Software which generates tile sets for the main construction from Turing machine definitions and tile assembly simulation software are freely available for download from the homepage for the ISU Laboratory for Nanoscale Self-Assembly:

http://www.cs.iastate.edu/~lnsa

Summary

In this paper we showed:

Summary

In this paper we showed:

1. A new characterization of decidable languages in terms of self-assembly

Summary

In this paper we showed:

1. A new characterization of decidable languages in terms of self-assembly
2. A lower bound on the space requirements of our construction

Summary

In this paper we showed:

1. A new characterization of decidable languages in terms of self-assembly
2. A lower bound on the space requirements of our construction

Thank you!