Small semi-weakly universal Turing machines

Damien Woods, Turlough Neary

University College Cork
Coláiste na hOllscoile Corcaigh

Department of Computer Science
University College Cork
Ireland

NUI MAYNOOTH
Ollscoil na hÉireann Má Nuad

Department of Computer Science
National University of Ireland, Maynooth
Ireland

MCU 2007

Acknowledgements

- MCU'07 co-chairs: Jérôme Durand-Lose, Maurice Margenstern
- Science Foundation Ireland (grant number 04/IN3/1524)
- Irish Research Council for Science, Engineering & Technology
- Cork Complex Systems Group, University College Cork, Ireland

Motivations

Some surprisingly simple processes are computationally powerful

Aim 1: To find the border between decidability and undecidability, with respect to the complexity measure of universal program size, e.g. by finding small universal machines or showing that none exist

Aim 2: To categorise trade-offs between universal-program size, and time, space and encoding complexity

Standard model & generalisations

Turing machine definition

One bi-infinite tape, one head, deterministic, usual blank symbol

- Most of the published work on the size of universal Turing machines has been for the standard model
  - A good (benchmark) model to compare new techniques
  - A lot of open questions remain
- There have also been results for restrictions and generalisations of the standard model
- Significant restrictions tend to have much larger programs
- Significant generalisations lead to smaller programs
- E.g. more tapes, dimensions, heads; couple with a FA, etc.
- Here we look at a generalisation called weakness

Standard, semi-weak, & weak machines

Turing machine Definition

One bi-infinite tape, one head, one-dimensional, deterministic, usual blank symbol

Semi-weak Turing machine definition

One bi-infinite tape, one head, one-dimensional, deterministic, (constant) blank word in one direction, blank symbol in the other

Weak Turing machine definition

One bi-infinite tape, one head, deterministic, (constant) blank word repeated to left and another blank word to right

Small machines

Some known results:

- small (standard) universal Turing machines
  - 2-tag simulators
  - bi-tag simulators
- small semi-weakly universal Turing machines
  - simulate Turing machines directly
- small weakly universal Turing machines
  - rule 110 simulators (new results since proceedings version)

New result:

- new small semi-weakly universal Turing machines
  - cyclic tag system simulators

Small machines

Some known results:

- small (standard) universal Turing machines
  - 2-tag simulators
  - bi-tag simulators
- small semi-weakly universal Turing machines
  - simulate Turing machines directly
- small weakly universal Turing machines
  - rule 110 simulators (new results since proceedings version)

New result:

- new small semi-weakly universal Turing machines
  - cyclic tag system simulators

History of weak machines

- Watanabe and Minksy
- Watanabe found smaller machines by generalising the model
- No results for a number of years
- Cook, Wolfram, Eppstein found significantly smaller machines
- Recently improved by Neary, Woods on the size of all weak machines (since proceedings; see arXiv)

Figure

New semi-weakly universal machines

- Here we give two new semi-weak machines
  - (4,5) and (3,7)
- Cyclic tag simulators
- Symmetric with Watanabe's semi-weak machines
  - (5,4) and (7,3)

Figure

Time complexity of small machines

- Until recently, almost all of the smallest universal Turing machines simulated via 2-tag systems
  - (modified tag systems, cyclic tag systems, rule 110, etc.)
- The Cocke-Minsky 2-tag universality proof: exponentially slow
- Now we know that 2-tag systems, bi-tag systems, cyclic tag systems, rule 110, run in polynomial time
- The smallest known universal Turing machines are efficient!
- Previous belief: there is an exponential trade-off between UTM program size complexity, and time/space complexity
- Current belief: there is little evidence for such a claim, as now *all* of the smallest UTMs are efficient

Time complexity: a plea

- Why are efficient simulations important?
- Universality of physical/biological/other systems that embed simple systems/UTMs
- Simple/small machines may be easier/better to embed in other systems
- Exponentially slow computations are exponentially boring!
- Clearer understanding of computation process
- P. van Emde Boas Class 1/Poly time Church-Turing
- It would be interesting to prove there is a polynomial (or even exponential) trade-off between program size and simulation time!

Time complexity: a plea

- Difficulty of prediction: NC versus P-completeness
- Machine prediction problem: Given an initial configuration and a time bound t in unary, will the machine ever reach a given state/configuration/etc. in time t?
- Exp slow simulators can not even solve in P problems in poly time
- Prediction problem has finite initial condition; even for weak machines!

Cyclic tag systems

Cyclic tag systems were used by M. Cook, S. Wolfram:

Turing machine $\mapsto$ 2-tag system
$\mapsto$ cyclic tag system $\mapsto$ Rule 110 $\mapsto$ small weak UTM

Cyclic tag systems

Cyclic tag system

A list of p binary words (appendants), $C = \alpha_0, \alpha_1, \ldots, \alpha_{p-1}$.

Configuration:

$$\alpha_0, \ldots, \boldsymbol{\alpha_m}, \ldots, \alpha_{p-1} \qquad x_0 x_1 \ldots x_\ell$$

Computation step:

1. $x_0 = 0 \Rightarrow$ delete x_0, increment marker to $m + 1 \mod p$.
2. $x_0 = 1 \Rightarrow$ delete x_0, increment marker to $m + 1 \mod p$, and append α_m to the right of the dataword.

Cyclic tag systems

Cyclic tag system

A list of p binary words (appendants), $C = \alpha_0, \alpha_1, \ldots, \alpha_{p-1}$.

Configuration:

$$\alpha_0, \ldots, \boldsymbol{\alpha_m}, \ldots, \alpha_{p-1} \qquad x_0 x_1 \ldots x_\ell$$

Computation step:

1. $x_0 = 0 \Rightarrow$ delete x_0, increment marker to $m+1 \mod p$.
2. $x_0 = 1 \Rightarrow$ delete x_0, increment marker to $m+1 \mod p$, and append α_m to the right of the dataword.

$$\begin{array}{rll} & \alpha_0, \ldots, \boldsymbol{\alpha_m}, \ldots, \alpha_{p-1} & x_0 x_1 \ldots x_\ell \\ \vdash & \alpha_0, \ldots, \alpha_m, \boldsymbol{\alpha_{m+1}}, \ldots, \alpha_{p-1} & x_1 \ldots x_\ell \end{array}$$

Cyclic tag systems

Cyclic tag system

A list of p binary words (appendants), $C = \alpha_0, \alpha_1, \ldots, \alpha_{p-1}$.

Configuration:

$$\alpha_0, \ldots, \boldsymbol{\alpha_m}, \ldots, \alpha_{p-1} \qquad x_0 x_1 \ldots x_\ell$$

Computation step:

1. $x_0 = 0 \Rightarrow$ delete x_0, increment marker to $m+1 \mod p$.
2. $x_0 = 1 \Rightarrow$ delete x_0, increment marker to $m+1 \mod p$, and append α_m to the right of the dataword.

$$\begin{array}{rll} & \alpha_0, \ldots, \boldsymbol{\alpha_m}, \ldots, \alpha_{p-1} & x_0 x_1 \ldots x_\ell \\ \vdash & \alpha_0, \ldots, \alpha_m, \boldsymbol{\alpha_{m+1}}, \ldots, \alpha_{p-1} & x_1 \ldots x_\ell \alpha_m \end{array}$$

Cyclic tag system

Example (Appendants: $C = 00, 010, 11$ Input word: 0010010)

	appendants	dataword
	00, 1010, 10	0010010
$\vdash$	00, **1010**, 10	010010
$\vdash$	00, 1010, **10**	10010
$\vdash$	**00**, 1010, 10	001010
$\vdash$	00, **1010**, 10	01010
$\vdash$	00, 1010, **10**	1010
$\vdash$	**00**, 1010, 10	01010
$\vdash$	00, **1010**, 10	1010
$\vdash$	00, 1010, **10**	0101010
$\vdash$	...	

Cyclic tag system

Example (Appendants: $C = 00, 010, 11$ Input word: 0010010)

	appendants	dataword
	00, 1010, 10	0010010
$\vdash$	00, **1010**, 10	010010
$\vdash$	00, 1010, **10**	10010
$\vdash$	**00**, 1010, 10	001010
$\vdash$	00, **1010**, 10	01010
$\vdash$	00, 1010, **10**	1010
$\vdash$	**00**, 1010, 10	01010
$\vdash$	00, **1010**, 10	1010
$\vdash$	00, 1010, **10**	0101010
$\vdash$	...	

Cyclic tag system

Example (Appendants: $C = 00, 010, 11$ Input word: 0010010)

	appendants	dataword
	00, 1010, 10	0010010
$\vdash$	00, **1010**, 10	010010
$\vdash$	00, 1010, **10**	10010
$\vdash$	**00**, 1010, 10	001010
$\vdash$	00, **1010**, 10	01010
$\vdash$	00, 1010, **10**	1010
$\vdash$	**00**, 1010, 10	01010
$\vdash$	00, **1010**, 10	1010
$\vdash$	00, 1010, **10**	0101010
$\vdash$	...	

Cyclic tag system

Example (Appendants: $C = 00, 010, 11$ Input word: 0010010)

	appendants	dataword
	00, 1010, 10	0010010
$\vdash$	00, **1010**, 10	010010
$\vdash$	00, 1010, **10**	10010
$\vdash$	**00**, 1010, 10	001010
$\vdash$	00, **1010**, 10	01010
$\vdash$	00, 1010, **10**	1010
$\vdash$	**00**, 1010, 10	01010
$\vdash$	00, **1010**, 10	1010
$\vdash$	00, 1010, **10**	0101010
$\vdash$	...	

Cyclic tag system

Example (Appendants: $C = 00, 010, 11$ Input word: 0010010)

	appendants	dataword
	00, 1010, 10	0010010
$\vdash$	00, **1010**, 10	010010
$\vdash$	00, 1010, **10**	10010
$\vdash$	**00**, 1010, 10	001010
$\vdash$	00, **1010**, 10	01010
$\vdash$	00, 1010, **10**	1010
$\vdash$	**00**, 1010, 10	01010
$\vdash$	00, **1010**, 10	1010
$\vdash$	00, 1010, **10**	0101010
$\vdash$	...	

4-state, 5-symbol machine

- Simulates cyclic tag systems where dataword and appendants are from $\{0, 10\}^*$, i.e. no consecutive 1 symbols
- Simulates in polynomial time
- The start state is u_1 and the blank symbol is $\emptyset$
- Table of behaviour for $U_{4,5}$:

	u_1	u_2	u_3	u_4
0	$\lambda \mathrm{L} u_1$	$\lambda \mathrm{L} u_2$	$\emptyset \mathrm{R} u_3$	$\emptyset \mathrm{R} u_4$
1	$\emptyset \mathrm{R} u_2$	$1 \mathrm{L} u_2$	$1 \mathrm{R} u_3$	$1 \mathrm{R} u_4$
λ	$0 \mathrm{R} u_2$	$0 \mathrm{R} u_1$	$0 \mathrm{R} u_4$	$0 \mathrm{R} u_3$
$\emptyset$		$0 \mathrm{L} u_2$	$0 \mathrm{L} u_2$	$1 \mathrm{L} u_2$
~~1~~	$0 \mathrm{L} u_2$	$\lambda \mathrm{L} u_3$		

Encoding for 4-state, 5-symbol machine

Given a CTS configuration

Example ($C = 00, 010, 11$, input word 011)

$$\mathbf{00}, 1010, 10 \qquad 0010010$$

Reverse order of symbols in each appendant

$$\mathbf{00}, 0101, 01 \qquad 0010010$$

Reverse order of appendants

$$01, 0101, \mathbf{00} \qquad 0010010$$

Dataword requires no special encoding

Encoding for 4-state, 5-symbol machine

Cyclic tag system, with dataword:

$$01, 0101, \mathbf{00} \qquad 0010010$$

Encode appendants via

- $\langle 0 \rangle = 0\lambda\not{1}0$
- $\langle 1 \rangle = 00\lambda\not{1}$

And separate with λ

$$\lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0} \qquad 0010010$$

Appendants repeated on left. Blank symbol $\emptyset$ repeated on right

$$\ldots \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}00\lambda\not{1}0\ \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0}\ 0010010\ \emptyset\emptyset\emptyset \ldots$$

Encoding for 4-state, 5-symbol machine

Cyclic tag system, with dataword:

$$01, 0101, \mathbf{00} \qquad 0010010$$

Encode appendants via

- $\langle 0 \rangle = 0\lambda\not{1}0$
- $\langle 1 \rangle = 00\lambda\not{1}$

And separate with λ

$$\lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0} \qquad 0010010$$

Appendants repeated on left. Blank symbol $\emptyset$ repeated on right

$$\ldots \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}00\lambda\not{1}0\ \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0}\ 0010010\ \emptyset\emptyset\emptyset \ldots$$

Encoding for 4-state, 5-symbol machine

Cyclic tag system, with dataword:

$$01, 0101, \mathbf{00} \qquad 0010010$$

Encode appendants via

- $\langle 0 \rangle = 0\lambda\not{1}0$
- $\langle 1 \rangle = 00\lambda\not{1}$

And separate with λ

$$\lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0} \qquad 0010010$$

Appendants repeated on left. Blank symbol $\emptyset$ repeated on right

$$\ldots\lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}00\lambda\not{1}0\ \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0}\ 0010010\ \emptyset\emptyset\emptyset\ldots$$

Encoding for 4-state, 5-symbol machine

Cyclic tag system, with dataword:

$$01, 0101, \mathbf{00} \qquad 0010010$$

Encode appendants via

- $\langle 0 \rangle = 0\lambda\not{1}0$
- $\langle 1 \rangle = 00\lambda\not{1}$

And separate with λ

$$\lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0} \qquad 0010010$$

Appendants repeated on left. Blank symbol $\emptyset$ repeated on right

$$\ldots \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}00\lambda\not{1}0\ \lambda 0\lambda\not{1}000\lambda\not{1}\ \lambda 0\lambda\not{1}000\lambda\not{1}0\lambda\not{1}000\lambda\not{1}\ \boldsymbol{\lambda 0\lambda\not{1}00\lambda\not{1}0}\ 0010010\ \emptyset\emptyset\emptyset \ldots$$

Simulation of reading 0

$$\begin{array}{lll} & \mathbf{00}, 1010, 10 & 0010010 \\ \vdash & 00, \mathbf{1010}, 10 & 010010 \end{array}$$

$$\lambda 0 \lambda \not 1 000 \lambda \not 1 \quad \lambda 0 \lambda \not 1 000 \lambda \not 1 0 \lambda \not 1 000 \lambda \not 1 \quad \boldsymbol{\lambda 0 \lambda \not 1 00 \lambda \not 1 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0\lambda\not 1 0$ or $\langle 1 \rangle = 00\lambda\not 1$ in state u_1 it gets deleted:

$$\boldsymbol{u_1}, \quad \ldots 0 \lambda \not 1 \underline{0}$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{lll} & \mathbf{00}, 1010, 10 & 0010010 \\ \vdash & 00, \mathbf{1010}, 10 & 010010 \end{array}$$

$$\lambda 0 \lambda \not 1 000 \lambda \not 1 \quad \lambda 0 \lambda \not 1 000 \lambda \not 1 0 \lambda \not 1 000 \lambda \not 1 \quad \boldsymbol{\lambda 0 \lambda \not 1 00 \lambda \not 1 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0\lambda\not 1 0$ or $\langle 1 \rangle = 00\lambda\not 1$ in state u_1 it gets deleted:

$$\boldsymbol{u_1}, \quad \ldots 0 \lambda \not 1 \lambda$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\lambda 0 \lambda \not{1} 000 \lambda \not{1} \quad \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \quad \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0 \lambda \not{1} 0$ or $\langle 1 \rangle = 00 \lambda \not{1}$ in state u_1 it gets deleted:

$$\boldsymbol{u_2}, \quad \ldots 0 \underline{\lambda} 0 \lambda$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\lambda 0 \lambda \not{1} 000 \lambda \not{1} \quad \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \quad \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0 \lambda \not{1} 0$ or $\langle 1 \rangle = 00 \lambda \not{1}$ in state u_1 it gets deleted:

$$\boldsymbol{u_1}, \quad \ldots 00\underline{0}\lambda$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\lambda 0 \lambda \not{1} 000 \lambda \not{1} \quad \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \quad \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0 \lambda \not{1} 0$ or $\langle 1 \rangle = 00 \lambda \not{1}$ in state u_1 it gets deleted:

$$\boldsymbol{u_1}, \quad \ldots 0 \underline{0} \lambda \lambda$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\lambda 0 \lambda \not{1} 000 \lambda \not{1} \quad \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \quad \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0 \lambda \not{1} 0$ or $\langle 1 \rangle = 00 \lambda \not{1}$ in state u_1 it gets deleted:

$$\boldsymbol{u_1}, \quad \ldots \underline{0} \lambda \lambda \lambda$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\lambda 0 \lambda \not{1} 000 \lambda \not{1} \quad \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \quad \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \quad 0010010$$

If we 'enter' $\langle 0 \rangle = 0 \lambda \not{1} 0$ or $\langle 1 \rangle = 00 \lambda \not{1}$ in state u_1 it gets deleted:

$$\boldsymbol{u_1}, \quad \ldots \_ \lambda \lambda \lambda \lambda$$

We exit left in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \qquad \underline{0}010010$$

If we 'enter' $\langle 0 \rangle = 0\lambda\not{1}0$ or $\langle 1 \rangle = 00\lambda\not{1}$ in state u_1 it gets deleted

$$\boldsymbol{u_1}, \quad \ldots 0 \lambda \not{1} \underline{0} \quad \vdash^* \quad \boldsymbol{u_1}, \quad \ldots \_ \lambda\lambda\lambda\lambda$$

We exit in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 00 \underline{\lambda \not{1} 0}} \qquad \lambda 010010$$

If we 'enter' $\langle 0 \rangle = 0\lambda\not{1}0$ or $\langle 1 \rangle = 00\lambda\not{1}$ in state u_1 it gets deleted

$$\boldsymbol{u_1}, \quad \ldots 0 \lambda \not{1} \underline{0} \quad \vdash^* \quad \boldsymbol{u_1}, \quad \ldots \_ \lambda\lambda\lambda\lambda$$

We exit in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda \underline{0 \lambda \not{1} 0} \lambda \lambda \lambda \lambda} \qquad \lambda 010010$$

If we 'enter' $\langle 0 \rangle = 0 \lambda \not{1} 0$ or $\langle 1 \rangle = 00 \lambda \not{1}$ in state u_1 it gets deleted

$$\boldsymbol{u_1}, \quad \ldots 0 \lambda \not{1} \underline{0} \quad \vdash^* \quad \boldsymbol{u_1}, \quad \ldots \_ \lambda \lambda \lambda \lambda$$

We exit in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \underline{\boldsymbol{\lambda}} \boldsymbol{\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \qquad \lambda 010010$$

If we 'enter' $\langle 0 \rangle = 0\lambda\not{1}0$ or $\langle 1 \rangle = 00\lambda\not{1}$ in state u_1 it gets deleted

$$\boldsymbol{u_1}, \quad \ldots 0 \lambda \not{1} \underline{0} \quad \vdash^* \quad \boldsymbol{u_1}, \quad \ldots \_ \lambda\lambda\lambda\lambda$$

We exit in state u_1

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{0\underline{\lambda}\lambda\lambda\lambda\lambda\lambda\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{00\underline{\lambda}\lambda\lambda\lambda\lambda\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{000\underline{\lambda}\lambda\lambda\lambda\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{0000\underline{\lambda}\lambda\lambda\lambda\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{00000\underline{\lambda}\lambda\lambda\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{000000\underline{\lambda}\lambda\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{0000000\underline{\lambda}\lambda} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{00000000\underline{\lambda}} \qquad \lambda 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{ll} & \mathbf{00}, 1010, 10 \quad 0010010 \\ \vdash & 00, \mathbf{1010}, 10 \quad 010010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{000000000} \qquad \underline{\lambda} 010010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{lll} & \mathbf{00}, 1010, 10 & 0010010 \\ \vdash & 00, \mathbf{1010}, 10 & 010010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; \mathbf{000000000} \qquad 0\underline{0}10010$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 0

$$\begin{array}{lll} & 00, \mathbf{1010}, 10 & 010010 \\ \vdash & 00, 1010, \mathbf{10} & 10010 \end{array}$$

$$\begin{array}{lll}
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1}} \; 000000000 & 0\underline{0}10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1}} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} \underline{0} \lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} \underline{0} \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\underline{\lambda} \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010
\end{array}$$

If we 'enter' $\langle 0 \rangle = 0\lambda\not{1}0$ or $\langle 1 \rangle = 00\lambda\not{1}$ in state u_1 it gets deleted

Simulation of reading 0

$$\begin{array}{ll} & 00, \mathbf{1010}, 10 \quad 010010 \\ \vdash & 00, 1010, \mathbf{10} \quad 10010 \end{array}$$

$$\begin{array}{lll}
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1}} \; 000000000 & 0\underline{0}10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \underline{\not{1}}} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} \underline{0} \lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \underline{\not{1}} \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\lambda 0 \lambda \not{1} \underline{0} \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1} \; \boldsymbol{\underline{\lambda} \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda} \; \lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda\lambda & \lambda\lambda 10010 \\
\boldsymbol{u_1}, & \ldots \boldsymbol{\lambda 0 \lambda \not{1} 000 \lambda \not{1}} \; 0000000000000000 \; 000000000 & 00\underline{1}0010
\end{array}$$

Move right, switching between two states, exit in state u_1, iterate.

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \boldsymbol{\lambda} 0 \boldsymbol{\lambda} 1 0 00 \boldsymbol{\lambda} 1 \; 0^{17} \; 0^{9} \qquad 00\underline{1}0010 \; \emptyset\emptyset\emptyset \ldots$$

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\begin{array}{lll} \boldsymbol{u_1}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & 00\underline{1}0010\ \emptyset\emptyset\emptyset\ldots \\ \boldsymbol{u_2}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & 00\emptyset\underline{0}010\ \emptyset\emptyset\emptyset\ldots \end{array}$$

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\begin{array}{lll} \boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 00\underline{1}0010\ \emptyset\emptyset\emptyset\ldots \\ \boldsymbol{u_2}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 00\emptyset\underline{0}010\ \emptyset\emptyset\emptyset\ldots \\ \boldsymbol{u_2}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 00\underline{\emptyset}\lambda 010\ \emptyset\emptyset\emptyset\ldots \end{array}$$

Simulation of reading 10

$$
\begin{array}{ll}
 & 00, 1010, \mathbf{10} \quad 10010 \\
\vdash^2 & 00, \mathbf{1010}, 10 \quad 01010
\end{array}
$$

$$
\begin{array}{lll}
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 00\underline{1}0010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 00\emptyset\underline{0}010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 00\underline{\emptyset}\lambda 010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda \not{1} 000 \lambda \not{1}\ 0^{17}\ 0^9 & 0\underline{0}0\lambda 010\ \emptyset\emptyset\emptyset\ldots
\end{array}
$$

Simulation of reading 10

$$
\begin{array}{ll}
 & 00, 1010, \mathbf{10} \quad 10010 \\
\vdash^2 & 00, \mathbf{1010}, 10 \quad 01010
\end{array}
$$

$$
\begin{array}{lll}
\boldsymbol{u_1}, & \ldots \lambda 0 \lambda 1 0 00 \lambda 1 \; 0^{17} \; 0^9 & 00\underline{1}0010 \;\; \emptyset\emptyset\emptyset \ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda 1 0 00 \lambda 1 \; 0^{17} \; 0^9 & 00\emptyset\underline{0}010 \;\; \emptyset\emptyset\emptyset \ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda 1 0 00 \lambda 1 \; 0^{17} \; 0^9 & 00\underline{\emptyset}\lambda 010 \;\; \emptyset\emptyset\emptyset \ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda 1 0 00 \lambda 1 \; 0^{17} \; 0^9 & 0\underline{0}0\lambda 010 \;\; \emptyset\emptyset\emptyset \ldots \\
\boldsymbol{u_2}, & \ldots \lambda 0 \lambda 1 0 00 \lambda 1 \; 0^{17} \; 0^9 & \underline{0}\lambda 0\lambda 010 \;\; \emptyset\emptyset\emptyset \ldots
\end{array}
$$

Simulation of reading 10

$$
\begin{array}{ll}
 & 00, 1010, \mathbf{10} \quad 10010 \\
\vdash^2 & 00, \mathbf{1010}, 10 \quad 01010
\end{array}
$$

$$
\begin{array}{lll}
\boldsymbol{u_1}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & 00\underline{1}0010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & 00\emptyset\underline{0}010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & 00\emptyset\underline{\lambda}010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & 0\underline{0}\lambda010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ 0^{17}\ 0^9 & \underline{0}\lambda0\lambda010\ \emptyset\emptyset\emptyset\ldots \\
\boldsymbol{u_2}, & \ldots \boldsymbol{\lambda 0 \lambda 1 0 0 0 \lambda 1}\ \lambda^{17}\ \lambda^9 & \lambda\lambda0\lambda010\ \emptyset\emptyset\emptyset\ldots
\end{array}
$$

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$u_2, \quad \ldots \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} \quad \lambda^{17} \quad \lambda^9 \qquad \lambda\lambda 0 \lambda 010 \quad \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} u_2, & \ldots 00\lambda \not{1} \\ u_3, & \ldots 00\underline{\lambda}\lambda \\ u_4, & \ldots 000\underline{\lambda} \\ u_3, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} u_2, & \ldots 0\lambda \not{1} \underline{0} \\ u_2, & \ldots 0\lambda \not{1} \underline{\lambda}\lambda \\ u_3, & \ldots 0\underline{0}\lambda\lambda \\ u_4, & \ldots 00\underline{0}\lambda \\ u_3, & \ldots 0000 \\ u_4, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_3}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda 1 0 0 0 \underline{\lambda} \lambda} \;\; \lambda^{17} \;\; \lambda^{9} \qquad \lambda\lambda 0 \lambda 010 \;\; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{1} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{0} \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{\lambda}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 0000 \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_4}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda 1 0 0 0 0 \underline{\lambda}} \;\; \lambda^{17} \;\; \lambda^{9} \qquad \lambda\lambda 0 \lambda 010 \;\; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{1} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}0 \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_3}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda 1 0 0 0 0} \; 0^{17} \; 0^9 \qquad \underline{\lambda} \lambda 0 \lambda 010 \; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{1} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}0 \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_4}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda 1 0 0 0 0} \; 0^{17} \; 0^9 \qquad 0\underline{\lambda}0\lambda 010 \; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{1} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}0 \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_3}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda \not{1} 0 0000} \;\; 0^{17} \;\; 0^9 \qquad 00\underline{0}\lambda 010 \;\; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{\not{1}} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}0} \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_3}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda \not{1} 0 0000} \;\; 0^{17} \;\; 0^9 \qquad 00\emptyset\underline{\lambda}010 \;\; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{\not{1}} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}0} \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 0000 \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_4}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda \not{1} 0 0000} \;\; 0^{17} \;\; 0^9 \qquad 00\emptyset 0\underline{0}10 \;\; \emptyset\emptyset\emptyset \ldots$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{\not{1}} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}0} \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_4}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda \not{1} 0 0000} \;\; 0^{17} \;\; 0^9 \qquad 00\emptyset 0\emptyset 1\emptyset \;\; \underline{\emptyset}\emptyset\emptyset$$

$\langle 1 \rangle$

$$\begin{array}{rl} \boldsymbol{u_2}, & \ldots 00\lambda \underline{\not{1}} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{rl} \boldsymbol{u_2}, & \ldots 0\lambda \not{1} \underline{0} \\ \boldsymbol{u_2}, & \ldots 0\lambda \underline{\not{1}} \lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda \not{1} 0 0000} \;\; 0^{17} \;\; 0^9 \qquad 00\emptyset 0\emptyset 1\underline{\emptyset} \;\; 1\emptyset\emptyset$$

$\langle 1 \rangle$

$$\begin{array}{rl} \boldsymbol{u_2}, & \ldots 00\lambda\underline{\not{1}} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{rl} \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}0} \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{\not{1}}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 000\underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \; \boldsymbol{\lambda 0 \lambda \not{1} \underline{0} \lambda \lambda \lambda \lambda} \;\; \lambda^{17} \;\; \lambda^{9} \qquad \lambda\lambda 0 \lambda 010 \;\; 1\emptyset\emptyset$$

$$\begin{array}{ll} & \langle 1 \rangle \\ \boldsymbol{u_2}, & \ldots 00 \lambda \not{\underline{1}} \\ \boldsymbol{u_3}, & \ldots 00 \underline{\lambda} \lambda \\ \boldsymbol{u_4}, & \ldots 000 \underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$$\begin{array}{ll} & \langle 0 \rangle \\ \boldsymbol{u_2}, & \ldots 0 \lambda \not{\underline{1}} 0 \\ \boldsymbol{u_2}, & \ldots 0 \lambda \not{\underline{1}} \lambda \\ \boldsymbol{u_3}, & \ldots 0 \underline{0} \lambda \lambda \\ \boldsymbol{u_4}, & \ldots 00 \underline{0} \lambda \\ \boldsymbol{u_3}, & \ldots 000 \underline{0} \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_3}, \quad \ldots \; \boldsymbol{\lambda 0000000} \;\; 0^{17} \;\; 0^9 \qquad 00\emptyset 0\emptyset 1\emptyset \;\; 1\underline{\emptyset}\emptyset$$

$$\begin{array}{rl} & \langle 1 \rangle \\ \boldsymbol{u_2}, & \ldots 00\lambda\underline{1'} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$$\begin{array}{rl} & \langle 0 \rangle \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{1'0} \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{1'}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 0000 \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \; \boldsymbol{\lambda 0000000} \; 0^{17} \; 0^9 \qquad 00\emptyset 0\emptyset 1\emptyset \; \underline{1}0\emptyset$$

$$\begin{array}{rl} & \langle 1 \rangle \\ \boldsymbol{u_2}, & \ldots 00\lambda \not{\underline{1}} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$$\begin{array}{rl} & \langle 0 \rangle \\ \boldsymbol{u_2}, & \ldots 0\lambda \not{\underline{1}} 0 \\ \boldsymbol{u_2}, & \ldots 0\lambda \not{\underline{1}} \lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 0000 \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_2}, \quad \ldots \underline{\boldsymbol{\lambda}}\boldsymbol{\lambda}\boldsymbol{\lambda}\boldsymbol{\lambda}\boldsymbol{\lambda}\boldsymbol{\lambda}\boldsymbol{\lambda} \ \lambda^{17} \ \lambda^{9} \qquad \lambda\lambda 0\lambda 010 \ 10\emptyset$$

$\langle 1 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 00\lambda\underline{1} \\ \boldsymbol{u_3}, & \ldots 00\underline{\lambda}\lambda \\ \boldsymbol{u_4}, & \ldots 000\underline{\lambda} \\ \boldsymbol{u_3}, & \ldots 0000\_ \end{array}$$

$\langle 0 \rangle$

$$\begin{array}{ll} \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}0 \\ \boldsymbol{u_2}, & \ldots 0\lambda\underline{1}\lambda \\ \boldsymbol{u_3}, & \ldots 0\underline{0}\lambda\lambda \\ \boldsymbol{u_4}, & \ldots 00\underline{0}\lambda \\ \boldsymbol{u_3}, & \ldots 0000 \\ \boldsymbol{u_4}, & \ldots 0000\_ \end{array}$$

So we exit the encoded symbol in one of two states:
$u_3 \Rightarrow$ append 1; and $u_4 \Rightarrow$ append 0

Simulation of reading 10

$$\begin{array}{lll} & 00, 1010, \mathbf{10} & 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 & 01010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \; \mathbf{0^9} \;\; 0^{17} \;\; 0^9 \qquad 00\underline{0}\lambda 010 \;\; 10\emptyset$$

Simulation of reading 10

$$
\begin{array}{lll}
 & 00, 1010, \mathbf{10} & 10010 \\
\vdash^2 & 00, \mathbf{1010}, 10 & 01010
\end{array}
$$

$$
\boldsymbol{u_1}, \quad \ldots \quad \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \quad \boldsymbol{\lambda 0 \lambda \not{1} 00 \lambda \not{1} 0} \quad \mathbf{0^9} \quad 0^{17} \quad 0^9 \qquad 00\underline{0}\lambda 010 \quad 10\emptyset
$$

Simulation of reading 10

$$\begin{array}{ll} & 00, 1010, \mathbf{10} \quad 10010 \\ \vdash^2 & 00, \mathbf{1010}, 10 \quad 01010 \end{array}$$

$$\boldsymbol{u_1}, \quad \ldots \; \lambda 0 \lambda \not{1} 000 \lambda \not{1} 0 \lambda \not{1} 000 \lambda \not{1} \; 0^9 \;\; 0^9 \;\; 0^{17} \;\; 0^9 \qquad 0000\underline{0}10 \;\; 10\emptyset \ldots$$

After the simulation of reading 10:

- we have appended the appendant 10
- we have indexed two further appendants
- dataword is easily decodable (from head to blank symbol)

Summary

- Use of parity in $U_{4,5}$
- Key point: encode symbols via shifted subwords & parity
- For both machines:
- Simulation runs in polynomial time $O(t^4 \log^2 t)$
- Encoding in logspace (actually constant space & linear time)
- Prediction is P-complete (even for finite configurations)

Conclusion & future work

- Find more semi-weak machines to complete the 'semi-weak universal curve'
- Explore encoding and decoding issues in a formal way?
- Exploring the 'open' space for standard machines (via weakness and other generalisations)
- Lower bounds on size for weak/semi-weak machines?
- Time complexity upper and lower bounds?
- New techniques for even smaller semi-weak machines?