

Busy Beaver Scores and Alphabet Size

Holger Petersen
FCT 2017, Bordeaux

Reminder: Busy Beaver Game (1)

Defined by T. Rado in 1962

Based on deterministic single-tape Turing machines with binary alphabet, steps characterized by:

1. Current state
2. Symbol scanned
3. Symbol written
4. Head movement (mandatory)
5. New state (including halt-state “for free”)

“quintuple Turing machine variant”



Reminder: Busy Beaver Game (2)

Start TM on all-blank (0's) tape, if it stops:

- Number of steps is **activity**
- Number of ones is **productivity**

Functions defined by Rado:

- $S(n)$ = maximum activity of any n -state TM
- $\Sigma(n)$ = maximum productivity of any n -state TM

Rado showed that $S(n)$ and $\Sigma(n)$ grow faster than any computable function



$S(n)$ and $\Sigma(n)$ are not computable (or recursive)

Reminder: Busy Beaver Game (3)

Generalization to alphabets with $m \geq 2$ symbols:

- $S(n, m)$ = max. activity of n -state, m -symbol TM
- $\Sigma(n, m)$ = max. productivity n -state, m -symbol TM

Productivity: number of non-blanks

Trivial observation:

$$S(n, m) \geq S(n, 2) \text{ and } \Sigma(n, m) \geq \Sigma(n, 2)$$



$S(n, m)$ and $\Sigma(n, m)$ are not computable

Goal of the game: Find machines of maximum activity/productivity (Busy Beavers)

Motivation

On a simple source for non-computable functions
(Rado's contribution to: Symposium on
Mathematical Theory of Automata, April 1962)

Metamathematical interest:

Knowing $\Sigma(n)$ or $S(n)$ for sufficiently large n would settle Goldbach's conjecture and other conjectures disproved by counterexamples (Chaitin 1987)

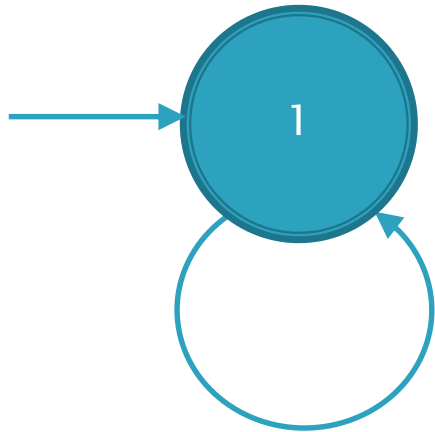
Concrete value of n (Yedidia and Aaronson 2016):
 $n = 4888$ suffices for Goldbach's conjecture

Improved to 47 and then 31 in Aaronson's blog



Warm-Up: 1-State Busy Beaver

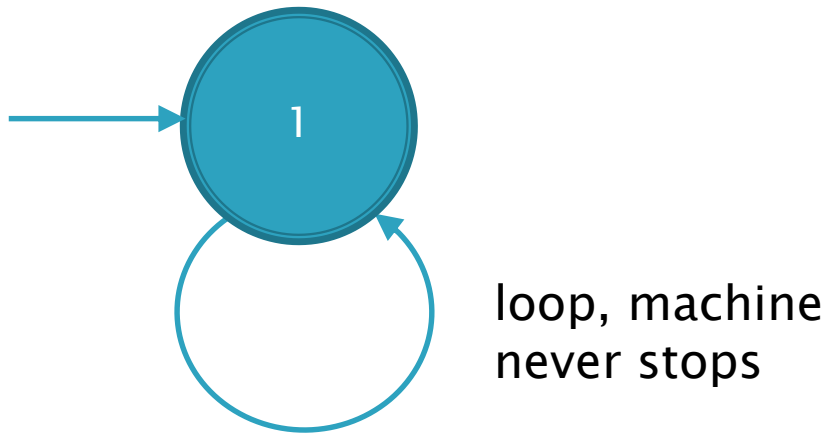
Let's compute $\Sigma(1)$:



0 <symbol written> <head movement>

Warm-Up: 1-State Busy Beaver

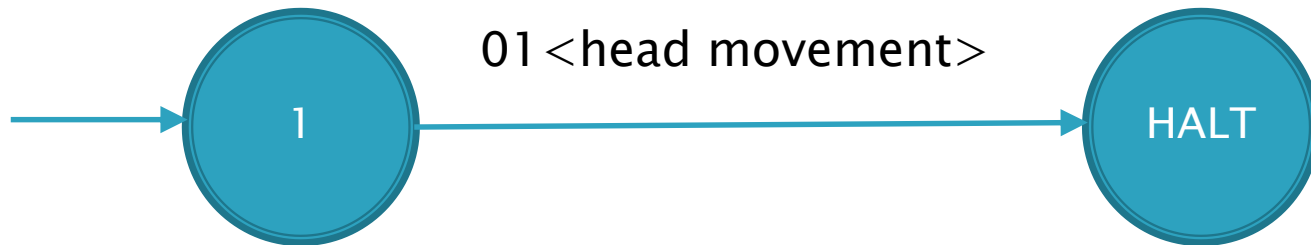
Let's compute $\Sigma(1)$:



0 <symbol written><head movement>

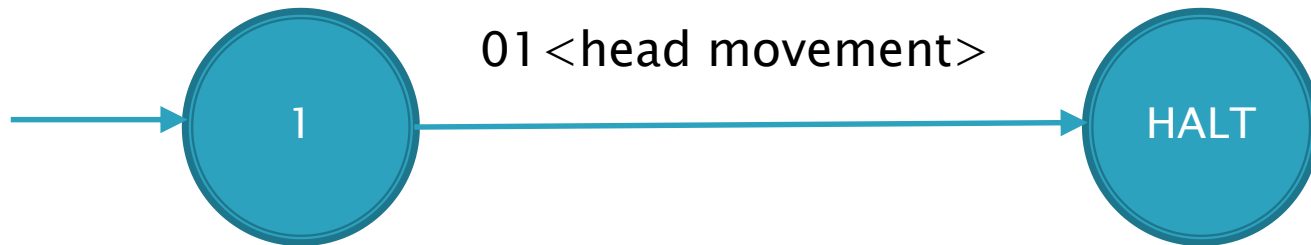
Warm-Up: 1-State Busy Beaver

Let's compute $\Sigma(1)$:



Warm-Up: 1-State Busy Beaver

Let's compute $\Sigma(1)$:



$$\Rightarrow \Sigma(1) = 1$$

Chasing Busy Beavers for Two Symbols

Results	Authors	Year
$\Sigma(2, 2) = 4$	T. Rado	1962
$\Sigma(3, 2) \geq 6$	R. Hegelman	1962
$\Sigma(3, 2) = 6, S(3, 2) = 21$	S. Lin, T. Rado	1965
$\Sigma(4, 2) \geq 13, S(4, 2) \geq 107$	A. H. Brady	1964
$\Sigma(4, 2) = 13, S(4, 2) = 107$	A. H. Brady	1974
		1983
	R. J. Kopp	1981

Chasing the Busy Beaver (5, 2)

Early Results:

Author	Year	Scores
reported by M. W. Green	1964	17 ones
D.S. Lynn	1972	22 ones 435 steps
B. Weimann	1973	40 ones
D.S. Lynn	1974	112 ones 7,707 steps

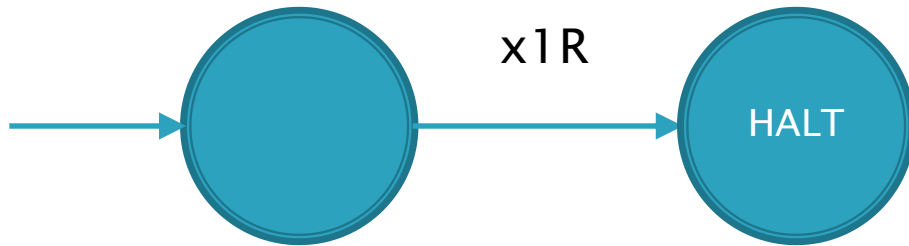
Chasing the Busy Beaver (5, 2)

Results after competition initiated at 6th GI-Conference on TCS, Dortmund:

Author	Year	Scores
U. Schult (winner out of 133 submissions)	1983	501 ones 134,467 steps
G. Uhing	1985	1,915 ones 2,358,063 steps
H. Marxen, J. Buntrock	1989	4,098 ones 47,176,870 steps

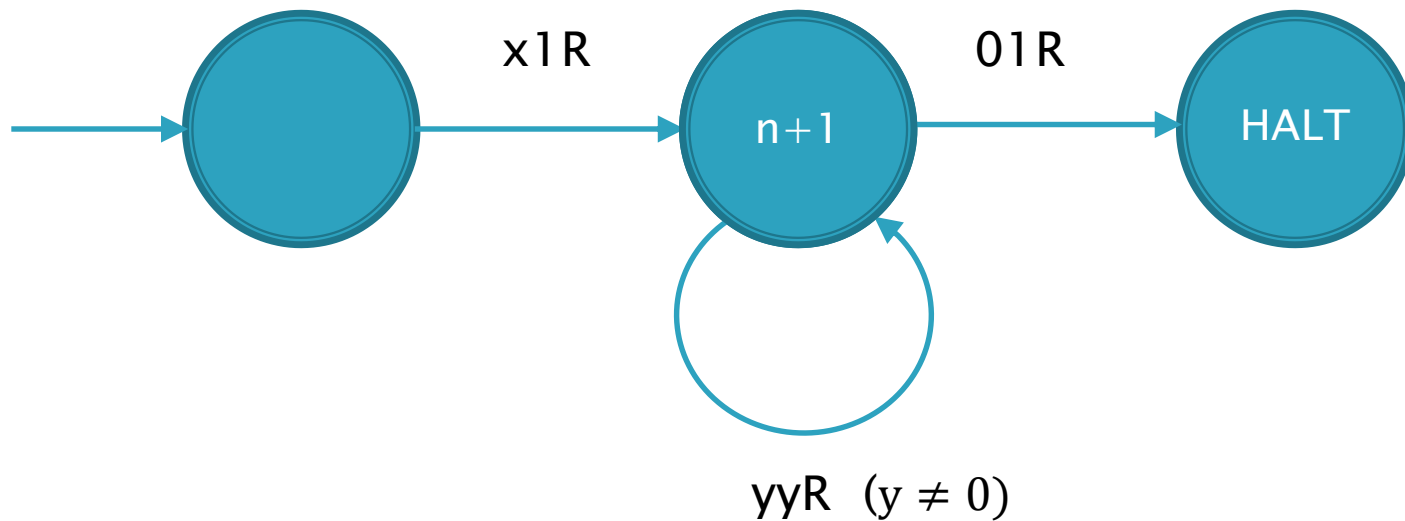
Monotonicity

Adding a state increases scores:



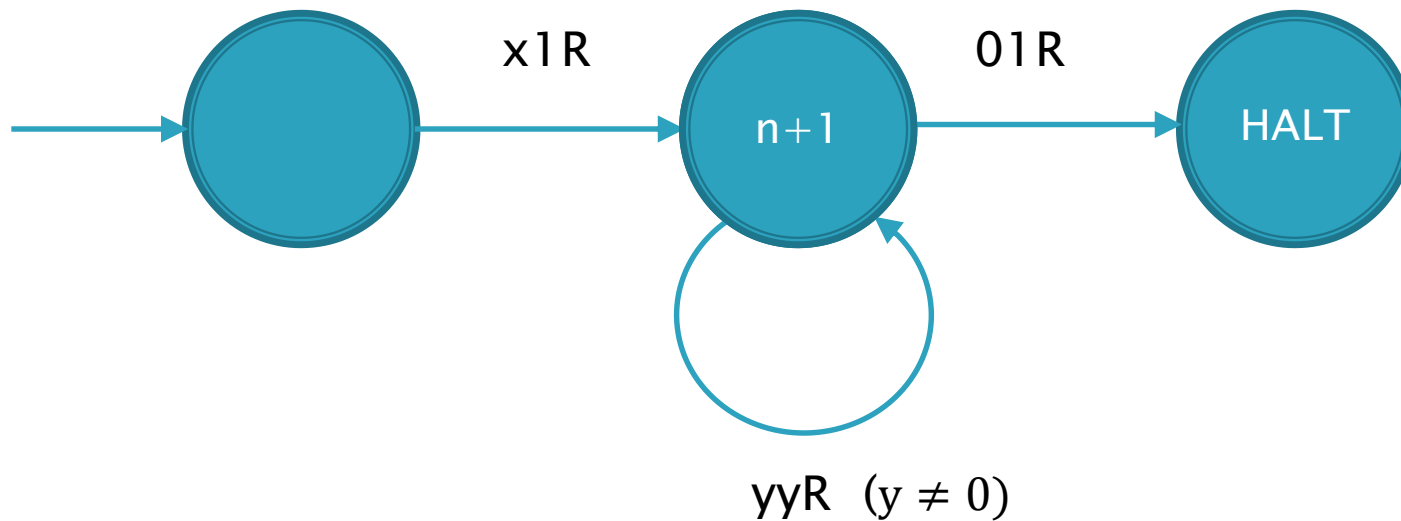
Monotonicity

Adding a state increases scores:



Monotonicity

Adding a state increases scores:



$$\Rightarrow \Sigma(n + 1, m) > \Sigma(n, m),$$
$$S(n + 1, m) > S(n, m)$$

Conjecture (Harland 2016)

Let M be a k -halting* Turing machine with n states and m symbols for some $k \geq 1$ with finite activity.

Then there is a k -halting n -state $(m + 1)$ -symbol Turing machine M' with finite activity such that:

$$\begin{aligned}\text{activity}(M') &> \text{activity}(M) \\ \text{productivity}(M') &> \text{productivity}(M)\end{aligned}$$

* k transitions to the halt state



Evidence Supporting Harland's Conjecture

$$\Sigma(2, 2) = 4 < \Sigma(2, 3) = 9 < 2,050 < \Sigma(2, 4)$$

$$S(2, 2) = 6 < S(2, 3) = 38 < 3,932 < S(2, 4)$$

$$\Sigma(3, 2) = 6 < 374,676,383 \leq \Sigma(3, 3)$$

$$S(3, 2) = 21 < 119,112,334,170,342,540 \leq S(3, 3)$$

Evidence Supporting Harland's Conjecture

$$\Sigma(2, 2) = 4 < \Sigma(2, 3) = 9 < 2,050 < \Sigma(2, 4)$$

$$S(2, 2) = 6 < S(2, 3) = 38 < 3,932 < S(2, 4)$$

$$\Sigma(3, 2) = 6 < 374,676,383 \leq \Sigma(3, 3)$$

$$S(3, 2) = 21 < 119,112,334,170,342,540 \leq S(3, 3)$$

↑
one hundred nineteen quadrillion...

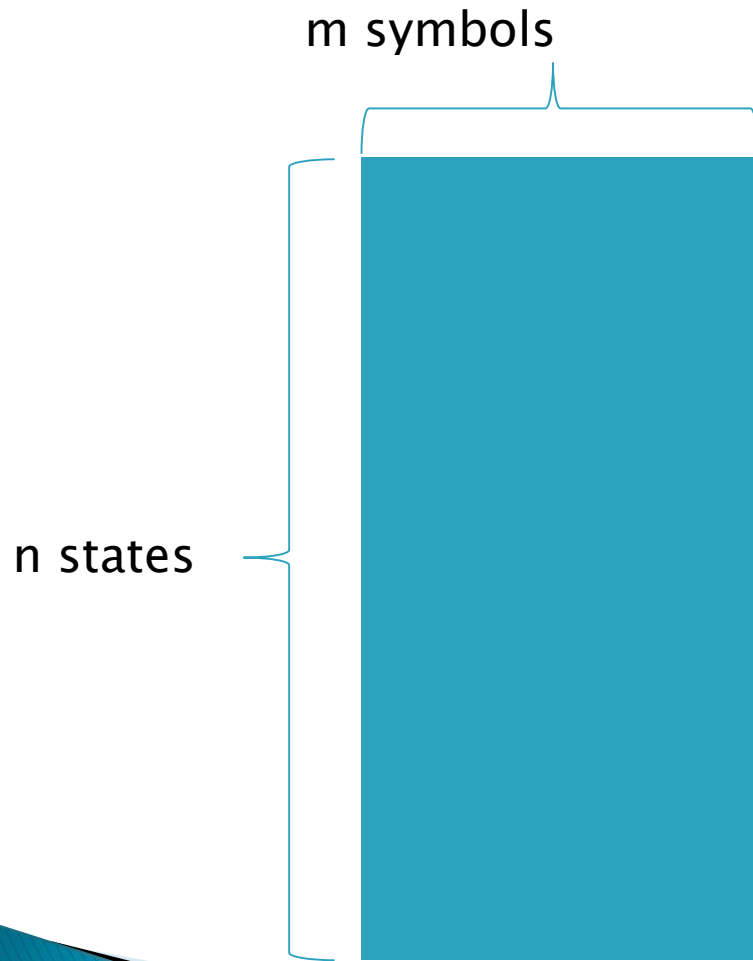
Results of Rado, Lin, Lafitte, Papazian,
T. Ligocki and S. Ligocki

Theorem 1

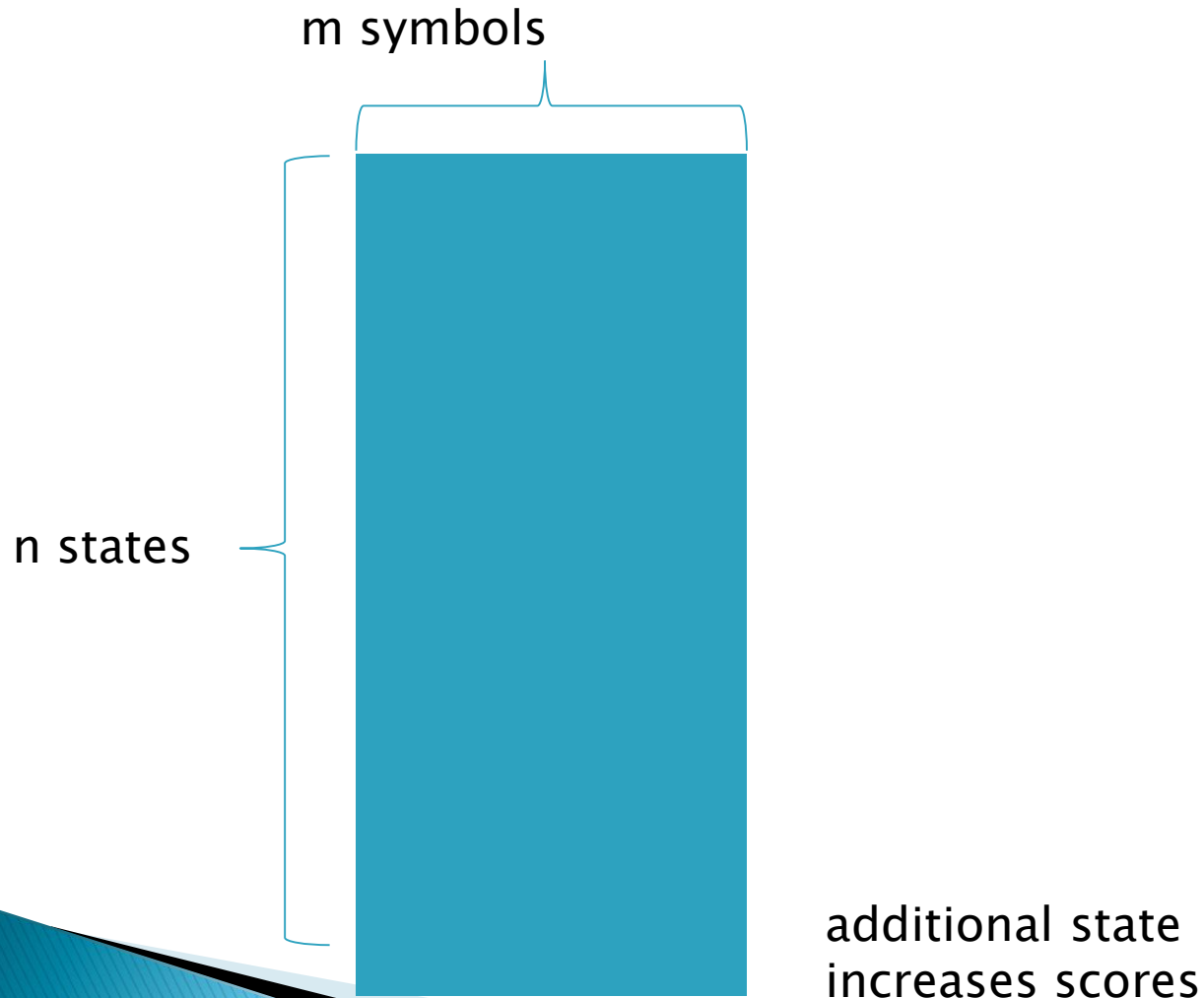
For every $m \geq 2$ and $k \geq 1$ there is an $N_{m, k}$ such that for every k –halting Turing machine M with $n \geq N_{m, k}$ states and m symbols with finite activity there is an n –state, $(m + 1)$ –symbol k –halting Turing machine M' with finite activity such that

$$\begin{aligned} \text{activity}(M') &> \text{activity}(M) \\ \text{productivity}(M') &> \text{productivity}(M). \end{aligned}$$

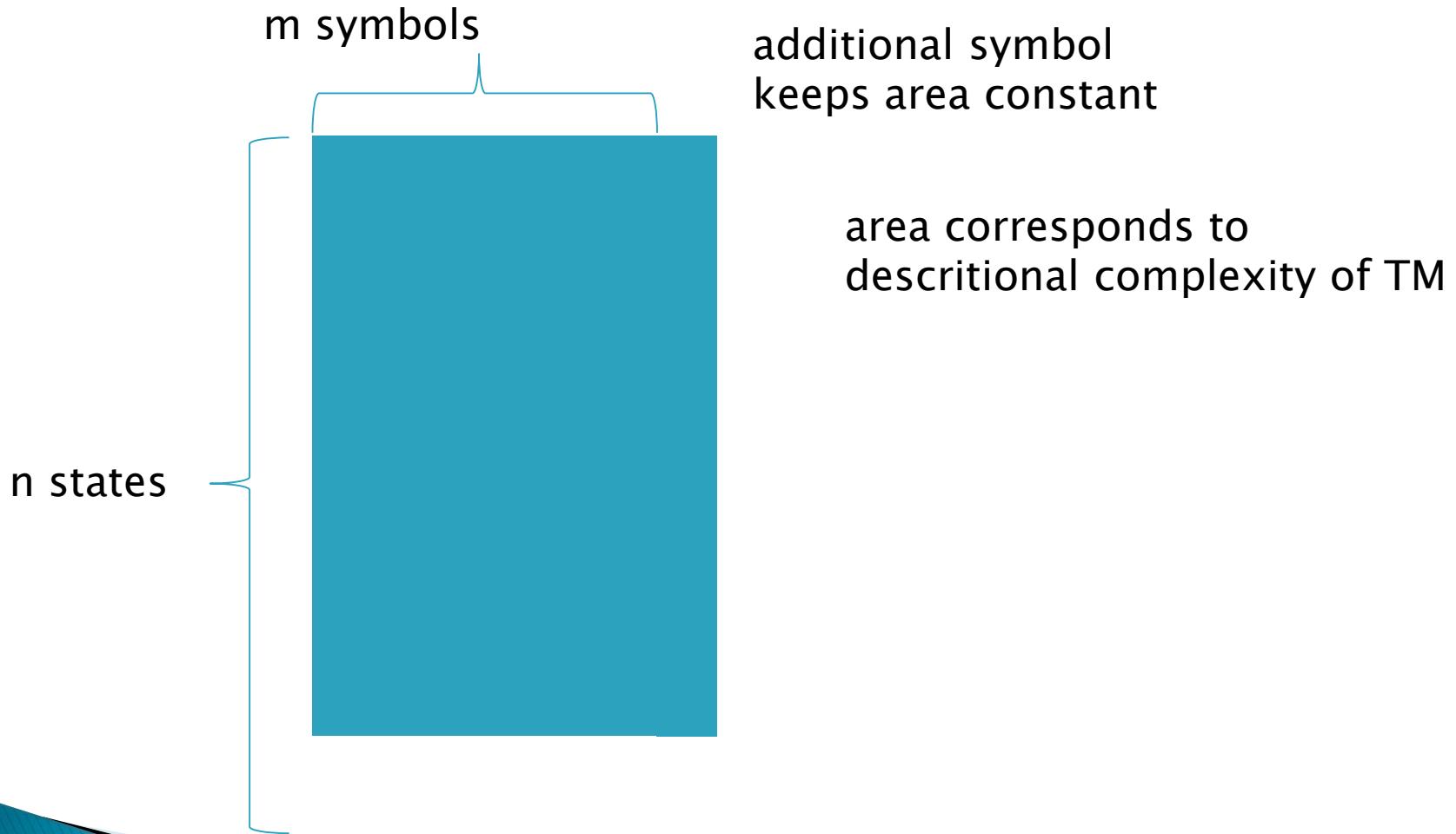
Visual Proof of Theorem 1



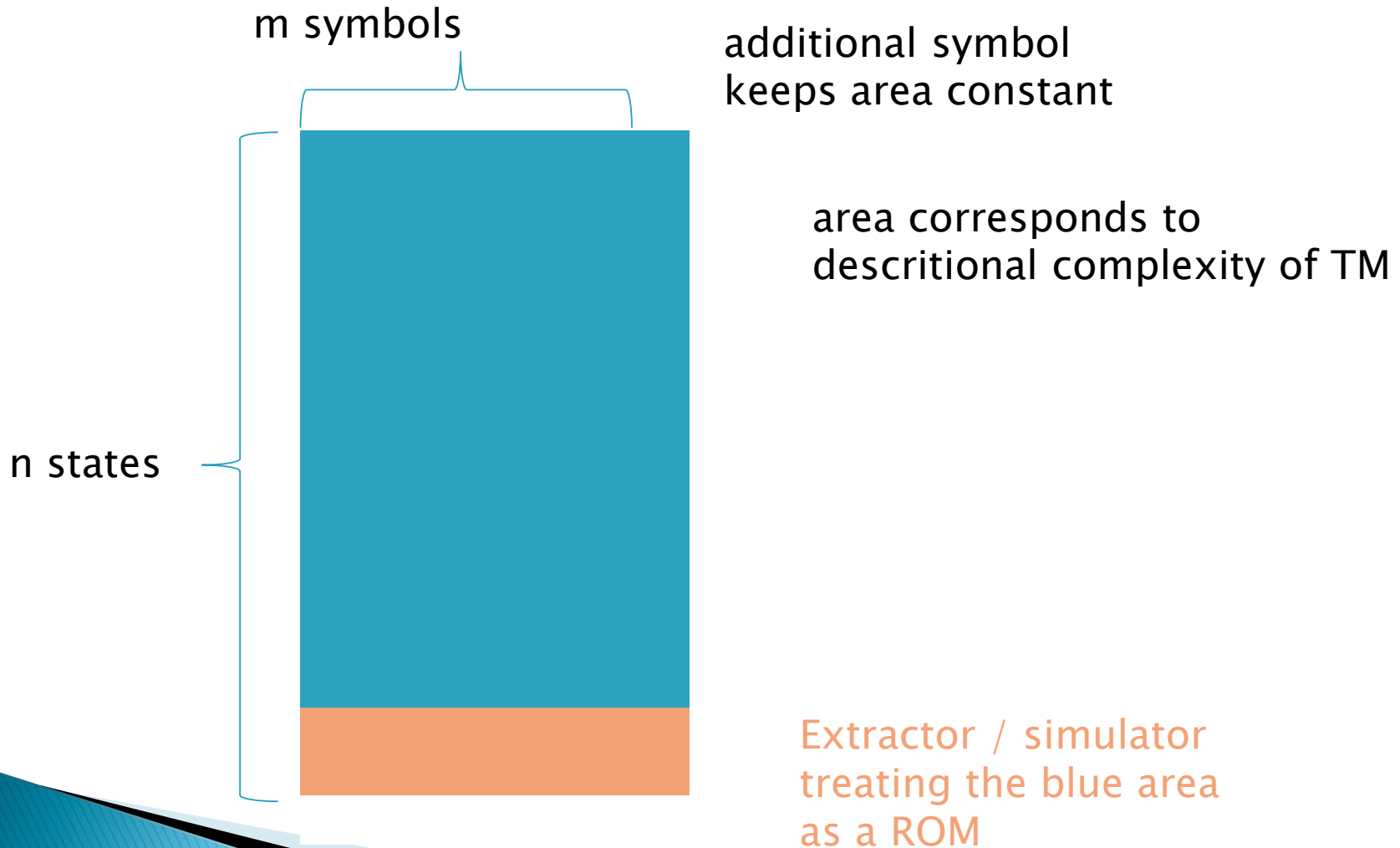
Visual Proof of Theorem 1



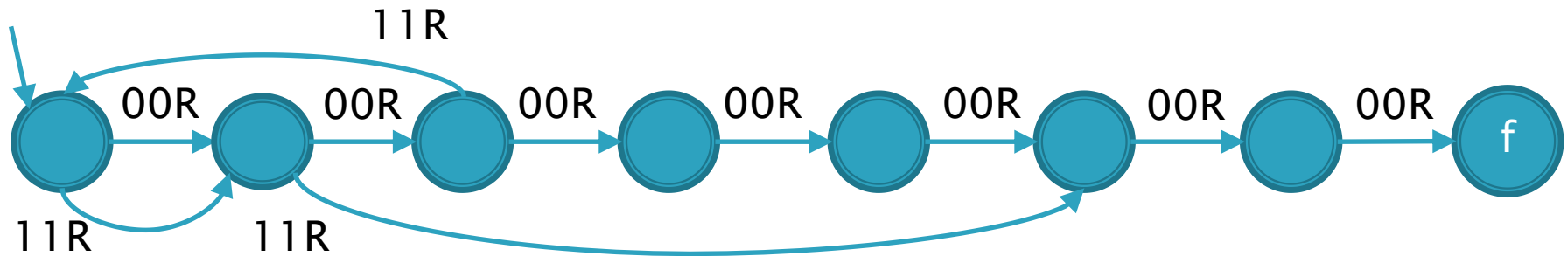
Visual Proof of Theorem 1



Visual Proof of Theorem 1



Introspective Computing (Ben-Amram, Petersen 2002)



State f reached after reading:

- 1000000 → bits 110 encoded
- 0100 → bits 010
- 0010000000 → bits 111

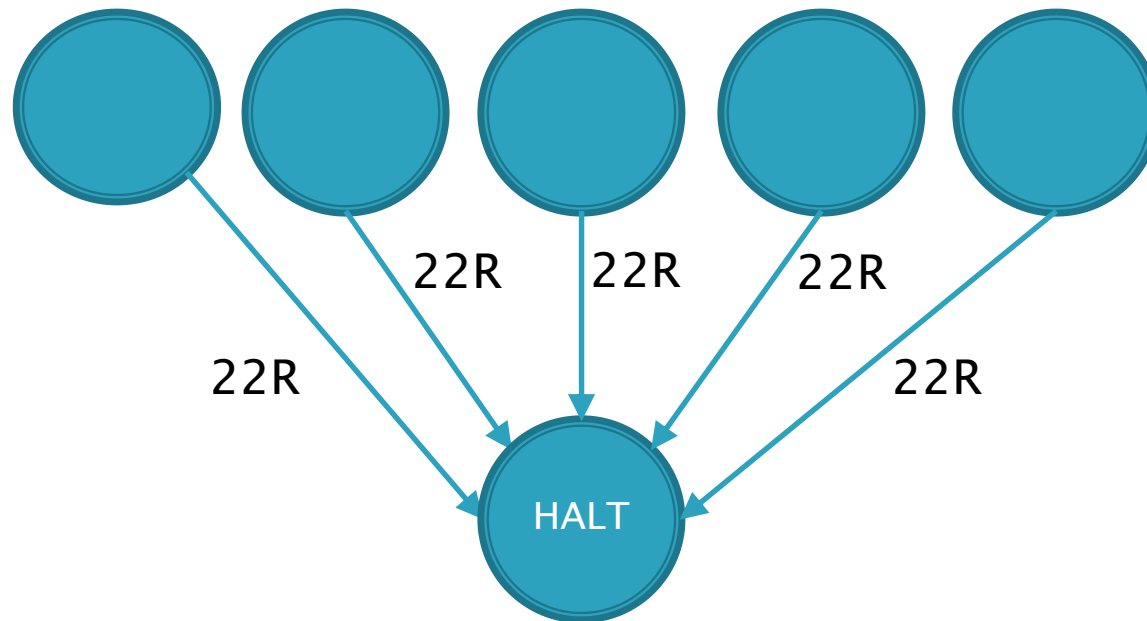
Transition encodes log. number of bits

Theorem 2

For every Turing machine M with $n \geq 2$ states and two symbols having finite activity there is an n -state, 3-symbol Turing machine M with finite activity such that

$$\text{activity}(M') > \text{activity}(M).$$

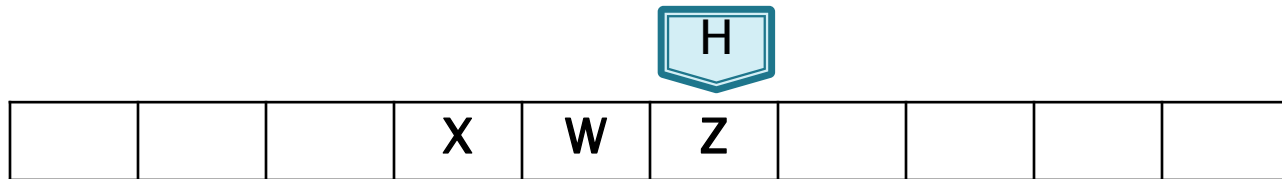
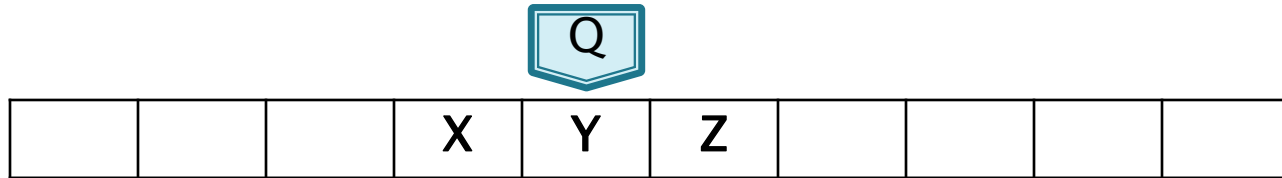
Proof of Theorem 2



Add halting transitions on new symbol „2“

Proof of Theorem 2 (cont.)

Consider halting transition:



Modify transition depending on X, Z

Proof of Theorem 2 (cont.)

$X = 0$, w.l.o.g. initial state 1 moves right on 0:

Q

			0	Y	Z				
--	--	--	---	---	---	--	--	--	--

1

			0	2	Z				
--	--	--	---	---	---	--	--	--	--

S

			0	2	Z				
--	--	--	---	---	---	--	--	--	--

H

			0	2	Z				
--	--	--	---	---	---	--	--	--	--

Proof of Theorem 2 (cont.)

$X = 1$, some state R moves right on 1:



			1	Y	Z				
--	--	--	---	---	---	--	--	--	--



			1	2	Z				
--	--	--	---	---	---	--	--	--	--



			W	2	Z				
--	--	--	---	---	---	--	--	--	--



			W	2	Z				
--	--	--	---	---	---	--	--	--	--

Proof of Theorem 2 (cont.)

$X = 1$, $Z = 1$, all states move left on 1:

Q

			1	Y	1				
--	--	--	---	---	---	--	--	--	--

L

			1	2	1				
--	--	--	---	---	---	--	--	--	--

S

			1	2	W				
--	--	--	---	---	---	--	--	--	--

H

			1	2	W				
--	--	--	---	---	---	--	--	--	--

Proof of Theorem 2 (cont.)

$X = 1$, $Z = 0$, some state L moves left on 0:

Q

			1	Y	0				
--	--	--	---	---	---	--	--	--	--

L

			1	2	0				
--	--	--	---	---	---	--	--	--	--

S

			1	2	W				
--	--	--	---	---	---	--	--	--	--

H

			1	2	W				
--	--	--	---	---	---	--	--	--	--

Proof of Theorem 2 (cont.)

If all states move right on 0, then activity is bounded by n .

For all $n \geq 2$ we have $S(n, 2) > n$

→ Take n -state Busy Beaver for 2 symbols



Theorem 3

For every Turing machine M with $n \geq 2$ states and $m \geq 2$ symbols having finite activity there is a 2-state, $(4nm + 5m)$ -symbol Turing machine M' with finite activity such that

$$\begin{aligned} \text{activity}(M') &> \text{activity}(M) \\ \text{productivity}(M') &> \text{productivity}(M). \end{aligned}$$

Proof of Theorem 3

By monotonicity in number of states, M with $n + 1$ states and m symbols increases activity and productivity.

Classical construction of Shannon transforms it into equivalent 2-state machine with

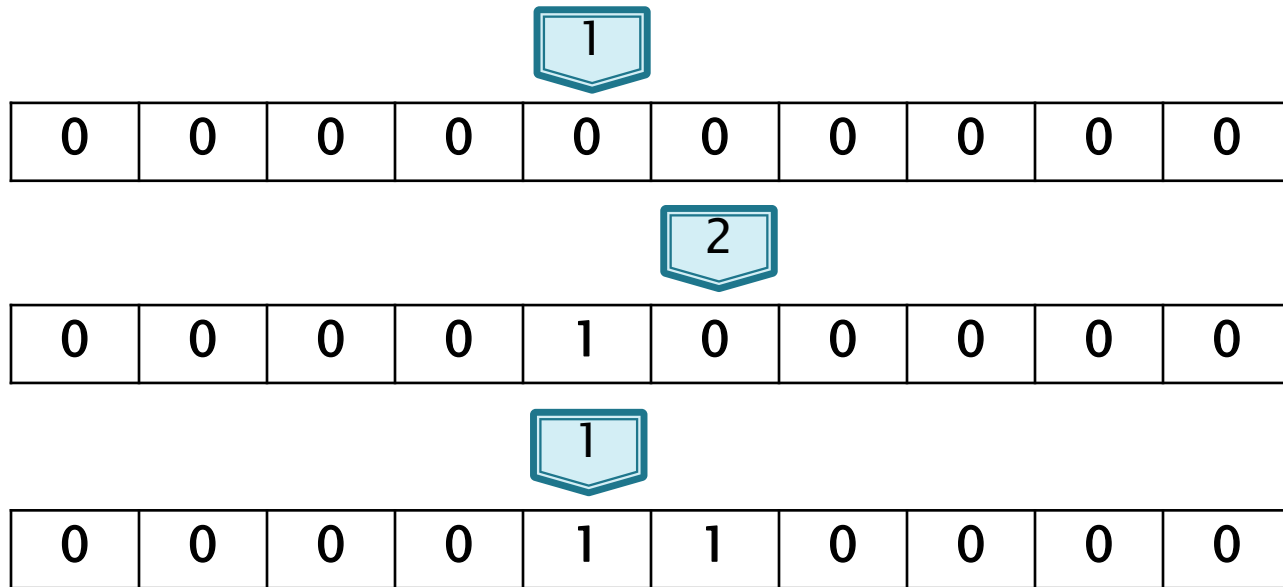
$$4m(n + 1) + m = 4nm + 5m$$
symbols.

Theorem 4

For every Turing machine M with $n \geq 2$ states and $m \geq 3$ symbols having finite activity there is an n –state, $3m$ –symbol Turing machine M' with finite activity such that

$$\text{activity}(M') > \text{activity}(M).$$

Proof of Theorem 4 by Example



Idea: Let head bounce on additional symbols with subscripts L, R

Proof of Theorem 4 (cont.)

1

0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

2

0	0	0	0	1_L	0	0	0	0	0
---	---	---	---	-------	---	---	---	---	---

1

0	0	0	0	1_L	1_R	0	0	0	0
---	---	---	---	-------	-------	---	---	---	---

1

0	0	0	0	1	1_R	0	0	0	0
---	---	---	---	---	-------	---	---	---	---

1

0	0	0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Thank you for your attention!



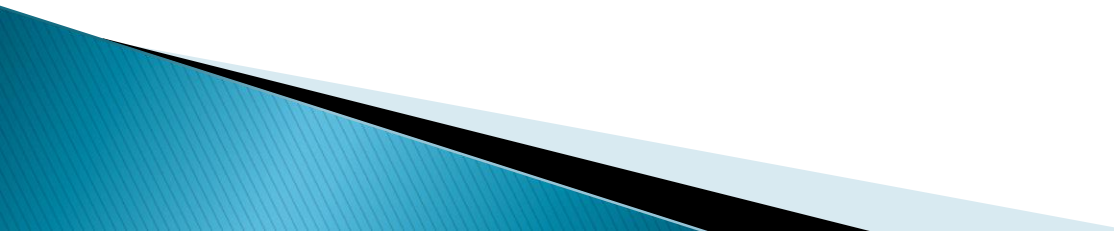
Attribution: By Steve from washington, dc, usa (American Beaver) [CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0>)], via Wikimedia Commons

Digression: In Praise of Beaverness

COMPUTING THE BUSY BEAVER FUNCTION (Chaitin 1987):

„...to information theorists it is clear that the correct measure is bits, not states. ..

...to deal with Σ (number of bits) one would need a model of a binary computer as simple and compelling as the Turing machine model, and no obvious natural choice is at hand.”



In Praise of Beaverness

- ▶ Reasonable programming languages encode n -state Turing machines (fixed m) in $O(n \log n)$ bits (not sure about Befunge and Ook!)
- ▶ Turing machines encode $n \log n$ bits of programme code in $O(n)$ states by introspective computing