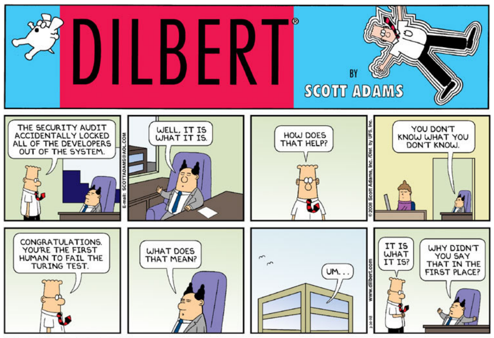
Class 19: Undecidability in Theory and Practice



cs302: Theory of Computation
University of Virginia Computer Science

David Evans
http://www.cs.virginia.edu/evans

Menu

- PS5, Problem 5
- Erudite Discussion on Undecidability
- Busy Beaver Problem

PS5, Problem 5

Consider a one-tape Turing Machine that is identical to a regular Turing machine except the input may not be overwritten. That is, the symbol in any square that is non-blank in the initial configuration must never change. Otherwise, the machine may read and write to the rest of the tape with no constraints (beyond those that apply to a regular Turing Machine).

a. What is the set of languages that can be recognized by an unmodifiable-input TM?
b. Is $HALT_{UTM}$ decidable?

Impossibility of Copying

0	1	0	0	1	1	1	0	0	0										

input (unmodifiable)

How can the TM keep track of which input square to copy next?

Option 1: Use the writable part of the tape.
Problem: can't read it without losing head position
Option 2: Use the FSM states.
Problem: there is a finite number of them!

Hence, it is equivalent to a DFA $\Rightarrow$ regular languages

Nevertheless, $HALT_{\mathrm{UTM}}$ is Undecidable

Prove by reducing $HALT_{\mathrm{TM}}$ to $HALT_{\mathrm{UTM}}$:

$HALT_{\mathrm{TM}}(<M, w>) = HALT_{\mathrm{UTM}}(<MUw, \varepsilon>)$ where
MUw = an unmodifiable-TM that ignores the input, writes a **#** on the tape, followed by w, then, simulates M on the tape starting at the first square after the **#**, treating the **#** as if it is the left edge of the tape.

Computability in Theory and Practice

(Intellectual Computability Discussion on TV Video)

Ali G Problem

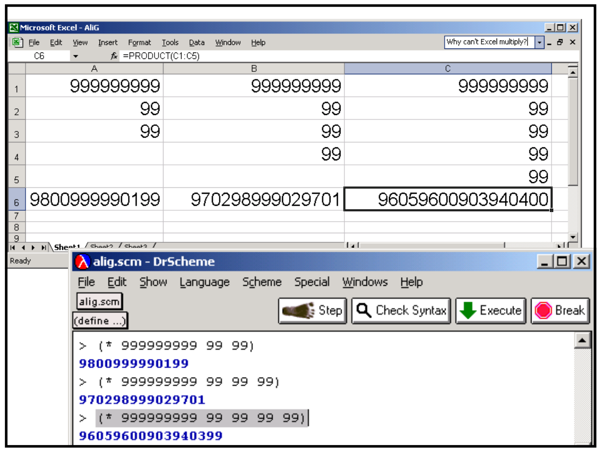
- **Input:** a list of numbers (mostly 9s)
- **Output:** the product of the numbers

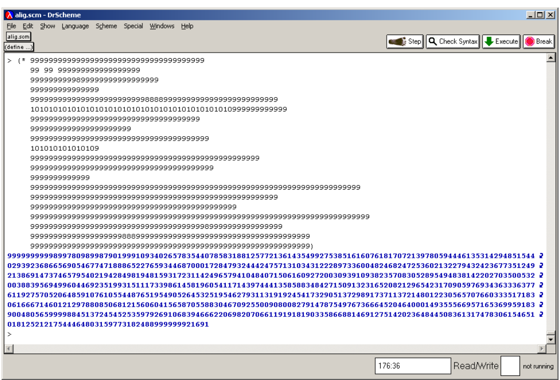
L_{ALIG} = { < k_0, k_1, …, k_n, p> | each k_i represents a number and p represents a number that is the product of all the k_is. numbers

Is L_{ALIG} decidable?

Yes. It is easy to see a simple algorithm (e.g., elementary school multiplication) that solves it.

Can *real* computers solve it?





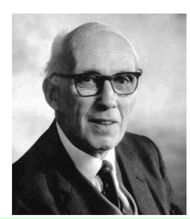
Ali G was Right!

- Theory assumes ideal computers:
  - Unlimited, perfect memory
  - Unlimited (finite) time
- Real computers have:
  - Limited memory, time, power outages, flaky programming languages, etc.
  - There are many decidable problems we cannot solve with real computer: the actual inputs ***do*** matter (in practice, but not in theory!)

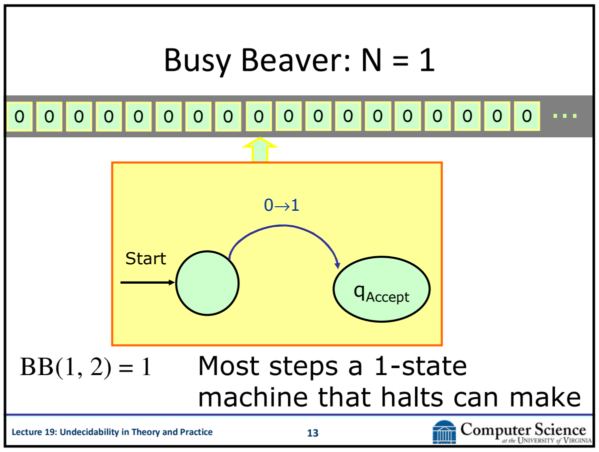
Busy Beavers

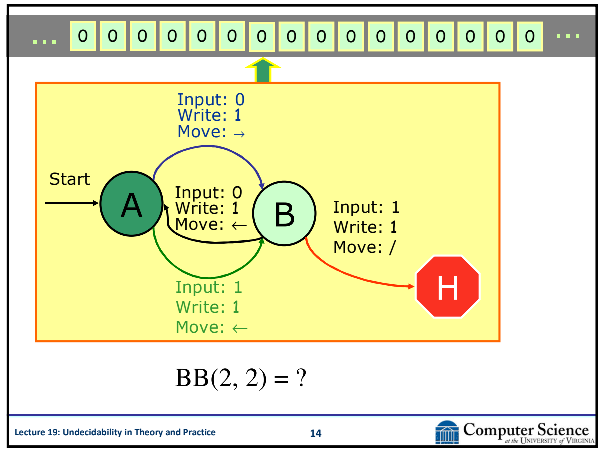
The "Busy Beaver" Game

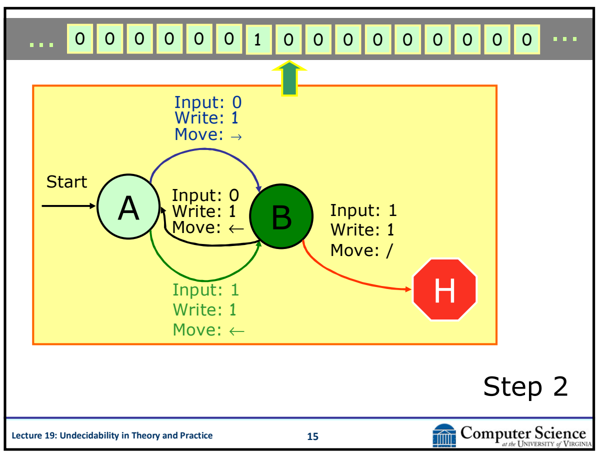
- Design a Turing Machine that:
  - Uses k symbols (e.g., "0" and "1")
  - Starts with a tape of all "0"s
  - Eventually halts (can't run forever)
  - Has n states (not counting q_{Accept} and q_{Reject})
- Goal is to run for as many steps as possible (before halting)
- 2-way infinite tape TM

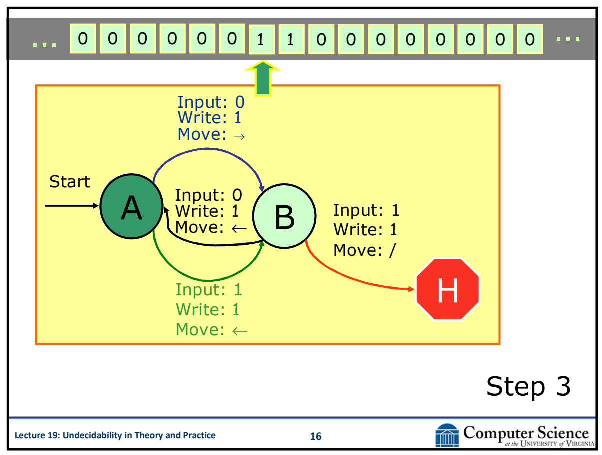


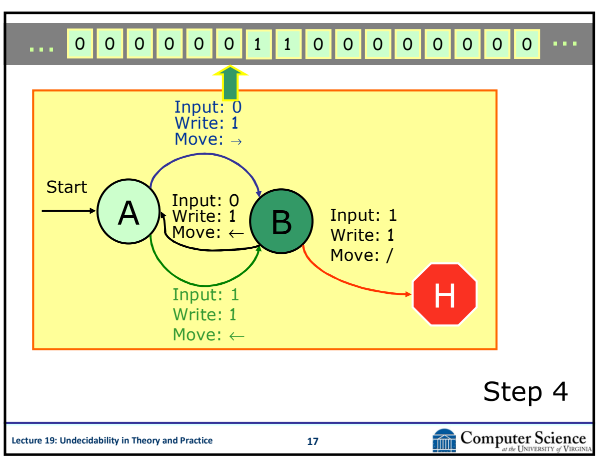
Tibor Radó, 1962

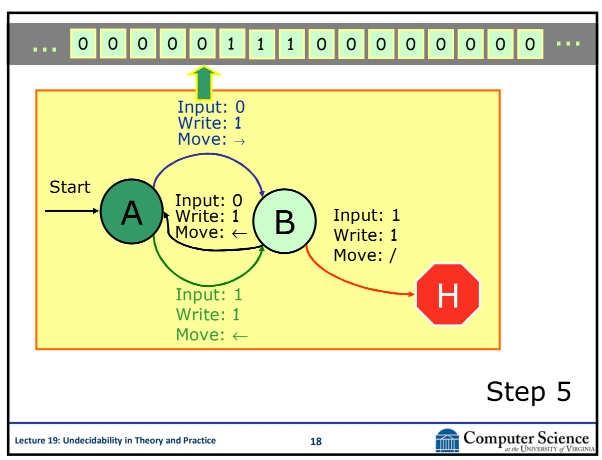

Busy Beaver: N = 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 …
0→1
Start
qAccept
BB(1, 2) = 1
Most steps a 1-state machine that halts can make
Lecture 19: Undecidability in Theory and Practice
13
Computer Science at the University of Virginia

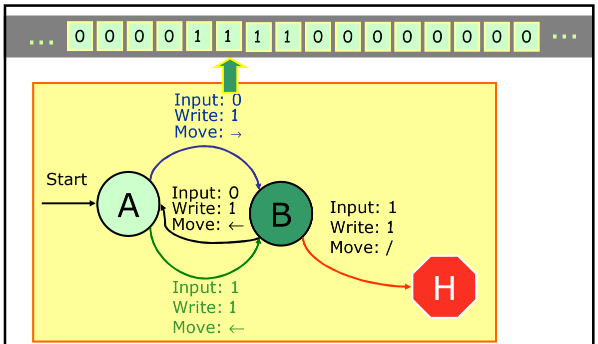

… 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 …
Input: 0
Write: 1
Move: →
Start
A
Input: 0
Write: 1
Move: ←
B
Input: 1
Write: 1
Move: /
H
Input: 1
Write: 1
Move: ←
BB(2, 2) = ?
Lecture 19: Undecidability in Theory and Practice
14
Computer Science at the University of Virginia


… 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 …
Input: 0
Write: 1
Move: →
Start
A
Input: 0
Write: 1
Move: ←
B
Input: 1
Write: 1
Move: /
H
Input: 1
Write: 1
Move: ←
Step 2
Lecture 19: Undecidability in Theory and Practice
15
Computer Science at the University of Virginia

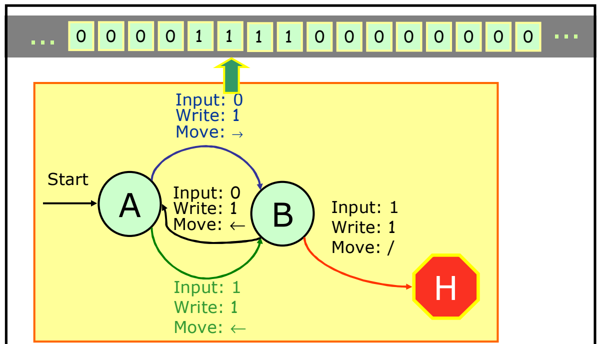

… 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 …
Input: 0
Write: 1
Move: →
Start
A
Input: 0
Write: 1
Move: ←
B
Input: 1
Write: 1
Move: /
H
Input: 1
Write: 1
Move: ←
Step 3
Lecture 19: Undecidability in Theory and Practice
16
Computer Science at the University of Virginia


… 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 …
Input: 0
Write: 1
Move: →
Start
A
Input: 0
Write: 1
Move: ←
B
Input: 1
Write: 1
Move: /
H
Input: 1
Write: 1
Move: ←
Step 4
Lecture 19: Undecidability in Theory and Practice
17
Computer Science at the University of Virginia


… 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 …
Input: 0
Write: 1
Move: →
Start
A
Input: 0
Write: 1
Move: ←
B
Input: 1
Write: 1
Move: /
H
Input: 1
Write: 1
Move: ←
Step 5
Lecture 19: Undecidability in Theory and Practice
18
Computer Science at the University of Virginia



Step 6

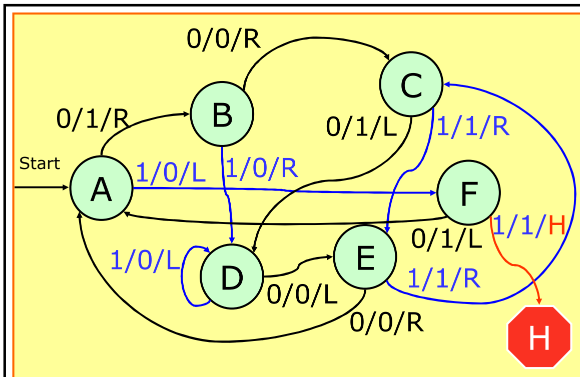


$BB(2, 2) \geq 6$

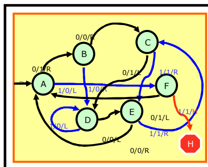
Halted

(Actually, known that $BB(2,2) = 6$)

What is $BB(6, 2)$?



6-state machine found by Buntrock and Marxen, 2001



30023277165235628289551030183413401851477543372467525003
73381801735214240760383265881912082978202876698984017860
71345848280422383492822716051848585583668153797251438618
56173020941548768557007853865875730485748722204003076984
40450988713670876150791383110343531646410779192098908371
64477363289374225531955126023251172259034570155087303683
65463087415599082251612993842583069137860727367070819016
05255340770400392265930739979231701547753586298504217125
13378527086223112680677973751790032937578520017666792246
83990885592036293376774476087012844688345547780631649160
18557844268607690279445427980061526931674528213366899174
60886106486574189015401194034857577718253065541632656334
31424232559248670011850671658130342327174896542616040979
71730737166888272814359046394456059281752540483211093060
02474658968108793381912381812336227992839930833085933478
85317657470277606285828915656839229596358626365413938385
67647280513949655544096884565781227432963199608083680945
36421039149584946758006509160985701328997026301708760235
50023959811941059214262166961455282724442921741646549436
38916971139653168926606117092900485806775661787157523545
94049016719278069832866522332923541370293059667996001319
37669855168384885147462515209456711061545198683989449088
56870822449787745514532043585886615939797639351028965232
95803940023673203101744986550732496850436999753711343067
32867615814626929272337566201561282692410545484965841096
15740312114406110889753498991567148886819523660180862466
87712098553077054825367434062671756760070388922117434932
63344477313878371402373589871279027828837719826038006510
50757929252394534506229992082975795848934488862781276290
44163292251815410053522460845527615133839346231290832660
9493773809504666431216897465119968476812750763132606

(1730 digits)

Best found before 2001, only 925 digits!

http://drb9.drb.insel.de/~heiner/BB/index.html

Busy Beaver Numbers

- BB(1) = 1
- BB(2) = 6
- BB(3) = 21
- BB(4) = 107
- BB(5) = Unknown!
  - The best found so far is 47,176,870
- $BB(6) > 10^{2879}$

Is there a language problem?

L_{BB}= { <n, k, s> | where n and k represent integers and s is the maximum number of steps a TM with n non-final states and k tape symbols can run before halting }

Is L_{BB} Decidable?

L_{BB} is Undecidable

Proof by reduction:
Assume M_{BB} exists that decides L_{BB}
$HALT_{TM}$(<M, w>) =
  n = number of states in M
  k = number of symbols in M's tape alphabet
  find s by trying s = 1, 2, ... until M_{BB} accepts <n, k, s>
  simulate M on w for up to s steps
  if it halts, **accept**
  if it doesn't complete, **reject**

Challenges

- The standard Busy Beaver problem is defined for a doubly-infinite tape TM. For the one-way infinite tape TM, what is $BB(4, 2)$?
- Find a new record BB number
  - http://drb9.drb.insel.de/~heiner/BB/index.html

Exam 2: one week from today.