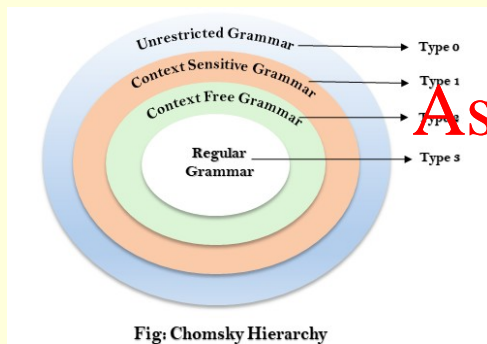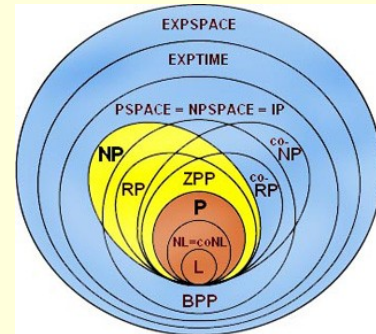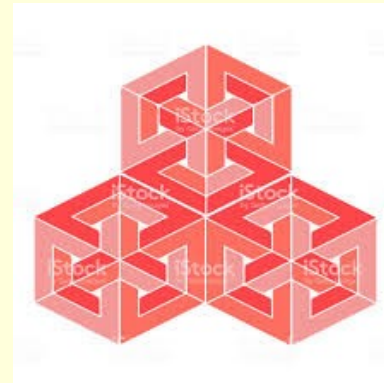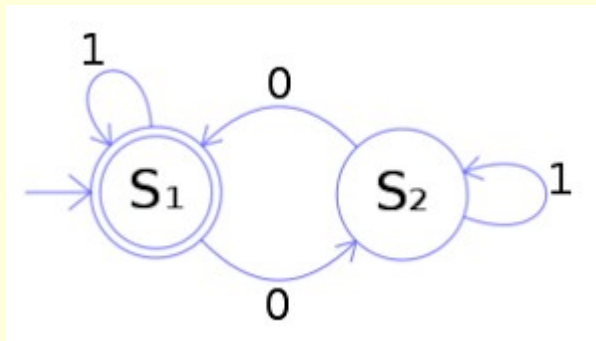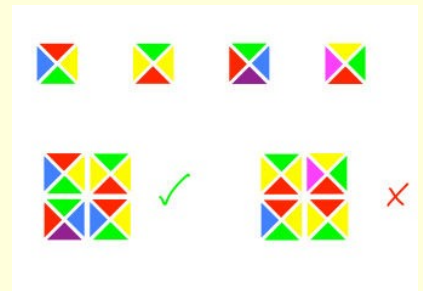# COSC1107 Computing Theory

(We will commence soon. We are just allowing a few minutes for people to join and set up. *Please mute your microphone unless you are speaking.* You can raise your hand or use the chat at any time.)

COSC1107

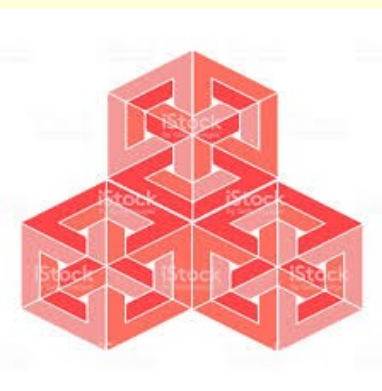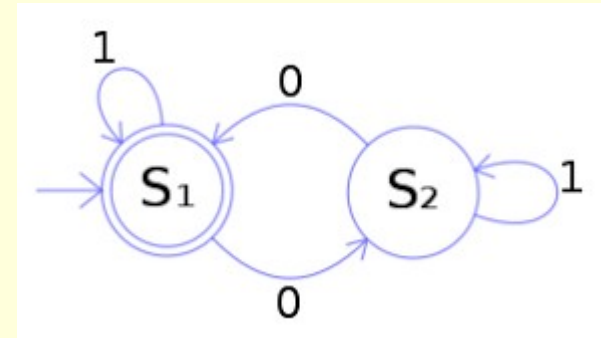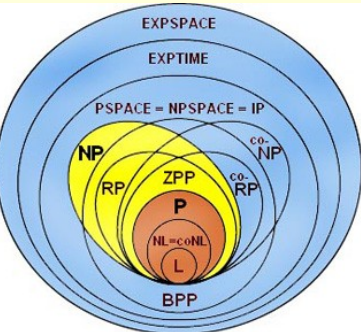Assignment Project Exam Help

Computing Theory

https://powcoder.com

Research and Requests

Add WeChat powcoder
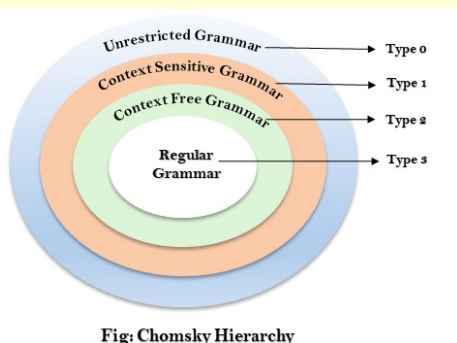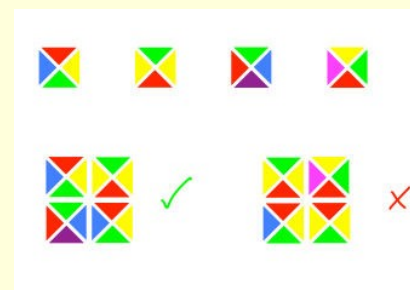
Week 12

James Harland

james.harland@rmit.edu.au

* With thanks to Sebastian Sardina

Intro music 'Far Over' playing now …

Acknowledgement

RMIT University acknowledges the people of the Woi wurrung and Boon wurrung language groups of the eastern Kulin Nations on whose unceded lands we conduct the business of the University. RMIT University respectfully acknowledges their Ancestors and Elders, past and present

RMIT also acknowledges the Traditional Custodians and their Ancestors of the lands and waters across Australia where we conduct our business.

(add your name here to volunteer for this or email me)
(my personal Acknowledgement of Country is here)

Acknowledgement

Acknowledgement of Traditional Owners
Original artwork by Amanda Power

As we gather virtually, from all areas around South Central Victoria, we acknowledge the people of the Eastern and Western Kulin Nations.

In this time when our Wilam (camp or meeting place) is dispersed across many separate homes, we hope that you all feel a strong sense of Noogal (belonging) in meeting with your colleagues, friends, and family using alternative and accessible communication methods.

We acknowledge that the lands we are conducting our business today remains unceded. We respectfully acknowledge the first nations people of the five Kulin Nations, their Ancestors and Elders, past, present and emerging. (thanks to Nick Balkin for this acknowledgement)

Computing Theory

Weekly Schedule

	Lectorial	Tutorial	Assessment
12	Research and requests	Sample exercise	Assignment 2
14-16	--	--	Final exercise

- ~~Quiz 10 deadline is 11.59pm Monday 11ᵗʰ October~~

- Assignment 2 deadline now **11.59pm Tuesday 19ᵗʰ October**

- Questions 4b, 4e of Assignment 2 will be submitted **via a special quiz on Canvas** (not as part of the PDF report)

- Other questions & csv files to be submitted as files

- Do not use zip files!

Week 12   Computing Theory

Weekly Schedule

	Lectorial	Tutorial	Assessment
12	Research and requests	Sample exercise	Assignment 2
14-16	--	--	Final exercise

Final exercise

- Released at 9.00am on **Thursday 4th November**
- Due by 9.00am on **Friday 5th November**
- Time is Melbourne time (UTC +11; see here)
- Expected time spent on the task is 4-6 hours
- Sample exercise will covered in tutorials in Week 12
- An additional practice exercise will be released soon

Computing Theory

Questions?

Questions?

Questions?

KEEP CALM AND ASK ME QUESTIONS

KeepCalmAndPosters.com

Week 12

Computing Theory
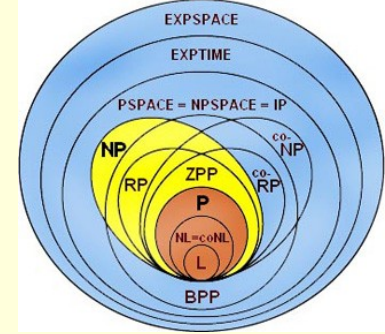
Requests

Requests for Week 12 class received by Friday 8th October:

Zilch!

Zip!

Zero!

Nada!

Nothing!

Request? What request? I thought you were making the request?

CES Survey

We want to hear your feedback!

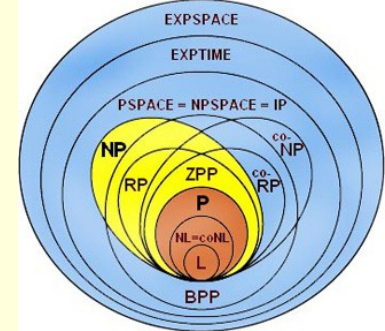This is your opportunity to tell us about your experiences related to:

- Assessments
- Methods of teaching
- Technology use
- Learning materials

Head to **rmit.edu.au/surveys** to see what student feedback have helped change in the past

Deadline is **Sunday 24th October**

Week 12

Computing Theory

SCAN ME

CES Survey
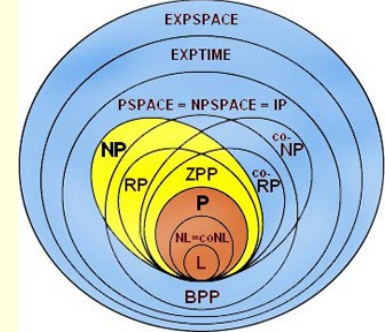
- **Your feedback is important!**

- CES scores are used for evaluation of courses and staff

Assignment Project Exam Help

- Developments in this course due to student feedback

https://powcoder.com

- Weekly Quizzes
- Less assessment Tasks Add WeChat powcoder
- Design of exercise

- All of these were new for 2021

PLEASE FILL IN YOUR SURVEY!

Week 12   Computing Theory

'Far Over'

Lilypond (from http://lilypond.org/)
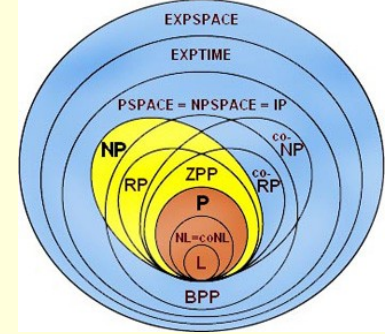
- Free "music engraving tool"
- "Programmer's" way to write sheet music
  ("more similar to a programming language than a graphical score editing program")
- Arrangement of Tune
- Generated score
- Generated MIDI track

Far Over the Misty Mountains Cold

Plan 9 and Howard Shore

```
\header{title = "Far Over …"}

global = {
  \key d \minor
  \time 4/4
  \dynamicUp
}

Tenornotes = \transpose d d \relative c' {
  d,1~d4 r4 a4 c4 d2. f4 g4 a8 (g8) f4 e4 d1~d4
  a4 d4 e4 e1~e4 f4 g4 f8 (e8) d1~d4
  f4 g4. e8 a1~a4 f4 g4. d8 e1~e4
  a,4 c4 e4 f1~f4 g8 (f8) e4 c4 d1
…
```

Week 12   Computing Theory

'Far Over'

Audacity

- Free multi-track audio editor and recorder
- Play 'click track' (MIDI version from Lilypond) in headphones
- Sing each individual part in time with the click track.
- Mix tracks together
- Repeat previous two steps until satisfied
- Export project as WAV (lossless, for later editing if need be)
- Export project as MP3 (because that makes it easy to play)

"Entire choir"

Questions?

Questions?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Questions?

KEEP CALM AND ASK ME QUESTIONS

KeepCalmAndPosters.com

Week 12

Computing Theory

Think music! (This will take 1 minute!)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com
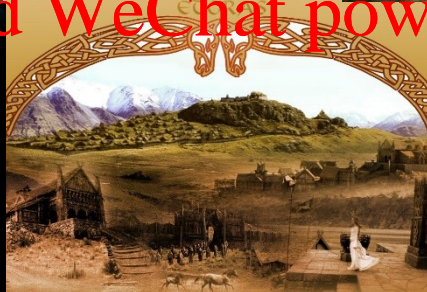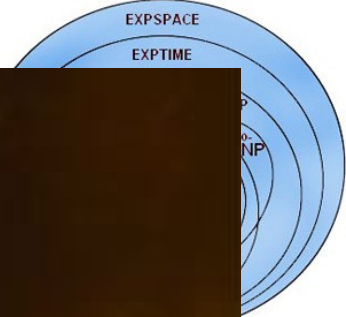
Add WeChat powcoder

GET READY!

Computable Functions
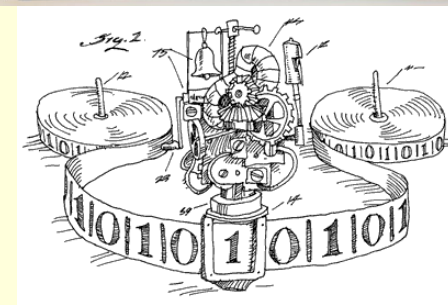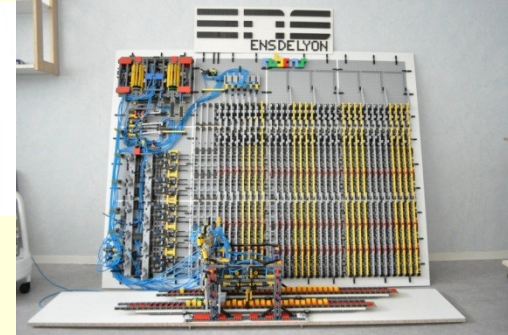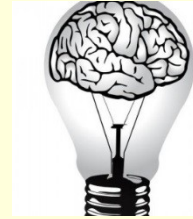
Some functions are not computable!

Turing Machines of a particular type:

- Deterministic Assignment Project Exam Help

- Symbols are only 0 (blank) and 1
  https://powcoder.com

- Only consider blank input
  Add WeChat powcoder

- n states plus a halt state means size is n
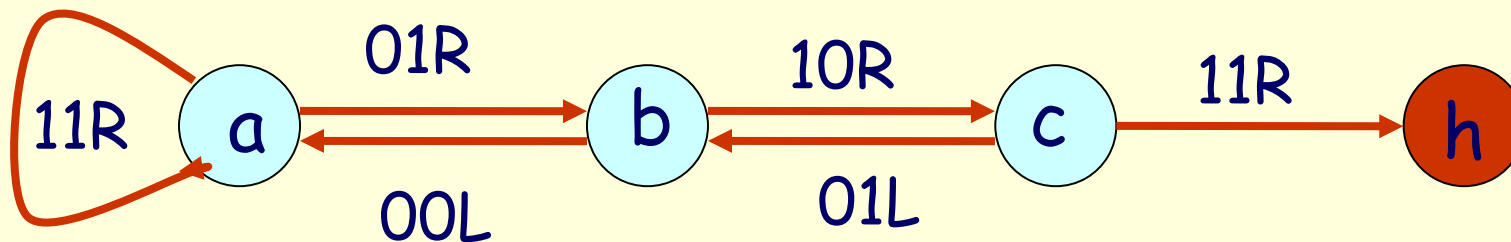
Busy Beaver
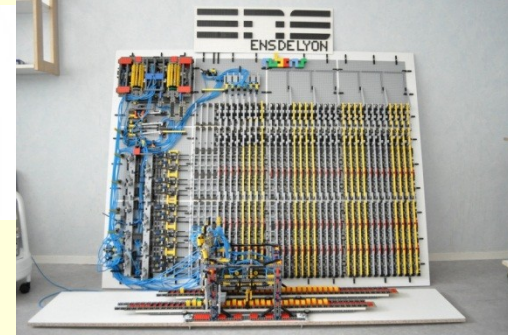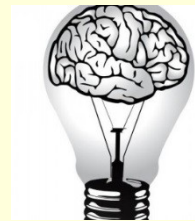
What is the largest number of 1's that can be printed by a terminating n-state machine?
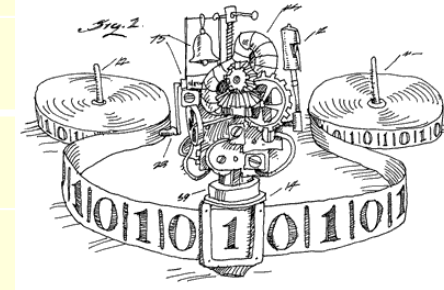
n	#1's (productivity)	#steps
1	1	1
2	4	6
3	6	21
4	13	107
5	≥ 4098	≥ 47,176,870 (??)
6	≥ 3.51×10^{18,276}	≥ 7.41×10^{36,534} (!!)

Busy Beaver
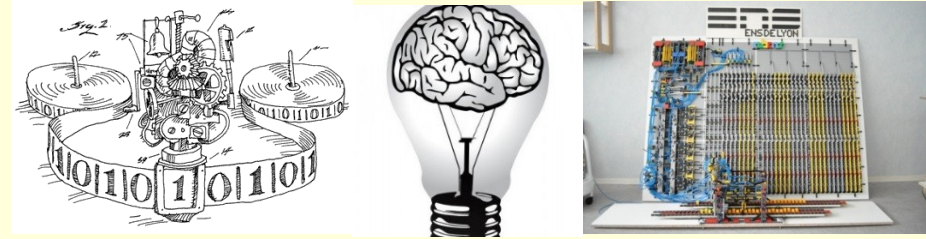
- Busy Beaver function is **non-computable; it grows faster than any computable function (!!)**

- Various mathematical bounds known

- All surpassed in practice

- Seems hopeless for $n \geq 7$

- Values for $n \leq 5$ seem settled (but as yet unproven)

Computing Theory

Busy Beavers
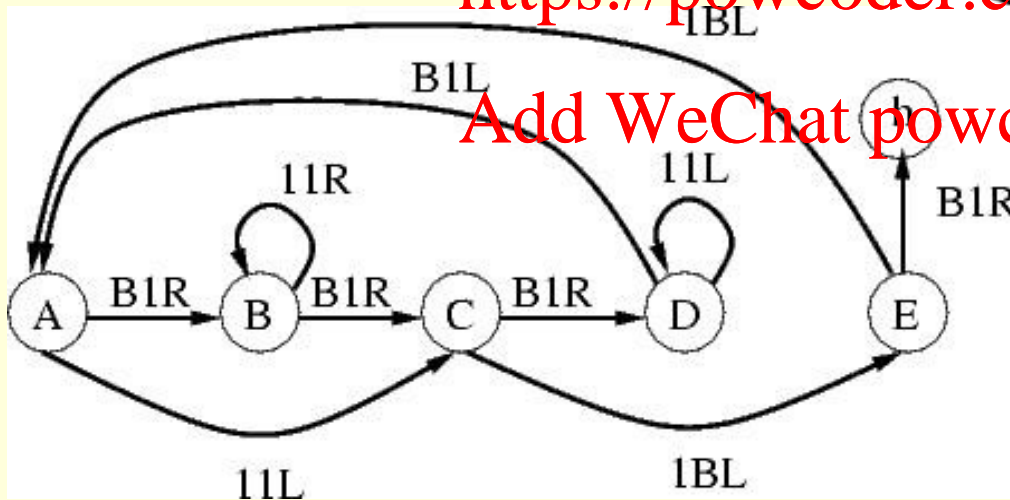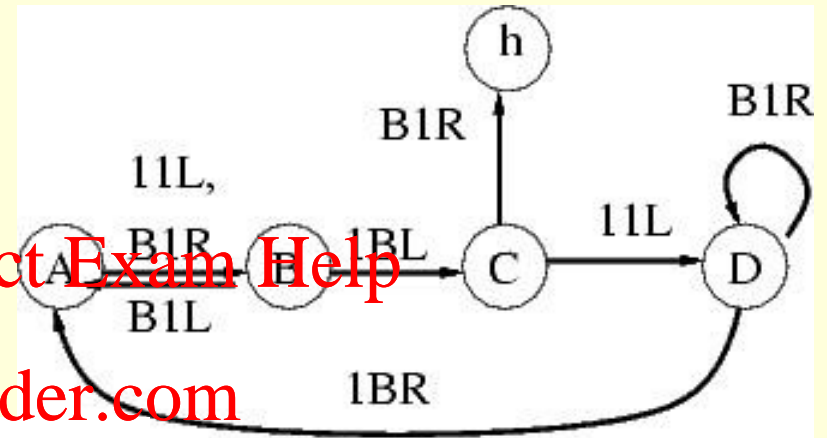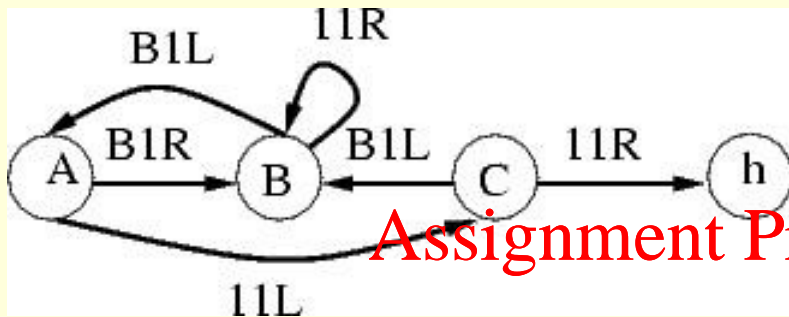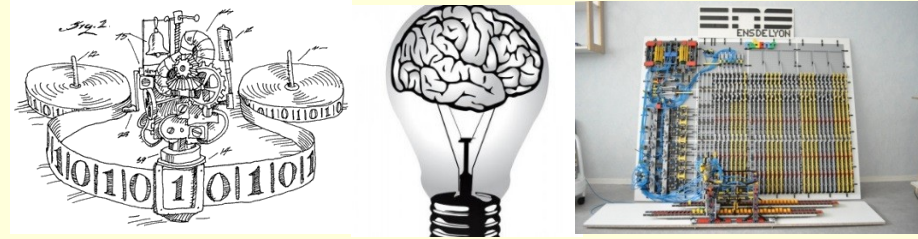
Busy Beavers
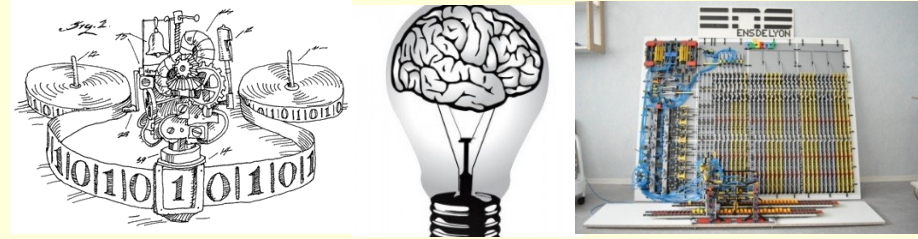
- n = 1,2,3 solved by Lin and Rado in 1960's
- n = 4 solved by Brady in 1970's
  - "proof" unsatisfactory; 200+ cases "checked by hand"
- n = 5, 6
  - monster machines found in 1990's and 2000's,
  - proof still not complete
  - Bigger monsters could be out there!

- Much evidence missing and is being re-created

  See Heiner Marxen's web page for more

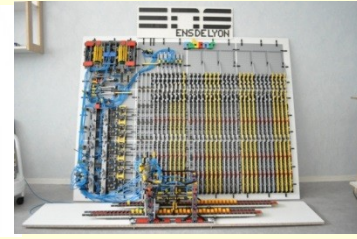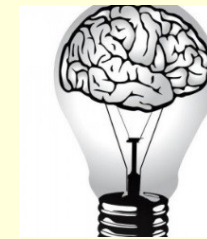# Questions?

Questions?

Questions?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

KEEP
CALM
AND
ASK ME
QUESTIONS

KeepCalmAndPosters.com

Computing Theory

Busy Beaver Grows FAST!

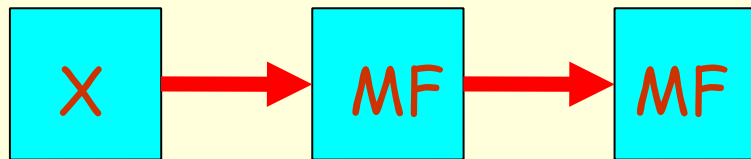The busy beaver function is non-computable, because it grows faster than any computable function!

Proof: Let f be any computable function.
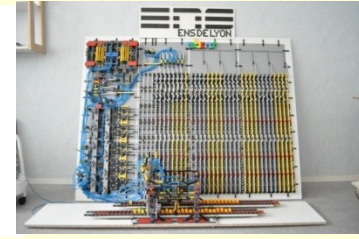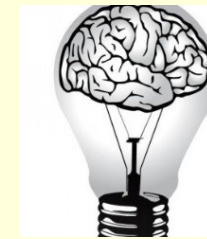
As f computable, so is

$F(x) = \sum_{0 \le i \le x} f(i) + i^2$ and $F(x) > f(x)$

So there is a k-state machine MF : x 1's $\rightarrow$ F(x) 1's

Consider M: X then MF then MF where X: blank $\rightarrow$ x 1's.
Note X has x states.
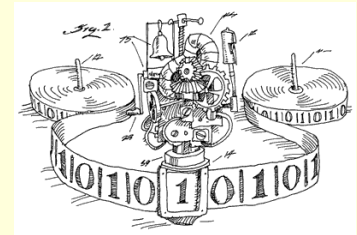
Busy Beaver Grows FAST!

M behaves as follows:

- M first writes x 1's
- M mimics MF, writing F(x) 1's on the tape
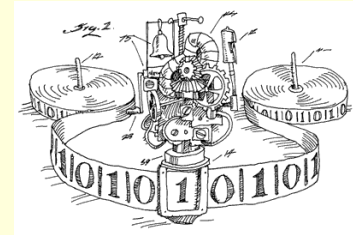- M mimics MF again, writing F(F(x)) 1's on the tape

M has x + 2k states, so
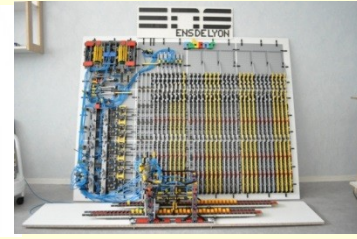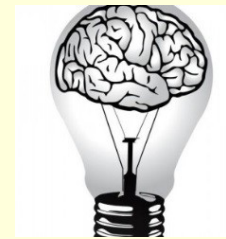bb(n+2k) ≥ 1's output by M = x + F(x) + F(F(x)) > F(F(x))

Now $F(x) ≥ x^2 > x + 2k$ for x > m, and F(x) > F(y) when x > y, and so F(F(x)) > F(x+2k) > f(x+2k)

So bb(x+2k) ≥ x + F(x) + F(F(x)) > F(F(x)) > F(x+2k) > f(x+2k)

Busy Beaver Grows FAST!

This means that bb(n) grows faster than any computable function (!)

Hence bb(n) is not computable

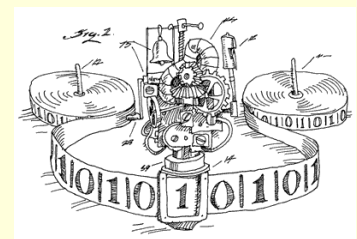Assignment Project Exam Help

https://powcoder.com

$n^{n!} + 12$ is computable ...
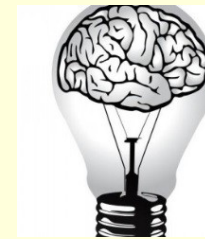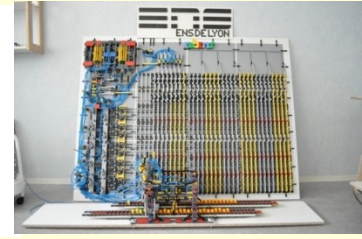
Add WeChat powcoder

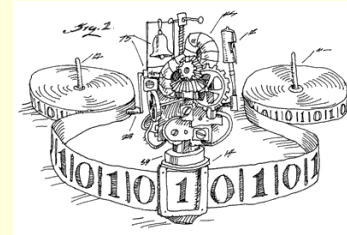(insert your "worst nightmare" computable function here)

I WIN! I ALWAYS WIN!

Finding Busy Beavers

- Generate all machines of a given size
- Remove those which do not terminate
- Take maximum of the rest

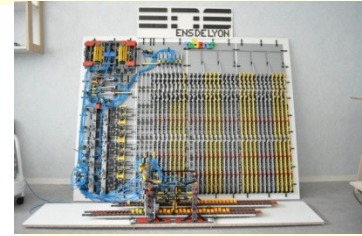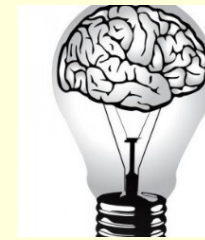Problem 1: There are $(2n-1) \times (4n)^{(2n-2)}$ machines with n states (n= 5 gives 'only' 230,400,000 machines (!) ☺)

Problem 2: How can we write a program to classify machines into terminating and non-terminating?

No **general** method, but ...

Monsters are rare ...

prod	5	6	7	8	9	10	11	12	13
number	73,617	13,029	1981	475	79	13	6	5	2

Of 117,440,512 4-state machines:
- 89,207 irredundant and terminate with prod ≥ 5
- only 2,561 machines with prod > bb(3)
- loops abound!

Computing Theory
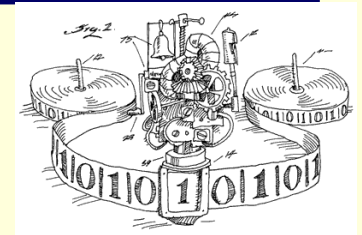
5-state monsters …
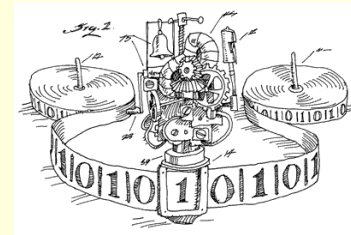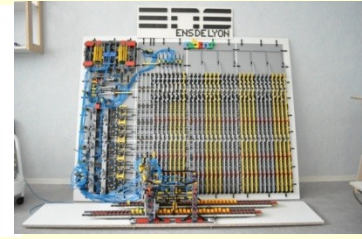
prod	max	transitions
4098	12,288	47,176,870
4098	6,144	11,798,826
4097	6,143	23,554,764
4097	6,143	11,798,796
4096	6,143	11,804,910
4096	6,143	11,804,896
1471	1,474	2,358,064

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Computing Theory
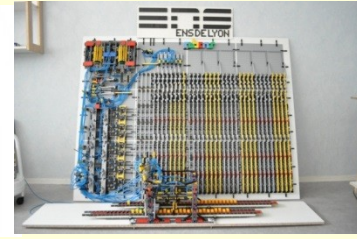
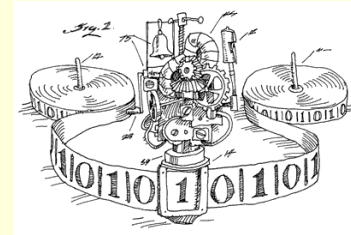# Platypus Machines

An n-state machine of productivity m shows
- bb(n) ≥ m
- at most n states are needed to print m 1's
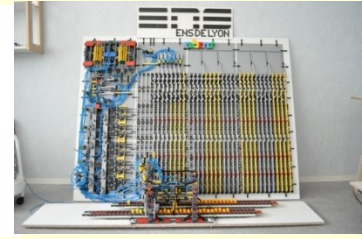
Question: what is the minimum number of states needed to print m 1's?
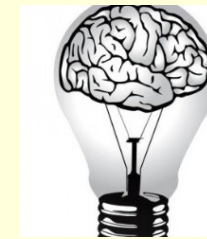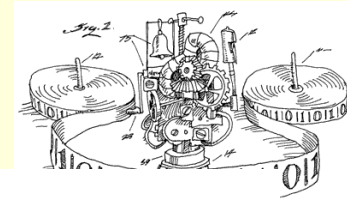
We call this the placid platypus or pp(m)

Known Platypus values

- 1-83 except 46, 48, 50, 74, 75, 77, 80, 82
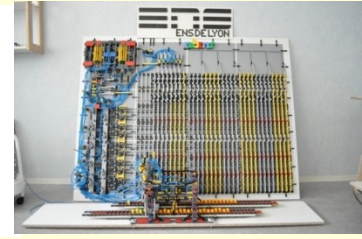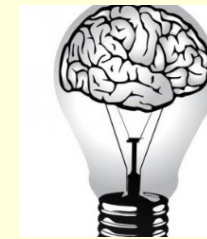- 87,88,89,91,99,112,...
- ...,1471, (..?...), 3996, 4097, 4098

Question: Is it true that there is a 5-state machine which prints m 1's for each bb(4) ≤ m ≤ bb(5)?

This is certainly false for bb(5) to bb(6).

Platypus questions
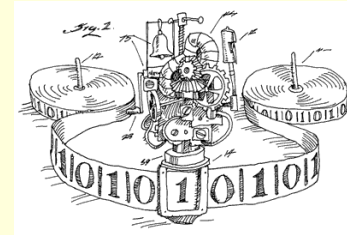
- Distribution of platypus machines for n = 5
- Largest interval [m1,m2] of **existence?**
- Largest interval [m1,m2] of **non-existence?**
- Smallest m s.t. pp(m) ≥ 6?
- Distribution of platypus machines for n = 6 …
- Smallest m s.t. pp(m) ≥ 7? (!!!)

Questions?

Questions?

Questions?

KEEP CALM AND ASK ME QUESTIONS

KeepCalmAndPosters.com

Week 12

Computing Theory
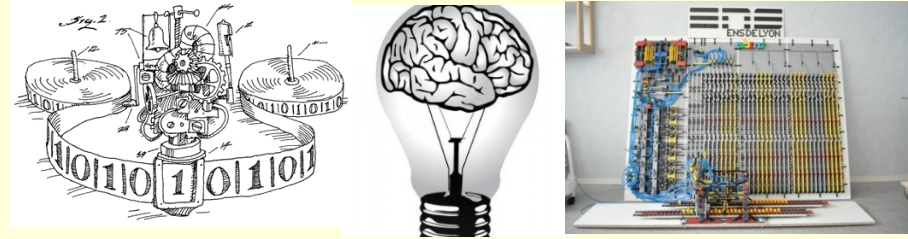
Universal TMs

- Quest for the smallest universal TM goes on ...
- Involves searching similar (but larger) spaces

Alain Colmerauer (KR'08 talk)

- U on code(M)code(w) simulates M on w
- Let M = U
- U on code(U)code(w) simulates U on w
- Let w = blank (and assume code(blank) = blank)
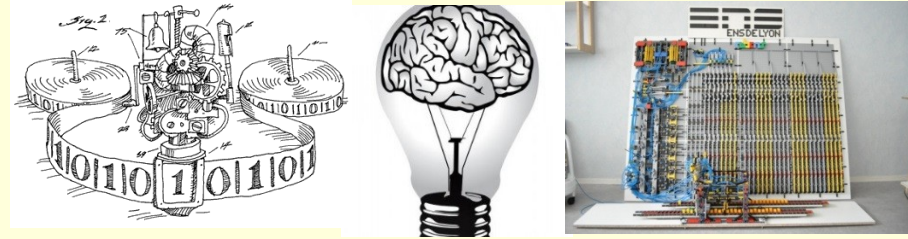- U on code(U) simulates U on blank

Hence pseudo-universality test: M is pseudo-universal if M on code(M) simulates M on blank

Computing Theory

Universal TMs

- What exactly is the definition of a universal Turing machine?

- How can such definitions be used to identify universal machines "in the wild"?

- What constraints are there on the coding function?

- Does a UTM have to terminate?
- Must a UTM terminate on code(M)code(w) exactly when M terminates on w?

- What is an appropriate "architecture" for a UTM? (code(M)code(w) vs code(w)code(M))

Computing Theory

Questions?

Questions?

Questions?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

KEEP CALM AND ASK ME QUESTIONS

KeepCalmAndPosters.com

Week 12

Computing Theory

The Platypus Game

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder
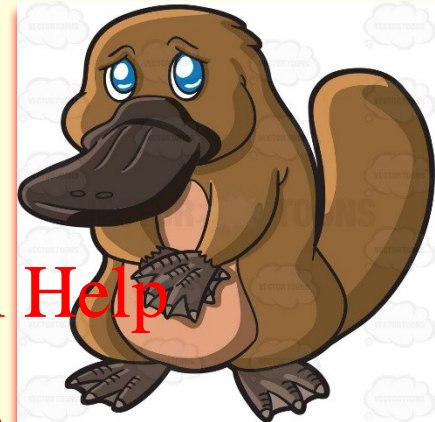
The Platypus Game

This is a research project!
- You have done a lot of initial experimentation
- Rules and scoring much improved
- Still need to find champion machines
- 2-animal and 3-animal cases?
- Eliminate unfair machines from 268,435,456
- Exercise in Turing machine concepts
- Exercise in dealing with intractability

- More work to be done!

That's it!

Computing Theory

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Break time! (We resume when all the pictures are gone! This will take 3 minutes!)