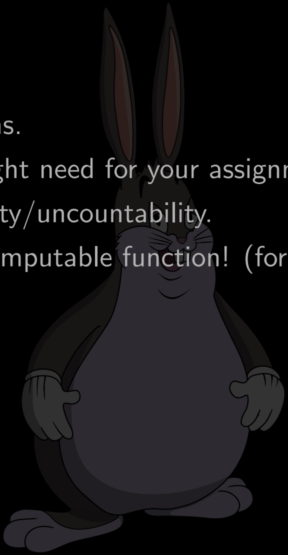
CSC363 Tutorial #5

“Review” (and some useful facts for A3?)

February 16, 2022

Learning objectives this tutorial

- Review *m*-reductions and 1-reductions.
- Prove some useful facts that you might need for your assignment...
- Recall/prove properties of countablility/uncountability.
- Show that "busy beaver" is a non-computable function! (for fun, of course)

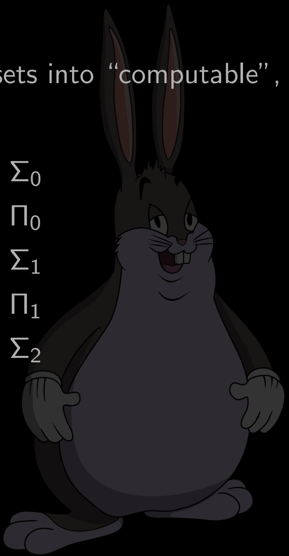


Stuff from like, the past few weeks

Task: Categorize these terms describing sets into “computable”, “CE”, or “neither”.

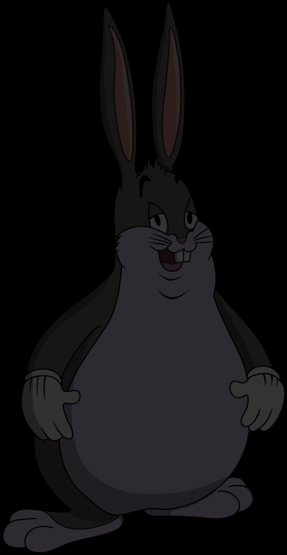
- Listable
- Diophantine
- Countable
- Uncountable
- Decidable

- Σ_0
- Π_0
- Σ_1
- Π_1
- Σ_2



Stuff from last lecture?

Question: What is the definition of $A \leq_m B$ and $A \equiv_1 B$?

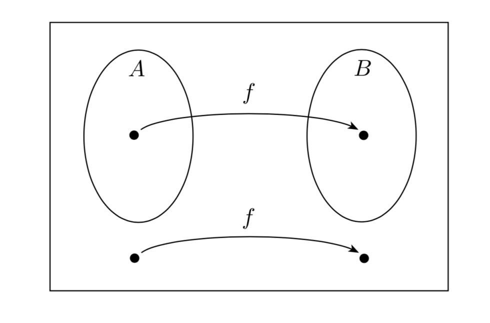


Stuff from last lecture?

Question: What is the definition of $A \leq_m B$ and $A \equiv_1 B$?

Ans: Given two sets $A, B \subseteq \mathbb{N}$, we say that $A \leq_m B$ if there exists a *computable* $f : \mathbb{N} \to \mathbb{N}$ such that

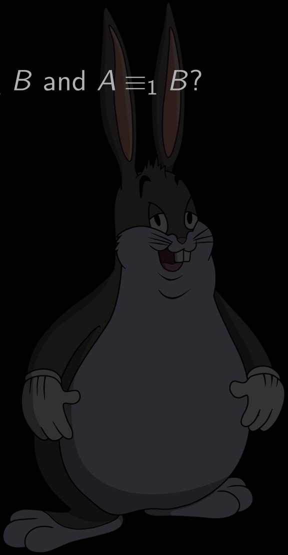
$$x \in A \Leftrightarrow f(x) \in B.$$



If $A \leq_m B$ and $B \leq_m A$, we say that $A \equiv_m B$.

Stuff from last lecture?

Question: What is the definition of $A \leq_1 B$ and $A \equiv_1 B$?

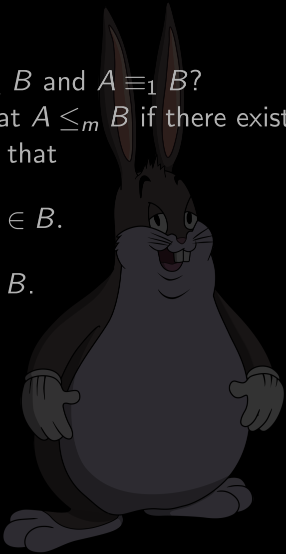


Stuff from last lecture?

Question: What is the definition of $A \leq_1 B$ and $A \equiv_1 B$?
Ans: Given two sets $A, B \subseteq \mathbb{N}$, we say that $A \leq_m B$ if there exists a *computable and injective* $f : \mathbb{N} \to \mathbb{N}$ such that

$$x \in A \Leftrightarrow f(x) \in B.$$

If $A \leq_1 B$ and $B \leq_1 A$, we say that $A \equiv_1 B$.

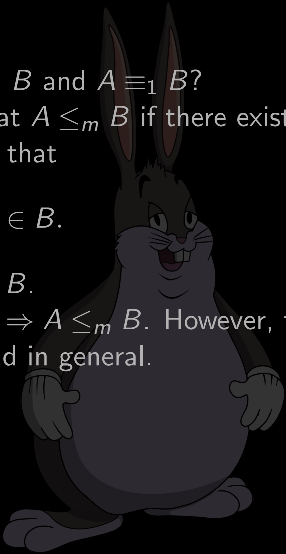


Stuff from last lecture?

Question: What is the definition of $A \leq_1 B$ and $A \equiv_1 B$?
Ans: Given two sets $A, B \subseteq \mathbb{N}$, we say that $A \leq_m B$ if there exists a *computable and injective* $f : \mathbb{N} \to \mathbb{N}$ such that

$$x \in A \Leftrightarrow f(x) \in B.$$

If $A \leq_1 B$ and $B \leq_1 A$, we say that $A \equiv_1 B$.
Immediately from the definitions, $A \leq_1 B \Rightarrow A \leq_m B$. However, the converse $A \leq_m B \Rightarrow A \leq_1 B$ does not hold in general.

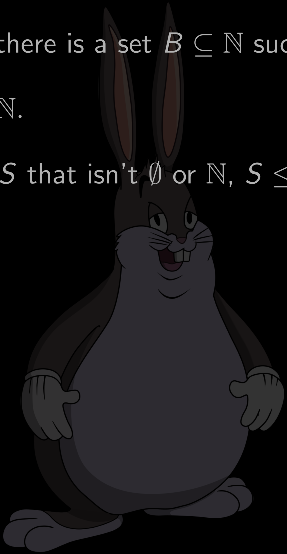


Assignment 3

Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

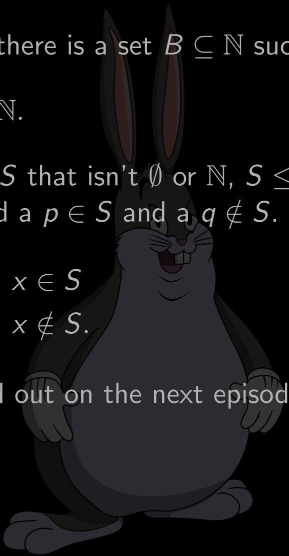
$$S \equiv_1 B \times \mathbb{N}.$$

Task: Show that for any computable set S that isn't $\emptyset$ or $\mathbb{N}$, $S \leq_1 S \times \mathbb{N}$.



[1] Or you may just prove it yourself instead.

Assignment 3



Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

$$S \equiv_1 B \times \mathbb{N}.$$

Task: Show that for any computable set S that isn't $\emptyset$ or $\mathbb{N}$, $S \leq_1 S \times \mathbb{N}$.
Ans: Since $S \neq \emptyset$ and $S \neq \mathbb{N}$, we can find a $p \in S$ and a $q \notin S$. Define

$$f(x) = \begin{cases} \langle p, x \rangle & x \in S \\ \langle q, x \rangle & x \notin S. \end{cases}$$

f is computable and injective. Why? Find out on the next episode.[1]

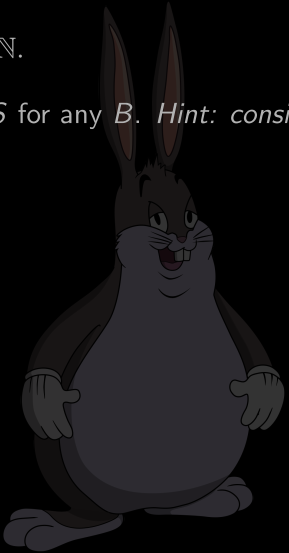
[1] Or you may just prove it yourself instead.

Assignment 3 

Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

$$S \equiv_1 B \times \mathbb{N}.$$

Task: Find a set S such that $B \times \mathbb{N} \not\leq_1 S$ for any B. *Hint: consider the cardinality of* S.

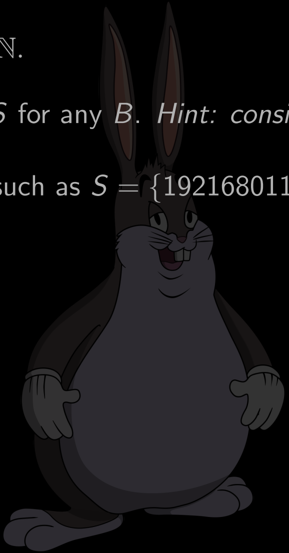


Assignment 3

Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

$$S \equiv_1 B \times \mathbb{N}.$$

Task: Find a set S such that $B \times \mathbb{N} \not\leq_1 S$ for any B. *Hint: consider the cardinality of S.*
Ans: Any nonempty finite set S will do, such as $S = \{192168011\}$.
Consider any set B:



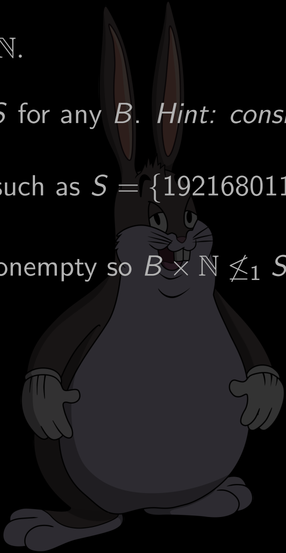
Assignment 3

Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

$$S \equiv_1 B \times \mathbb{N}.$$

Task: Find a set S such that $B \times \mathbb{N} \not\leq_1 S$ for any B. *Hint: consider the cardinality of S.*
Ans: Any nonempty finite set S will do, such as $S = \{192168011\}$.
Consider any set B:

- If $B = \emptyset$ then $B \times \mathbb{N} = \emptyset$, but S is nonempty so $B \times \mathbb{N} \not\leq_1 S$ (Why?).



Assignment 3

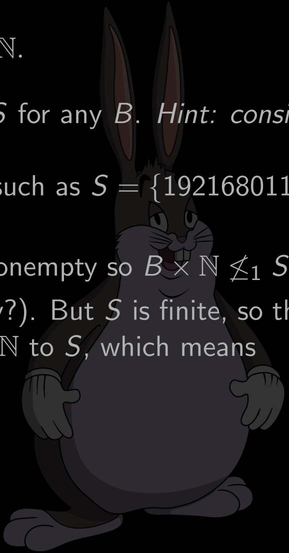
Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

$$S \equiv_1 B \times \mathbb{N}.$$

Task: Find a set S such that $B \times \mathbb{N} \not\leq_1 S$ for any B. *Hint: consider the cardinality of S.*

Ans: Any nonempty finite set S will do, such as $S = \{192168011\}$.
Consider any set B:

- If $B = \emptyset$ then $B \times \mathbb{N} = \emptyset$, but S is nonempty so $B \times \mathbb{N} \not\leq_1 S$ (Why?).
- If $B \neq \emptyset$, then $B \times \mathbb{N}$ is infinite (Why?). But S is finite, so there is no way to create an injection from $B \times \mathbb{N}$ to S, which means $B \times \mathbb{N} \not\leq_1 S$.



Assignment 3

Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

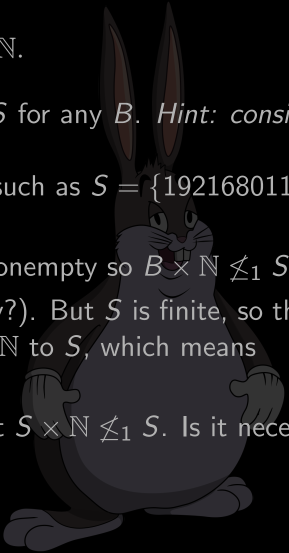
$$S \equiv_1 B \times \mathbb{N}.$$

Task: Find a set S such that $B \times \mathbb{N} \not\leq_1 S$ for any B. *Hint: consider the cardinality of S.*

Ans: Any nonempty finite set S will do, such as $S = \{192168011\}$. Consider any set B:

- If $B = \emptyset$ then $B \times \mathbb{N} = \emptyset$, but S is nonempty so $B \times \mathbb{N} \not\leq_1 S$ (Why?).
- If $B \neq \emptyset$, then $B \times \mathbb{N}$ is infinite (Why?). But S is finite, so there is no way to create an injection from $B \times \mathbb{N}$ to S, which means $B \times \mathbb{N} \not\leq_1 S$.

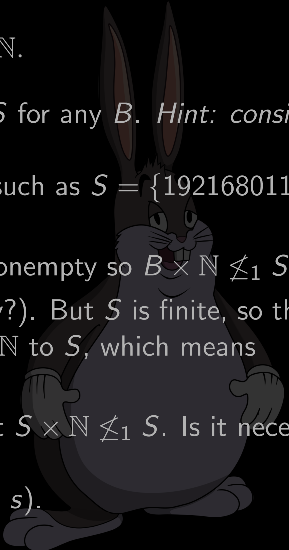
Question: We've found a set S such that $S \times \mathbb{N} \not\leq_1 S$. Is it necessarily true that for any set S, $S \times \mathbb{N} \leq_m S$?



Assignment 3

Definition: A set $S \subseteq \mathbb{N}$ is a **cylinder** if there is a set $B \subseteq \mathbb{N}$ such that

$$S \equiv_1 B \times \mathbb{N}.$$



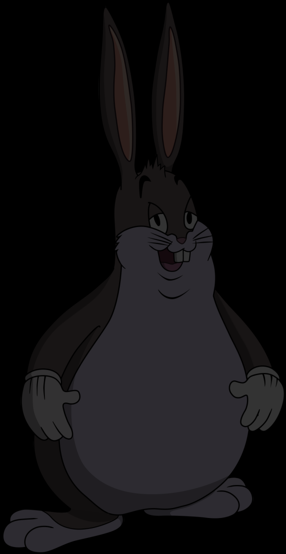
Task: Find a set S such that $B \times \mathbb{N} \not\leq_1 S$ for any B. *Hint: consider the cardinality of S.*
Ans: Any nonempty finite set S will do, such as $S = \{192168011\}$.
Consider any set B:

- If $B = \emptyset$ then $B \times \mathbb{N} = \emptyset$, but S is nonempty so $B \times \mathbb{N} \not\leq_1 S$ (Why?).
- If $B \neq \emptyset$, then $B \times \mathbb{N}$ is infinite (Why?). But S is finite, so there is no way to create an injection from $B \times \mathbb{N}$ to S, which means $B \times \mathbb{N} \not\leq_1 S$.

Question: We've found a set S such that $S \times \mathbb{N} \not\leq_1 S$. Is it necessarily true that for any set S, $S \times \mathbb{N} \leq_m S$?
Ans: Yes (consider the function $f\langle s, n\rangle = s$).

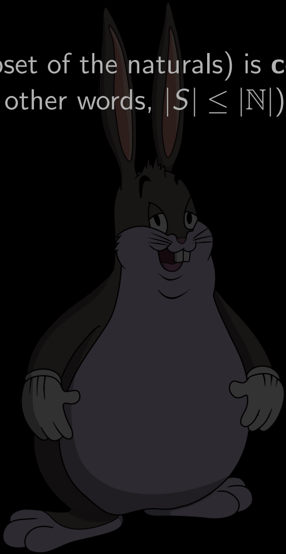
Speaking of cardinality...

Hope you remember your MAT102 stuff!



Speaking of cardinality...

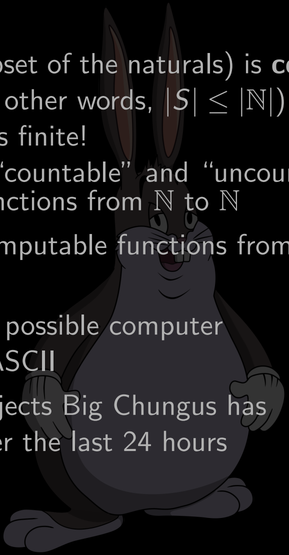
Hope you remember your MAT102 stuff!
Definition: A set S (not necessarily a subset of the naturals) is **countable** if there is an injection from S to $\mathbb{N}$ (or in other words, $|S| \leq |\mathbb{N}|$).



Speaking of cardinality...

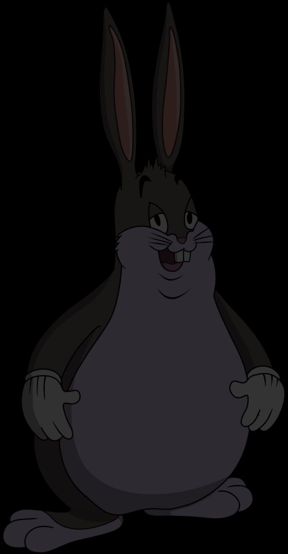
Hope you remember your MAT102 stuff!
Definition: A set S (not necessarily a subset of the naturals) is **countable** if there is an injection from S to $\mathbb{N}$ (or in other words, $|S| \leq |\mathbb{N}|$).
Note: “Countable” in this course includes finite!
Task: Categorize the following sets into “countable” and “uncountable”.

- $\mathbb{N}$
- $\mathbb{R}$
- $\mathbb{N} \times \mathbb{N}$
- $\mathbb{Q}$
- $\mathbb{N}^n$
- $P(\mathbb{N})$ (power set)
- The set of functions from $\mathbb{N}$ to $\mathbb{N}$
- The set of computable functions from $\mathbb{N}$ to $\mathbb{N}$
- The set of all possible computer programs in ASCII
- The set of objects Big Chungus has consumed over the last 24 hours



Speaking of cardinality...

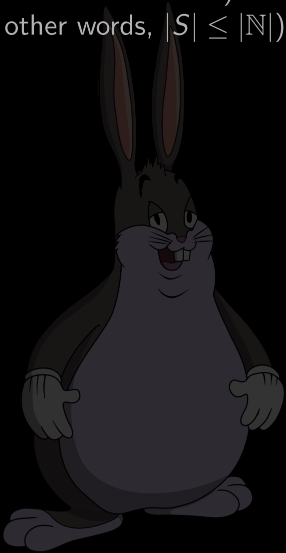
Hope you remember your MAT102 stuff!



Speaking of cardinality...

Hope you remember your MAT102 stuff!

Definition: A set S (not necessarily a subset of the naturals) is **countable** if there is an injection from S to $\mathbb{N}$ (or in other words, $|S| \leq |\mathbb{N}|$).



Speaking of cardinality...

Hope you remember your MAT102 stuff!

Definition: A set S (not necessarily a subset of the naturals) is **countable** if there is an injection from S to $\mathbb{N}$ (or in other words, $|S| \leq |\mathbb{N}|$).

Task: Let $S_0, S_1, \ldots$ be a countably infinite collection of countably infinite and pairwise disjoint sets. Show that

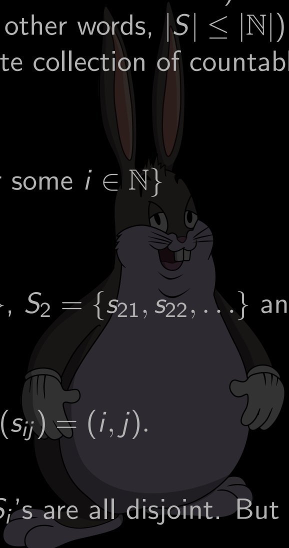
$$\bigcup_{i=0}^{\infty} S_i = \{s : s \in S_i \text{ for some } i \in \mathbb{N}\}$$

is countable.

Ans: Write it out! Let $S_1 = \{s_{11}, s_{12}, \ldots\}$, $S_2 = \{s_{21}, s_{22}, \ldots\}$ and so on. We can then define the injection

$$f : \bigcup_{i=0}^{\infty} S_i \to \mathbb{N} \times \mathbb{N}, f(s_{ij}) = (i, j).$$

This is well-defined and injective, as the S_i's are all disjoint. But we know $\mathbb{N} \times \mathbb{N}$ is countable already!



BusyBeaver

Not to be confused with the Android game *Busy Beaver*...

Here's the BusyBeaver(n) game!

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *(*$\square$ *being the blank symbol), with* **exactly** n **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

BusyBeaver

Here's the BusyBeaver(n) game! (for fun only, not needed for this course)

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *($\square$ being the blank symbol), with* **exactly** *n* **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Task: Play BusyBeaver(1). That is, build a 1-state (excluding the halting state) TM that writes as many '1's to the tape as possible before halting.

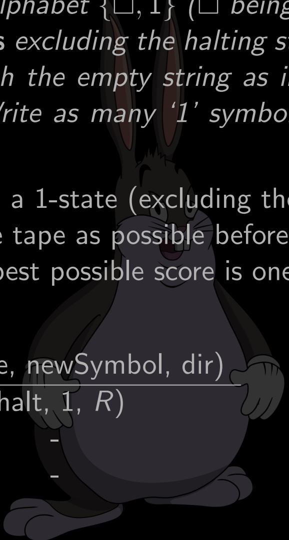
BusyBeaver

Here's the BusyBeaver(n) game! (for fun only, not needed for this course)

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *(*$\square$ *being the blank symbol), with* **exactly** *n* **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Task: Play BusyBeaver(1). That is, build a 1-state (excluding the halting state) TM that writes as many '1's to the tape as possible before halting.
Ans: There isn't much we can do... the best possible score is one '1' symbol.

State	Symbol	(newState, newSymbol, dir)
start	$\square$	(halt, 1, R)
start	1	-
halt	-	-

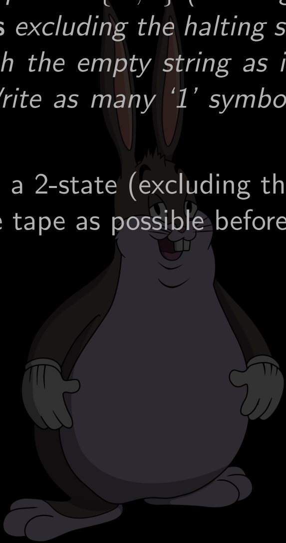


BusyBeaver

Here's the BusyBeaver(n) game!

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *(*$\square$ *being the blank symbol), with* **exactly** *n* **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Task: Play BusyBeaver(2). That is, build a 2-state (excluding the halting state) TM that writes as many '1's to the tape as possible before halting.



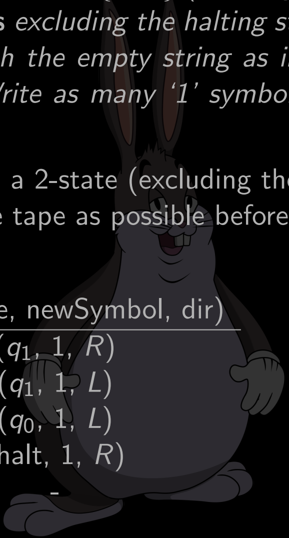
BusyBeaver

Here's the BusyBeaver(n) game!

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *($\square$ being the blank symbol), with* **exactly** n **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Task: Play BusyBeaver(2). That is, build a 2-state (excluding the halting state) TM that writes as many '1's to the tape as possible before halting.
Ans: The best possible score is 4.

State	Symbol	(newState, newSymbol, dir)
start	$\square$	(q_1, 1, R)
start	1	(q_1, 1, L)
q_1	$\square$	(q_0, 1, L)
q_1	1	(halt, 1, R)
halt	-	-

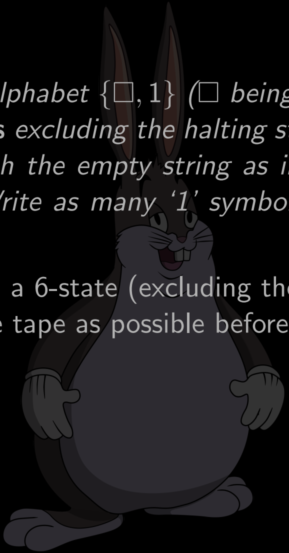


BusyBeaver

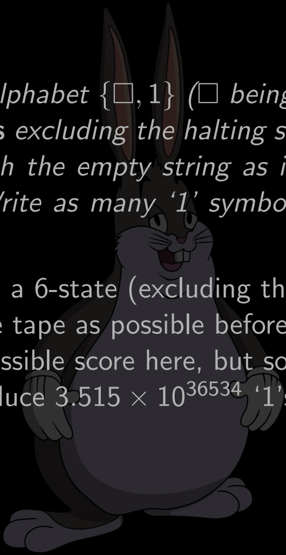
Here's the BusyBeaver(n) game!

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *(*$\square$ *being the blank symbol), with* **exactly** *n* **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Task: Play BusyBeaver(6). That is, build a 6-state (excluding the halting state) TM that writes as many '1's to the tape as possible before halting.



BusyBeaver



Here's the BusyBeaver(n) game!

> *Design a Turing machine with tape alphabet $\{\square, 1\}$ ($\square$ being the blank symbol), with* **exactly** *n* **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all $\square$ on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Task: Play BusyBeaver(6). That is, build a 6-state (excluding the halting state) TM that writes as many '1's to the tape as possible before halting.
Ans: We actually don't know the best possible score here, but someone came up with a 6-state TM that can produce 3.515×10^{36534} '1's.

BusyBeaver

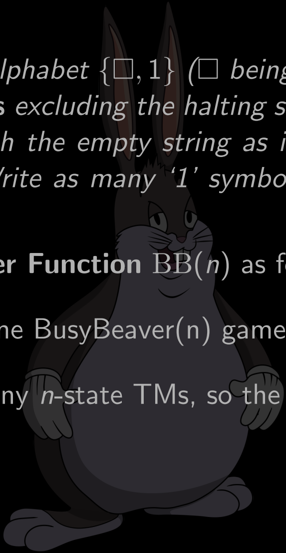
Here's the BusyBeaver(n) game!

> *Design a Turing machine with tape alphabet* $\{\square, 1\}$ *(*$\square$ *being the blank symbol), with* **exactly** *n* **states** *excluding the halting state. This Turing machine* **must halt** *with the empty string as input (starting will all* $\square$ *on the tape). Write as many '1' symbols as you can to the tape, before halting.*

Of course, we can define the **Busy Beaver Function** $\mathrm{BB}(n)$ as follows:

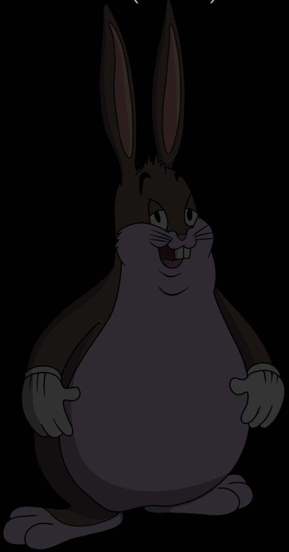
$$\mathrm{BB}(n) = \text{Best possible score in the BusyBeaver(n) game.}$$

For each $n \in \mathbb{N}$, there are only finitely many n-state TMs, so the maximum possible score exists.



BusyBeaver

Task: Prove that $BB(n)$ is not a computable function. That is, there is no program that can compute $BB(n)$ for all $n \in \mathbb{N}$. (Hard!)



BusyBeaver

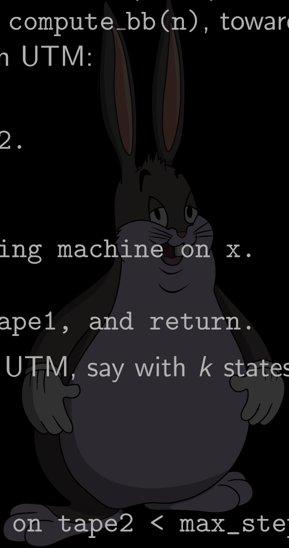
Task: Prove that $BB(n)$ is not a computable function. That is, there is no program that can compute $BB(n)$ for all $n \in \mathbb{N}$. (Hard!)

Ans: Suppose there were such a program `compute_bb(n)`, towards a contradiction. Build the following program UTM:

```
UTM(e, x):
  Create two tapes tape1 and tape2.
  While True:
    On tape1:
      Run one step of the eth turing machine on x.
    On tape2, write a 1.
    If the eth TM halted, erase tape1, and return.
```

We can build a Turing machine that runs UTM, say with k states. But then we can write the following program:

```
H(e, x):
  max_steps = compute_bb(k)
  Run UTM(e, x) while the # of 1s on tape2 < max_steps.
  If it halts, return True. Else, return False.
```



BusyBeaver

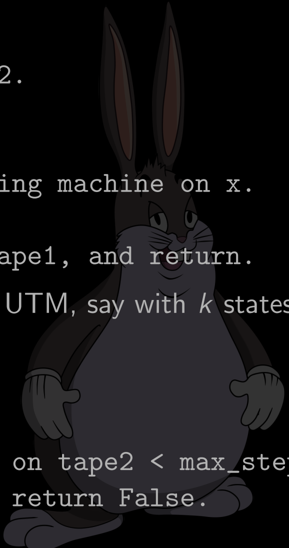
Ans: Suppose there were such a program compute_bb(n), towards a contradiction. Build the following program UTM:

```
UTM(e, x):
  Create two tapes tape1 and tape2.
  While True:
    On tape1:
      Run one step of the eth turing machine on x.
    On tape2, write a 1.
    If the eth TM halted, erase tape1, and return.
```

We can build a Turing machine that runs UTM, say with k states. But then we can write the following program:

```
H(e, x):
  max_steps = compute_bb(k)
  Run UTM(e, x) while the # of 1s on tape2 < max_steps.
  If it halts, return True. Else, return False.
```

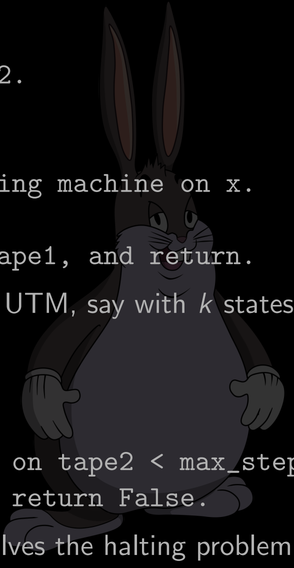
Task: Spot the contradiction.



BusyBeaver

Ans: Suppose there were such a program compute_bb(n), towards a contradiction. Build the following program UTM:

```
UTM(e, x):
  Create two tapes tape1 and tape2.
  While True:
    On tape1:
      Run one step of the eth turing machine on x.
    On tape2, write a 1.
    If the eth TM halted, erase tape1, and return.
```



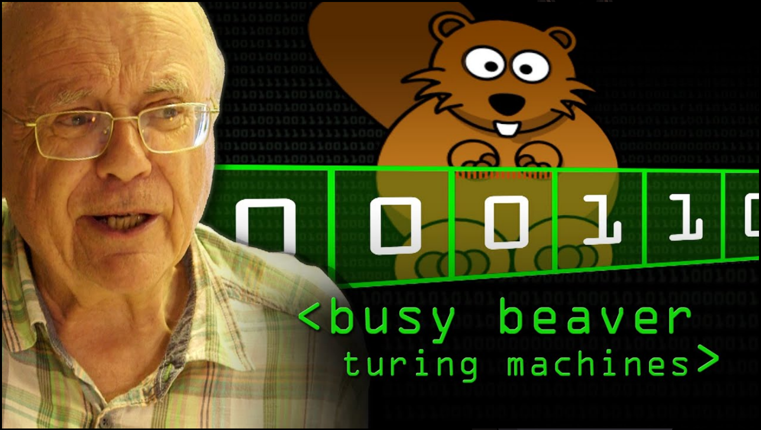
We can build a Turing machine that runs UTM, say with k states. But then we can write the following program:

```
H(e, x):
  max_steps = compute_bb(k)
  Run UTM(e, x) while the # of 1s on tape2 < max_steps.
  If it halts, return True. Else, return False.
```

Task: Spot the contradiction. **Ans:** H solves the halting problem!

BusyBeaver

Some video on BusyBeaver by Computerphile:
`https://www.youtube.com/watch?v=CE8UhcyJS0I`



You get to listen to this guy talk about BB!