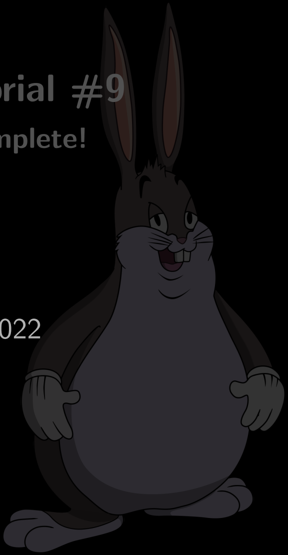
CSC363 Tutorial #9

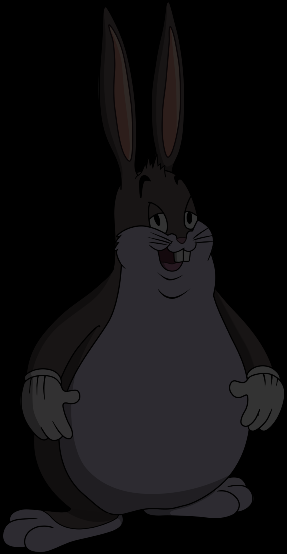
SAT is NP-complete!

March 23, 2022



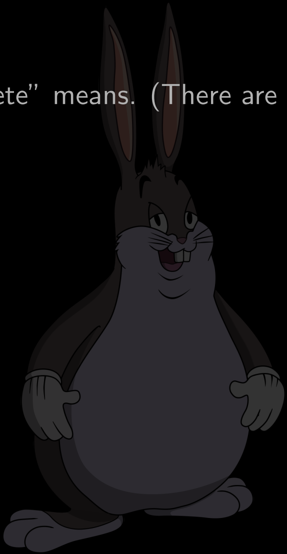
Learning objectives this tutorial

- Review the SAT problem.
- Review NP-completeness.
- Prove that SAT is NP-complete!



NP-completeness

Task: Recall what "$S \subseteq \Sigma^*$ is NP-complete" means. (There are two conditions!)



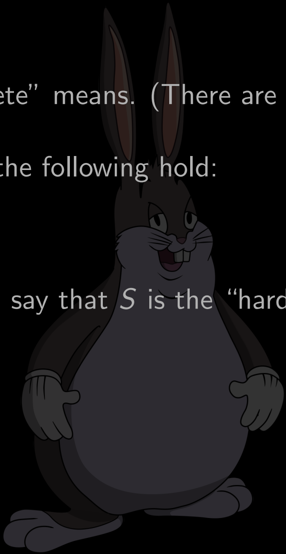
NP-completeness

Task: Recall what "$S \subseteq \Sigma^*$ is NP-complete" means. (There are two conditions!)

Ans: A set S is NP-complete iff both of the following hold:

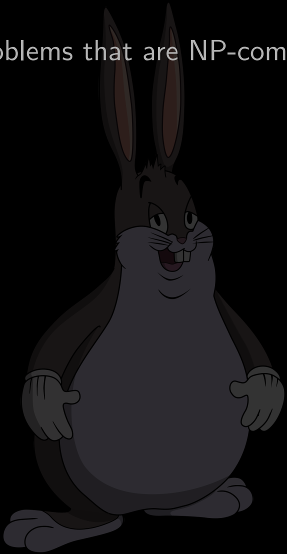
- $S \in \mathrm{NP}$;
- For every $A \in \mathrm{NP}$, $A \leq_P S$.

These two conditions, combined together, say that S is the "hardest" problem in NP.



NP-completeness

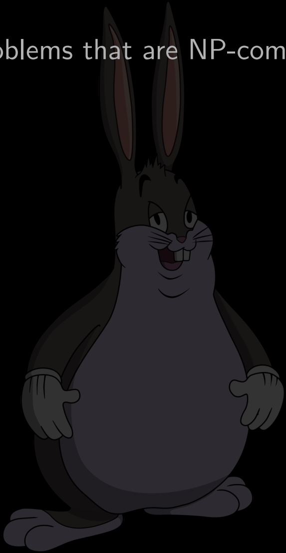
Task: Name some languages/decision problems that are NP-complete.



NP-completeness

Task: Name some languages/decision problems that are NP-complete.
Ans: (I might be missing some here)

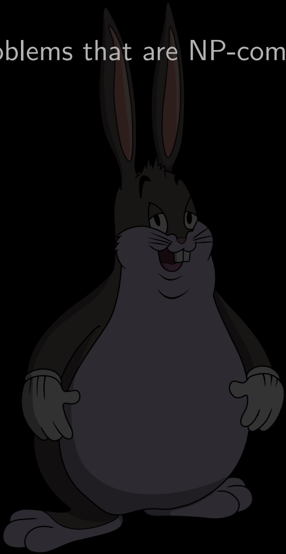
- SAT.



NP-completeness

Task: Name some languages/decision problems that are NP-complete.
Ans: (I might be missing some here)

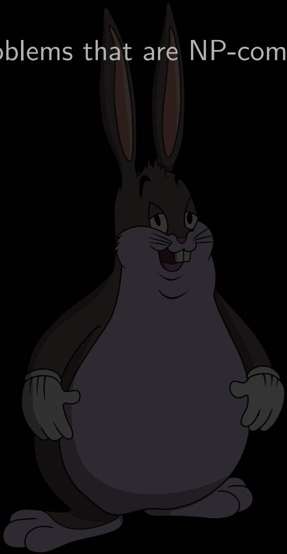
- SAT.
- 3SAT.



NP-completeness

Task: Name some languages/decision problems that are NP-complete.
Ans: (I might be missing some here)

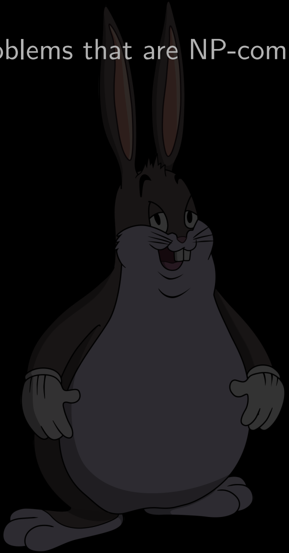
- SAT.
- 3SAT.
- Subset Sum.



NP-completeness

Task: Name some languages/decision problems that are NP-complete.
Ans: (I might be missing some here)

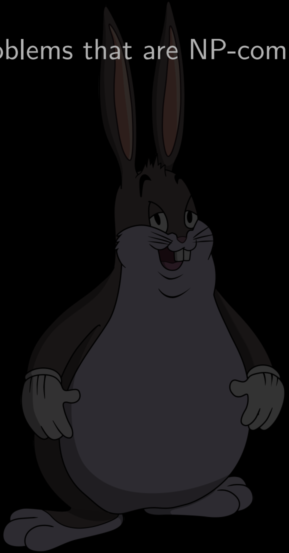
- SAT.
- 3SAT.
- Subset Sum.
- Hamiltonian Path/Cycle Problem.



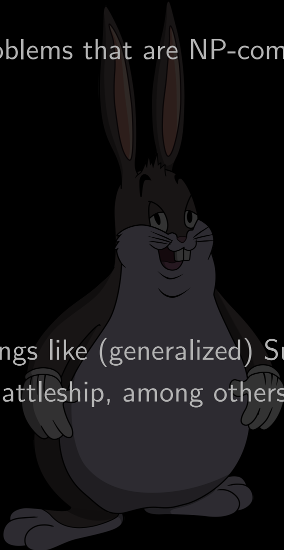
NP-completeness

Task: Name some languages/decision problems that are NP-complete.
Ans: (I might be missing some here)

- SAT.
- 3SAT.
- Subset Sum.
- Hamiltonian Path/Cycle Problem.
- Knapsack Problem.



NP-completeness

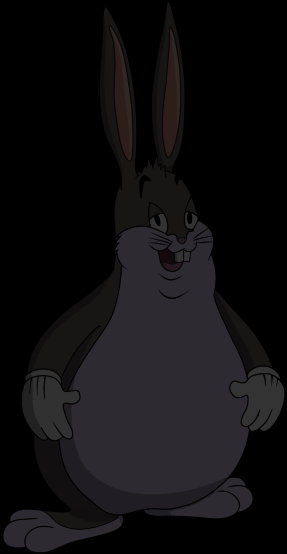


Task: Name some languages/decision problems that are NP-complete.
Ans: (I might be missing some here)

- SAT.
- 3SAT.
- Subset Sum.
- Hamiltonian Path/Cycle Problem.
- Knapsack Problem.
- Many more (click me!), including things like (generalized) Sudoku, optimal solutions for Rubik's cube, Battleship, among others...

Boolean Formulae

Question: What's a *boolean formula*?

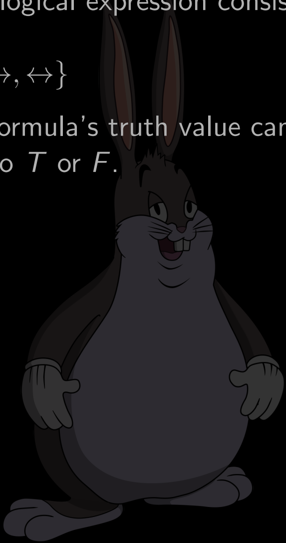


Boolean Formulae

Question: What's a *boolean formula*?
Ans: A *boolean formula* is a well-formed logical expression consisting of symbols from

$$\{(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$$

and one or more "variables". A boolean formula's truth value can be evaluated once all variables are assigned to T or F.



Boolean Formulae

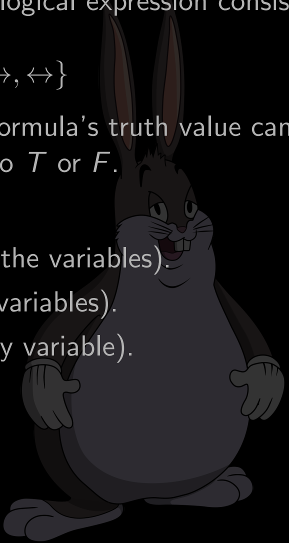
Question: What's a *boolean formula*?
Ans: A *boolean formula* is a well-formed logical expression consisting of symbols from

$$\{(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$$

and one or more "variables". A boolean formula's truth value can be evaluated once all variables are assigned to T or F.

The following are boolean formulas:

- $(x_1 \vee x_2) \rightarrow x_3$ (with x_1, x_2, x_3 being the variables).
- $(x \wedge \neg y) \wedge x$ (with x, y, z being the variables).
- $x \wedge x \wedge x \wedge \neg x$ (with x being the only variable).



Boolean Formulae

Question: What's a *boolean formula*?
Ans: A *boolean formula* is a well-formed logical expression consisting of symbols from

$$\{(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$$

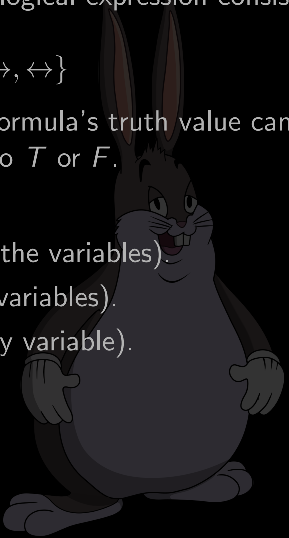
and one or more "variables". A boolean formula's truth value can be evaluated once all variables are assigned to T or F.

The following are boolean formulas:

- $(x_1 \vee x_2) \rightarrow x_3$ (with x_1, x_2, x_3 being the variables).
- $(x \wedge \neg y) \wedge x$ (with x, y, z being the variables).
- $x \wedge x \wedge x \wedge \neg x$ (with x being the only variable).

The following are not boolean formulas:

- $x_1 x_2^2 = x_3$.
- $()()()x_1 \rightarrow \neg$

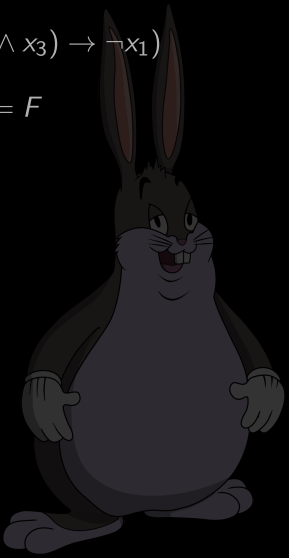


Boolean Formulae

Task: Evaluate the boolean formula

$$(x_1 \vee x_2 \vee x_3) \rightarrow ((x_2 \wedge x_3) \rightarrow \neg x_1)$$

with the assignments $x_1 = T, x_2 = F, x_3 = F$



Boolean Formulae

Task: Evaluate the boolean formula

$$(x_1 \vee x_2 \vee x_3) \rightarrow ((x_2 \wedge x_3) \rightarrow \neg x_1)$$

with the assignments $x_1 = T, x_2 = F, x_3 = F$

Ans: We convert the boolean formula to

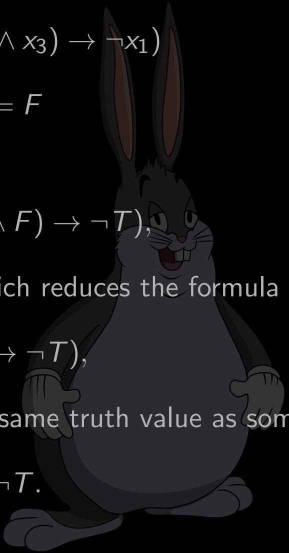
$$(T \vee F \vee F) \rightarrow ((F \wedge F) \rightarrow \neg T),$$

and observe that $(T \vee F \vee F)$ is true, which reduces the formula to

$$T \rightarrow ((F \wedge F) \rightarrow \neg T),$$

or further (since $T \rightarrow$ something has the same truth value as something),

$$(F \wedge F) \rightarrow \neg T.$$



Boolean Formulae

$$(F \wedge F) \rightarrow \neg T.$$

Continuing, $(F \wedge F)$ is false, so we reduce to

$$F \rightarrow \neg T$$

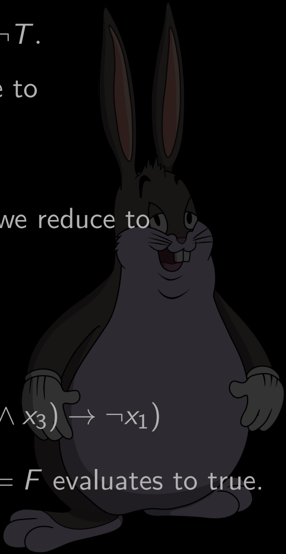
and since $F \rightarrow$ something is always true, we reduce to

$$T.$$

Thus, we conclude that

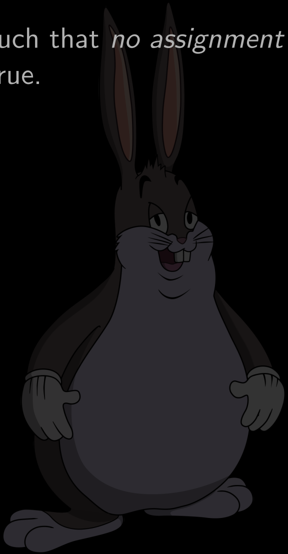
$$(x_1 \vee x_2 \vee x_3) \rightarrow ((x_2 \wedge x_3) \rightarrow \neg x_1)$$

with the assignments $x_1 = T, x_2 = F, x_3 = F$ evaluates to true.



SAT

Task: Come up with a boolean formula such that *no assignment of its variables* makes the formula evaluate to true.

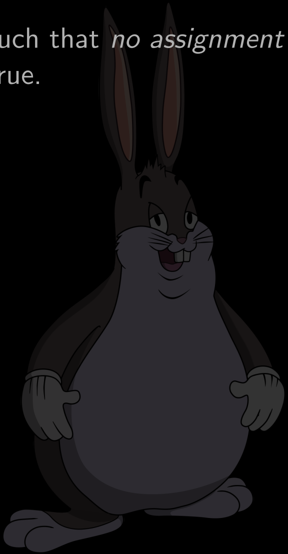


SAT

Task: Come up with a boolean formula such that *no assignment of its variables* makes the formula evaluate to true.
Ans: Something like

$$x \wedge \neg x$$

should work.



SAT

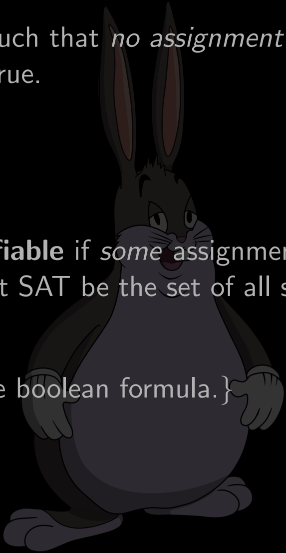
Task: Come up with a boolean formula such that *no assignment of its variables* makes the formula evaluate to true.
Ans: Something like

$$x \wedge \neg x$$

should work.

Definition: A boolean formula ϕ is **satisfiable** if *some* assignment of its variables makes ϕ evaluate to true. We let SAT be the set of all satisfiable boolean formulas:

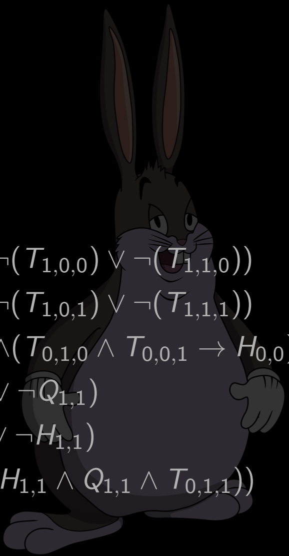
$$\mathrm{SAT} = \{\phi : \phi \text{ is a satisfiable boolean formula.}\}$$



SAT

Task: Determine which of the following formulas are satisfiable.

- $(x_1 \vee x_2) \rightarrow x_3$.
- $(x \wedge \neg y) \wedge x$.
- $x \wedge x \wedge x \wedge \neg x$.
- $$\begin{aligned}
& (T_{0,0,0} \wedge T_{1,0,0}) \\
& \wedge (Q_{0,0}) \wedge (H_{0,0}) \\
& \wedge (\neg(T_{0,0,0}) \vee \neg(T_{0,1,0})) \wedge (\neg(T_{1,0,0}) \vee \neg(T_{1,1,0})) \\
& \wedge (\neg(T_{0,0,1}) \vee \neg(T_{0,1,1})) \wedge (\neg(T_{1,0,1}) \vee \neg(T_{1,1,1})) \\
& \wedge (T_{0,0,0} \wedge T_{0,1,1} \rightarrow H_{0,0}) \wedge \wedge (T_{0,1,0} \wedge T_{0,0,1} \rightarrow H_{0,0}) \\
& \wedge (\neg Q_{0,0} \vee \neg Q_{1,0}) \wedge (\neg Q_{0,1} \vee \neg Q_{1,1}) \\
& \wedge (\neg H_{0,0} \vee \neg H_{1,0}) \wedge (\neg H_{0,1} \vee \neg H_{1,1}) \\
& \wedge ((H_{0,0} \wedge Q_{0,0} \wedge T_{0,0,0}) \rightarrow (H_{1,1} \wedge Q_{1,1} \wedge T_{0,1,1})) \\
& \wedge Q_{1,0} \vee Q_{1,1}
\end{aligned}$$



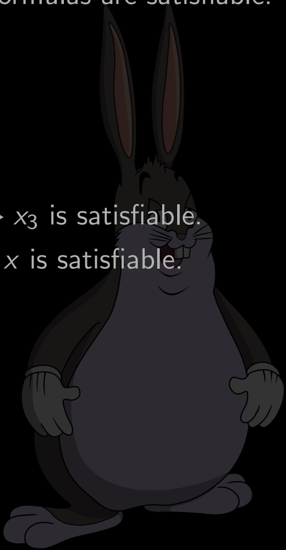
SAT

Task: Determine which of the following formulas are satisfiable.

- $(x_1 \vee x_2) \to x_3$.
- $(x \wedge \neg y) \wedge x$.
- $x \wedge x \wedge x \wedge \neg x$.

Ans:

- $(T \vee T) \to T$ is true, so $(x_1 \vee x_2) \to x_3$ is satisfiable.
- $(T \wedge \neg F) \wedge T$ is true, so $(x \wedge \neg y) \wedge x$ is satisfiable.
- $x \wedge x \wedge x \wedge \neg x$ is not satisfiable.



SAT

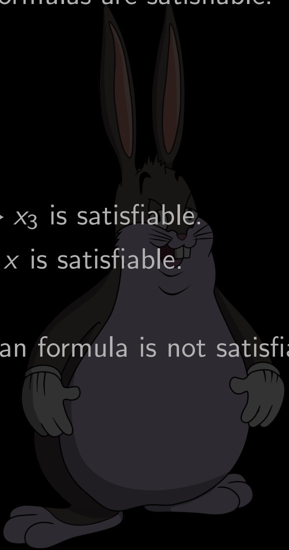
Task: Determine which of the following formulas are satisfiable.

- $(x_1 \vee x_2) \rightarrow x_3$.
- $(x \wedge \neg y) \wedge x$.
- $x \wedge x \wedge x \wedge \neg x$.

Ans:

- $(T \vee T) \rightarrow T$ is true, so $(x_1 \vee x_2) \rightarrow x_3$ is satisfiable.
- $(T \wedge \neg F) \wedge T$ is true, so $(x \wedge \neg y) \wedge x$ is satisfiable.
- $x \wedge x \wedge x \wedge \neg x$ is not satisfiable.

Question: How do we know that a boolean formula is not satisfiable?



SAT

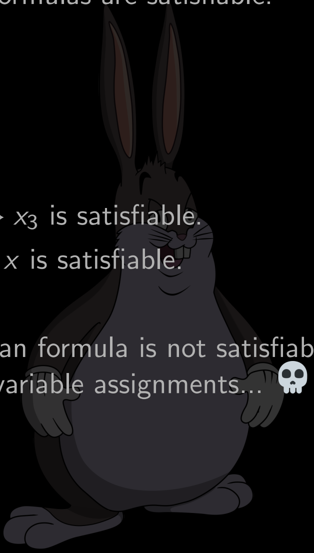
Task: Determine which of the following formulas are satisfiable.

- $(x_1 \vee x_2) \to x_3$.
- $(x \wedge \neg y) \wedge x$.
- $x \wedge x \wedge x \wedge \neg x$.

Ans:

- $(T \vee T) \to T$ is true, so $(x_1 \vee x_2) \to x_3$ is satisfiable.
- $(T \wedge \neg F) \wedge T$ is true, so $(x \wedge \neg y) \wedge x$ is satisfiable.
- $x \wedge x \wedge x \wedge \neg x$ is not satisfiable.

Question: How do we know that a boolean formula is not satisfiable?
Ans: Check through all combinations of variable assignments... 💀



SAT

Task: Determine which of the following formulas are satisfiable.

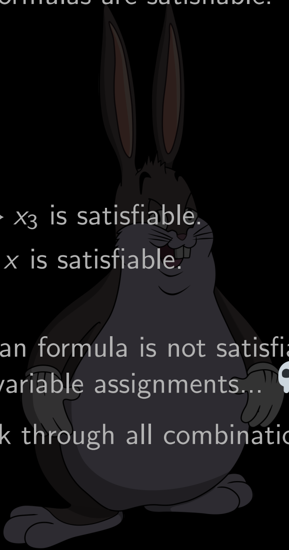
- $(x_1 \vee x_2) \to x_3$.
- $(x \wedge \neg y) \wedge x$.
- $x \wedge x \wedge x \wedge \neg x$.

Ans:

- $(T \vee T) \to T$ is true, so $(x_1 \vee x_2) \to x_3$ is satisfiable.
- $(T \wedge \neg F) \wedge T$ is true, so $(x \wedge \neg y) \wedge x$ is satisfiable.
- $x \wedge x \wedge x \wedge \neg x$ is not satisfiable.

Question: How do we know that a boolean formula is not satisfiable?
Ans: Check through all combinations of variable assignments... 💀

Question: How long does it take to check through all combinations?



SAT

Task: Determine which of the following formulas are satisfiable.

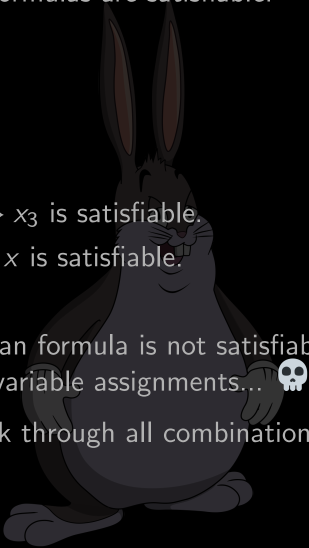
- $(x_1 \lor x_2) \to x_3$.
- $(x \land \neg y) \land x$.
- $x \land x \land x \land \neg x$.

Ans:

- $(T \lor T) \to T$ is true, so $(x_1 \lor x_2) \to x_3$ is satisfiable.
- $(T \land \neg F) \land T$ is true, so $(x \land \neg y) \land x$ is satisfiable.
- $x \land x \land x \land \neg x$ is not satisfiable.

Question: How do we know that a boolean formula is not satisfiable?
Ans: Check through all combinations of variable assignments... 💀

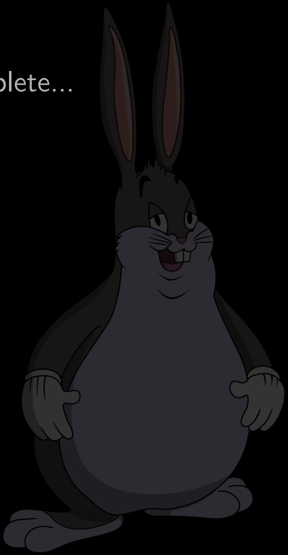
Question: How long does it take to check through all combinations?
Ans: $O(2^n)$... 💀



SAT is another easy-to-verify, hard-to-solve (unproven) problem.

SAT is NP-complete!

But... we can prove that SAT is NP-complete...



SAT is NP-complete!

But... we can prove that SAT is NP-complete...

Theorem (Cook-Levin): SAT is NP-complete.

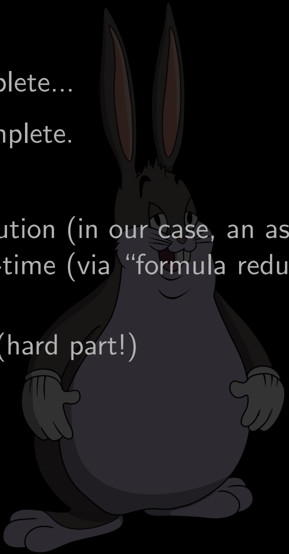
SAT is NP-complete!

But... we can prove that SAT is NP-complete...

Theorem (Cook-Levin): SAT is NP-complete.

Proof:

- $\mathrm{SAT} \in \mathrm{NP}$, since we can verify a solution (in our case, an assignment of variables resulting in true) in poly-time (via "formula reduction", as we have done earlier).
- For any $A \in \mathrm{NP}$, $A \leq \mathrm{SAT}$, since... (hard part!)

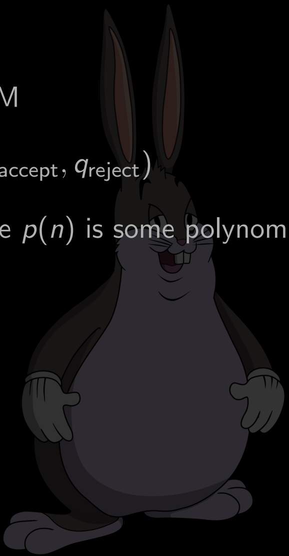


SAT is NP-complete!

(Proof taken from Wikipedia)
Let $A \in \mathrm{NP}$. By definition, there is a NTM

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

that decides A in $p(n)$ steps or less (where $p(n)$ is some polynomial).



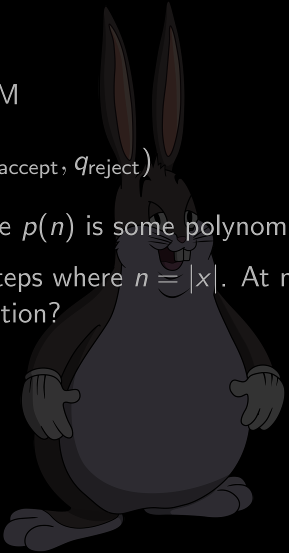
SAT is NP-complete!

(Proof taken from Wikipedia)
Let $A \in \mathrm{NP}$. By definition, there is a NTM

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

that decides A in $p(n)$ steps or less (where $p(n)$ is some polynomial).

Question: Suppose $M(x)$ halts in $p(n)$ steps where $n = |x|$. At most, how many *cells* are accessed during execution?



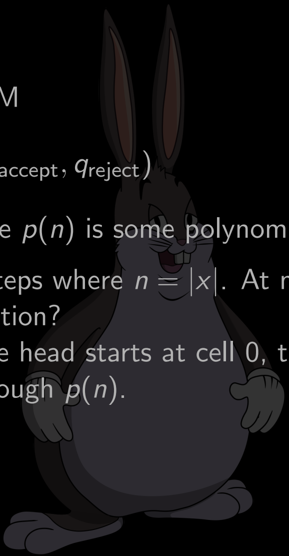
SAT is NP-complete!

(Proof taken from Wikipedia)
Let $A \in \mathrm{NP}$. By definition, there is a NTM

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

that decides A in $p(n)$ steps or less (where $p(n)$ is some polynomial).

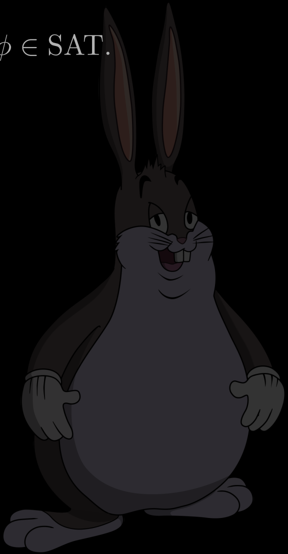
Question: Suppose $M(x)$ halts in $p(n)$ steps where $n = |x|$. At most, how many *cells* are accessed during execution?
Ans: At most $p(n)$ cells: if our read/write head starts at cell 0, then we could only access cell numbers $-p(n)$ through $p(n)$.



SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

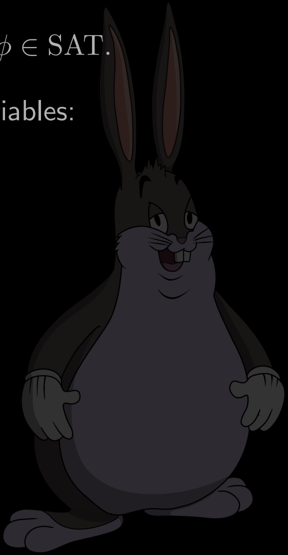


SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:



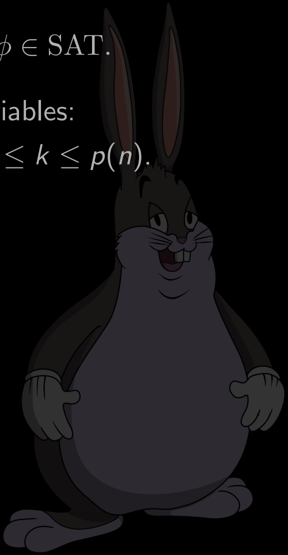
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \le i \le p(n)$, $j \in \Gamma$, $0 \le k \le p(n)$.



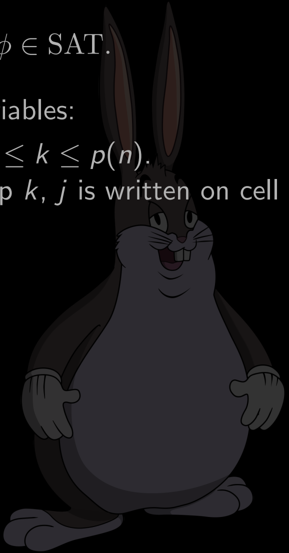
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.



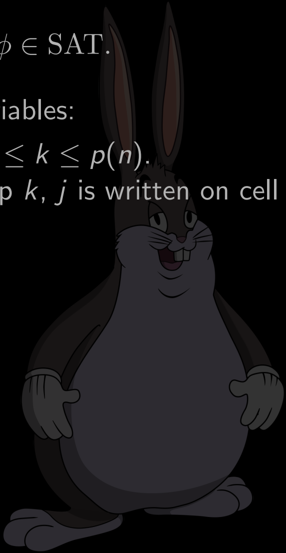
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.



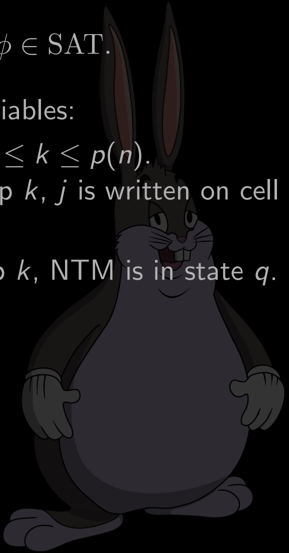
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.



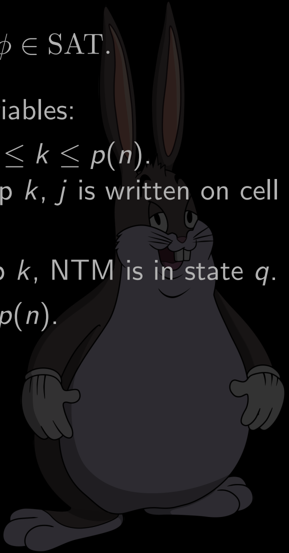
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.



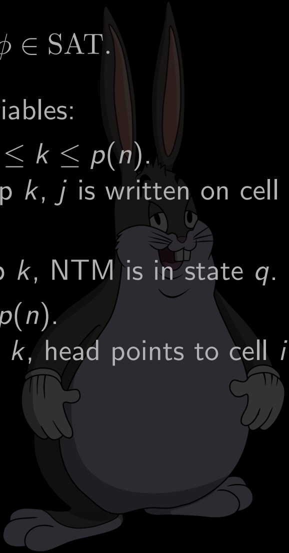
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.



SAT is NP-complete!

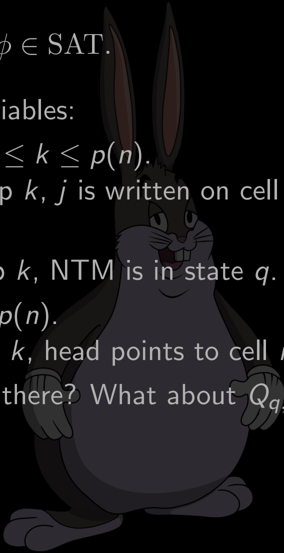
We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.

Question: How many $T_{i,j,k}$ variables are there? What about $Q_{q,k}$ and $H_{i,k}$?



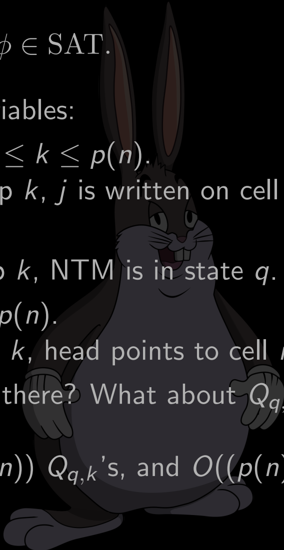
SAT is NP-complete!

We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.



Question: How many $T_{i,j,k}$ variables are there? What about $Q_{q,k}$ and $H_{i,k}$?
Ans: There are $O((p(n))^2)$ $T_{i,j,k}$'s, $O(p(n))$ $Q_{q,k}$'s, and $O((p(n))^2)$ $H_{i,k}$'s.

SAT is NP-complete!

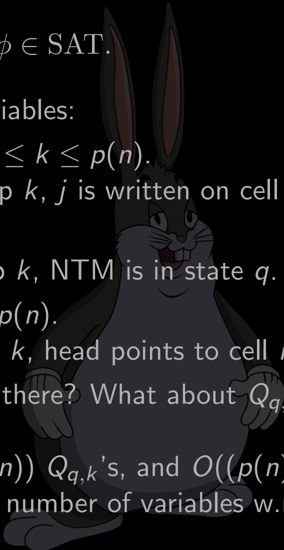
We will construct a formula ϕ in polynomial-time (w.r.t n) such that

$$M(A) \text{ accepts} \Leftrightarrow \phi \in \mathrm{SAT}.$$

This formula ϕ will have the following variables:

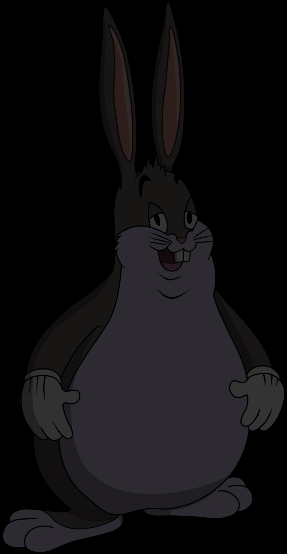
- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.

Question: How many $T_{i,j,k}$ variables are there? What about $Q_{q,k}$ and $H_{i,k}$?
Ans: There are $O((p(n))^2)$ $T_{i,j,k}$'s, $O(p(n))$ $Q_{q,k}$'s, and $O((p(n))^2)$ $H_{i,k}$'s. Either way, there are a polynomial number of variables w.r.t n.



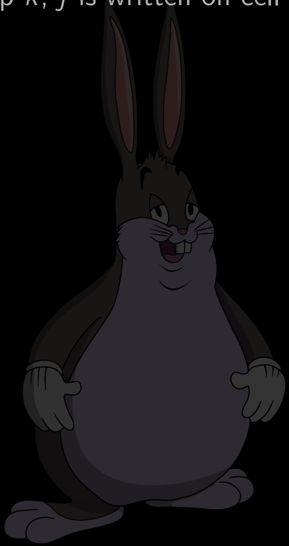
SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.



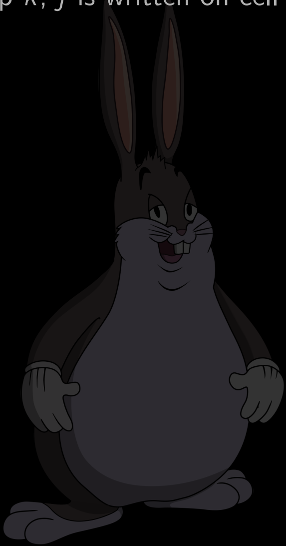
SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.



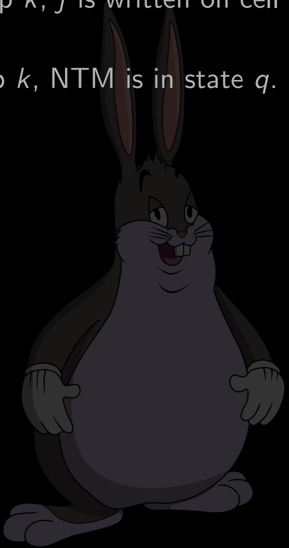
SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.



SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.

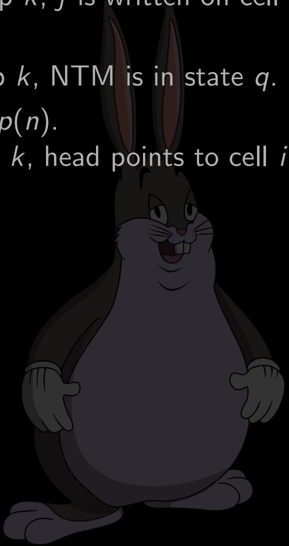


SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.

SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.



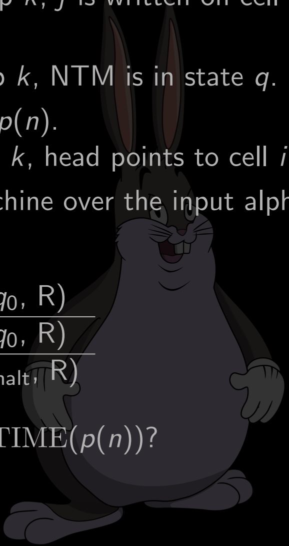
SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.

Task: Suppose M were the following machine over the input alphabet $\{0, 1\}$:

$(0,\ q_0)$	$(1,\ q_0,\ \mathrm{R})$
$(1,\ q_0)$	$(1,\ q_0,\ \mathrm{R})$
$(\square,\ q_0)$	$(1,\ q_{\mathrm{halt}},\ \mathrm{R})$

- Can you find a $p(n)$ such that $M \in \mathrm{TIME}(p(n))$?



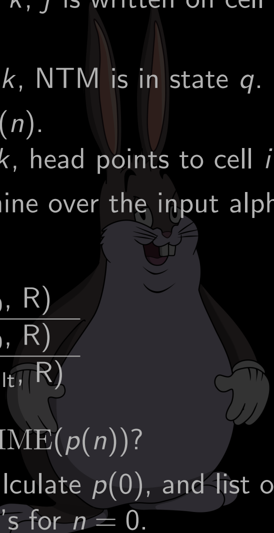
SAT is NP-complete!

- $T_{i,j,k}$ for $-p(n) \leq i \leq p(n)$, $j \in \Gamma$, $0 \leq k \leq p(n)$.
  "Interpretation": $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in Q$, $0 \leq k \leq p(n)$.
  "Interpretation": $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-p(n) \leq i \leq p(n)$, $0 \leq k \leq p(n)$.
  "Interpretation": $H_{i,k}$ true iff at step k, head points to cell i.

Task: Suppose M were the following machine over the input alphabet $\{0, 1\}$:

$(0,\ q_0)$	$(1,\ q_0,\ \text{R})$
$(1,\ q_0)$	$(1,\ q_0,\ \text{R})$
$(\square,\ q_0)$	$(1,\ q_{\text{halt}},\ \text{R})$

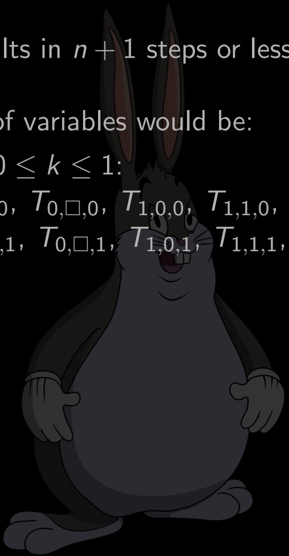
- Can you find a $p(n)$ such that $M \in \mathrm{TIME}(p(n))$?
- Using the $p(n)$ you've found above, calculate $p(0)$, and list out all the $T_{i,j,k}$'s, all the $Q_{q,k}$'s, and all the $H_{i,k}$'s for $n = 0$.



SAT is NP-complete!

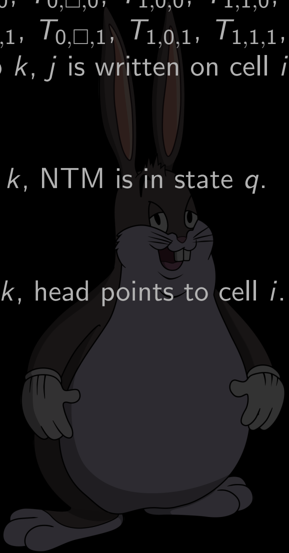
Ans: Our NTM, on an input of size n, halts in $n+1$ steps or less. Thus if $p(n) = n+1$, $M \in \mathrm{TIME}(p(n))$.
For $n = 0$, we have $p(n) = 1$, so our list of variables would be:

- $T_{i,j,k}$ for $-1 \leq i \leq 1$, $j \in \{0, 1, \Box\}$, $0 \leq k \leq 1$:
  $T_{-1,0,0}$, $T_{-1,1,0}$, $T_{-1,\Box,0}$ $T_{0,0,0}$, $T_{0,1,0}$, $T_{0,\Box,0}$, $T_{1,0,0}$, $T_{1,1,0}$, $T_{1,\Box,0}$,
  $T_{-1,0,1}$, $T_{-1,1,1}$, $T_{-1,\Box,1}$, $T_{0,0,1}$, $T_{0,1,1}$, $T_{0,\Box,1}$, $T_{1,0,1}$, $T_{1,1,1}$, $T_{1,\Box,1}$
- $Q_{q,k}$ for $q \in \{q_0, q_{\text{halt}}\}$, $0 \leq k \leq 1$:
  $Q_{q_0,0}$, $Q_{q_{\text{halt}},0}$, $Q_{q_0,1}$, $Q_{q_{\text{halt}},1}$
- $H_{i,k}$ for $-1 \leq i \leq 1$, $0 \leq k \leq 1$:
  $H_{-1,0}$, $H_{0,0}$, $H_{1,0}$, $H_{-1,1}$, $H_{0,1}$, $H_{1,1}$.



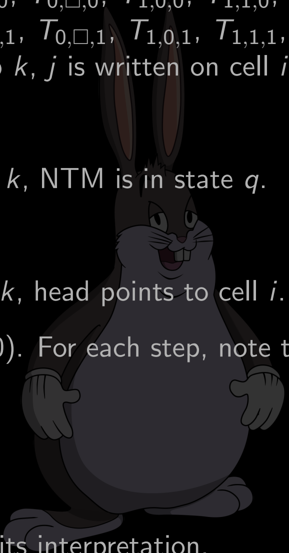
SAT is NP-complete!

- $T_{i,j,k}$ for $-1 \leq i \leq 1$, $j \in \{0, 1, \Box\}$, $0 \leq k \leq 1$:
  $T_{-1,0,0}$, $T_{-1,1,0}$, $T_{-1,\Box,0}$ $T_{0,0,0}$, $T_{0,1,0}$, $T_{0,\Box,0}$, $T_{1,0,0}$, $T_{1,1,0}$, $T_{1,\Box,0}$,
  $T_{-1,0,1}$, $T_{-1,1,1}$, $T_{-1,\Box,1}$, $T_{0,0,1}$, $T_{0,1,1}$, $T_{0,\Box,1}$, $T_{1,0,1}$, $T_{1,1,1}$, $T_{1,\Box,1}$
  Interpretation: $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in \{q_0, q_{\text{halt}}\}$, $0 \leq k \leq 1$:
  $Q_{q_0,0}$, $Q_{q_{\text{halt}},0}$, $Q_{q_0,1}$, $Q_{q_{\text{halt}},1}$
  Interpretation: $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-1 \leq i \leq 1$, $0 \leq k \leq 1$:
  $H_{-1,0}$, $H_{0,0}$, $H_{1,0}$, $H_{-1,1}$, $H_{0,1}$, $H_{1,1}$.
  Interpretation: $H_{i,k}$ true iff at step k, head points to cell i.



SAT is NP-complete!

- $T_{i,j,k}$ for $-1 \leq i \leq 1$, $j \in \{0, 1, \Box\}$, $0 \leq k \leq 1$:
  $T_{-1,0,0}$, $T_{-1,1,0}$, $T_{-1,\Box,0}$ $T_{0,0,0}$, $T_{0,1,0}$, $T_{0,\Box,0}$, $T_{1,0,0}$, $T_{1,1,0}$, $T_{1,\Box,0}$,
  $T_{-1,0,1}$, $T_{-1,1,1}$, $T_{-1,\Box,1}$, $T_{0,0,1}$, $T_{0,1,1}$, $T_{0,\Box,1}$, $T_{1,0,1}$, $T_{1,1,1}$, $T_{1,\Box,1}$
  Interpretation: $T_{i,j,k}$ true iff at step k, j is written on cell i.
- $Q_{q,k}$ for $q \in \{q_0, q_{\text{halt}}\}$, $0 \leq k \leq 1$:
  $Q_{q_0,0}$, $Q_{q_{\text{halt}},0}$, $Q_{q_0,1}$, $Q_{q_{\text{halt}},1}$
  Interpretation: $Q_{q,k}$ true iff at step k, NTM is in state q.
- $H_{i,k}$ for $-1 \leq i \leq 1$, $0 \leq k \leq 1$:
  $H_{-1,0}$, $H_{0,0}$, $H_{1,0}$, $H_{-1,1}$, $H_{0,1}$, $H_{1,1}$.
  Interpretation: $H_{i,k}$ true iff at step k, head points to cell i.



Task: Trace M on an empty input ($n = 0$). For each step, note the following:

- What is written on cells $-1, 0, 1$?
- Which state is the TM in?
- Where is the head pointing?

For each variable listed above, determine its interpretation.

SAT is NP-complete!

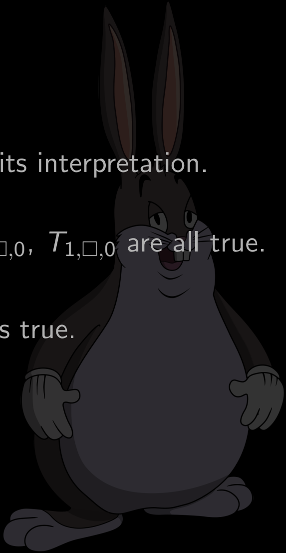
Task: Trace M on an empty input ($n = 0$). For each step, note the following:

- What is written on cells $-1, 0, 1$?
- Which state is the TM in?
- Where is the head pointing?

For each variable listed above, determine its interpretation.

Ans: At step $k = 0$:

- Cells $-1, 0, 1$ are all $\square$: $T_{-1,\square,0}$, $T_{0,\square,0}$, $T_{1,\square,0}$ are all true.
- We are in state q_0: $Q_{q_0,0}$ is true.
- The head is pointing to cell 0: $H_{0,0}$ is true.



SAT is NP-complete!

Task: Trace M on an empty input ($n = 0$). For each step, note the following:

- What is written on cells $-1, 0, 1$?
- Which state is the TM in?
- Where is the head pointing?

For each variable listed above, determine its interpretation.

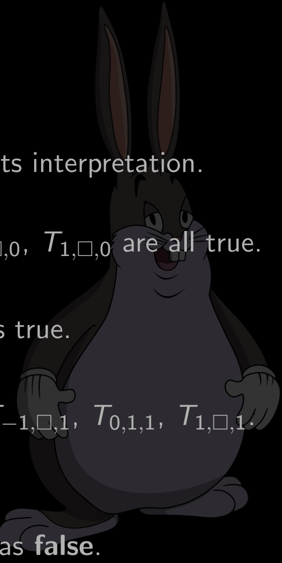
Ans: At step $k = 0$:

- Cells $-1, 0, 1$ are all $\square$: $T_{-1,\square,0}$, $T_{0,\square,0}$, $T_{1,\square,0}$ are all true.
- We are in state q_0: $Q_{q_0,0}$ is true.
- The head is pointing to cell 0: $H_{0,0}$ is true.

At step $k = 1$:

- Cells $-1, 1$ are $\square$, while cell 0 is 1: $T_{-1,\square,1}$, $T_{0,1,1}$, $T_{1,\square,1}$.
- We are in state q_{halt}: $Q_{q_{\text{halt}},1}$.
- The head is pointing to cell 1: $H_{1,1}$.

Any variable not listed here is interpreted as **false**.



SAT is NP-complete!

Now, back to describing ϕ:

Define the Boolean expression ϕ to be the conjunction of the sub-expressions in the following table, for all $-p(n) \leq i \leq p(n)$ and $0 \leq k \leq p(n)$:

Expression	Conditions	Interpretation	How many?
$T_{i,j,0}$	Tape cell *i* initially contains symbol *j*	Initial contents of the tape. For *i* > *n*-1 and *i* < 0, outside of the actual input I, the initial symbol is the special default/blank symbol.	O(*p*(*n*))
$Q_{s,0}$		Initial state of *M*.	1
$H_{0,0}$		Initial position of read/write head.	1
$\neg T_{i,j,k} \vee \neg T_{i,j',k}$	$j \neq j'$	At most one symbol per tape cell.	O(*p*(*n*)^2)
$\bigvee_{j\in\Sigma} T_{i,j,k}$		At least one symbol per tape cell.	O(*p*(*n*)^2)
$T_{i,j,k} \wedge T_{i,j',k+1} \to H_{i,k}$	$j \neq j'$	Tape remains unchanged unless written.	O(*p*(*n*)^2)
$\neg Q_{q,k} \vee \neg Q_{q',k}$	$q \neq q'$	Only one state at a time.	O(*p*(*n*))
$\neg H_{i,k} \vee \neg H_{i',k}$	$i \neq i'$	Only one head position at a time.	O(*p*(*n*)^3)
$(H_{i,k} \wedge Q_{q,k} \wedge T_{i,\sigma,k}) \to \bigvee_{((q,\sigma),(q',\sigma',d))\in\delta}(H_{i+d,\,k+1} \wedge Q_{q',\,k+1} \wedge T_{i,\,\sigma',\,k+1})$	$k<p(n)$	Possible transitions at computation step *k* when head is at position *i*.	O(*p*(*n*)^2)
$\bigvee_{0\leq k\leq p(n)} Q_{f,k}$		Must finish in an accepting state, not later than in step *p*(*n*).	1

TL;DR: by choosing an assignment for this statement, we are specifying an *execution path* that the NTM *M* takes on an input, and vice versa.