# Recursion and Decidability

Dr. Chuck Rocca
roccac@wcsu.edu
http://sites.wcsu.edu/roccac


WESTERN CONNECTICUT STATE UNIVERSITY
MACRICOSTAS SCHOOL OF ARTS & SCIENCES

# Table of Contents

# Table of Contents

# The Printing Machine

Define $q(w) = Q$ run on $w$ where

$Q =$ "On input $w$:

1. Construct the following Turing machine $P_w$:
   $P_w =$ "On any input:
   1. Erase the input.
   2. Write $w$ on the tape.
   3. Halt."
2. Output $\langle P_w \rangle$ and halt."

Thus $q(w) = \langle P_w \rangle$.

# The Machining Machine

For any portion $M$ of a Turing machine, define

$B =$ "On input $\langle M \rangle$, where $M$ is a portion of a TM:

1. Compute $q(\langle M \rangle) = P_{\langle M \rangle}$.
2. Combine this with $\langle M \rangle$ to make the TM $P_{\langle M \rangle} M$.
3. Print a description of $\left\langle P_{\langle M \rangle} M \right\rangle$ and halt"

# The SELF Machine

Finally, combining these we get

$SELF =$"On any input:

1. First Run $A = q(\langle B \rangle) = P_{\langle B \rangle}$ which prints $\langle B \rangle$.
2. Input the result, $\langle B \rangle$, into $B$.
3. B then computes $q(\langle B \rangle) = P_{\langle B \rangle} = A$ and combines it with the input $\langle B \rangle$ to form $\left\langle P_{\langle B \rangle} B \right\rangle = \langle AB \rangle = \langle SELF \rangle$.
4. B ends by printing $\langle SELF \rangle$ and halting."

# The SELF Machine

Finally, combining these we get

$SELF =$ "On any input:

1. First Run $A = q(\langle B \rangle) = P_{\langle B \rangle}$ which prints $\langle B \rangle$.
2. Input the result, $\langle B \rangle$, into $B$.
3. B then computes $q(\langle B \rangle) = P_{\langle B \rangle} = A$ and combines it with the input $\langle B \rangle$ to form $\left\langle P_{\langle B \rangle} B \right\rangle = \langle AB \rangle = \langle SELF \rangle$.
4. B ends by printing $\langle SELF \rangle$ and halting."

Compare this to something like the function:

*Print out two copies of the following, the second one in quotes: "$\langle STRING \rangle$"*

# The SELF Machine

Finally, combining these we get

$SELF =$ "On any input:

1. First Run $A = q(\langle B \rangle) = P_{\langle B \rangle}$ which prints $\langle B \rangle$.
2. Input the result, $\langle B \rangle$, into $B$.
3. B then computes $q(\langle B \rangle) = P_{\langle B \rangle} = A$ and combines it with the input $\langle B \rangle$ to form $\langle P_{\langle B \rangle} B \rangle = \langle AB \rangle = \langle SELF \rangle$.
4. B ends by printing $\langle SELF \rangle$ and halting."

Which gives output:

$\langle STRING \rangle$
"$\langle STRING \rangle$"

# The SELF Machine

Finally, combining these we get

$SELF =$ "On any input:

1. First Run $A = q(\langle B \rangle) = P_{\langle B \rangle}$ which prints $\langle B \rangle$.
2. Input the result, $\langle B \rangle$, into $B$.
3. B then computes $q(\langle B \rangle) = P_{\langle B \rangle} = A$ and combines it with the input $\langle B \rangle$ to form $\left\langle P_{\langle B \rangle} B \right\rangle = \langle AB \rangle = \langle SELF \rangle$.
4. B ends by printing $\langle SELF \rangle$ and halting."

But, if you feed the function itself:

*Print out two copies of the following, the second one in quotes:*
*"Print out two copies of the following, the second one in quotes:"*

# Table of Contents

# Recursion Theorem

**Theorem**

*Let $T$ be a Turing machine that computes a function $t : \Sigma^* \times \Sigma^* \to \Sigma^*$. There exists a Turing machine $R$ that computes a function $r : \Sigma^* \to \Sigma^*$, where for every $w$,*

$$r(w) = t(\langle R \rangle, w).$$

# Recursion Theorem

## Theorem

*Let $T$ be a Turing machine that computes a function $t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$. There exists a Turing machine $R$ that computes a function $r : \Sigma^* \rightarrow \Sigma^*$, where for every $w$,*

$$r(w) = t(\langle R \rangle, w).$$

The desired $R$ is constructed similar to $SELF$:

- $w \rightarrow A \rightarrow w \langle BT \rangle$ with $B$ as above and $T$ as given
- $w \langle BT \rangle \rightarrow B \rightarrow \langle ABT \rangle$ call this $\langle R \rangle$
- $\langle R, w \rangle \rightarrow T = t(\langle R \rangle, w)$

# Recursion Theorem

**Theorem**

*Let $T$ be a Turing machine that computes a function $t : \Sigma^* \times \Sigma^* \to \Sigma^*$. There exists a Turing machine $R$ that computes a function $r : \Sigma^* \to \Sigma^*$, where for every $w$,*

$$r(w) = t(\langle R \rangle, w).$$

The desired $R$ is constructed similar to *SELF*:

- $w \to A \to w \langle BT \rangle$ with $B$ as above and $T$ as given
- $w \langle BT \rangle \to B \to \langle ABT \rangle$ call this $\langle R \rangle$
- $\langle R, w \rangle \to T = t(\langle R \rangle, w)$

This is an extension of *SELF* that allows a machine to do more than just print with itself.

# *SELF* Revisited

Using the recursion theorem we can diagram SELF as

$$w \rightarrow R \rightarrow \langle R, w \rangle \rightarrow T \rightarrow \langle R \rangle$$

where $T$ is a machine that prints $\langle M \rangle$ on input $\langle M, w \rangle$

# $A_{TM}$ is Undecidable (again)

Recall

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM that accepts } w\},$$

and assume that $H$ is a TM that decides $A_{TM}$. Construct

$B =$ "On input $w$:

1. Obtain description $\langle B\rangle$ via the recursion theorem.
2. Run $H$ on $\langle B, w\rangle$.
3. If $H$ accepts, *reject*, else if $H$ rejects, *accept*."

Since $B$ does the opposite of $H$, we have constructed an undecidable machine.

# $MIN_{TM}$ is not Turing-Recognizable

Define

$$MIN_{TM} = \{\langle M \rangle \,|M \text{ is a TM with minimum length description}\}.$$

Assume TM $E$ can enumerate $MIN_{TM}$ and construct TM $C$ as follows

$C =$ "On input $w$:

1. Obtain description $\langle C \rangle$ via the recursion theorem.
2. Run $E$ until we get some machine $D$ with a longer description than $C$.
3. Simulate $D$ on input $w$."

TM $D$ must exists since there are infinitely many Turing machines. That $C$ can simulate $D$ but has a shorter description is a contradiction to the assumption $D \in MIN_{TM}$.

# Fixed Point Theorem

Let $t : \Sigma^* \to \Sigma^*$ be any computable function and define $F$ as follows

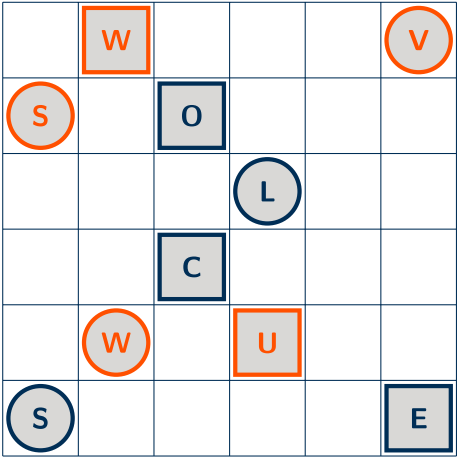
$F =$ "On input $w$:

1. Obtain description $\langle F \rangle$ via the recursion theorem.
2. Compute $G = t(\langle F \rangle)$.
3. Simulate $G$ on input $w$."

Clearly $\langle F \rangle \equiv \langle G \rangle = \langle t(F) \rangle$ because $F$ computes and then simulates $G$.

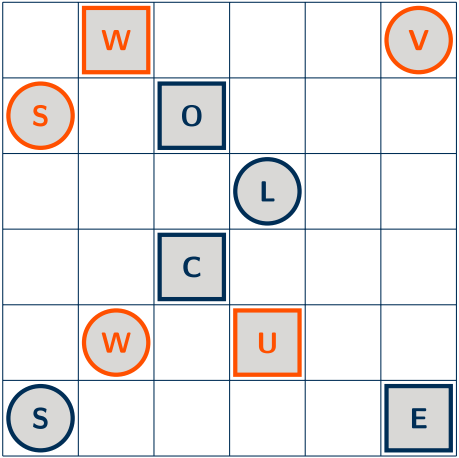
# Table of Contents

# Review of Quantifiers (Tarski's World)



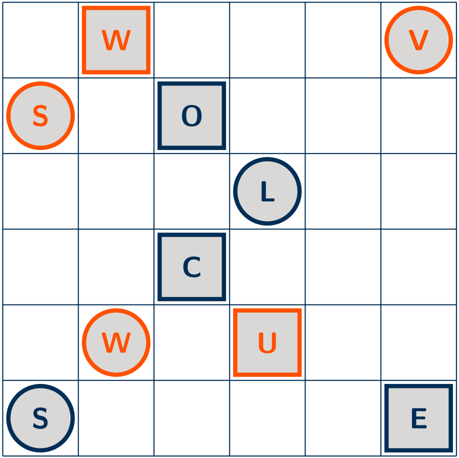

- $C(x) \equiv x$ is a circle

# Review of Quantifiers (Tarski's World)



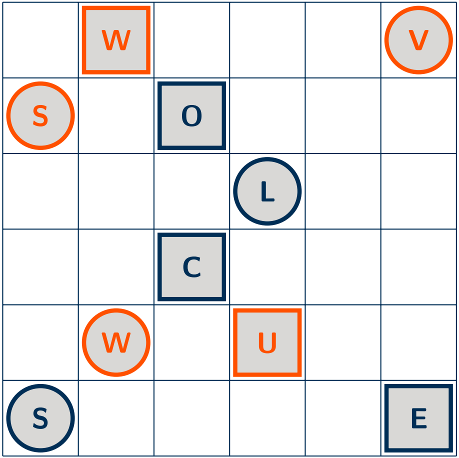

- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel

# Review of Quantifiers (Tarski's World)



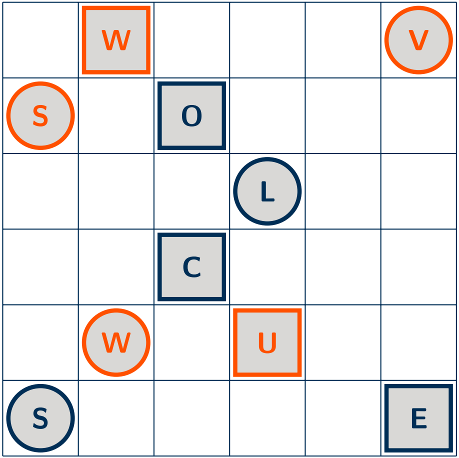

- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$

# Review of Quantifiers (Tarski's World)



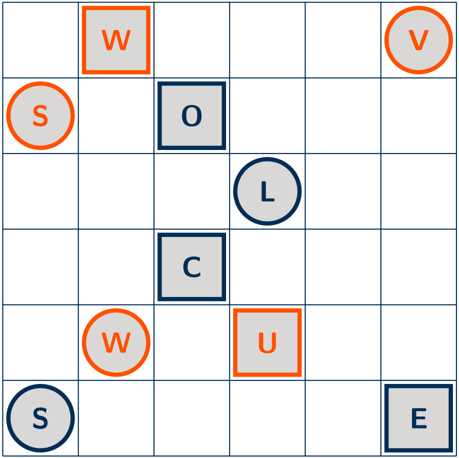

- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$
- $L(x, y) \equiv x$ is left of $y$

# Review of Quantifiers (Tarski's World)



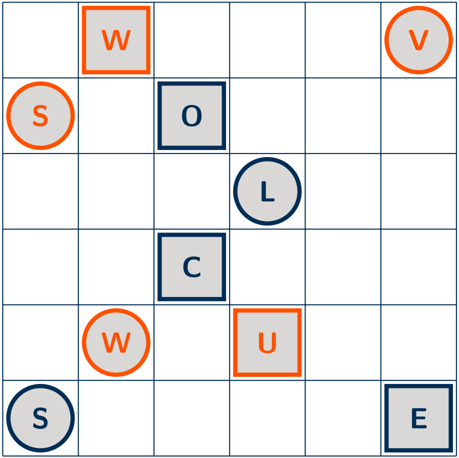

- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$
- $L(x, y) \equiv x$ is left of $y$
- $\forall x : \neg V(x) \rightarrow C(x)$

# Review of Quantifiers (Tarski's World)



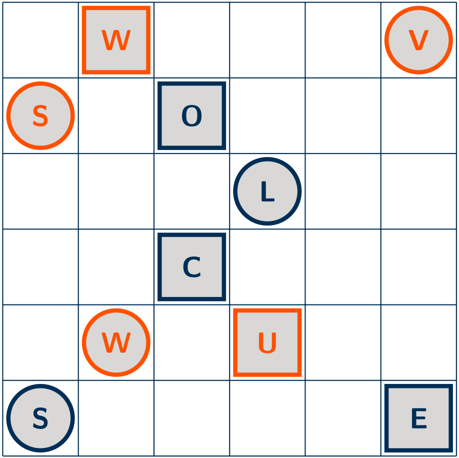

- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$
- $L(x, y) \equiv x$ is left of $y$
- $\forall x : \neg V(x) \to C(x)$
- $\exists x : \neg V(x) \wedge \neg C(x)$

# Review of Quantifiers (Tarski's World)



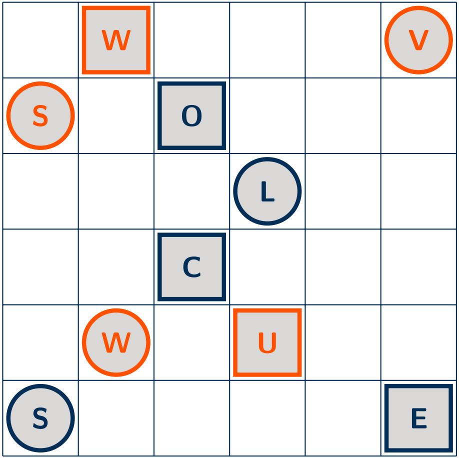

- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$
- $L(x, y) \equiv x$ is left of $y$
- $\forall x: \neg V(x) \rightarrow C(x)$
- $\exists x: \neg V(x) \wedge \neg C(x)$
- $\forall x \exists y: A(x, y) \vee A(y, x)$

# Review of Quantifiers (Tarski's World)




- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$
- $L(x, y) \equiv x$ is left of $y$
- $\forall x : \neg V(x) \rightarrow C(x)$
- $\exists x : \neg V(x) \wedge \neg C(x)$
- $\forall x \exists y : A(x, y) \vee A(y, x)$
- $\forall x \exists y : V(x) \wedge L(y, x)$

# Review of Quantifiers (Tarski's World)




- $C(x) \equiv x$ is a circle
- $V(x) \equiv x$ is a vowel
- $A(x, y) \equiv x$ is above $y$
- $L(x, y) \equiv x$ is left of $y$
- $\forall x : \neg V(x) \rightarrow C(x)$
- $\exists x : \neg V(x) \wedge \neg C(x)$
- $\forall x \exists y : A(x, y) \vee A(y, x)$
- $\forall x \exists y : V(x) \wedge L(y, x)$
- $\exists y \forall x : V(x) \wedge L(y, x)$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$
- $\exists x \forall y : G(x, y)$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$
- $\exists x \forall y : G(x, y)$ (False)

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$
- $\exists x \forall y : G(x, y)$ (False)
- $\neg(\exists x \forall y : G(x, y)) \equiv \forall x \exists y : \neg G(x, y)$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$
- $\exists x \forall y : G(x, y)$ (False)
- $\neg(\exists x \forall y : G(x, y)) \equiv \forall x \exists y : \neg G(x, y)$
- $\exists x : \neg G(x^2, x)$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$
- $\exists x \forall y : G(x, y)$ (False)
- $\neg(\exists x \forall y : G(x, y)) \equiv \forall x \exists y : \neg G(x, y)$
- $\exists x : \neg G(x^2, x)$
- $\forall x, y\, \exists n : E(xy, n)$

# Review of Quantifiers (Arithmetic)

Let $x, y, n \in \mathbb{N}$:

- $G(x, y) \equiv x > y$
- $E(x, y) \equiv x = y$
- $\forall y \exists x : G(x, y)$
- $\exists x \forall y : G(x, y)$ (False)
- $\neg(\exists x \forall y : G(x, y)) \equiv \forall x \exists y : \neg G(x, y)$
- $\exists x : \neg G(x^2, x)$
- $\forall x, y\, \exists n : E(xy, n)$
- $\forall q\, \exists p\, \forall x, y : G(p, q) \wedge (G(\{x, y\}, 1) \rightarrow \neg E(xy, p))$

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.
- Boolean Operators: *And*, $\wedge$, *or*, $\vee$, and *negation*, $\neg$

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.
- Boolean Operators: *And*, $\wedge$, *or*, $\vee$, and *negation*, $\neg$
- Predicate: $P(x_1, x_2, \ldots, x_l)$ is a "sentence" containing a finite number of variables

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.
- Boolean Operators: *And*, $\wedge$, *or*, $\vee$, and *negation*, $\neg$
- Predicate: $P(x_1, x_2, \ldots, x_l)$ is a "sentence" containing a finite number of variables
- Relation: $R(x_1, x_2, \ldots, x_l)$ is a predicate which also a *Boolean function*

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.
- Boolean Operators: *And*, $\wedge$, *or*, $\vee$, and *negation*, $\neg$
- Predicate: $P(x_1, x_2, \ldots, x_l)$ is a "sentence" containing a finite number of variables
- Relation: $R(x_1, x_2, \ldots, x_l)$ is a predicate which also a *Boolean function*
- Alphabet: $\Sigma = \{\wedge, \vee, \neg, (, ), R_1, R_2, \ldots, R_k\}$

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.
- Boolean Operators: *And*, $\wedge$, *or*, $\vee$, and *negation*, $\neg$
- Predicate: $P(x_1, x_2, \ldots, x_l)$ is a "sentence" containing a finite number of variables
- Relation: $R(x_1, x_2, \ldots, x_l)$ is a predicate which also a *Boolean function*
- Alphabet: $\Sigma = \{\wedge, \vee, \neg, (, ), R_1, R_2, \ldots, R_k\}$
- Formula: *Well-formed string* in $\Sigma^*$, i.e.

$$R_1(x_1) \wedge \neg R_2(x_2, x_3)$$

# Notation/Vocabulary

- Quantifiers: *For all*, $\forall$, and *there exists*, $\exists$.
- Boolean Operators: *And*, $\wedge$, *or*, $\vee$, and *negation*, $\neg$
- Predicate: $P(x_1, x_2, \ldots, x_l)$ is a "sentence" containing a finite number of variables
- Relation: $R(x_1, x_2, \ldots, x_l)$ is a predicate which also a *Boolean function*
- Alphabet: $\Sigma = \{\wedge, \vee, \neg, (, ), R_1, R_2, \ldots, R_k\}$
- Formula: *Well-formed string* in $\Sigma^*$, i.e.

$$R_1(x_1) \wedge \neg R_2(x_2, x_3)$$

- Statement: A formula with no free variables, i.e.

$$\forall x_1 \, \exists x_2, x_3 : R_1(x_1) \wedge \neg R_2(x_2, x_3)$$

# Equivalencies

- DeMorgan's Laws:

$$\neg(R_1 \land R_2) \equiv \neg R_1 \lor \neg R_2$$
$$\neg(R_1 \lor R_2) \equiv \neg R_1 \land \neg R_2$$

# Equivalencies

- DeMorgan's Laws:

$$\neg(R_1 \wedge R_2) \equiv \neg R_1 \vee \neg R_2$$
$$\neg(R_1 \vee R_2) \equiv \neg R_1 \wedge \neg R_2$$

- Negating Quantifiers:

$$\neg\left(\forall x_i : R(x_1, x_2, \ldots, x_l)\right) \equiv \left(\exists x_i : \neg R(x_1, x_2, \ldots, x_l)\right)$$
$$\neg\left(\exists x_i : R(x_1, x_2, \ldots, x_l)\right) \equiv \left(\forall x_i : \neg R(x_1, x_2, \ldots, x_l)\right)$$

# Equivalencies

- DeMorgan's Laws:

$$\neg(R_1 \wedge R_2) \equiv \neg R_1 \vee \neg R_2$$
$$\neg(R_1 \vee R_2) \equiv \neg R_1 \wedge \neg R_2$$

- Negating Quantifiers:

$$\neg\left(\forall x_i : R(x_1, x_2, \ldots, x_l)\right) \equiv \left(\exists x_i : \neg R(x_1, x_2, \ldots, x_l)\right)$$
$$\neg\left(\exists x_i : R(x_1, x_2, \ldots, x_l)\right) \equiv \left(\forall x_i : \neg R(x_1, x_2, \ldots, x_l)\right)$$

- Implications:

$$R_1 \rightarrow R_2 \equiv \neg(R_1 \wedge \neg R_2) \equiv \neg R_1 \vee R_2$$

# $\mathrm{Th}(\mathbb{N}, +)$ is Decidable

**Theorem**

*Let $M = (\mathbb{N}, +)$ be the* **model** *consisting of all relations over $\mathbb{N}$ using the operation $+$ and $\mathrm{Th}(\mathbb{N}, +)$ be the* **theory of M**, *that is the set of all true statements in $M$. Then, $\mathrm{Th}(\mathbb{N}, +)$ is decidable.*

- $\psi$ is a well-formed formula in $M = (\mathbb{N}, +)$ with $l$ free variables

# $\mathrm{Th}(\mathbb{N},+)$ is Decidable

**Theorem**

*Let $M = (\mathbb{N},+)$ be the* **model** *consisting of all relations over $\mathbb{N}$ using the operation $+$ and $\mathrm{Th}(\mathbb{N},+)$ be the* **theory of M**, *that is the set of all true statements in $M$. Then, $\mathrm{Th}(\mathbb{N},+)$ is decidable.*

- $\psi$ is a well-formed formula in $M = (\mathbb{N},+)$ with $l$ free variables
- $Q_l x_l : \psi$ has $l-1$ free variables

# $\mathrm{Th}(\mathbb{N},+)$ is Decidable

**Theorem**

*Let $M = (\mathbb{N},+)$ be the* **model** *consisting of all relations over $\mathbb{N}$ using the operation $+$ and $\mathrm{Th}(\mathbb{N},+)$ be the* **theory of M**, *that is the set of all true statements in $M$. Then, $\mathrm{Th}(\mathbb{N},+)$ is decidable.*

- $\psi$ is a well-formed formula in $M = (\mathbb{N},+)$ with $l$ free variables
- $Q_l x_l : \psi$ has $l-1$ free variables
- $Q_{l-1} x_{l-1} Q_l x_l : \psi$ has $l-2$ free variables

# $\mathrm{Th}(\mathbb{N}, +)$ is Decidable

**Theorem**

*Let $M = (\mathbb{N}, +)$ be the* **model** *consisting of all relations over $\mathbb{N}$ using the operation $+$ and $\mathrm{Th}(\mathbb{N}, +)$ be the* **theory of M***, that is the set of all true statements in $M$. Then, $\mathrm{Th}(\mathbb{N}, +)$ is decidable.*

- $\psi$ is a well-formed formula in $M = (\mathbb{N}, +)$ with $l$ free variables
- $Q_l x_l : \psi$ has $l - 1$ free variables
- $Q_{l-1} x_{l-1} Q_l x_l : \psi$ has $l - 2$ free variables
- $\vdots$

# $\mathrm{Th}(\mathbb{N}, +)$ is Decidable

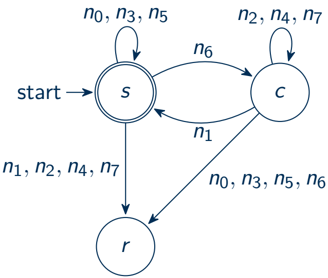
**Theorem**

*Let $M = (\mathbb{N}, +)$ be the* **model** *consisting of all relations over $\mathbb{N}$ using the operation $+$ and $\mathrm{Th}(\mathbb{N}, +)$ be the* **theory of M***, that is the set of all true statements in $M$. Then, $\mathrm{Th}(\mathbb{N}, +)$ is decidable.*

- $\psi$ is a well-formed formula in $M = (\mathbb{N}, +)$ with $l$ free variables
- $Q_l x_l : \psi$ has $l - 1$ free variables
- $Q_{l-1} x_{l-1} Q_l x_l : \psi$ has $l - 2$ free variables
- $\vdots$
- $Q_0 x_0 q_1 x_1 \cdots Q_l x_l : \psi$ has 0 free variables

# Problem 1.32: $a_1 + a_2 = a_3$ is Decidable

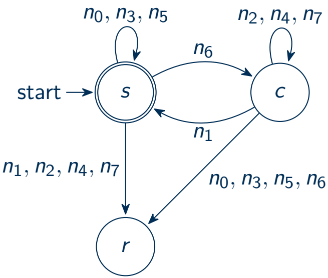
Machine $A_3$:




- $n_0 = [0,0,0]$
- $n_1 = [0,0,1]$
- $n_2 = [0,1,0]$
- $n_3 = [0,1,1]$
- $n_4 = [1,0,0]$
- $n_5 = [1,0,1]$
- $n_6 = [1,1,0]$
- $n_7 = [1,1,1]$

# Problem 1.32: $a_1 + a_2 = a_3$ is Decidable

Machine $A_3$:



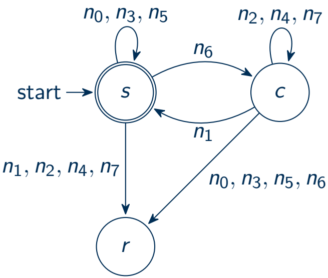

- $n_0 = [0, 0, 0]$
- $n_1 = [0, 0, 1]$
- $n_2 = [0, 1, 0]$
- $n_3 = [0, 1, 1]$
- $n_4 = [1, 0, 0]$
- $n_5 = [1, 0, 1]$
- $n_6 = [1, 1, 0]$
- $n_7 = [1, 1, 1]$

Binary Sums:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$

# Problem 1.32: $a_1 + a_2 = a_3$ is Decidable

Machine $A_3$:




- $n_0 = [0, 0, 0]$
- $n_1 = [0, 0, 1]$
- $n_2 = [0, 1, 0]$
- $n_3 = [0, 1, 1]$
- $n_4 = [1, 0, 0]$
- $n_5 = [1, 0, 1]$
- $n_6 = [1, 1, 0]$
- $n_7 = [1, 1, 1]$

Binary Sums:

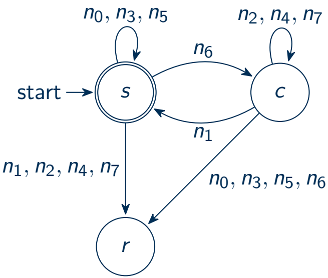
- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$

Example 1: Read from right to left,

$$\begin{matrix} n_1 & n_7 & n_6 & n_3 \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \end{matrix}$$

# Problem 1.32: $a_1 + a_2 = a_3$ is Decidable

Machine $A_3$:




- $n_0 = [0, 0, 0]$
- $n_1 = [0, 0, 1]$
- $n_2 = [0, 1, 0]$
- $n_3 = [0, 1, 1]$
- $n_4 = [1, 0, 0]$
- $n_5 = [1, 0, 1]$
- $n_6 = [1, 1, 0]$
- $n_7 = [1, 1, 1]$

Binary Sums:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$

Example 2: Read from right to left,

$$\begin{matrix} n_1 & n_6 & n_7 & n_3 \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \end{matrix}$$

# The Sentence $\exists a_1 \, \exists a_2 \, \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$

# The Sentence $\exists a_1 \exists a_2 \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$

# The Sentence $\exists a_1 \exists a_2 \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$
  Machine $A_2$:

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \xrightarrow{\varepsilon} \begin{bmatrix} 0 & b_{11} & b_{12} \\ 0 & b_{21} & b_{22} \\ z_0 & z_1 & z_2 \end{bmatrix} \longrightarrow \text{Machine } A_3$$

# The Sentence $\exists a_1 \, \exists a_2 \, \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
  Machine $A_2$:

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \xrightarrow{\varepsilon} \begin{bmatrix} 0 & b_{11} & b_{12} \\ 0 & b_{21} & b_{22} \\ z_0 & z_1 & z_2 \end{bmatrix} \longrightarrow \text{Machine } A_3$$

# The Sentence $\exists a_1 \exists a_2 \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
- $\exists a_2 \exists a_3 : \psi$

# The Sentence $\exists a_1 \exists a_2 \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
- $\exists a_2 \exists a_3 : \psi$
  Machine $A_1$:

$$\begin{bmatrix} b_{11} \end{bmatrix} \xrightarrow{\varepsilon} \begin{bmatrix} 0 & b_{11} \\ z_0 & z_1 \end{bmatrix} \longrightarrow \text{Machine } A_2$$

# The Sentence $\exists a_1 \, \exists a_2 \, \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
- $\exists a_2 \, \exists a_3 : \psi$ is decidable by $A_1$
  Machine $A_1$:

$$\begin{bmatrix} b_{11} \end{bmatrix} \xrightarrow{\varepsilon} \begin{bmatrix} 0 & b_{11} \\ z_0 & z_1 \end{bmatrix} \longrightarrow \text{Machine } A_2$$

# The Sentence $\exists a_1 \exists a_2 \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
- $\exists a_2 \exists a_3 : \psi$ is decidable by $A_1$
- $\exists a_1 \exists a_2 \exists a_3 : \psi$

# The Sentence $\exists a_1 \, \exists a_2 \, \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
- $\exists a_2 \, \exists a_3 : \psi$ is decidable by $A_1$
- $\exists a_1 \, \exists a_2 \, \exists a_3 : \psi$
  Machine $A_0$:

$$[\quad] \xrightarrow{\varepsilon} [z_0] \longrightarrow \text{Machine } A_1$$

# The Sentence $\exists a_1\, \exists a_2\, \exists a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \equiv a_1 + a_2 = a_3$ is decidable by machine $A_3$
- $\exists a_3 : \psi$ is decidable by $A_2$
- $\exists a_2\, \exists a_3 : \psi$ is decidable by $A_1$
- $\exists a_1\, \exists a_2\, \exists a_3 : \psi$ is decidable by $A_0$
  Machine $A_0$:

$$[\quad] \xrightarrow{\varepsilon} [z_0] \longrightarrow \text{Machine } A_1$$

# $\mathrm{Th}(\mathbb{N}, +)$ is Decidable

**Theorem**

*Let $M = (\mathbb{N}, +)$ be the* **model** *consisting of all relations over $\mathbb{N}$ using the operation $+$ and $\mathrm{Th}(\mathbb{N}, +)$ be the* **theory of M**, *that is the set of all true statements in $M$. Then, $\mathrm{Th}(\mathbb{N}, +)$ is decidable.*

- $\psi$ is a well-formed formula in $M = (\mathbb{N}, +)$ with $l$ free variables
- $Q_l x_l : \psi$ has $l - 1$ free variables
- $Q_{l-1} x_{l-1} Q_l x_l : \psi$ has $l - 2$ free variables
- $\vdots$
- $Q_0 x_0 q_1 x_1 \cdots Q_l x_l : \psi$ has 0 free variables

# Table of Contents

# $\mathrm{Th}(\mathbb{N}, +, \times)$ is Undecidable

**Theorem**

*Let* $M = (\mathbb{N}, +, \times)$ *be the* model *consisting of all relations over* $\mathbb{N}$ *using the operations* $+$ *and* $\times$*, and* $\mathrm{Th}(\mathbb{N}, +, \times)$ *be the* theory of M*, that is the set of all true statement in* $M$*. Then,* $\mathrm{Th}(\mathbb{N}, +, \times)$ *is undecidable.*

# Reducing $A_{TM}$ (again)

**Lemma**

*Given machine $M$ and word $w$ construct a formula $\phi_{M,w} \in (\mathbb{N}, +, \times)$ such that the sentence $\exists x : \phi_{M,w}$ is true if and only if $\langle M, w \rangle \in A_{TM}$.*

# Reducing $A_{TM}$ (again)

**Lemma**

*Given machine $M$ and word $w$ construct a formula $\phi_{M,w} \in (\mathbb{N}, +, \times)$ such that the sentence $\exists x : \phi_{M,w}$ is true if and only if $\langle M, w \rangle \in A_{TM}$.*

- Let the $x$ be potential computation histories for $M$ run on $w$ encoded as integers.

# Reducing $A_{TM}$ (again)

**Lemma**

*Given machine $M$ and word $w$ construct a formula $\phi_{M,w} \in (\mathbb{N}, +, \times)$ such that the sentence $\exists x : \phi_{M,w}$ is true if and only if $\langle M, w \rangle \in A_{TM}$.*

- Let the $x$ be potential computation histories for $M$ run on $w$ encoded as integers.
- Construct $\phi_{M,w} \in (\mathbb{N}, +\times)$ to check if $x$ is a valid computation history.

# Reducing $A_{TM}$ (again)

**Lemma**

*Given machine $M$ and word $w$ construct a formula $\phi_{M,w} \in (\mathbb{N}, +, \times)$ such that the sentence $\exists x : \phi_{M,w}$ is true if and only if $\langle M, w \rangle \in A_{TM}$.*

- Let the $x$ be potential computation histories for $M$ run on $w$ encoded as integers.
- Construct $\phi_{M,w} \in (\mathbb{N}, +\times)$ to check if $x$ is a valid computation history.
- $\therefore \exists x : \phi_{M,w}$ is true if and only if $\langle M, w \rangle \in A_{TM}$.

# $\mathrm{Th}(\mathbb{N}, +, \times)$ is Undecidable

**Theorem**

*Let $M = (\mathbb{N}, +, \times)$ be the* model *consisting of all relations over $\mathbb{N}$ using the operations $+$ and $\times$, and $\mathrm{Th}(\mathbb{N}, +, \times)$ be the* theory of M, *that is the set of all true statement in $M$. Then, $\mathrm{Th}(\mathbb{N}, +, \times)$ is undecidable.*

- The previous lemma is a mapping reduction from $A_{TM}$ to $\mathrm{Th}(\mathbb{N}, +, \times)$

# $\mathrm{Th}(\mathbb{N}, +, \times)$ is Undecidable

**Theorem**

*Let $M = (\mathbb{N}, +, \times)$ be the* model *consisting of all relations over $\mathbb{N}$ using the operations $+$ and $\times$, and $\mathrm{Th}(\mathbb{N}, +, \times)$ be the* theory of M, *that is the set of all true statement in $M$. Then, $\mathrm{Th}(\mathbb{N}, +, \times)$ is undecidable.*

- The previous lemma is a mapping reduction from $A_{TM}$ to $\mathrm{Th}(\mathbb{N}, +, \times)$
- $A_{TM}$ is undecidable

# $\mathrm{Th}(\mathbb{N}, +, \times)$ is Undecidable

**Theorem**

*Let $M = (\mathbb{N}, +, \times)$ be the* model *consisting of all relations over $\mathbb{N}$ using the operations $+$ and $\times$, and $\mathrm{Th}(\mathbb{N}, +, \times)$ be the* theory of M, *that is the set of all true statement in $M$. Then, $\mathrm{Th}(\mathbb{N}, +, \times)$ is undecidable.*

- The previous lemma is a mapping reduction from $A_{TM}$ to $\mathrm{Th}(\mathbb{N}, +, \times)$
- $A_{TM}$ is undecidable
- $\therefore$ $\mathrm{Th}(\mathbb{N}, +, \times)$ is undecidable

# Provability

**Definition**

A statement $\phi$ is *provable* if there is a sequence of statements $S_i$ such that

$$S_1 \to S_2 \to \cdots \to S_l = \phi,$$

this sequence is called a *formal proof*, $\pi$, of $\phi$. Given some reasonable definition of proof:

1. Correctness of a proof is decidable.
2. Systems of proofs are *sound*, i.e. if a statement is provable then it is true.

# More on $\mathrm{Th}(\mathbb{N}, +, \times)$

**Theorem**

*The collection of provable statements in* $\mathrm{Th}(\mathbb{N}, +, \times)$ *is Turing-recognizable.*

# More on $\mathrm{Th}(\mathbb{N}, +, \times)$

**Theorem**

*The collection of provable statements in* $\mathrm{Th}(\mathbb{N}, +, \times)$ *is Turing-recognizable.*

Just test every possible proof.

# More on $\mathrm{Th}(\mathbb{N}, +, \times)$

**Theorem**

*The collection of provable statements in* $\mathrm{Th}(\mathbb{N}, +, \times)$ *is Turing-recognizable.*

**Theorem**

*There exists statements in* $\mathrm{Th}(\mathbb{N}, +, \times)$ *which are true and not provable.*

# More on $\mathrm{Th}(\mathbb{N}, +, \times)$

**Theorem**

*The collection of provable statements in* $\mathrm{Th}(\mathbb{N}, +, \times)$ *is Turing-recognizable.*

**Theorem**

*There exists statements in* $\mathrm{Th}(\mathbb{N}, +, \times)$ *which are true and not provable.*

If every statement is provable, then every statement is decidable. Thus, there exists an unprovable statement.

# An Unprovable Sentence

$S =$ "On any input:

1. Obtain a copy of $\langle S \rangle$ via the recursion theorem.
2. Construct the sentence

$$\psi = \neg \exists c : [\phi_{S,0}] \text{ (i.e. } S \text{ doesn't accept 0)}$$

   using previous techniques.
3. Run the proof recognizer on $\psi$, if it accepts, *accept*."

# An Unprovable Sentence

$S =$ "On any input:

1. Obtain a copy of $\langle S \rangle$ via the recursion theorem.
2. Construct the sentence

   $$\psi = \neg \exists c : [\phi_{S,0}] \text{ (i.e. } S \text{ doesn't accept 0)}$$

   using previous techniques.
3. Run the proof recognizer on $\psi$, if it accepts, *accept*."

- If $S$ finds a proof, then $S$ accepts 0 and $\psi$ is false and so not provable, thus a contradiction.

# An Unprovable Sentence

$S =$ "On any input:

1. Obtain a copy of $\langle S \rangle$ via the recursion theorem.
2. Construct the sentence

$$\psi = \neg \exists c : [\phi_{S,0}] \text{ (i.e. } S \text{ doesn't accept 0)}$$

   using previous techniques.
3. Run the proof recognizer on $\psi$, if it accepts, *accept*."

- If $S$ finds a proof, then $S$ accepts 0 and $\psi$ is false and so not provable, thus a contradiction.
- $\therefore$ $S$ fails to find a proof, $S$ will not accept 0, and $\psi$ is true.

# Table of Contents

# Next Class

- Turing Reducibility

# Next Class

- Turing Reducibility
- Information

# Recursion and Decidability

Dr. Chuck Rocca
roccac@wcsu.edu
http://sites.wcsu.edu/roccac


WESTERN CONNECTICUT STATE UNIVERSITY
MACRICOSTAS SCHOOL OF ARTS & SCIENCES