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 $\langle P_{\langle M \rangle} M \rangle$ 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 $\langle P_{\langle B \rangle} B \rangle = \langle AB \rangle = \langle SELF \rangle$.
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 $\left\langle P_{\langle B \rangle} B \right\rangle = \langle AB \rangle = \langle SELF \rangle$.
4. B ends by printing $\langle SELF \rangle$ and halting."

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 $\langle P_{\langle B \rangle} B \rangle = \langle AB \rangle = \langle SELF \rangle$.
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

$$w \ ! \ R \ ! \ hR, wi \ ! \ T \ ! \ hRi$$

where T is a machine that prints hMi on input hM, wi

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

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

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 $t : \Sigma \quad ! \;\; \Sigma$ be any computable function and define F as follows

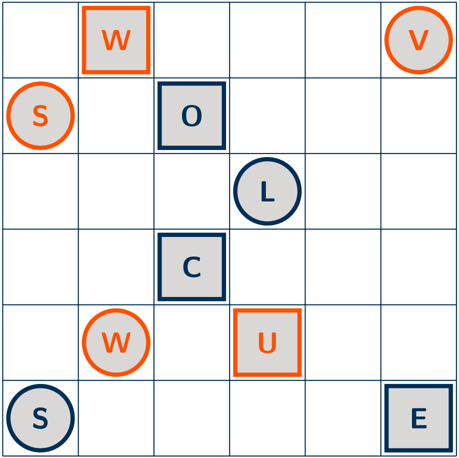
$F =$ "On input w:

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

Clearly $hFi \quad hGi = ht(F)i$ because F computes and then simulates G.

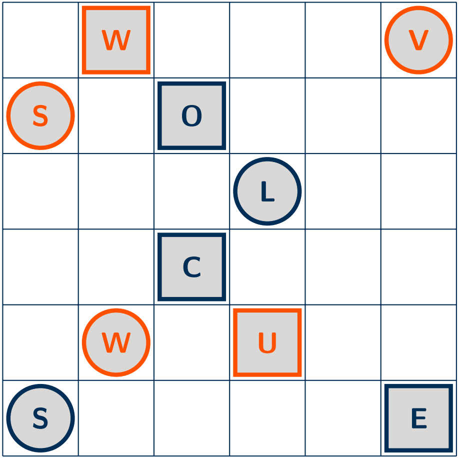
Table of Contents

Review of Quantifiers (Tarski's World)



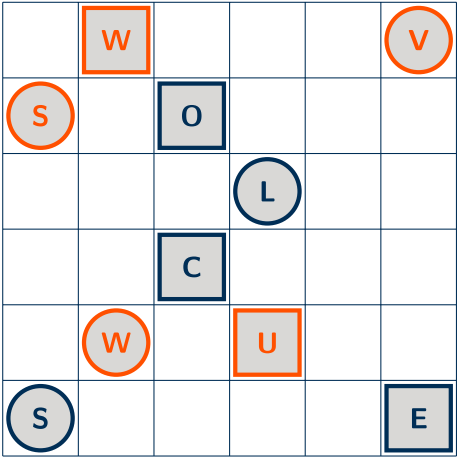
- $C(x)$ $\quad x$ is a circle

Review of Quantifiers (Tarski's World)



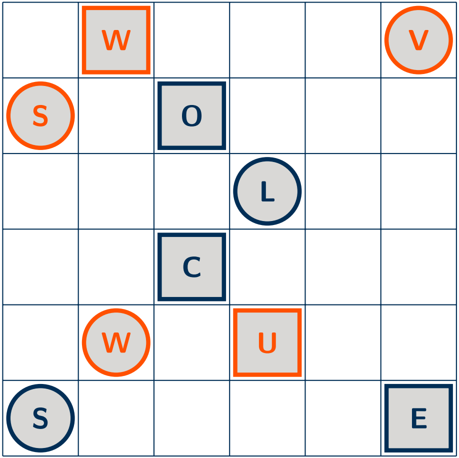
- $C(x)$ x is a circle
- $V(x)$ x is a vowel

Review of Quantifiers (Tarski's World)



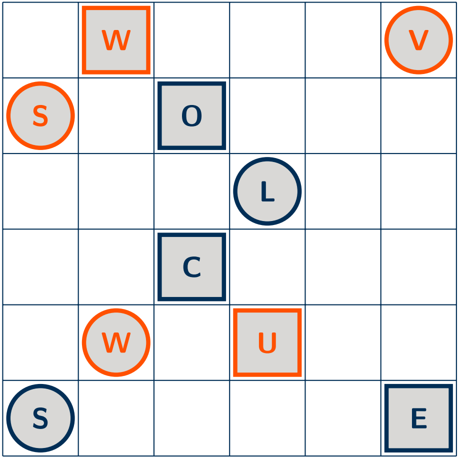
- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x, y)$ x is above y

Review of Quantifiers (Tarski's World)



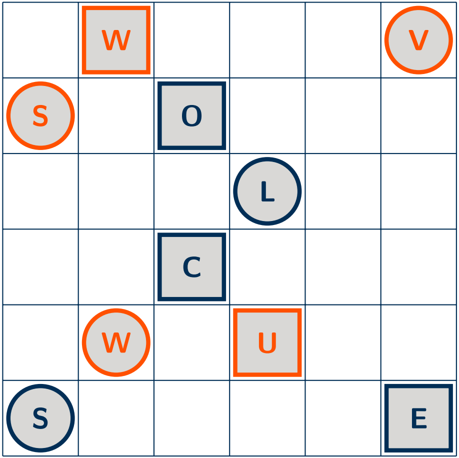
- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x,y)$ x is above y
- $L(x,y)$ x is left of y

Review of Quantifiers (Tarski's World)



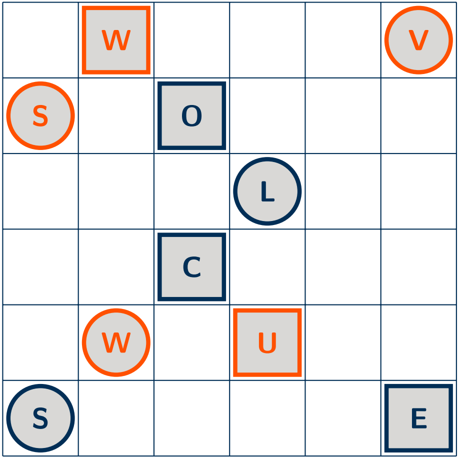
- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x, y)$ x is above y
- $L(x, y)$ x is left of y
- $8x : \ : V(x) \ ! \ C(x)$

Review of Quantifiers (Tarski's World)



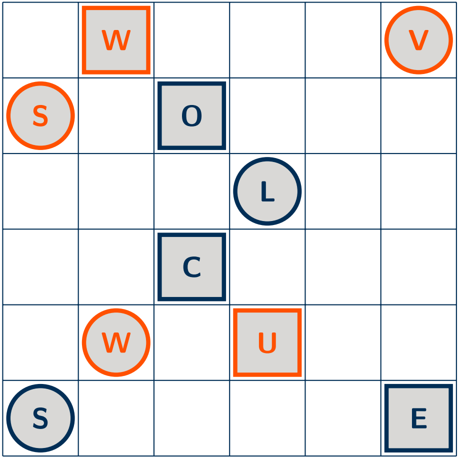
- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x, y)$ x is above y
- $L(x, y)$ x is left of y
- $8x : \; : V(x) \; ! \; C(x)$
- $9x : \; : V(x) \wedge : C(x)$

Review of Quantifiers (Tarski's World)



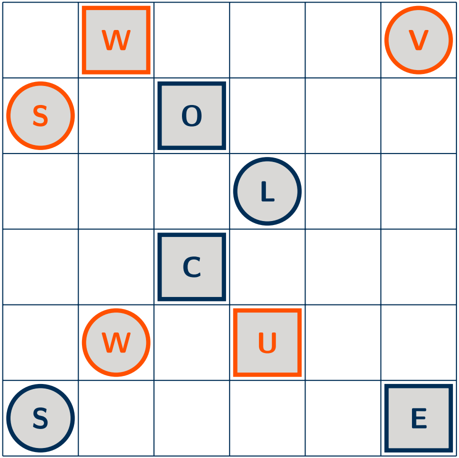
- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x, y)$ x is above y
- $L(x, y)$ x is left of y
- $8x :\ :V(x)\ !\ C(x)$
- $9x :\ :V(x) \wedge :C(x)$
- $8x9y :\ A(x, y) \_ A(y, x)$

Review of Quantifiers (Tarski's World)



- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x, y)$ x is above y
- $L(x, y)$ x is left of y
- $8x : \; :V(x) \; ! \; C(x)$
- $9x : \; :V(x) \wedge :C(x)$
- $8x9y : \; A(x, y) \_ A(y, x)$
- $8x9y : \; V(x) \wedge L(y, x)$

Review of Quantifiers (Tarski's World)



- $C(x)$ x is a circle
- $V(x)$ x is a vowel
- $A(x, y)$ x is above y
- $L(x, y)$ x is left of y
- $8x :\ \ : V(x)\ !\ \ C(x)$
- $9x :\ \ : V(x) \wedge\ : C(x)$
- $8x9y :\ \ A(x, y) \_\ A(y, x)$
- $8x9y :\ \ V(x) \wedge L(y, x)$
- $9y8x :\ \ V(x) \wedge L(y, x)$

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

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

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

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

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

- $G(x, y) \quad x > y$
- $E(x, y) \quad x = y$
- $8y9x : G(x, y)$

Review of Quantifiers (Arithmetic)

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

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

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

- $G(x, y) \quad x > y$
- $E(x, y) \quad x = y$
- $8y9x : G(x, y)$
- $9x8y : G(x, y)$ (False)

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

- $G(x, y) \quad x > y$
- $E(x, y) \quad x = y$
- $8y9x : G(x, y)$
- $9x8y : G(x, y)$ (False)
- $:(9x8y : G(x, y)) \quad 8x9y : :G(x, y)$

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

- $G(x, y) \quad x > y$
- $E(x, y) \quad x = y$
- $8y9x : G(x, y)$
- $9x8y : G(x, y)$ (False)
- $:(9x8y : G(x, y)) \quad 8x9y : :G(x, y)$
- $9x : :G(x^2, x)$

Review of Quantifiers (Arithmetic)

Let x, y, n 2 N:

- $G(x, y) \quad x > y$
- $E(x, y) \quad x = y$
- $8y9x : G(x, y)$
- $9x8y : G(x, y)$ (False)
- $: (9x8y : G(x, y)) \quad 8x9y : : G(x, y)$
- $9x : : G(x^2, x)$
- $8x, y\, 9n : E(xy, n)$

Review of Quantifiers (Arithmetic)

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

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

Notation/Vocabulary

- Quantifiers: *For all*, *8*, and *there exists*, *9*.

Notation/Vocabulary

- Quantifiers: *For all*, *8*, and *there exists*, *9*.
- Boolean Operators: *And*, ^, *or*, _, and *negation*, *:*

Notation/Vocabulary

- Quantifiers: *For all*, *8*, and *there exists*, *9*.
- Boolean Operators: *And*, ^, *or*, _, and *negation*, *:*
- Predicate: $P(x_1, x_2, \ldots, x_l)$ is a "sentence" containing a finite number of variables

Notation/Vocabulary

- Quantifiers: *For all*, *8*, and *there exists*, *9*.
- Boolean Operators: *And*, ^, *or*, _, and *negation*, *:*
- 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*, *8*, and *there exists*, *9*.
- Boolean Operators: *And*, ^, *or*, _, and *negation*, *:*
- 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: $\Sigma = f\,\hat{}\,, \_, :, (,), R_1, R_2, \ldots, R_k g$

Notation/Vocabulary

- Quantifiers: *For all*, *8*, and *there exists*, *9*.
- Boolean Operators: *And*, ^, *or*, _, and *negation*, *:*
- 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: $\Sigma = f$ ^, _, :, (,), $R_1, R_2, \ldots, R_k g$
- Formula: *Well-formed string* in Σ , i.e.

$$R_1(x_1) \text{ ^ } : R_2(x_2, x_3)$$

Notation/Vocabulary

- Quantifiers: *For all*, *8*, and *there exists*, *9*.
- Boolean Operators: *And*, ^, *or*, _, and *negation*, *:*
- 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: $\Sigma = f\,^\wedge, \_, :, (,), R_1, R_2, \ldots, R_k g$
- Formula: *Well-formed string* in Σ , i.e.

$$R_1(x_1)\ ^\wedge\ :R_2(x_2, x_3)$$

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

$$8x_1\, 9x_2, x_3 : R_1(x_1)\ ^\wedge\ :R_2(x_2, x_3)$$

Equivalencies

- DeMorgan's Laws:

$$: (R_1 \wedge R_2) \qquad : R_1 \_ : R_2$$
$$: (R_1 \_ R_2) \qquad : R_1 \wedge : R_2$$

Equivalencies

- DeMorgan's Laws:

$$:(R_1 \wedge R_2) \qquad :R_1 \_ \; :R_2$$
$$:(R_1 \_ R_2) \qquad :R_1 \wedge \; :R_2$$

- Negating Quantifiers:

$$:\,(8x_i : R(x_1, x_2, \ldots, x_l)) \qquad (9x_i : \; :R(x_1, x_2, \ldots, x_l))$$
$$:\,(9x_i : R(x_1, x_2, \ldots, x_l)) \qquad (8x_i : \; :R(x_1, x_2, \ldots, x_l))$$

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:

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

- Implications:

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

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

Theorem

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

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

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

Theorem

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

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

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

Theorem

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

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

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

Theorem

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

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

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

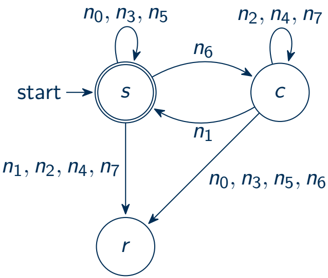
Theorem

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

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

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

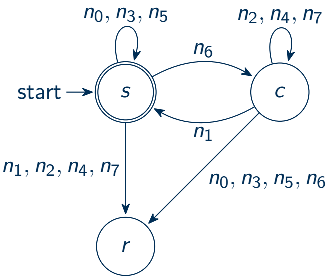
Machine A_3:



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

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

Machine A_3:



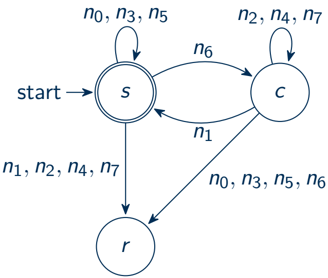
- $n_0 = [0, 0, 0]$
- $n_1 = [0, 0, 1]$
- $n_2 = [0, 1, 0]$
- $n_3 = [0, 1, 1]$
- $n_4 = [1, 0, 0]$
- $n_5 = [1, 0, 1]$
- $n_6 = [1, 1, 0]$
- $n_7 = [1, 1, 1]$

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:

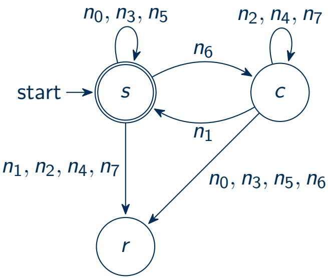
- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$

Example 1: Read from right to left,

$$\overset{n_1}{\begin{bmatrix}0\\0\\1\end{bmatrix}} \quad \overset{n_7}{\begin{bmatrix}1\\1\\1\end{bmatrix}} \quad \overset{n_6}{\begin{bmatrix}1\\1\\0\end{bmatrix}} \quad \overset{n_3}{\begin{bmatrix}0\\1\\1\end{bmatrix}}$$

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,

$$\underset{n_1}{\begin{bmatrix}0\\0\\1\end{bmatrix}} \quad \underset{n_6}{\begin{bmatrix}1\\1\\0\end{bmatrix}} \quad \underset{n_7}{\begin{bmatrix}1\\1\\1\end{bmatrix}} \quad \underset{n_3}{\begin{bmatrix}0\\1\\1\end{bmatrix}}$$

The Sentence $9a_1\, 9a_2\, 9a_3 : a_1 + a_2 = a_3$ is Decidable

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

The Sentence $9a_1\, 9a_2\, 9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$

The Sentence $9a_1\,9a_2\,9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$
  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 $9a_1\,9a_2\,9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ 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 $9a_1 \, 9a_2 \, 9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ is decidable by A_2
- $9a_2 \, 9a_3 : \psi$

The Sentence $9a_1\, 9a_2\, 9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ is decidable by A_2
- $9a_2\, 9a_3 : \psi$
  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 $9a_1\, 9a_2\, 9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ is decidable by A_2
- $9a_2\, 9a_3 : \psi$ 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 $9a_1\,9a_2\,9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ is decidable by A_2
- $9a_2\,9a_3 : \psi$ is decidable by A_1
- $9a_1\,9a_2\,9a_3 : \psi$

The Sentence $9a_1\,9a_2\,9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ is decidable by A_2
- $9a_2\,9a_3 : \psi$ is decidable by A_1
- $9a_1\,9a_2\,9a_3 : \psi$
  Machine A_0:

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

The Sentence $9a_1\,9a_2\,9a_3 : a_1 + a_2 = a_3$ is Decidable

- $\psi \quad\quad a_1 + a_2 = a_3$ is decidable by machine A_3
- $9a_3 : \psi$ is decidable by A_2
- $9a_2\,9a_3 : \psi$ is decidable by A_1
- $9a_1\,9a_2\,9a_3 : \psi$ is decidable by A_0
  Machine A_0:

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

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

Theorem

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

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

Table of Contents

$\mathrm{Th}(\mathsf{N}, +, \quad)$ is Undecidable

Theorem

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

Reducing A_{TM} (again)

Lemma

Given machine M and word w construct a formula $\phi_{M,w} \in (\mathbb{N}, +, \ )$ 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 $\phi_{M,w} \in (\mathbb{N}, +, \ )$ 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 $\phi_{M,w} \in (\mathbb{N}, +, \ )$ 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 $\phi_{M,w} \in (\mathbb{N}, + \ )$ to check if x is a valid computation history.

Reducing A_{TM} (again)

Lemma

Given machine M and word w construct a formula $\phi_{M,w}$ 2 $(\mathtt{N}, +, \ )$ such that the sentence 9$x : \phi_{M,w}$ is true if and only if hM, wi 2 A_{TM}.

- Let the x be potential computation histories for M run on w encoded as integers.
- Construct $\phi_{M,w}$ 2 $(\mathtt{N}, + \ )$ to check if x is a valid computation history.
-) 9$x : \phi_{M,w}$ is true if and only if hM, wi 2 A_{TM}.

$\mathrm{Th}(\mathsf{N}, +, \ )$ is Undecidable

Theorem

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

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

$\mathrm{Th}(\mathtt{N}, +, \ )$ is Undecidable

Theorem

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

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

$\mathrm{Th}(\mathsf{N}, +, \quad)$ is Undecidable

Theorem

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

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

Provability

Definition

A statement ϕ is *provable* if there is a sequence of statements S_i such that

$$S_1 \; ! \; S_2 \; ! \qquad ! \; S_l = \phi,$$

this sequence is called a *formal proof*, π, of ϕ. 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 $\mathrm{Th}(\mathsf{N}, +, \quad)$

Theorem

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

More on $\mathrm{Th}(\mathsf{N}, +, \quad)$

Theorem

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

Just test every possible proof.

More on $\mathrm{Th}(\mathsf{N}, +, \quad)$

Theorem

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

Theorem

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

More on $\mathrm{Th}(\mathsf{N}, +, \ )$

Theorem

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

Theorem

There exists statements in $\mathrm{Th}(\mathsf{N}, +, \ )$ *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 hSi via the recursion theorem.
2. Construct the sentence

$$\psi = \ :9c : [\phi_{S,0}] \ \text{(i.e. } S \text{ doesn't accept 0)}$$

   using previous techniques.

3. Run the proof recognizer on ψ, 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 ψ, if it accepts, *accept*."

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

An Unprovable Sentence

$S =$ "On any input:

1. Obtain a copy of hSi via the recursion theorem.
2. Construct the sentence

$$\psi = \, :9c : [\phi_{S,0}] \text{ (i.e. } S \text{ doesn't accept 0)}$$

   using previous techniques.

3. Run the proof recognizer on ψ, if it accepts, *accept*."

- If S finds a proof, then S accepts 0 and ψ is false and so not provable, thus a contradiction.
-) S fails to find a proof, S will not accept 0, and ψ 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