Final Review

CSE 311 Summer 2025
Lecture 24

Announcements

- HW7 Feedback and resubmission is out
  - Due **Thursday at 11:59pm—no late submissions!**

- Remember to submit course feedback by Friday at 11:59 pm to earn an additional E
  - See the post on Ed for more information

- Our Final is this Thursday (8/21) and Friday (8/22) in section and class!
  - Exam logistics and practice exams are posted on the "Exams" page of the course website

Problem 1: Regex/CFGs

Regex

Let $\Sigma = \{1, 2, 3, 4\}$ and let L be the language containing all strings w where w is **nondecreasing**.

We define a string w as **nondecreasing** if for all k, the character at index k is greater than or equal to all the numbers to the left, meaning the number at index k is greater than or equal to each of the numbers at indices 0 to $k-1$.

Some example **nondecreasing** strings are:

- ε (this string vacuously meets the definition of nondecreasing)
- 13 $(3 \geq 1)$
- 11223334

An example on an invalid string (one not in the language) is 132, as $3 > 2$.

Regex

Write a regular expression that matches L. (No explanation required).

Regex

Write a regular expression that matches L. (No explanation required).

$$(1)^*(2)^*(3)^*(4)^*$$

Regex

Write a CFG that generates L.
Be sure to tell us which symbol is the start symbol; also include a sentence or two of explanation of how yo
CFG works.

Regex

Write a CFG that generates L.
Be sure to tell us which symbol is the start symbol; also include a sentence or two of explanation of how yo
CFG works.

$$\mathbf{S} \rightarrow \mathbf{ABCD} \mid \varepsilon$$
$$\mathbf{A} \rightarrow 1\mathbf{A} \mid \varepsilon$$
$$\mathbf{B} \rightarrow 2\mathbf{B} \mid \varepsilon$$
$$\mathbf{C} \rightarrow 3\mathbf{C} \mid \varepsilon$$
$$\mathbf{D} \rightarrow 4\mathbf{D} \mid \varepsilon$$

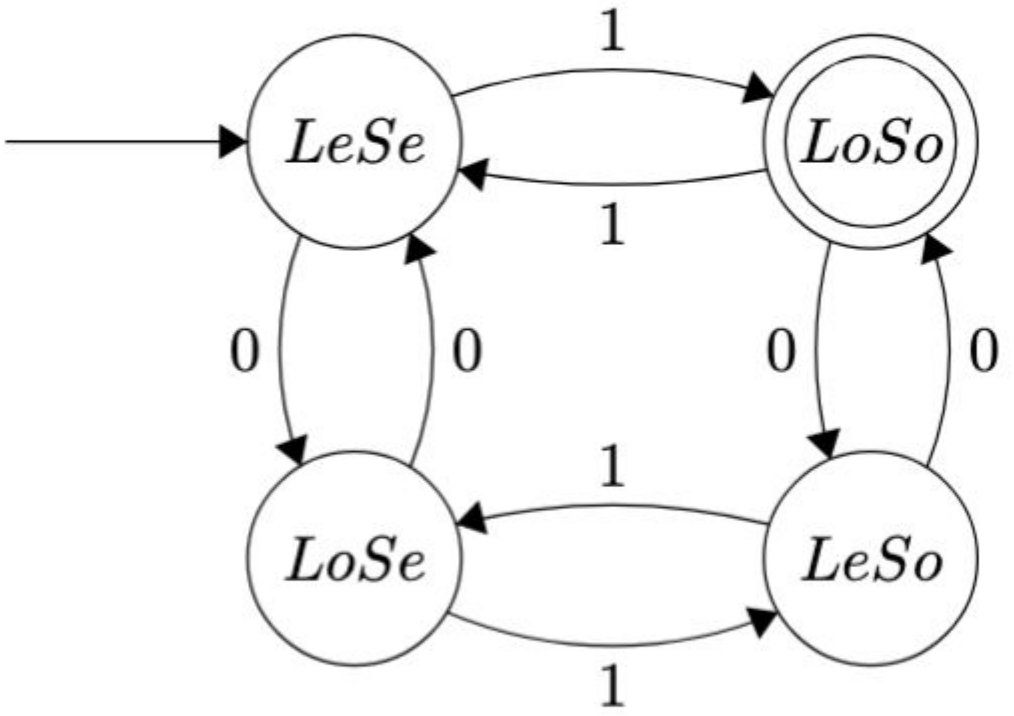
Problem 2: NFAs/DFAs

DFAs

Draw a **DFA** that accepts all binary strings that meet both of the following conditions: the string has an odd length **and** the digits of the string sum to an odd number.

DFAs

Draw a **DFA** that accepts all binary strings that meet both of the following conditions: the string has an odd length **and** the digits of the string sum to an odd number.

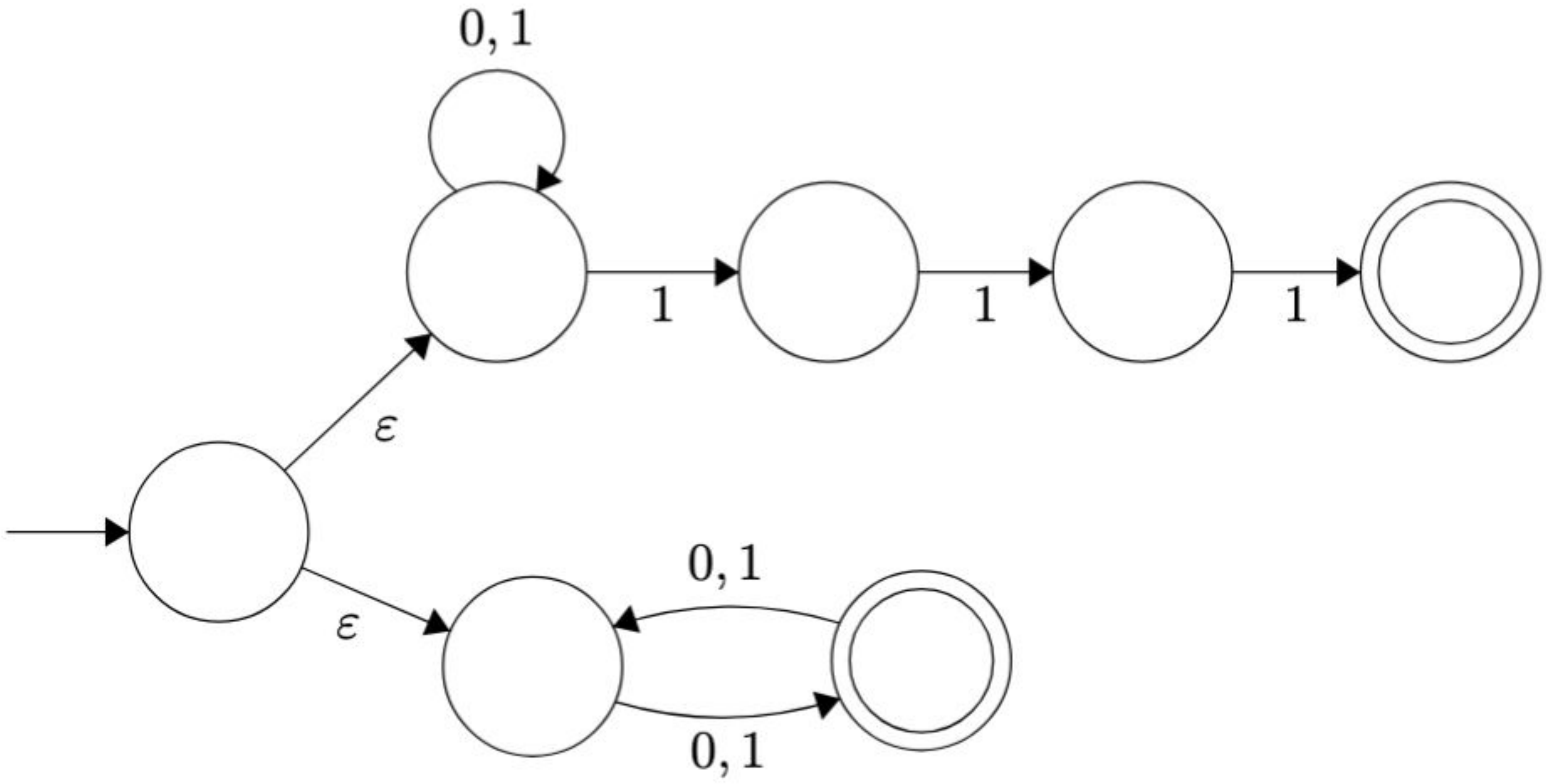


NFAs

Draw an **NFA** that accepts all binary strings with an odd length **or** end in 111 (or both).

NFAs

Draw an **NFA** that accepts all binary strings with an odd length **or** end in 111 (or both).



Problem 3: Structural Induction

Structural Induction

We define simple binary trees as the recursive set $\mathcal{B}$:

Basis Step: $\bullet \in \mathcal{B}$.
Recursive Step: If $L, R \in \mathcal{B}$, then $(L, \bullet, R) \in \mathcal{B}$.

Define the following functions on simple binary trees:

$$\texttt{edges}(t) = \begin{cases} 0 & \text{if } t = \bullet \\ 2 + \texttt{edges}(L) + \texttt{edges}(R) & \text{if } t = (L, \bullet, R) \end{cases}$$

$$\texttt{degree}(t) = \begin{cases} 1 & \text{if } t = \bullet \\ 3 & \text{if } t = (L, \bullet, R) \end{cases}$$

$$\texttt{sum}(t) = \begin{cases} \texttt{degree}(t) & \text{if } t = \bullet \\ \texttt{degree}(t) + \texttt{sum}(L) + \texttt{sum}(R) & \text{if } t = (L, \bullet, R) \end{cases}$$

Prove that for all $t \in \mathcal{B}$, $\texttt{sum}(t) = 2 \cdot \texttt{edges}(t) + 1$.
You must use structural induction for this problem.

Structural Induction

Structural Induction

Let $P(t) :=$"$\texttt{sum}(t) = 2 \cdot \texttt{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Structural Induction

Let $P(t) :=$"$\texttt{sum}(t) = 2 \cdot \texttt{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\texttt{sum}(\bullet) = \texttt{degree}(\bullet) = 1$$

and

$$2 \cdot \texttt{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Structural Induction

Let $P(t)$:=“$\texttt{sum}(t) = 2 \cdot \texttt{edges}(t) + 1$”. We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\texttt{sum}(\bullet) = \texttt{degree}(\bullet) = 1$$

and

$$2 \cdot \texttt{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Structural Induction

Let $P(t)$:="$\texttt{sum}(t) = 2 \cdot \texttt{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\texttt{sum}(\bullet) = \texttt{degree}(\bullet) = 1$$

and

$$2 \cdot \texttt{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Structural Induction

Let $P(t) :=$"$\mathtt{sum}(t) = 2 \cdot \mathtt{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathtt{sum}(\bullet) = \mathtt{degree}(\bullet) = 1$$

and

$$2 \cdot \mathtt{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\mathtt{sum}(t) = \mathtt{degree}(t) + \mathtt{sum}(L) + \mathtt{sum}(R)$$

Structural Induction

Let $P(t) :=$ "$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned} \mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\ &= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \end{aligned}$$

Structural Induction

Let $P(t)$:="$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned} \mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\ &= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \\ &= 3 + 2 \cdot \mathsf{edges}(L) + 1 + 2 \cdot \mathsf{edges}(R) + 1 && \text{by IH} \end{aligned}$$

Structural Induction

Let $P(t)$:="$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned}
\mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + 2 \cdot \mathsf{edges}(L) + 1 + 2 \cdot \mathsf{edges}(R) + 1 && \text{by IH} \\
&= 5 + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R)
\end{aligned}$$

Structural Induction

Let $P(t)$:="$\texttt{sum}(t) = 2 \cdot \texttt{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\texttt{sum}(\bullet) = \texttt{degree}(\bullet) = 1$$

and

$$2 \cdot \texttt{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned}
\texttt{sum}(t) &= \texttt{degree}(t) + \texttt{sum}(L) + \texttt{sum}(R) \\
&= 3 + \texttt{sum}(L) + \texttt{sum}(R) \\
&= 3 + 2 \cdot \texttt{edges}(L) + 1 + 2 \cdot \texttt{edges}(R) + 1 && \text{by IH} \\
&= 5 + 2 \cdot \texttt{edges}(L) + 2 \cdot \texttt{edges}(R) \\
&= 1 + 2(2) + 2 \cdot \texttt{edges}(L) + 2 \cdot \texttt{edges}(R)
\end{aligned}$$

Structural Induction

Let $P(t) :=$"$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned}
\mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + 2 \cdot \mathsf{edges}(L) + 1 + 2 \cdot \mathsf{edges}(R) + 1 && \text{by IH} \\
&= 5 + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2) + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2 + \mathsf{edges}(L) + \mathsf{edges}(R))
\end{aligned}$$

Structural Induction

Let $P(t)$:="$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned}
\mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + 2 \cdot \mathsf{edges}(L) + 1 + 2 \cdot \mathsf{edges}(R) + 1 && \text{by IH} \\
&= 5 + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2) + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2 + \mathsf{edges}(L) + \mathsf{edges}(R)) \\
&= 1 + 2 \cdot \mathsf{edges}(t)
\end{aligned}$$

Structural Induction

Let $P(t)$:="$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned}
\mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + 2 \cdot \mathsf{edges}(L) + 1 + 2 \cdot \mathsf{edges}(R) + 1 && \text{by IH} \\
&= 5 + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2) + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2 + \mathsf{edges}(L) + \mathsf{edges}(R)) \\
&= 1 + 2 \cdot \mathsf{edges}(t)
\end{aligned}$$

This proves $P(t)$.

Structural Induction

Let $P(t) :=$"$\mathsf{sum}(t) = 2 \cdot \mathsf{edges}(t) + 1$". We will prove $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Basis Step: From the definitions,

$$\mathsf{sum}(\bullet) = \mathsf{degree}(\bullet) = 1$$

and

$$2 \cdot \mathsf{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary simple binary trees L, R.

Inductive Step: Let $t = (L, \bullet, R)$. We will show $P(t)$.

$$\begin{aligned}
\mathsf{sum}(t) &= \mathsf{degree}(t) + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + \mathsf{sum}(L) + \mathsf{sum}(R) \\
&= 3 + 2 \cdot \mathsf{edges}(L) + 1 + 2 \cdot \mathsf{edges}(R) + 1 && \text{by IH} \\
&= 5 + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2) + 2 \cdot \mathsf{edges}(L) + 2 \cdot \mathsf{edges}(R) \\
&= 1 + 2(2 + \mathsf{edges}(L) + \mathsf{edges}(R)) \\
&= 1 + 2 \cdot \mathsf{edges}(t)
\end{aligned}$$

This proves $P(t)$.

Conclusion: Therefore, $P(t)$ holds for all $t \in \mathcal{B}$ by structural induction.

Problem 4: Translations

Training Wheels

`PerfectSquare`(x) returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

`Divides`(x, y) returns true if and only if $x|y$.

`Even`(x) returns true if and only if x is even.

If x is even and x is a perfect square then $4|x$.

Training Wheels

$\texttt{PerfectSquare}(x)$ returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

$\texttt{Divides}(x, y)$ returns true if and only if $x|y$.

$\texttt{Even}(x)$ returns true if and only if x is even.

If x is even and x is a perfect square then $4|x$.

$$\forall x([\mathsf{Even}(x) \wedge \texttt{PerfectSquare}(x)] \rightarrow \texttt{Divides}(4, x))$$

Training Wheels

$\texttt{PerfectSquare}(x)$ returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

$\texttt{Divides}(x, y)$ returns true if and only if $x|y$.

$\texttt{Even}(x)$ returns true if and only if x is even.

Every even number is divisible by two different integers.

Training Wheels

$\texttt{PerfectSquare}(x)$ returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

$\texttt{Divides}(x, y)$ returns true if and only if $x|y$.

$\texttt{Even}(x)$ returns true if and only if x is even.

Every even number is divisible by two different integers.

$$\forall x \exists y \exists z (\texttt{Even}(x) \rightarrow [\texttt{Divides}(y, x) \land \texttt{Divides}(z, x) \land y \neq z])$$

Training Wheels

$\texttt{PerfectSquare}(x)$ returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

$\texttt{Divides}(x, y)$ returns true if and only if $x|y$.

$\mathsf{Even}(x)$ returns true if and only if x is even.

Negate the following sentence (leave your answer as symbols; negations must be applied to single predicates, but you do not need to reformat your answer to show domain restriction).

$$\forall x \forall y([\mathsf{Even}(x) \wedge \mathsf{Even}(y)] \rightarrow [\texttt{Divides}(x, y) \vee \texttt{Divides}(y, x)])$$

Training Wheels

$\texttt{PerfectSquare}(x)$ returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

$\texttt{Divides}(x, y)$ returns true if and only if $x | y$.

$\mathsf{Even}(x)$ returns true if and only if x is even.

Negate the following sentence (leave your answer as symbols; negations must be applied to single predicates, but you do not need to reformat your answer to show domain restriction).

$$\forall x \forall y ([\mathsf{Even}(x) \wedge \mathsf{Even}(y)] \rightarrow [\texttt{Divides}(x, y) \vee \texttt{Divides}(y, x)])$$

$$\exists x \exists y (\mathsf{Even}(x) \wedge \mathsf{Even}(y) \wedge \neg \texttt{Divides}(x, y) \wedge \neg \texttt{Divides}(y, x))$$

Training Wheels

$\texttt{PerfectSquare}(x)$ returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

$\texttt{Divides}(x, y)$ returns true if and only if $x|y$.

$\textsf{Even}(x)$ returns true if and only if x is even.

State **in English** the contrapositive of the **original** statement in the last part (not your answer). Your English contrapositive must use domain restriction.

$$\forall x \forall y([\textsf{Even}(x) \wedge \textsf{Even}(y)] \rightarrow [\texttt{Divides}(x, y) \vee \texttt{Divides}(y, x)])$$

Training Wheels

PerfectSquare(x) returns true if and only if x is a perfect square (that is $x = z^2$ for some $z \in \mathbb{Z}$).

Divides(x, y) returns true if and only if $x|y$.

Even(x) returns true if and only if x is even.

State **in English** the contrapositive of the **original** statement in the last part (not your answer). Your English contrapositive must use domain restriction.

$$\forall x \forall y([\text{Even}(x) \wedge \text{Even}(y)] \rightarrow [\texttt{Divides}(x, y) \vee \texttt{Divides}(y, x)])$$

For all x and y if x does not divide y and y does not divide x then at least one of x and y is not even.

Todo

Tonight:

- HW7 resubmissions are due Thursday night!
- Read the final logistics on the Exams page of the course website and post on the Ed board if you have any questions
- Submit course feedback if you haven't already
- Get enough sleep and good luck studying!