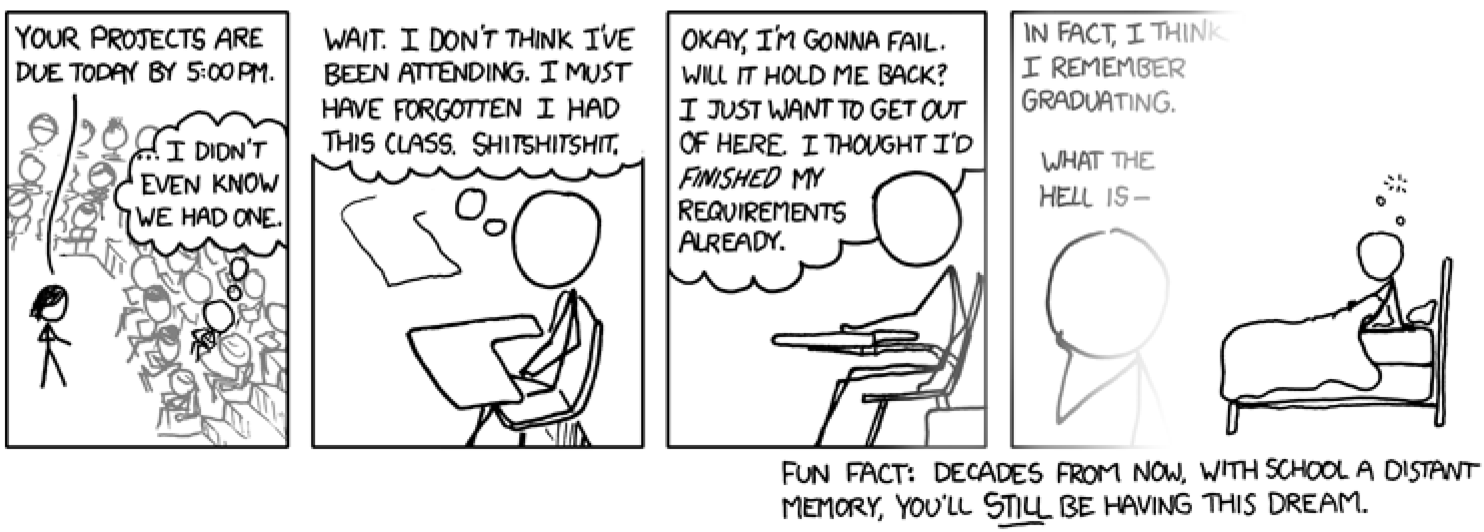
cse 311: foundations of computing

Spring 2015

Final Exam review session



irregular languages

Let $\Sigma = \{1, +, =\}$.

Define $L_1 = \{x + y = z \ : \ x, y, z \in \{1\}^* \text{ and } (x)_1 + (y)_1 = (z)_1\}$. That is, L_1 is all the strings of true statements of the form $x + y = z$ where $+$ and $=$ are characters and x, y, and z are strings of 1's interpreted as unary numbers. For example, "$111 + 11 = 11111$" $\in L_1$, because $3 + 2 = 5$. But, "$111 + 11 = 11$" $\notin L_1$, because $3 + 2 \neq 2$.

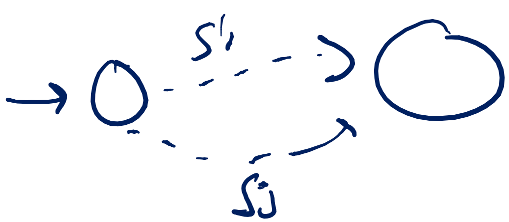
Prove that L_1 is not a regular language.

L_1 is regular $\Longrightarrow$ accepted by some DFA M

Prefixes $= \{1, 11, 111, 1111, \dots\}$ $\{1+, 11+, 111+, \dots\}$ $\{1+1, 1+11, 1+111, \dots\}$

$\exists\, s_i, s_j$ that lead to the same state of M

s_i - i 1's, s_j - j 1's, $i < j$



$\overbrace{s_i + 1 = 11\cdots1}$ $\in L_1$ $\Longrightarrow$ M does not recognize L

$s_j + 1 = \underbrace{11\cdots1}_{i+1 \text{ 1's}}$ $\notin L_1$ $\Longrightarrow$ L_1 is irregular

irregular languages

Let $\Sigma = \{0, 1, +, =\}$.

Define $L_2 = \{x + y = z \ : \ x, y, z \in \{0,1\}^* \text{ and } (x)_2 + (y)_2 = (z)_2\}$. That is, L_2 is the same idea as L_1, except the numbers are interpreted as binary instead of unary. For example, "$111 + 11 = 1000$" $\in L_2$, because $5 + 3 = 8$. But, "$111 + 11 = 11$" $\notin L_2$, because $7 + 3 \neq 3$.

Prove that L_2 is not a regular language.

Prefixes $= \{1, 11, 111, \ldots\}$

S_i, S_j end at same state

$S_i + 1 = (S_i + 1)_2 \in L_2$

$S_j + 1 = (S_i + 1)_2 \notin L_2$

$\Rightarrow$ L_2 is irregular.

Prefixes $=$ $\{0,1\}^*$

$S_i \neq S_j$ $\in$ Prefixes

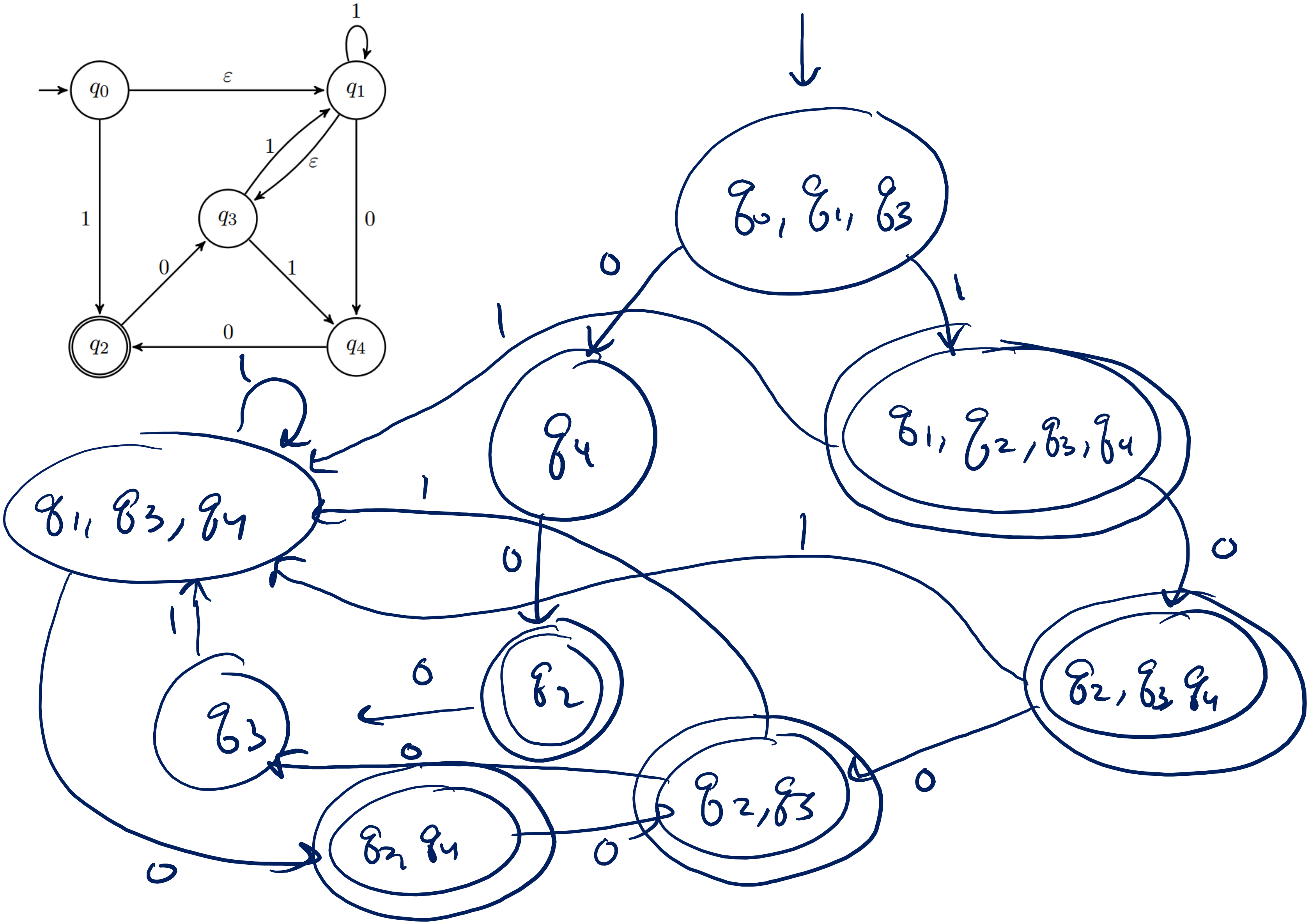
$i \neq j$

end at same state

$S_i + 0 = S_i \in L_2$

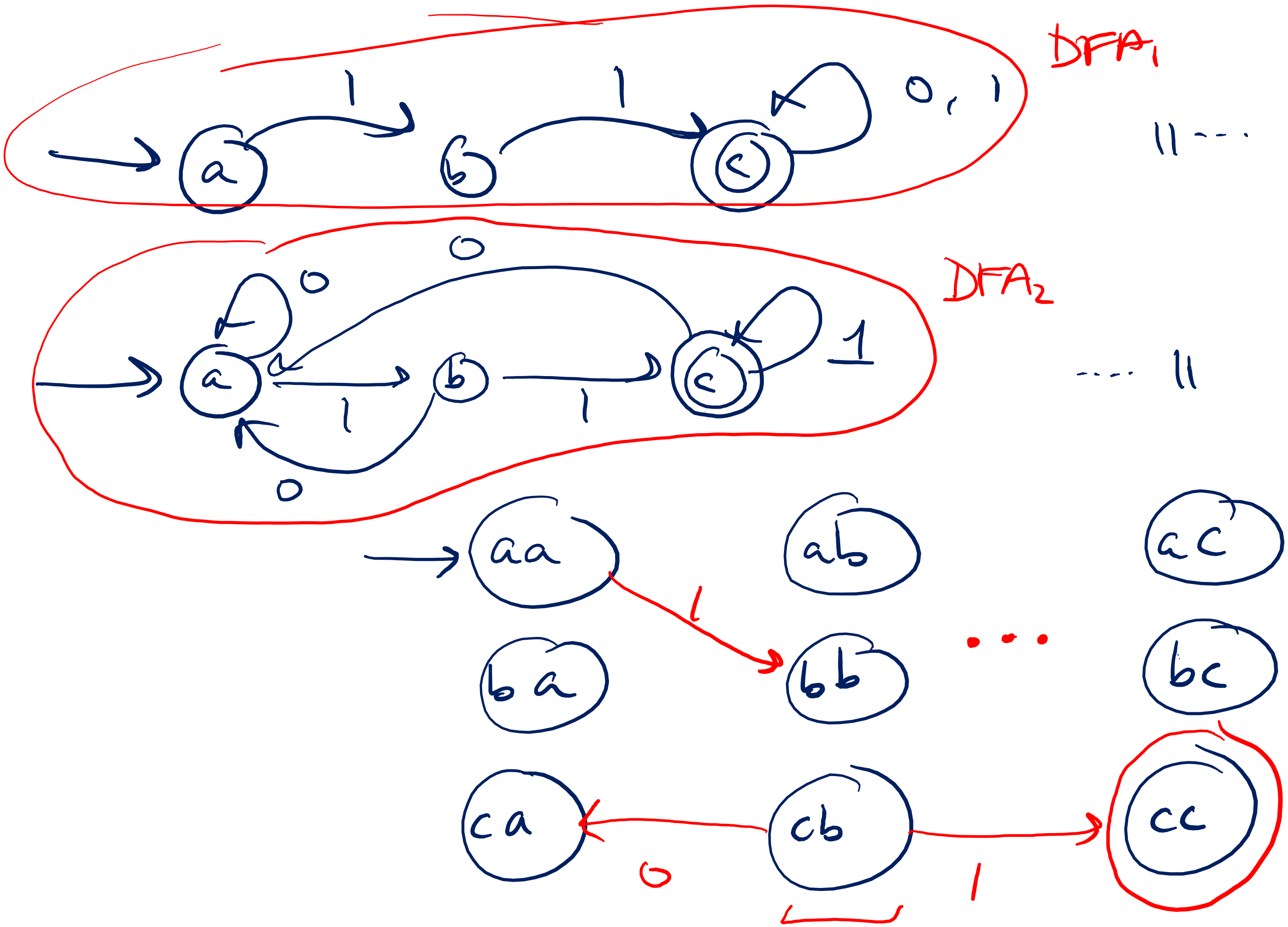
$S_j + 0 = S_i \notin L_2$

convert NFA to DFA



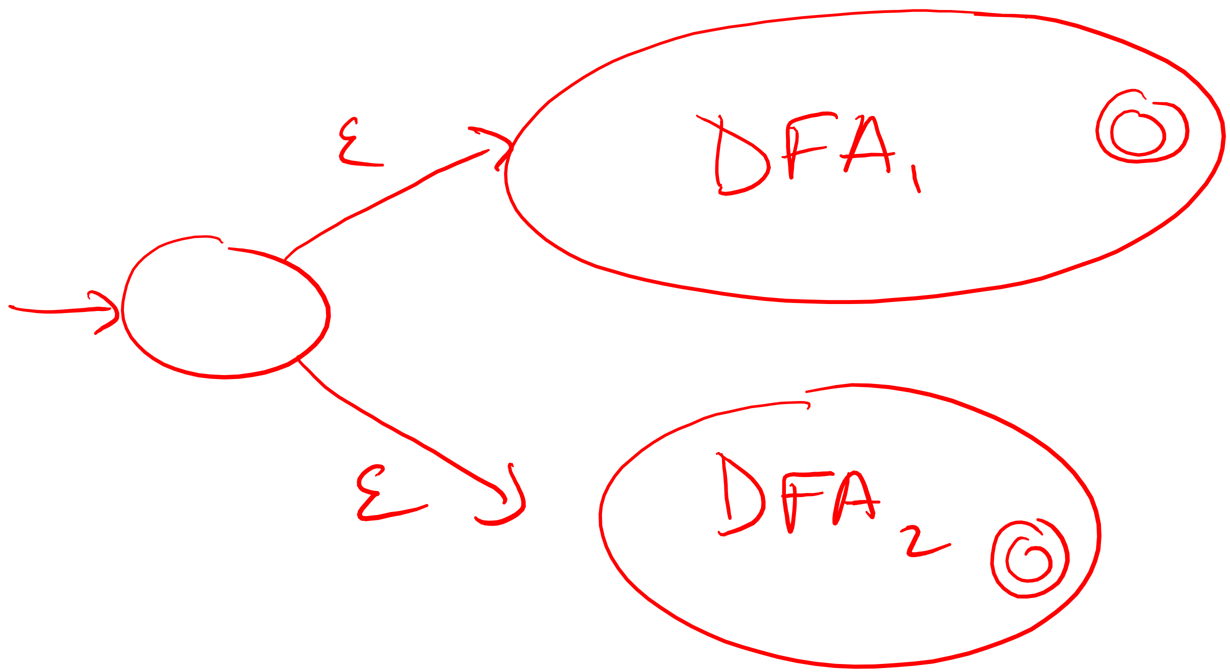
design an NFA

The set of all binary strings that start with two one's **and** end with two one's.



design an NFA

The set of all binary strings that start with two one's **or** end with two one's.



(see prev. slide)

design a CFG

The set of all binary strings that are of odd length and have 1 as their middle character.

$\Sigma = \{0, 1\}$

S, t s,t same length.

$S \rightarrow 1 \mid 0S0 \mid 1S1 \mid 0S1 \mid$

1S0

01111

$S \rightarrow 0S1 \rightarrow 01S11 \rightarrow 01111.$

example

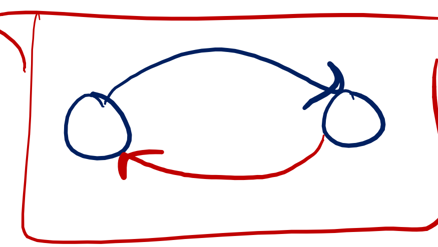
design a CFG

All binary strings that contain at least two 0's and at most two 1's.

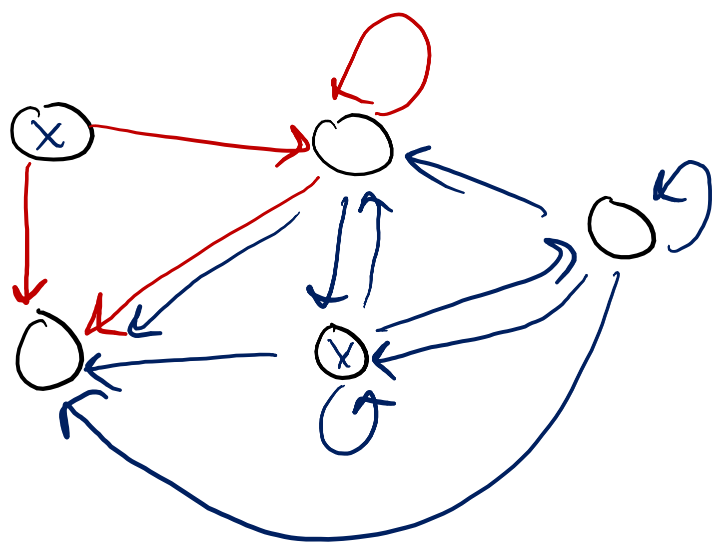
Let A be a set. Let R and S be transitive relations on A.

(a) Is $R \cup S$ necessarily transitive? Prove your answer.

(b) Is $R \cap S$ necessarily transitive? Prove your answer.



(a)



R trans.

S trans.

$R \cup S$ <u>not</u> trans.

(b) Yes. $(b,c), (a,b) \in R \cap S$

$\Rightarrow (a,c) \in R$ by trans, $(a,c) \in S$ by trans

$\Rightarrow (a,c) \in R \cap S$.

✓

A

We use $\mathbb{Z}^+$ to mean the set of positive integers.
Let $R \subseteq (\mathbb{Z}^+ \times \mathbb{Z}^+) \times (\mathbb{Z}^+ \times \mathbb{Z}^+)$ be the relation given by $((a,b),(c,d)) \in R$ if and only if $ad = bc$.
Prove that R is reflexive, symmetric and transitive.

Reflexive: $\underbrace{((a,b),(a,b)) \in R}_{ab=ba} \quad \forall (a,b) \in A$

Symmetric: $((a,b),(c,d)) \in R \implies ((c,d),(a,b)) \in R$

$\iff ad=bc \implies cb=da$ ✓

Trans: $((a,b),(c,d)) \in R$ and $((c,d),(e,f)) \in R$

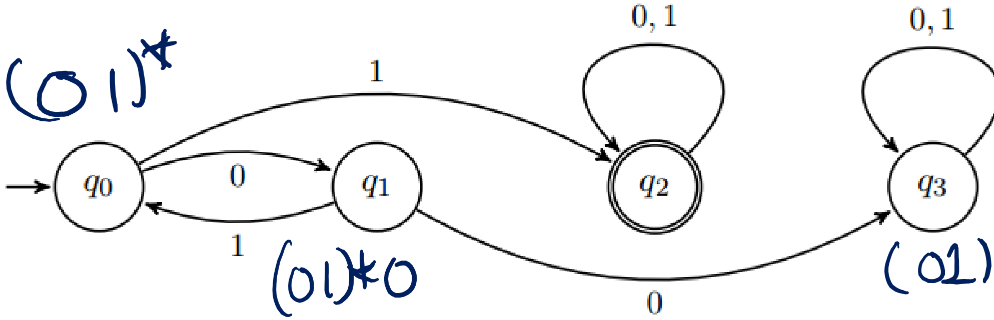
$\implies ((a,b),(e,f)) \in R \quad \frac{ad}{c}=b \quad \frac{cf}{d}=e$

$\iff ad=bc \wedge cf=de \implies af=be$

$(01)^*1(0|1)^*$

Consider the following DFA, M:

$(01)^*$



$(01)^*0$

$(01)^*00(0|1)^*$

For each of the four states, q, in M, write a regular expression that matches exactly the strings that end at the state q when starting from the initial state.

strong induction

Let $c > 0$ be an integer. The following recursive definition describes the running time of a recursive algorithm.

$$
\begin{aligned}
&T(0) = 0 \\
&T(n) \le c && \text{for all } n \le 20 \\
&T(n) = T\left(\left\lfloor \frac{3n}{4} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{5} \right\rfloor\right) + cn && \text{for all } n > 20
\end{aligned}
$$

Prove by strong induction that $T(n) \le 20cn$ for all $n \ge 0$.

$P(n) =$ "$T(n) \le 20cn$"

Base case: $T(0) = 0 \le 0 = 20c \cdot 0$.

For $n \le 20$, $T(n) \le c \le 20c \cdot n$.

Ind. hypoth: For some $n > 20$, $P(0), \ldots, P(n)$ hold.

Ind. step: $T(n+1) \overset{n>20}{=} T\left(\left\lfloor \frac{3(n+1)}{4} \right\rfloor\right) + T\left(\left\lfloor \frac{n+1}{5} \right\rfloor\right) + c(n+1)$

$\overset{IH}{\le} 20c\left\lfloor \frac{3(n+1)}{4} \right\rfloor + 20c\left\lfloor \frac{n+1}{5} \right\rfloor + c(n+1)$

$\le 20c(n+1)$.

Define f_n and g_n as follows for $n \in \mathbb{N}$:

$$f_0 = 1$$
$$f_1 = 5$$
$$f_2 = 10$$
$$f_n = 2f_{n-1} - 4f_{n-2} \text{ for } n \geq 3$$

$$g_0 = 1$$
$$g_1 = 5$$
$$g_2 = 10$$
$$g_3 = 0$$
$$g_4 = -40$$
$$g_n = 2g_{n-1} - 3g_{n-2} - 2g_{n-3} + 4g_{n-4} \text{ for } n \geq 5$$

Prove that $f_n = g_n$ for all $n \in \mathbb{N}$.

Give a recursive definition of the functions max and min so that $\max(a_1, a_2, \ldots, a_n)$ and $\min(a_1, a_2, \ldots, a_n)$ are the maximum and minimum of the n numbers $a_1, a_2, \ldots, a_n$, respectively.

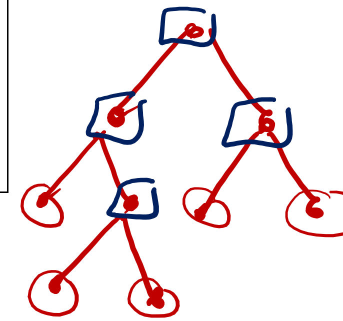
$$\max(a_1) = a_1$$

$$\max(a_1, \ldots, a_n) = \max(a_1, \max(a_2, \ldots, a_n)).$$

$$\max(a_1, a_2) = \max(a_1, \max(a_2)) = \max(a_1, a_2)$$

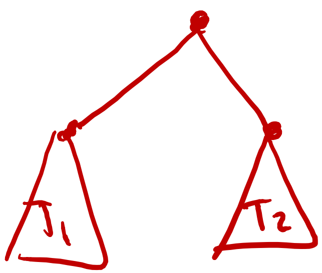
$$\max(a_1, a_2) = \begin{cases} a_1 & a_1 > a_2 \\ a_2 & a_1 \le a_2 \end{cases}$$

Use structural induction to show that $l(T)$, the number of leaves of a full binary tree T, is 1 more than $i(T)$, the number of internal vertices of T.

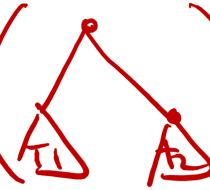


B: $T = \bullet$

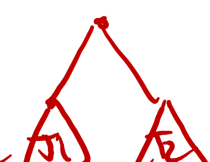
R:



B: $l(\cdot) = 1$

R: $l\Big($  $\Big) = l(T_1) + l(T_2)$

B: $i(\cdot) = 0$

R: $i\Big($  $\Big) = 1 + i(T_1) + i(T_2)$.

structural induction

Use structural induction to show that $l(T)$, the number of leaves of a full binary tree T, is 1 more than $i(T)$, the number of internal vertices of T.

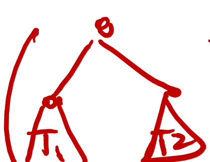
$$l(T) = 1 + i(T) \qquad \forall\, T$$

$P(T) =$ " $l(T) = 1 + i(T)$ "

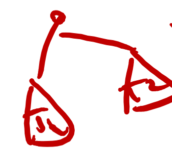
Pf by struct. ind.

Base: $l(\cdot) = 1 = 1 + 0 = 1 + i(\cdot)$.

Ind Hypoth: $P(T_1)$ and $P(T_2)$ hold

$l\Big($  $\Big) \overset{def}{=} l(T_1) + l(T_2) \overset{IH}{=} 2 + i(T_1) + i(T_2)$

$= 1 + (1 + i(T_1) + i(T_2))$

$\overset{def}{=} 1 + i\Big($  $\Big)$

By str. ind. $\forall T\ P(T)$.

$A = \mathbb{Z}$, $\phi \subseteq A$

short answer

- If a set A is countable, then every subset of A is countable.

  False. Subset could be finite.

- If a L is generated by a context-free grammar, then every subset of L is generated by a context-free grammar.

  False. $L = \Sigma^*$ has a CFG so any lang. is a subset L.

- There is a Java program that takes (P, x) as input and decides whether P halts on x within $2^{|x|}$ steps where $|x|$ is the length of x. True (just do it)

- If $P(x)$ is true for some x in the domain and false for others, and $Q(x)$ is always ~~true~~ false, then $\exists\, x\, \big(P(x) \to Q(x)\big)$ is true.

  True.

$0^{m+n} 1^{m} 0^{n} 1^{m+n}$

- If R and R' are relations on the same set and $R \subseteq R'$ then R' reflexive $\Rightarrow$ R reflexive

False.

- A relation R is anti-reflexive if $(x, x) \notin R$ for every x. If R is anti-reflexive, then R^2 is anti-reflexive.

$R = \{ (a,x), (x,a) \}$

$R^2 = \{ (a,a), (x,x) \}$ So False.

- If L is regular, then the language $\cancel{\{ xx : x \in L \}}$ has a CFG.

$\{ x_1 x_2 : x_1, x_2 \in \cancel{L} \}$

$S \to XX$

$X \to \dots$

- Every regular language is decidable.

True.