BILL, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

Regex: Closure Properties

Unfinished Business: Blow up for the $R(i, j, k)$ Method

Unfinished Business: Blow up for the $R(i, j, k)$ Method

Recall the $R(i, j, k)$ algorithm.

Unfinished Business: Blow up for the $R(i,j,k)$ Method

Recall the $R(i,j,k)$ algorithm.
For all $1 \le i,j \le n$:

$$R(i,j,0) = \begin{cases} \{\sigma : \delta(i,\sigma) = j\} & \text{if } i \neq j \ \} \\ \{\sigma : \delta(i,\sigma) = j\} \cup \{e\} & \text{if } i = j \ \} \end{cases} \tag{1}$$

Unfinished Business: Blow up for the $R(i,j,k)$ Method

Recall the $R(i,j,k)$ algorithm.
For all $1 \le i,j \le n$:

$$R(i,j,0) = \begin{cases} \{\sigma : \delta(i,\sigma) = j\} & \text{if } i \ne j \\ \{\sigma : \delta(i,\sigma) = j\} \cup \{e\} & \text{if } i = j \end{cases} \quad (1)$$

For all $1 \le i,j \le n$ and all $1 \le k \le n$

$$R(i,j,k) = R(i,j,k-1) \bigcup R(i,k,k-1)R(k,k,k-1)^* R(k,j,k-1)$$

Unfinished Business: Blow up for the $R(i,j,k)$ Method

Recall the $R(i,j,k)$ algorithm.
For all $1 \le i,j \le n$:

$$R(i,j,0) = \begin{cases} \{\sigma : \delta(i,\sigma) = j\} & \text{if } i \neq j \ \} \\ \{\sigma : \delta(i,\sigma) = j\} \cup \{e\} & \text{if } i = j \ \} \end{cases} \tag{1}$$

For all $1 \le i,j \le n$ and all $1 \le k \le n$

$$R(i,j,k) = R(i,j,k-1) \bigcup R(i,k,k-1)R(k,k,k-1)^* R(k,j,k-1)$$

If the DFA has n states, how long is the regex?

Length of $R(i, j, 0)$

Length of $R(i, j, 0)$

We assume $|\Sigma| = 2$.

Length of $R(i, j, 0)$

We assume $|\Sigma| = 2$.
For all $1 \leq i, j \leq n$:

$$R(i,j,0) = \begin{cases} \{\sigma : \delta(i,\sigma) = j\} & \text{if } i \neq j \ \} \\ \{\sigma : \delta(i,\sigma) = j\} \cup \{e\} & \text{if } i = j \ \} \end{cases} \tag{2}$$

Length of $R(i, j, 0)$

We assume $|\Sigma| = 2$.
For all $1 \leq i, j \leq n$:

$$R(i, j, 0) = \begin{cases} \{\sigma : \delta(i, \sigma) = j\} & \text{if } i \neq j \text{ } \} \\ \{\sigma : \delta(i, \sigma) = j\} \cup \{e\} & \text{if } i = j \text{ } \} \end{cases} \tag{2}$$

$R(i, j, 0)$ could be $\{a, b\} \cup \{e\}$ so length 9. I'll call it 10.

Length of $R(i, j, 0)$

We assume $|\Sigma| = 2$.
For all $1 \leq i, j \leq n$:

$$R(i,j,0) = \begin{cases} \{\sigma : \delta(i,\sigma) = j\} & \text{if } i \neq j \ \} \\ \{\sigma : \delta(i,\sigma) = j\} \cup \{e\} & \text{if } i = j \ \} \end{cases} \tag{2}$$

$R(i, j, 0)$ could be $\{a, b\} \cup \{e\}$ so length 9. I'll call it 10.

Let $f(k)$ be a bound on the length of all the $R(i, j, k)$.

Length of $R(i,j,0)$

We assume $|\Sigma| = 2$.
For all $1 \le i,j \le n$:

$$R(i,j,0) = \begin{cases} \{\sigma : \delta(i,\sigma) = j\} & \text{if } i \ne j \text{ } \} \\ \{\sigma : \delta(i,\sigma) = j\} \cup \{e\} & \text{if } i = j \text{ } \} \end{cases} \quad (2)$$

$R(i,j,0)$ could be $\{a,b\} \cup \{e\}$ so length 9. I'll call it 10.

Let $f(k)$ be a bound on the length of all the $R(i,j,k)$.

$f(0) = 10$.

Length of $R(i, j, k)$

Length of $R(i, j, k)$

$$R(i, j, k) = R(i, j, k-1) \bigcup R(i, k, k-1) R(k, k, k-1)^* R(k, j, k-1)$$

Let $f(k)$ be a bound on the length of all the $R(i, j, k)$.

Length of $R(i, j, k)$

$$R(i,j,k) = R(i,j,k-1) \bigcup R(i,k,k-1)R(k,k,k-1)^* R(k,j,k-1)$$

Let $f(k)$ be a bound on the length of all the $R(i,j,k)$.
$f(k) = f(k-1) + 3f(k-1) = 4f(k-1)$.

Length of $R(i,j,k)$

$$R(i,j,k) = R(i,j,k-1) \bigcup R(i,k,k-1)R(k,k,k-1)^{*}R(k,j,k-1)$$

Let $f(k)$ be a bound on the length of all the $R(i,j,k)$.
$f(k) = f(k-1) + 3f(k-1) = 4f(k-1)$.
(I am ignoring the * and it won't matter in the end since I will be using O-notation.)

Length of $R(i, j, n)$

Length of $R(i, j, n)$

$$f(0) = 10$$

Length of $R(i, j, n)$

$f(0) = 10$

For all $1 \leq k \leq n$,

$$f(k) = 4f(k-1) = 4^2 f(k-2) = \cdots = 4^k f(0) = 10 \times 4^k$$

Length of $R(i, j, n)$

$f(0) = 10$
For all $1 \leq k \leq n$,

$$f(k) = 4f(k-1) = 4^2 f(k-2) = \cdots = 4^k f(0) = 10 \times 4^k$$

Hence

$$f(n) = 10 \times 4^n = 10 \times 2^{2n}$$

Length of $R(i, j, n)$

$f(0) = 10$
For all $1 \leq k \leq n$,

$$f(k) = 4f(k-1) = 4^2 f(k-2) = \cdots = 4^k f(0) = 10 \times 4^k$$

Hence

$$f(n) = 10 \times 4^n = 10 \times 2^{2n}$$

We will call this $2^{O(n)}$ from now on.

Terminology: Regular Languages

Def We will say **regex** when we mean a lang generated by a regex. For example we might say:

regex is closed under Leo

if Leo was some operation on sets.

Terminology: Regular Languages

Def We will say **regex** when we mean a lang generated by a regex.
For example we might say:

regex is closed under Leo

if Leo was some operation on sets.

We prove (or disprove) closure properties of regex.

Terminology: Regular Languages

Def We will say **regex** when we mean a lang generated by a regex. For example we might say:

regex is closed under Leo

if Leo was some operation on sets.

We prove (or disprove) closure properties of regex.

We already know all of these closure properties since we did closure proofs with DFA's and NFA's.

Terminology: Regular Languages

Def We will say **regex** when we mean a lang generated by a regex. For example we might say:

regex is closed under Leo

if Leo was some operation on sets.

We prove (or disprove) closure properties of regex.

We already know all of these closure properties since we did closure proofs with DFA's and NFA's.

However, we are curious which ones can be proven **easily** with regex's.

Regex Closed Under Complementation

How do you complement a regular language (not a joke)?

Regex Closed Under Complementation

How do you complement a regular language (not a joke)?

While not a joke, there is no easy way to go from regex α to regex β such that $L(\beta) = \overline{L(\alpha)}$.

Regex Closed Under Complementation

How do you complement a regular language (not a joke)?

While not a joke, there is no easy way to go from regex α to regex β such that $L(\beta) = \overline{L(\alpha)}$.

You can do it. And you might on a later HW.

Regular Lang Closed Under Union

Easy The regex for $L(\alpha) \cup L(\beta)$ is $\alpha \cup \beta$.

Regular Lang Closed Under Intersection

Hard Need to convert to NFA's and do it there and convert back.

Regular Lang Closed Under Intersection

Hard Need to convert to NFA's and do it there and convert back. Might be on a HW or Exam.

Regex Closed Under Concatenation

Easy The regex for $L(\alpha) \cdot L(\beta)$ is $\alpha \cdot \beta$.

Regular Lang Closed Under $*$?

Easy The regex for $L(\alpha)^*$ is α^*.

Summary of Closure Properties and Proofs

X means **Can't Prove Easily**

$n_1 + n_2$ (and similar) is number of states in new machine if L_i reg via n_i-state machine.

$L_1 + L_2$ (and similar) is length of regex of L_i length of α_i.

Closure Property	DFA	NFA	Regex
$L_1 \cup L_2$	$n_1 n_2$	$n_1 + n_2$	$L_1 + L_2 + 1$
$L_1 \cap L_2$	$n_1 n_2$	$n_1 n_2$	X
$L_1 \cdot L_2$	X	$n_1 + n_2$	$L_1 + L_2$
$\overline{L}$	n	X	X
L^*	X	$n + 1$	$L + 1$

BILL, STOP RECORDING LECTURE!!!!

BILL STOP RECORDING LECTURE!!!