CSE 396 Lecture Tue Mar 1 Spring 2016 (1)

More MNT Examples

If I change this to "$n \geq 0$", allowing "1" $\in L$, does the proof become wrong?

$L = \{0^n 1 0^n : n \geq 1\}$. Same basic "script":

Take $S = 0^+$. Clearly S is infinite. Let any $x, y \in S$ ($x \neq y$) be given. Then there are numbers $m, n \geq 1$ with $m \neq n$ s.t. $x = 0^m$ and $y = 0^n$. Take $z = 10^m$. Then:

$xz = 0^m 1 0^m \in L$ but $yz = 0^n 1 0^m \notin L$ since $n \neq m$.

Hence S is an infinite PD set for L, so L is not regular. ☒

[Study Q: With L, (as above with $n \geq 1$), is $S = 0^*$ still PD? The proof would be wrong, but can the proof be fixed?]

Footnote: $\{0^m 0100^m : m \geq 0\}$ is an equivalent defn of L.

$L' = \{0^n 0^n : n \geq 1\}$? Take $S = 0^*$ Let any $x, y \in S$, $x \neq y$ be given. $x = 0^m$ $y = 0^n$. Take $z = 0^n$. Then $xz = 0^m 0^n \notin ? L'$ ... no: it can still be $\in L'$.

$L' = (00)^*$ which is regular.

$L'' = \{ww : w \in \{0,1\}^*\}$. "Critical Cases": $w = 0000\cdots01$

e.g. $ww = 0000001 0000001$.

Take $S = 0^* 1$. Clearly S is infinite.

Let any $x, y \in S$, $x \neq y$ be given. Then there are $m, n \geq 1$, $m \neq n$, such that $x = 0^m 1$ and $y = 0^n 1$. Take $z = 0^m 1$. Then $xz = 0^m 1 \cdot 0^m 1 \in L''$ by the division shown, but $yz = 0^n 1 0^m 1 \notin L''$ because the only possible div is after the first 1, but $n \neq m$ so it doesn't work.

$\therefore L''$ is not regular. ☒

II. The Class REG (or just REG) of Regular Languages.

string = list<char> — "First-order Objects" $x, y, z, w, v, u \ldots$

language = set<string> — "Second-order" $L, A, B, C, D, \ldots$

3rd order class = set<language> = set<set<list<char>>>

The last few weeks have proved the following **Theorem 1**:
For any language $L \subseteq \Sigma^*$, the following are equivalent:

(a) There is a regular expression R such that $L = L(R)$
(b) There is a DFA M such that $L = L(M)$
(c) There is an NFA N such that $L = L(N)$
(d) ... GNFA ... etc!

$L \in$ REG.

Proof: (a) → (c), (c) → (b), (b, c, d) → a. ⊠

Theorem 2: For any regular expression R, we can build a regular expression R' such that $L(R') = \sim L(R)$.

Abstractly: The class REG is closed under complements.

Abstract proof: Use (b) to take a DFA M s.t. $L(M) = L$, then simply and quickly build $M' = (Q, \Sigma, \delta, s, Q \setminus F)$.

Concrete proof of Theorem 2:
(a) → (b): Convert R into equivalent NFA N_R.
⚠ (c) → (b): Convert N_R into equivalent DFA M_R.
Stay in (b): Complement M_R to M'_R.
(b) → (a): Convert M'_R into final regexp R'.

Theorem 3: REG is closed under $\cap$.

I.e. for any regular languages $A, B \in$ REG, the language $A \cap B$ is also in REG.

Proof: Let any $A, B \in$ REG be given. We may take DFAs M_A, M_B s.t. $L(M_A) = A$ and $L(M_B) = B$. Then build a DFA M_C s.t. $L(M_C) = L(M_A) \cap L(M_B)$ using "Cartesian Product" for DFAs.

Thus $L(M_C) = A \cap B$ and M_C is a DFA, so $A \cap B$ is regular. ⊠

(3)

Suppose instead we are given regular expressions α and β, and we desire to build a regexp γ s.t. $L(\gamma) = L(\alpha) \cap L(\beta)$?

Proof: / **Algorithm:**

(1) Convert α, β to <u>equivalent</u> NFAs (with ϵ arcs) N_α and N_β: $L(N_\alpha) = L(\alpha)$, $L(N_\beta) = L(\beta)$.

(2) → (2) Convert N_α, N_β to equivalent DFAs M_α, M_β.

⚠ (3) "<u>Cart.</u> <u>Prod.</u>" M_γ s.t. $L(M_\gamma) = L(M_\alpha) \cap L(M_\beta)$.

Piazza Q: (4) Convert M_γ back to γ s.t. $L(\gamma) = L(M_\gamma)$.

Can we avoid exp blowup of NFA → DFA?

not to mention $\cup$

∴ Since REG is closed under $\sim$ and $\cap$, it is closed under all Boolean operations.

<u>Theorem</u>: All <u>finite</u> languages are regular. <u>Proof</u>: If $L = \{w_1, w_2, \dots w_m\}$, then L has the regexp $w_1 \cup w_2 \cup \dots \cup w_m$. ⊠

∴ If L is regular, then any language L' obtained by adding and/or taking away finitely many strings is also regular.

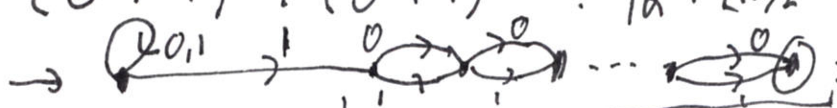
<u>Because</u>: if F is the finite language of strings whose status was changed, then $L' = \underbrace{L}_{\text{regular}} \underset{\uparrow \text{ Bool op.}}{\Delta} \underbrace{F}_{\text{regular}} = (L \setminus F) \cup (F \setminus L)$.

$L \oplus F$ in text (p 15 or so)

<u>Example of all three objects</u>:

For any $k \geq 1$, define <u>L_k</u> $= \{x \in \{0,1\}^* :$ bit k from the end is a '1'.$\}$

<u>Regular Exp^n</u>: $R_k = (0+1)^* 1 (0+1)^{k-1}$: $12 + \lceil \log_2 k \rceil$ chars.

NFA: N_k = 

: $2k+1$ <u>arcs</u>

DFA? Study Fact: $\{0,1\}^k$ is PD for L_k, so any DFA M_k s.t. $L_k = L(M_k)$ needs 2^k states!