Week 9 - Regular Expressions & CFGs

Billy Price

Definition A language, L, is *regular*, iff it can be recognised by some DFA

Regular Operations Let A and B, be languages. The *regular operations* are

Union: $A \cup B$

Kleene Star $A^* := \bigcup_{n \in \mathbb{N}} \{\underbrace{w_1 w_2 \dots w_n}_{\text{string concatenation}} \mid \forall_{1 \le i \le n}\ w_i \in A\}$

Concatenation: $A \circ B := \{xy \mid x \in A, y \in B\}$

Theorem 1 The class of regular languages is closed under the regular operations

Meaning If A, B, are any regular languages then $A \cup B$, $A \circ B$, A^* are all regular languages

How to Prove? Given any regular languages A, B, with NFAs D_A, D_B, construct

- An NFA which recognises the language $A \cup B$ using D_A and D_B
- An NFA which recognises the language $A \circ B$ using D_A and D_B
- An NFA which recognises the language A^* using D_A

Upshot Describe all the regular languages with just union, complement & Kleene star

Regular Expressions
- ϕ is a RE
- ε is an RE
- Any symbol a in Σ is a RE
- If x, y, are REs, then so are $(x \cup y)$, xy, x^*

The *language* of an RE r, written $L(r)$ is defined such that

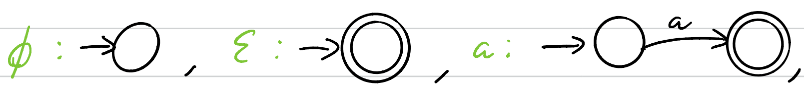
- $L(\phi) := \phi$
- $L(\varepsilon) := \{\varepsilon\}$
- $L(a) := \{a\}$ for $a \in \Sigma$
- $L(xy) := L(x) \circ L(y)$
- $L(x \cup y) := L(x) \cup L(y)$
- $L(x^*) := L(x)^*$

Examples $L(ab^*a) = \{a\} \circ \{b\}^* \circ \{a\}$ $L\big((0^* \cup 10)(0 \cup 1)^*\big) = \big(\{0\}^* \cup \overbrace{\{10\}}^{\{1\} \circ \{0\}}\big) \circ (\{0\} \cup \{1\})^*$

Theorem 2 A language is regular iff it can be described by a regular expression

$$A \text{ is regular} \iff \exists \overset{\text{RE}}{r},\ L(r) = A$$

How to prove?

RE to NFA: 

+ Theorem 1

NFA to RE: State Elimination Process (Week 8 Part 1 - Slide 8-12)

CONTEXT FREE GRAMMARS

Billy Price

Definition A Context-free language (a CFL) is a language described by a CFG

CFGs are to Context Free Languages what regular expressions are to Regular Languages.
They describe a recipe for creating strings of the language.

A context-free grammar (CFG) G is a 4-tuple (V, Σ, R, S), where

1. V is a finite set of variables,
2. Σ is a finite set of terminals,
3. R is a finite set of rules, each consisting of a variable (the left-hand side) and a string in $(V \cup \Sigma)^*$ (the right-hand side),
4. S is the start variable.

The binary relation $\Rightarrow$ on $(V \cup \Sigma)^*$ is defined as follows.

Let $u, v, w \in (V \cup \Sigma)^*$. Then $uAw \Rightarrow uvw$ iff $A \to v$ is a rule in R. That is, $\Rightarrow$ captures a single derivation step.

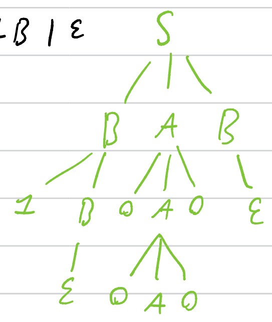
Let $\overset{*}{\Rightarrow}$ be the reflexive transitive closure of $\Rightarrow$.

$$L(G) = \{s \in \Sigma^* \mid S \overset{*}{\Rightarrow} s\}$$

Example $G := (\{S, A, B\}, \{0, 1\}, R, S)$

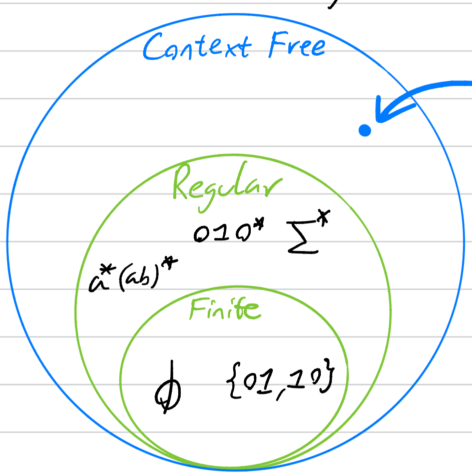
$S \to A \mid BAB$
$A \to 0A0 \mid \varepsilon$
$B \to 1B \mid \varepsilon$

$10000 \in L(G)$



$S \Rightarrow BAB$
$\Rightarrow 1BAB$
$\Rightarrow 1\varepsilon AB$
$\Rightarrow 1A\varepsilon$
$\Rightarrow 1040$
$\Rightarrow 100A00$
$S \overset{*}{\Rightarrow} 10000$

How would you describe this language?
Is it also regular?
Can it be described by a R.E.?



Why is every regular language context free?

What languages live here?

Example $\{0^n 1^n \mid n \in \mathbb{N}\}$

It's Context Free	It's NOT Regular
Produce a CFG that describes it	Pumping Lemma!

The Pumping Lemma For any language A,

$$A \text{ is Regular} \Rightarrow \exists p \geq 0\ \forall s \in A\ \exists x \exists y \exists z \left[(|s| \geq p \land s = xyz) \Rightarrow \begin{Bmatrix} |xy| \leq p \land \dots \\ y \neq \varepsilon \land \dots \\ \forall i \geq 0\ xy^iz \in A \end{Bmatrix} \right]$$

Equivalently, we can take the contrapositive...

$$\forall p \geq 0\ \exists s \in A\ \forall x \forall y \forall z \left[|s| \geq p \land s = xyz \land \text{NOT} \begin{Bmatrix} |xy| \leq p \land \dots \\ y \neq \varepsilon \land \dots \\ \forall i \geq 0\ xy^iz \in A \end{Bmatrix} \right] \Rightarrow A \text{ is NOT Regular}$$

① Take any p
② design an s at least length p
③ show <u>any</u> way of breaking down s into xyz cannot make all three statements true.

Do ③ by assuming two statements and explaining why the third is not true.
Often you will assume $|xy| \leq p$ & $y \neq \varepsilon$ and pick a $j \geq 0$ such that $xy^jz \notin A$