Basics

Basics Σ – Alphabet, Σ* – All possible strings

RegEx ε = match empty, ∅ = match none
() = precedence, * = match 0-∞
| = match either, ccat = match both

Language $L(r) = \{u \in \Sigma^* \mid u \text{ matches } r\}$
Equivalence of r determined by L(r)

Automata

Makeup Finite set $States_M$ of states
Finite set Σ_M of input symbols
For each $q \in States_M$ and $a \in \Sigma_M$, a subset $\Delta_M(q, a) \subseteq States_M$ which can be reached by that transition a
Element $S_M \in States_M$, the start state
$Accept_M \subseteq States_M$ of accepting states
DFA have the property that $\Delta_M(q, a)$ contains exactly one element, so:
$q \xrightarrow{a} q' \Leftrightarrow q' = \delta_M(q, a)$

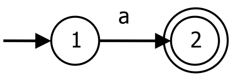
Subset Constr. $States_{PM} = \{S \mid S \subseteq States_M\}$
$\delta_{PM}(S, a) = \{q' \mid \exists q \in S.(\Delta_M(q, a) = q')\}$
$S_{PM} = \{q \mid S_M \xrightarrow{\varepsilon} q\}$
$Accept_{PM} =$
$\{S \in States_{PM} \mid \exists q \in S.(q \in Accept_M)\}$

Kleene's Theorem

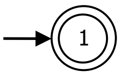
Regularity L is regular iff it's the set of strings accepted by some DFA

For any regular expression r, L(r) is regular

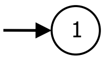
Symbol



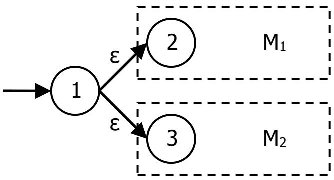
Empty



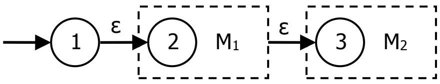
Nothing



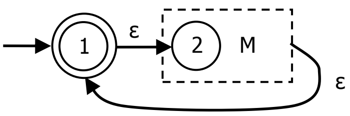
Union of M_1 and M_2



Concat of M_1 and M_2



Star of M



Every regular language is of the form L(r)

Lemma Given an NFA M, for each subset $Q \subseteq States_M$, there is a regular expression $r_{q,q'}^Q$ satisfying:
$L(r_{q,q'}^Q) = \{u \in \Sigma_M^* \mid q \xrightarrow{u} q' \text{ in M}$
with all intermedia te states in Q}

Prove By $r = r_{q,q'}^{Q\backslash\{q_0\}} \mid r_{q,q_0}^{Q\backslash\{q_0\}}(r_{q_0,q_0}^{Q\backslash\{q_0\}})^* r_{q_0,q'}^{Q\backslash\{q_0\}}$

Corollary Regular expressions have complements (find DFA, invert it, find expression)

The Pumping Lemma

For every regular language L there is a number l ≥ 1 such that all w ∈ L can be expressed as $w = u_1 v u_2$ where:

Length(v) ≥ 1
Length(u_1v) ≤ l
For all n ≥ 0 $u_1 v^n u_2 \in L$

Prove By Considering the # of state transitions as compared to the # of states

Acceptance

If a DFA M accepts any string at all, it accepts one whose length is less than the number of states in M

Prove By Similarly to the pumping lemma

Corollary We can test two expressions r_1, r_2 for equivalence:
$r_1 = r_2$ □ $L(r_1\&\sim r_2) = L(r_2\&\sim r_1) = 0$
And by this lemma there are only finitely many possible strings to check

Grammars

Components Terminals, non-terminals

Derivations Non-terminal□string of both

BNF Non-terminal ::= possible | possible

Regularity All regular languages are context-free (q□aq' iff transition is in M)
Context free grammar is regular iff all productions are of the form:
x□uy or x□u
For u being a string of terminals and x and y both non-terminals
The NFA you can build from this as the logically reverse process from the method by which regular languages are made context free