CMSC 141 Automata and Language Theory

Regular Languages

Mark Froilan B. Tandoc

September 12, 2014

Closure Properties

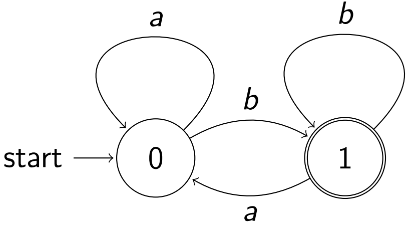
The set of regular languages is closed under *union*, *concatenation*, and *Kleene star*.

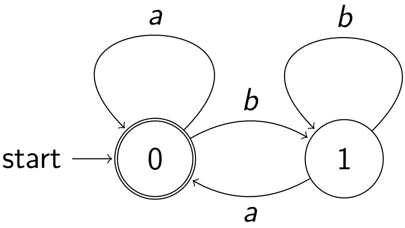
- Proof follows immediately from the definition of regular expressions
- If R and S are regular languages, then they can be described by some regular expressions r and s
- $r + s$, rs, r^* are also regular expressions that describe the regular languages $R \cup S$, RS, R^*

Closure Under Complementation

The set of regular languages is also closed under *complementation*. Proof is by FA construction.

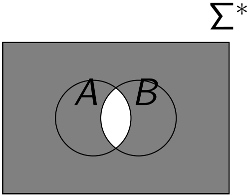
- Given a regular language L, build a DFA for L, then negate the status of all the states. All final states are made non-final and vise versa. The resulting DFA is the DFA for $L^{\complement} = \Sigma^* - L$.





Closure Under Intersection

The set of regular languages is closed under *intersection*.



- Proof is by De Morgan's Law
  $(A \cap B)^{\complement} = A^{\complement} \cup B^{\complement}$ and hence, $A \cap B = (A^{\complement} \cup B^{\complement})^{\complement}$
  If A and B are regular languages, then so are $A^{\complement}$, $B^{\complement}$, $A^{\complement} \cup B^{\complement}$, and $(A^{\complement} \cup B^{\complement})^{\complement}$ by the closure properties shown in the previous slides.

Closure Under Intersection

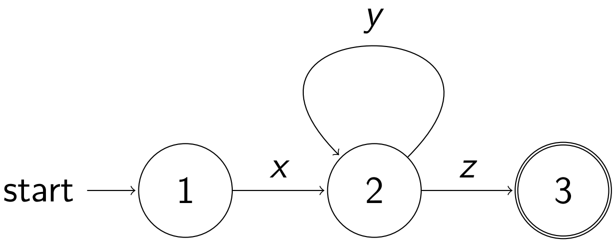
While the proof using De Morgan's Law is valid, actual construction of the FA or the regular expression for the proof is long.
A more efficient constructive proof for closure under intersection is available.
(Left as reading assignment :P)

Other Closure Properties

- Closure under string substitutions (homomorphism)?
- String reversal?
- Inverse homomorphisms?

Pumping Lemma
for regular languages

Short version: If L is an infinite, regular language, and we have a string w which is sufficiently long (pumping length p) string in L, then we can partition w into xyz where y should be non-empty and can be repeated (or "pumped") and preserves membership in L.



Example of Pumping Lemma

Consider: $a(bb)^*a$

Let's say that the string "abbbba" is long enough, we can partition the string into xyz.

- $x =$ "a"
- $y =$ "$bbbb$"
- $z =$ "a"

Pumping y will still retain membership in L.

- $xy^0z =$ "aa"
- $xy^1z =$ "$abbbba$"
- $xy^2z =$ "$abbbbbbbba$"
- ...

Proof of the Pumping Lemma

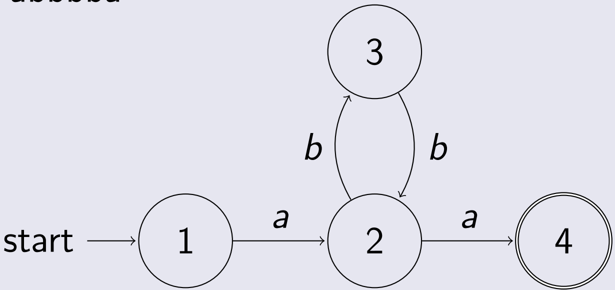
- Using pigeonhole principle: If there are more pigeons than pigeonholes, then at least one pigeonhole will contain more than one pigeon.

Proof of the Pumping Lemma

- If the minimum-state DFA for L has n states, and a string w in L has n or more symbols, then some state q must be revisited.
- The non-empty substring between the visit and revisit of a state is the string y that we pump.

$a(bb)^*a$

"abbbba"



A More Formal Definition

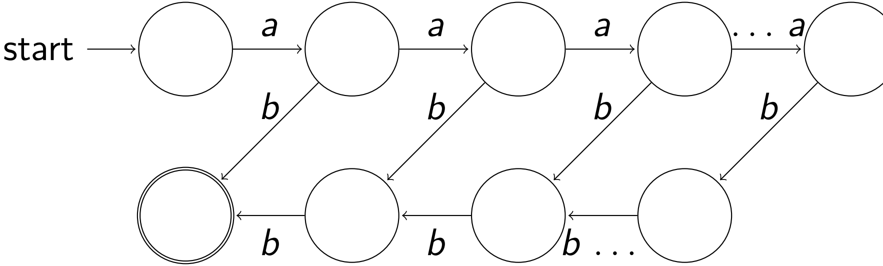
Pumping Lemma: If A is a regular language, then there is a number p (pumping length), and a string $w \in A$ of length at least p, where w can be divided into three pieces $w = xyz$, satisfying the conditions:

1. for each $i \geq 0, xy^iz \in A$
2. $\mid y \mid > 0$, and
3. $\mid xy \mid \leq p$

Beyond Regular Languages

- Not all languages are regular.
- Try creating a DFA for
  $= \{a^n b^n : n > 0\}$
  $= \{ab, aabb, aaabbb, ...\}$
- Or $\{(), (()), ((())), ...\}$

Candidate DFA for $\{a^n b^n : n > 0\}$



Why is this not a valid DFA for the language?
It contains infinite states

$\{a^n b^n\}$ is non-regular

- Just because we cannot find a DFA for a given language doesn't mean that the language is non-regular. Maybe we just didn't tried hard enough.
- Proving a language as non-regular needs real proof.

$\{a^n b^n\}$ is non-regular

- Pumping Lemma can help us using indirect proof.
  - Pumping Lemma: If L is infinite and regular, it should satisfy the pumping lemma.
  - Indirect proof: We know that L is infinite. Let's assume L is regular. Then let's try to reach a contradiction. Because of the contradiction, L cannot be regular.

Proof Using Pumping Lemma

- Let's assume $L = \{a^n b^n : n > 0\}$ is regular. Because it is also infinite, then it must satisfy the pumping lemma.
- Say $w = aaaabbbb$ is long enough.
- We must be able to partition w into xyz where $y \neq \varepsilon$ and $xy^j z \in L$ for any j
  - All a's for y, aa(aa)bbbb
  - All b's for y, aaaa(bbb)b
  - Mix of a's and b's for y, aa(aabb)bb
- In all cases, pumping y gives strings that are not in L
- Because finding the substring y is impossible, L **cannot be regular**

Other notes

- The Pumping Lemma is a *theorem*, but we call it a *lemma* because we use it to prove theorems on non-regularity of several languages.
- We **cannot** use the Pumping Lemma to prove that a certain language is regular
- We do not need to use Pumping Lemma every time we prove non-regularity of some languages. Sometimes, closure properties are enough

References

- Previous slides on CMSC 141
- M. Sipser. Introduction to the Theory of Computation. Thomson, 2007.
- J.E. Hopcroft, R. Motwani and J.D. Ullman. Introduction to Automata Theory, Languages and Computation. 2nd ed, Addison-Wesley, 2001.
- E.A. Albacea. Automata, Formal Languages and Computations, UPLB Foundation, Inc. 2005
- JFLAP, www.jflap.org
- Various online LaTeX tutorials