CS423 Finite Automata & Theory of Computation

TTh 9:30 - 10:50 in Blow 331 (section 2)

TTh 12:30 - 13:50 in Small Physics Lab 111 (section 1)

Prof. Weizhen Mao, wxmaox@wm.edu, wm@cs.wm.edu

General Information

- Office Hours: TTh 11:00 - 12:00 in 114 McGl and W 2:30 - 3:00 on zoom or by email
- Grader: Idema, Jacob for section 1 (office hour under instructor contact information on BB)
- Grader: Tran, Tung for section 2 (office hour under instructor contact information on BB)
- Textbook: Intro to the theory of computation (any edition), Michael Sipser. An e-book in PDF maybe available online.
- Prerequisites/background: Linear algebra, Data structures and algorithms, and Discrete math

Part 1: Mathematical foundation and proofs

Sets and Languages (*Sipser 0.2*)

Languages are central concepts in automata theory.

- Alphabet Σ: Finite and nonempty, e.g., $\Sigma = \{0, 1\}$ and $\Sigma = \{a, b, \ldots, z\}$.
- String (or word), e.g., $w = 01110$; empty string ε; length of a string, $|w|$; concatenation of two strings $w_1 w_2$; reverse of a string w^R, and substring of a string.
  Example: $a^R = a$, $(wa)^R = aw^R$, $(uv)^R = v^R u^R$, $(w^R)^R = w$
- Language: A language is a set of strings, e.g., $\{\varepsilon\}$, $\emptyset$, Σ, $A = \{w \mid w \text{ has an equal number of 0s and 1s}\} = \{\varepsilon, 01, 10, 0011, 0110, 1010, \cdots\}$, and $B = \{0^n 1^n | n \geq 1\} = \{01, 0011, 000111, \cdots\}$. Note $B \subset A$.

- Regular operators: Let A and B be two languages.
- Union: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.
- Concatenation: $A \cdot B = \{xy \mid x \in A \text{ and } y \in B\}$.
  Example: If $A = \{01, 10\}$ and $B = \{0, 1\}$, then
  $AB = \{010, 011, 100, 101\}$.
- Star:
  $A^* = \{x_1 x_2 \cdots x_k \mid \text{ all } k \geq 0 \text{ and each } x_i \in A \text{ for } i = 1, 2, \ldots, k\}$.
  Note $\varepsilon \in A^*$
  Example: If $A = \{0, 1\}$, then
  $A^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 111, \ldots\}$
  Example: If $A = \{01, 11\}$, then
  $A^* = \{\varepsilon, 01, 11, \underline{01}\ \underline{01}, \underline{11}\ \underline{01}, \ldots, \underline{11}\ \underline{11}\ \underline{01}\ \underline{11}, \ldots\}$
  Quiz 1: L^* is the concatenation of any strings in L in any order. If $L = \{ab, aa, baa\}$. Which of followings are in L^*?
  (1) abaabaaabaa (2) aaaabaaaa (3) baaaaabaaaab
  (4) baaaaabaa

- More operators
  - Intersection: $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
  - Complement: $\overline{A}$ is the set of all elements under consideration that are not in A. Usually, $\overline{A} = \Sigma^* - A$.
  - Difference: $A - B = A \cap \overline{B}$. (anything in A but not in B)
  - Power of language: $A^k = \{x_1 x_2 \cdots x_k \mid x_i \in A \text{ for } i = 1, 2, \cdots, k\} = AA^{k-1}$.
  - Observation:
    - $A^0 = \{\varepsilon\}$
    - $A^* = A^0 \cup A^1 \cup A^2 \cup \cdots$ (zero or more)
    - $A^+ = A^1 \cup A^2 \cup \cdots$ (one or more)
    - $A^* = A^+ \cup \{\varepsilon\}$

  Quiz 2: If $A = \{1, 01\}$, then $A^2 = ?$
  Quiz 3: If $A = \{01, 11, 0\}$, then $|A^3| = ?$
  Quiz 4: If $|A| = n$, then $|A^k| = |A|^k = n^k$. True or false?
  Quiz 5: $A^+ = A^* - \{\varepsilon\}$?

- Why languages, not problems:
  - Meta Claim: Computational problem → decision problem.
    Example: Traveling Saleman Problem (TSP)
    Input: $G = (V, E, w)$
    Goal: Find a tour (cycle) with minimun total weight
  - Example: Decision problem for TSP
    Input: $G = (V, E, w)$ and $B \geq 0$
    Question: Is there a tour in G such that the total weight of the tour is no more than B?
  - Decision problem: Given an input x, does x satisfy property P, or is $x \in \{y | y \text{ satisfies } P\}$?
    Input data of any form, such as matrix, graph, list, etc., can be coded into strings
  - Membership in a language: Given a language A and a string $w \in \Sigma^*$, is w a member of A, or is $w \in A$?

- Prove by contradiction: $H \to C$ is equivalent to $\overline{C} \to \overline{H}$ or $\overline{C} \wedge H \to \overline{T}$, where T is an axiom, a proven truth/fact.

  Example 1: The number of primes is infinite.

  1. Assume $\exists$ a finite number of primes: $p_1, p_2, \cdots, p_k$, where p_k is the largest.
  2. Define a new prime $p = \Pi_{i=1}^{k}(p_1 \cdots p_k) + 1 > p_k$.
  3. Since p cannot be divisible by all primes, then p must be a prime larger than p_k.
  4. A contradiction!

  Example 2: $\sqrt{2}$ is irrational.

  1. Assume $\sqrt{2}$ is not irrational, thus is rational. So $\sqrt{2} = \frac{a}{b}$
  2. Squaring both sides of the equation, we get $2 = \frac{a^2}{b^2}$ or $2b^2 = a^2$. So $a = 2k$ for some k.
  3. $2b^2 = a^2$
  4. $2b^2 = (2k)^2$
  5. $2b^2 = 4k^2$
  6. $b^2 = 2k^2$, So $b = 2j$ for some j
  7. So $gcd(a, b) \neq 1$. A contradiction!

- Prove by induction: Used to prove a statement $S(n) \forall n \geq c$.

The logic behind the method:
$S(n)$ for $n \geq c$ iff $S(c) \wedge \forall k(S(k) \rightarrow S(k+1))$.
The proof includes (1) basis step, (2) inductive hypothesis, and (3) inductive step.

Example 3: For $n \geq 1$, $\sum_{i=1}^{n} i^2 = \frac{1}{6}n(n+1)(2n+1)$.

1. Base case: $n = 1$ then $1 = \frac{1}{6} \cdot 1 \cdot 2 \cdot 3 = 1$ (left=right)
2. Inductive hypothesis: Assume equation holds for $n = k+1$, i.e., $\sum_{i=1}^{k+1} i^2 = (\sum_{i=1}^{k} i^2)\frac{1}{6}k(k+1)(2k+1)$
3. Consider $\sum_{i=1}^{k+1} i^2 = \sum_{i=1}^{k} i^2 + (k+1)^2 = \frac{1}{6}k(k+1)(2k+1) + (k+1)^2$
4. Induction: $n = k+1$
5. $L = \sum_{i=1}^{k+1} i^2 = \sum_{i=1}^{k} i^2 + (k+1)^2 = \frac{1}{6}k(k+1)(2k+1) + (k+1)^2$
6. $R = \frac{1}{6}(k+1)(k+2)(2(k+1)+1) = \frac{1}{6}(k+1)(k+2)(2k+3)$
7. $L = R$ Q.E.D.

A Quick Review of Languages

- Alphabet Σ, string or word, substring, language
- Common operations borrowed from set theory: union, intersection, difference, complement
- New operations: concatenation, complement, star, power
- Proof techniques: By contradiction, induction, and construction

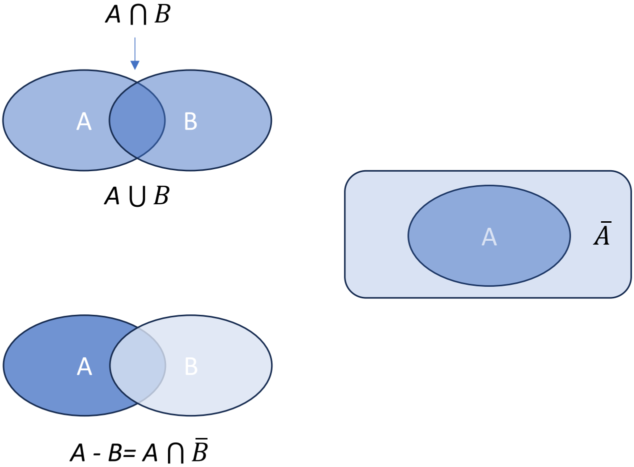


Figure 1: Regular operators in Venn diagrams

Part 2: DFA, NFA, and Regular Languages

2.1 Finite Automaton(*Sipser 1.1 - 1.2, pp.31- 58*)

- Finite automata are simple computational models for computers with an extremely limited amount of memory.
- Use of automata theory includes: study of behavior of digital circuits, lexical analyzer in compilers, text pattern matching, and verification of finite-state systems.
- They are designed to accept some strings, therefore to recognize a language, which is the set of accepted strings.
- Given an input string, a finite automaton reads the string, one symbol at a time, from left to right. Using a transition function, the FA changes from one state to another based on what symbol is being read. Eventually, it reads all symbols in the string. If at this time the FA enters a final state, then the FA halts and the string is accepted.

Example 1 of a DFA

- Alphabet: $\Sigma = \{0, 1\}$.
- Four states: q_0, q_1, q_2, q_3, in which q_0 is the start state and q_3 is a final state.
- Transition function δ:

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
$*q_3$	q_3	q_3

- What does δ tell us?
  $\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$
  $\delta(q_1, 0) = q_0$, $\delta(q_1, 1) = q_2$
  $\delta(q_2, 0) = q_0$, $\delta(q_2, 1) = q_3$
  $\delta(q_3, 0) = q_3$, $\delta(q_3, 1) = q_3$

Example 1 (continued)

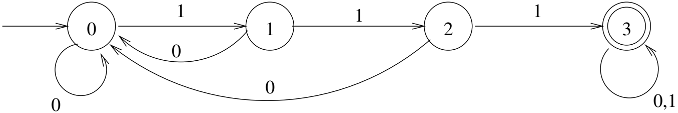


Figure 2: A DFA that accepts strings with substring 111

Try $w_1 = 100111010$ and $w_2 = 0011010001$
The language recognized/accepted is
$L_1 = \{w \in \{0,1\}^* \mid w \text{ contains substring } 111\}$

2.2 DFA (*Sipser 1.1, 35-44*)

- DFA $M = (Q, \Sigma, \delta, q_0, F)$, where
  - Q is a finite set of states
  - Σ is an alphabet
  - $q_0 \in Q$ is the start state
  - $F \subseteq Q$ is a set of accept/final states
  - $\delta : Q \times \Sigma \to Q$ is a transition function, where $\delta(q, a) = p$ is the next state of M if the current state is q and the current symbol is a
- Components of a DFA
  - A tape of squares, with the same length of the input string
  - A control unit that keeps track of the current state and follows the δ function
  - The head that reads and moves to the right, one square at a time
  - The DFA accepts the input string if the head reaches the end of the tape and the control unit sees the final state

- Extending δ to $\hat{\delta}: Q \times \Sigma^* \rightarrow Q$: For any $q \in Q$ and any $w = xa \in \Sigma^*$, define $\hat{\delta}(q,x)$ recursively as below:
  - $\hat{\delta}(q,\varepsilon) = q$ and
  - $\hat{\delta}(q,w) = \delta(\hat{\delta}(q,x),a)$
- Language recognized by DFA M: $L(M) = \{w|\hat{\delta}(q_0,w) \in F\}$.
- A language is called a regular language if some DFA recognizes it.
- How does a DFA accept a string? A path in the diagram that starts from q_0 and ends at $q_f \in F$ such that the concatenation of the symbols on the path matches the input string.

Example 2

Let $L_2 = \{w \in \{0,1\}^* \mid w \text{ has even numbers of 0s and 1s}\}$
Construct DFA M such that $L(M) = L_2$.

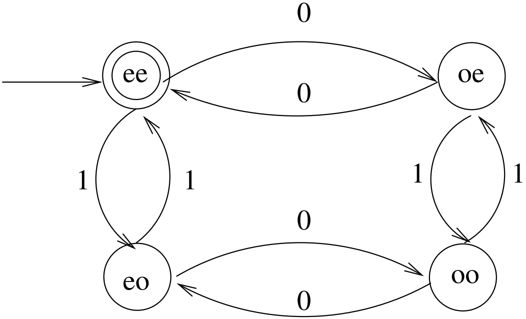


Figure 3: A DFA that accepts strings of even number of 0s and 1s

Example 3 (Sipser p. 36):
A DFA M is given below:

δ	0	1
$\rightarrow q_0$	q_0	q_1
$*q_1$	q_2	q_1
q_2	q_1	q_1

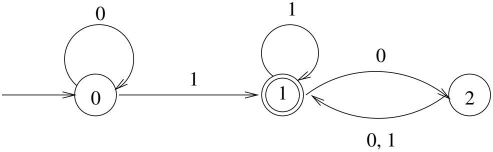


Figure 4: Describe the language of the DFA

Example 3 (continued)

The DFA in this example accepts strings that have at least one 1 and an even number of 0s after the last 1.

Reminder: More on DFA design in Sipser pp. 37-44.

Example 4 Design a DFA that accepts
$L_4 = \{w \in \{0,1\}^* | w \text{ contains substring 001}\}$

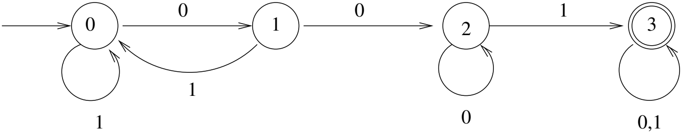


Figure 5: A DFA that accepts strings with substring 001

Example 5 Give a DFA that accepts $L_5 = \{w \in \{0,1\}^* |$ numerical value of w is a multiple of $3\}$, e.g., for string 0110, its numerical value is 6, then it is in the language, but for 101 with a numerical value of 5, it is not a multiple of 3, thus is not in the language.

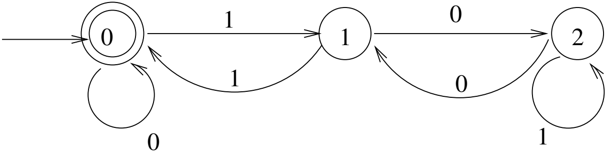


Figure 6: A DFA that accepts strings with numerical value that is a multiple of 3. Each state represents a possible remainder.

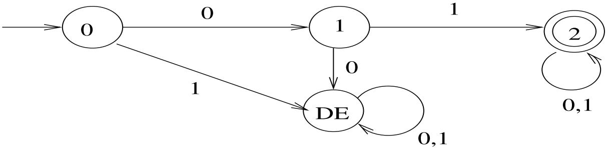
Example 5 (continued)

Let x be a part of the input string being read so far.
Let $[x]$ be the numerical value of x, e.g., if $x = 0010$, $[x] = 2$.
Entering state q_i means that the string read so far has a numerical value that has a remainder of i when divided by 3.

$[x] = 3k$	$[x] = 3k+1$	$[x] = 3k+2$
$[x0] = 6k$	$[x0] = 6k+2$	$[x0] = 6k+4$
$[x1] = 6k+1$	$[x1] = 6k+3$	$[x1] = 6k+5$

- Case 0: The current state is q_0, then $[x] = 3k$. If the next symbol is 0, then $[x0] = 2 \cdot 3k = 6k$, a multiple of 3. So the next state should be q_0. But if the next symbol is 1, then $[x1] = 2 \cdot 3k + 1 = 6k + 1$, with a remiander of 1 when divided by 3. So the next state should be q_1.
  $\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$
- Case 1: Similar to Case 0. $\delta(q_1, 0) = q_2$, $\delta(q_1, 1) = q_0$
- Case 2: Similar to above. $\delta(q_2, 0) = q_1$, $\delta(q_2, 1) = q_2$

- Definitions of $\delta : Q \times \Sigma \to Q$ and $\hat{\delta} : Q \times \Sigma^* \to Q$
- The language of DFA M:
  $L(M) = \{w \in \{0,1\}^* \mid \hat{\delta}(q_0, w) \in F\}$
- A language accepted by a DFA is a **regular language**
- In designing a DFA with n nodes and alphabet Σ, the following two properties must be satisfied:
  - (1) each node must have $|\Sigma|$ out-going arcs
  - (2) the total number of arcs must be $n \cdot |\Sigma|$
  - A dead-end state may be added to a DFA to satisfy the above properties
- An example of including a dead-end state in a DFA that accepts strings that start with a 01:



2.3 NFA (*Sipser 1.2, pp. 47-54*)

- NFA $N = (Q, \Sigma, \delta, q_0, F)$, where
  - Q is a finite set of states,
  - Σ is an alphabet,
  - $q_0 \in Q$ is the start state,
  - $F \subseteq Q$ is a set of accept/final states, and
  - $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ is a transition function, where $\delta(q, a) = P$ is the set of states that N may enter if the current state is q and the current symbol is a.
    In the case of $\delta(q, \varepsilon) = P$, N ignores the current symbol and goes from q to any state in P without reading any symbol.

- ε-closure: For any $P \subseteq Q$, $E(P)$ is the set of all states reachable from any state in P via ≥ 0 ε-transitions.
- Extending δ to $\hat{\delta} : Q \times \Sigma^* \to 2^Q$: For any $q \in Q$ and any $w = xa \in \Sigma^*$, define
  - $\hat{\delta}(q, \varepsilon) = E(\{q\})$ and
  - $\hat{\delta}(q, w) = E(\cup_{i=1}^{k} \delta(p_i, a))$ if $w = xa$ and $\hat{\delta}(q, x) = \{p_1, \ldots, p_k\}$.
- Language of NFA N: $L(N) = \{w | \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$.
- How does an NFA accept a string? Among all paths from q_0 to $q_f \in F$, there is some path such that the concatenation of symbols on the path matches the string.

Example 1: Language of strings that contain substring 111

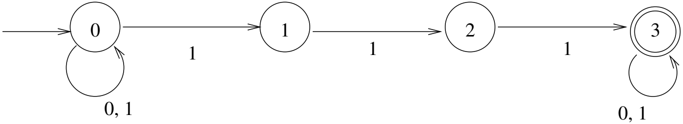


Figure 7: An NFA that accepts strings with substring 111

Example 2: $A = \{w \in \{0,1\}^* | w \text{ has 101 or 11 as substrings}\}$

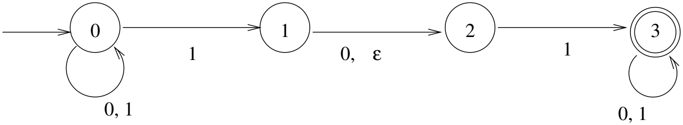


Figure 8: An NFA that accepts strings with substrings 101 or 11

Clarification:

- Consider string 001011. There are at least two paths from q_0 to q_3
- Examples of ε-closure (or E-closure): $E(\{q_0\}) = \{q_0\}$; $E(\{q_0, q_1\}) = \{q_0, q_1, q_2\}$

Example 3:
$B = \{w \in \{0, 1\}^* | w \text{ has a 1 in the 3rd position from the right end}\}$

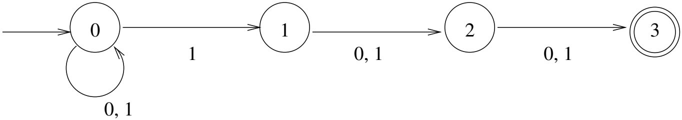


Figure 9: An NFA that accepts strings with a 1 in the third position from the right end

Example 4: An NFA that accepts decimal numbers (a number that may have $+$ or $-$ preceding it, but must have a decimal point, e.g., .123, 23., +1.2, -1.0).

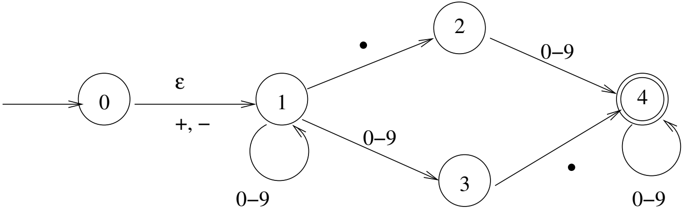


Figure 10: An NFA that accepts strings that are decimal numbers with or without signs

2.4 DFAs $\Leftrightarrow$ NFAs (*Sipser 1.2, pp.54-58*)
Subset construction method: Given NFA $N = (Q, \Sigma, \delta, q_0, F)$, construct a DFA $M = (Q', \Sigma, \delta', q_0', F')$ such that $L(M) = L(N)$.

- $Q' = 2^Q$ (power set), i.e., Q' contains all subsets of Q. Note that if $|Q| = n$ then $|Q'| = 2^n$. This is just the worst case. Since many states in M are inaccessible or dead-end states and thus may be thrown away, so in practice, $|Q'|$ may be much less than 2^n.
- $q_0' = E(\{q_0\})$.
- $F' = \{R \in Q' | R \cap F \neq \emptyset\}$.
- For each $R \in Q'$ and each $a \in \Sigma$, $\delta'(R, a) = E(\cup_{p \in R} \delta(p, a))$.

Definition: Any language that can be accepted by DFA or NFA is called a regular language (RL).
Theorem: The equivalence of DFAs, NFAs, and RLs.

Converting an NFA to a DFA with subset construction
Example 1:

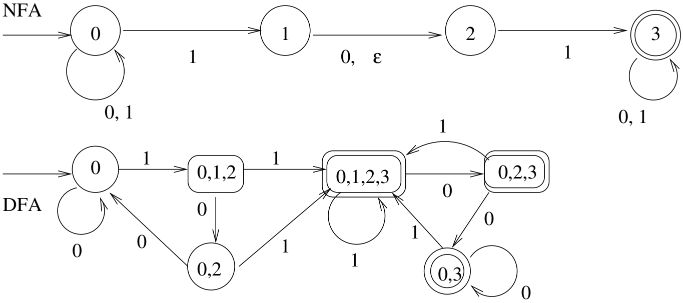


Figure 11: Converting an NFA to DFA with subset construction

Example 2:

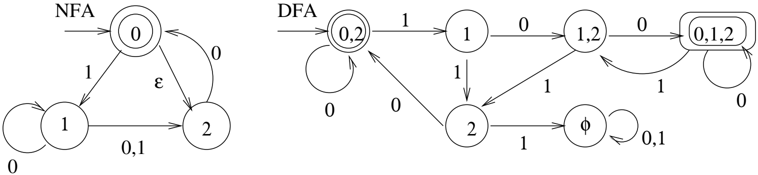


Figure 12: Converting an NFA to a DFA

Example 3:

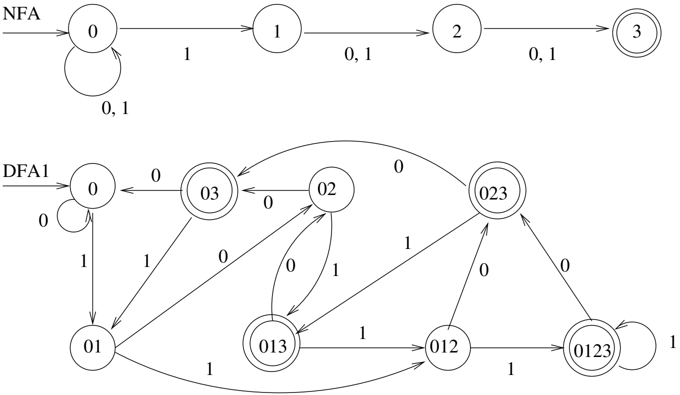


Figure 13: Converting an NFA to a DFA with subset construction

Example 3: (continued)
Without using the subset construction method, can a DFA be designed to accept all strings that has a 1 in the third position from the right end?

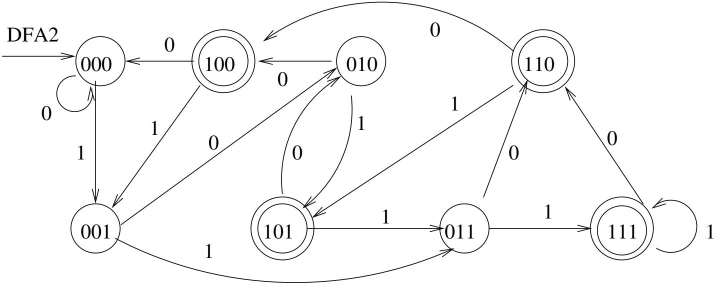


Figure 14: Design the same DFA from scratch

Example 4: A bad case for the subset construction: $|Q_N| = n+1$ and $|Q_D| = 2^n$.

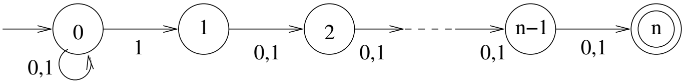


Figure 15: A case that converting an NFA to a DFA causes an exponential increase of states in the DFA

2.5 Closure Properties of RL’s (*Sisper 1.1, pp. 44-47 and 1.2 pp. 58-63*)

- Union: If A and B are regular, so is $A \cup B$.
- Concatenation: If A and B are regular, so is AB.
- Star: If A is regular, so is A^*. (Need a new start state.)

- Complementation: If A is regular, so is $\overline{A}$ (which is $\Sigma^* - A$).
- Intersection: If A and B are regular, so is $A \cap B$ (which is $\overline{\overline{A} \cup \overline{B}}$).
- Difference: If A and B are regular, so is $A - B$ (which is $A \cap \overline{B}$.
- Reverse: If A is regular, so is A^R.
- Homomorphism: If A is regular, so is $h(A)$ (which is $\{h(w) | w \in A\}$ for a homomorphism $h : \Sigma \to (\Sigma')^*$). (Discuss later)
- Inverse homomorphism: If A is regular, so is $h^{-1}(A)$ (where $h^{-1}(A) = \{w | h(w) \in A\}$).

Example: Prove that $A = \{w \in \{a,b\}^* | w \text{ is of odd length and contains an even number of } a\text{'s}\}$ is regular.

- Let $A_1 = \{w | w \text{ is of odd length}\}$ Let $A_2 = \{w | w \text{ has an even number of } a\text{'s}\}$
- Since DFAs exist to accept A_1 and A_2, both are RLs
- $A = A_1 \cap A_2$. By the CP of RLs under intersection, $A_1 \cap A_2$ is RL. So A is RL

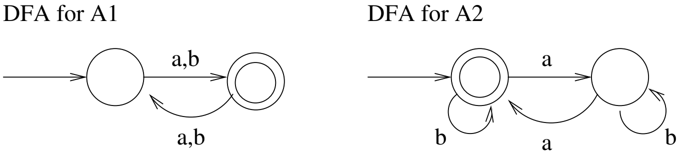


Figure 16: DFAs for A_1 and A_2

3.1 Definition of REs (*Sipser 1.3 (pp. 63-66)*)

Regular expressions (REs) are to represent regular languages. Let $L(R)$ be the language that regular expression R represents. A recursive definition is given below:

- **Basis**: ε and $\emptyset$ are REs, and $L(\varepsilon) = \{\varepsilon\}$ and $L(\emptyset) = \emptyset$. For any $a \in \Sigma$, a is an RE and $L(a) = \{a\}$.
- **Induction**: If R_1 and R_2 are REs, then
  - $R_1 \cup R_2$ is an RE, with $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$,
  - $R_1 R_2$ is an RE, with $L(R_1 R_2) = L(R_1)L(R_2)$,
  - R_1^* is an RE, with $L(R_1^*) = (L(R_1))^*$, and
  - (R_1) is an RE, with $L((R_1)) = L(R_1)$.

Remark:

- Precedence order for regular-expression operators: Star, concatenation, and finally union. () may override this order.
- Use of R^+ and R^k.
- Algebraic laws:
  - $R_1 \cup R_2 = R_2 \cup R_1$, $(R_1 \cup R_2) \cup R_3 = R_1 \cup (R_2 \cup R_3)$, and $(R_1 R_2) R_3 = R_1 (R_2 R_3)$.
  - $\emptyset \cup R = R \cup \emptyset = R$, $\varepsilon R = R\varepsilon = R$, $\emptyset R = R\emptyset = \emptyset$, and $R \cup R = R$.
  - $R_1 (R_2 \cup R_3) = R_1 R_2 \cup R_1 R_3$ and $(R_1 \cup R_2) R_3 = R_1 R_3 \cup R_2 R_3$.
  - $(R^*)^* = R^*$, $\emptyset^* = \varepsilon$, $R^+ = RR^* = R^* R$, and $R^* = R^+ \cup \varepsilon$.

3.2 Understanding REs

- RE is a pattern for all strings in a RL. The goal is to make the RE as simple and readable as possible. Consider the following examples to simplify REs
- $1 \cup 10^* \Rightarrow 10^*$
- $(0^*1^*)^* \Rightarrow (0 \cup 1)^*$
- $((0 \cup 1)(0 \cup 1)^*)^* \Rightarrow (0 \cup 1)^*$

Given a language, design its RE

Example 1: $\{w$ has no substring 10$\}$: 0^*1^*
Example 2: $\{w$ has even number of 1's$\}$: $(0^*10^*10^*)^* \cup 0^*$
Example 3: $\{w$ has odd length$\}$: $((0\cup 1)(0\cup 1))^*(0\cup 1)$
Example 4: $\{$There is a 1 in the 3rd position to the right end$\}$: $(0\cup 1)^*1(0\cup 1)(0\cup 1)$
Example 5: $\{w$ has a 1 in 3rd or 2nd position from right end$\}$:
$(0\cup 1)^*1(0\cup 1)(0\cup 1)\cup(0\cup 1)^*1(0\cup 1) \Rightarrow$
$(0\cup 1)^*1(0\cup 1)((0\cup 1)\cup\varepsilon) \Rightarrow$
$(0\cup 1)^*1(0\cup 1)(0\cup 1\cup\varepsilon)$
Example 6: A language of strings that consists of alternating 0s and 1s: $(01)^*\cup(10)^*\cup 0(10)^*\cup 1(01)^*$.

Example 7: $D = \{w$ has odd number of alternating blocks of 0s and 1's$\}$ (Note: $\varepsilon \notin D$)

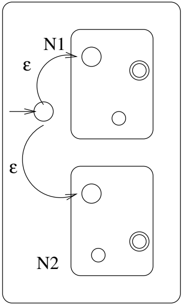
- $0|1|0 \in D$, $0|1|0|1 \notin D$, $11|00|111 \in D$, $0|11|000|11|0|1 \notin D$
- Observation 1: Odd number of blocks implies strings in D must start and end with the same symbol.
- RE for D based on ob.1: $0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1$
- Observation 2: Draw boundaries between blocks. Near each boundary, we see substring 01 or 10. A string in D must have equal number of substrings of 01 and 10.
- RE for D based on ob.2: $0^+(1^+0^+)^* \cup 1^+(0^+1^+)^*$

Example 8: $E = \{w|$ In w, each 1 is immediately preceeded by a 0 and followed by a 0$\}$

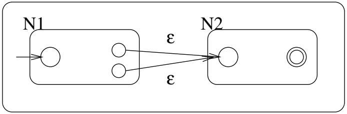
- Look for substring 010
- $0010001000010010 \in E$. Rewrite the string as $(001)(0001)(00001)(001)(0) \Rightarrow$ $(0^{+}1)(0^{+}1)(0^{+}1)(0^{+}1)(0^{+})$
- RE: $(0^{+}1)^{*}0^{+} \cup \varepsilon$ or equally correct, $0^{+}(10^{+})^{*} \cup \varepsilon$

Proof of Closure Properties of RLs

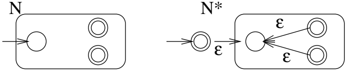
Union, concatenation and star:



Union

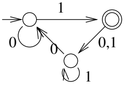


Concat

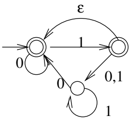


Star

An example:



100 not accepted



100 accepted

RE $\Rightarrow$ NFA (*Sipser 1.3 pp. 67-69*)

Since REs are defined recursively, it is suitable to construct the equivalent NFAs recursively.

- **Basis**: NFAs for simple RE: ε, $\emptyset$, and a for $a \in \Sigma$.
- **Induction**: Given the NFAs for REs R_1 and R_2, what are the NFAs for $R_1 \cup R_2$, $R_1 R_2$, and R_1^*?

Example 1: Converting RE $(0^*10^*10^*)^*$ to an NFA (which NFA is correct?)

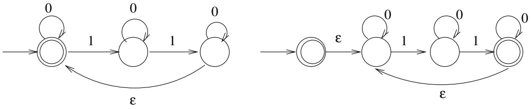


Figure 17: Converting a RE to an NFA

Example 2: Converting RE $(00^*1)^*(\varepsilon\cup 10^*)$

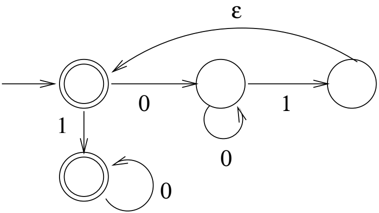


Figure 18: Converting a RE to an NFA

Theorem: DFA, NFA, and RE are equivalent ways to accept or represent RLs. In particular, RE⇒NFA, NFA⇒DFA, DFA⇒RE (will not discuss in our class)

4.1 Regular versus nonregular languages

- $A = \{0^*1^*\}$ (Regular)
- $B = \{0^n1^n | n \geq 0\}$ (Non-regular)
- $C = \{w \in \{0,1\}^* | w \text{ has an equal number of 0s and 1s}\}$ (Non-regular)
- $D = \{w \in \{0,1\}^* | w \text{ has an equal \# of substrings 01 and 10}\}$ (Regular)
  Surprisingly, D is regular, whose RE is
  $0^+(1^+0^+)^* \cup 1^+(0^+1^+)^*$ if $\varepsilon \notin D$.

The Pumping Lemma for RLs: (Sipser 1.4,77-82) For any RL A, there is p (whose value depends on A) such that $\forall s \in A$ with $|s| \geq p$, s can be partitioned into three substrings $s = xyz$ s.t.

1. $|y| > 0$; (y cannot be the empty string ε)
2. $|xy| \leq p$; and
3. $\forall i \geq 0$, string $xy^iz \in A$. (Note: $xy^0z = xz$, $xy^2z = xyyz$)

The pumping lemma for RLs is a lemma that describes an essential property of all RLs. Informally, any sufficiently long strings in a RL may be pumped, or have a middle section of the string repeated any number of times to produce a new string that is also part of the RL.

The pumping lemma is useful for disproving the regularity of a specific language in question. It was first proven by Michael Rabin and Dana Scott in 1959 and rediscovered later by Yehoshua Bar-Hillel, et. al. in 1961 as a simplification of their pumping lemma for context-free languages.

How to use the pumping lemma to prove that a language A is not regular:

- (1)Assume that A is regular by contradiction.
- (2)Then the pumping lemma applies to A.
- (3) Let p be the constant in the pumping lemma. (three steps to start the proof)
- Select $s \in A$ with $|s| = f(p) \geq p$.
- By the pumping lemma, $\exists x, y, z$ such that $s = xyz$ with $|y| > 0$, $|xy| \leq p$ and $xy^iz \in A$ for any $i \geq 0$.
- For any x, y, z such that $s = xyz$, $|y| > 0$, and $|xy| \leq p$, find $i \geq 0$ such that $xy^iz \notin A$. A contradiction!

Example 1 (Sipser p. 80): Prove $B = \{0^n 1^n | n \geq 0\}$ is non-RL.

How to use the PL to prove that a language B is not regular:

- (1) (2) (3) Three sentences to start the proof.
- Select $s \in B$ with $|s| = f(p) \geq p$.
- By the PL, $\exists x, y, z$ s.t. $s = xyz$ with $|y| > 0$, $|xy| \leq p$ and $xy^i z \in B$ for any $i \geq 0$.
- For any x, y, z such that $s = xyz$, $|y| > 0$, and $|xy| \leq p$, find $i \geq 0$ s.t. $xy^i z \notin B$. A contradiction!

  ****

- Assume B is RL. Then PL applies to B. Let p be the constant in PL.
- Select $s = 0^p 1^p \in B$ with $|s| = 2p > p$
- By PL, $s = \underbrace{0\ldots0}_{p}\underbrace{1\ldots1}_{p} = xyz$, $|y| > 0$ and $|xy| \leq p$
- Since $y \neq \varepsilon$ and $|xy| \leq p$, then $y = 0^+$
- Choose $i = 0$. Then $xy^0 z = xz = 0^{p'} 1^p$ for $p' < p$
- So $xy^0 z \notin B$. A contradiction to PL. So B is non-R.

Example 2 (Sipser p. 81): Prove that $F = \{ww | w \in \{0,1\}^*\}$ is not regular.

- Assume F is RL. Then PL applies to F. Let p be the constant in PL.
- Select $s = 0^p10^p1 \in F$ with $|s| = 2p+2 > p$
- By PL, $s = \underbrace{0\ldots0}_{p}1\underbrace{0\ldots0}_{p}1 = xyz$, $|y| > 0$, $|xy| \leq p$
- Since $y \neq \varepsilon$ and $|xy| \leq p$, then $y = 0^+$
- Choose $i = 2$. Then $xy^2z = xyyz = 0^{p'}10^p1$ for $p' > p$
- So $xy^2z \notin F$. A contradiction to PL. So F is non-R.

Example 3: Prove that $A = \{1^r | r \text{ is a prime}\}$ is not regular.

- Some strings in A: 11, 111, 11111, 1111111, etc..
- Assume A is RL. Then PL applies. Let p be the constant.
- Select $s = 1^q$, where q is a prime and $q \geq p$
- By PL, $s = \underbrace{1 \ldots 1}_{q} = xyz$, $x = 1^{h_1}$ ($h_1 \geq 0$), $y = 1^{h_2}$ ($h_2 > 0$), $z = 1^{q-h_1-h_2}$
- By PL, $\forall i \geq 0$, $xy^iz = 1^{h_1+i\cdot h_2+(q-h_1-h_2)} = 1^{(i-1)h_2+q} \in A$
- Choose $i = q+1$. Then $xy^iz = xy^{q+1}z = 1^{q(h_2+1)} \notin A$
- A contradiction to PL. So A is non-R.

Note: $|xy^iz| = xy^{q+1}z = 1^{h_1+ih_2+q-h_1-h_2} = 1^{q+h_2(i-1)} = 1^{q(1+h_2)}$

Example 4: (Sipser p. 82) Prove that $D = \{1^{n^2} | n \geq 1\}$ is non-R.

- Some strings in D: 1, 1111, 111111111, etc.
- Assume D is RL. Then PL applies. Let p be the constant
- Select $s = 1^{p^2} \in D$, $|s| = p^2 \geq p$
- By PL, $s = \underbrace{1 \ldots 1}_{p^2} = xyz$, $x = 1^{h_1}$ $(h_1 \geq 0)$, $y = 1^{h_2}$ $(h_2 > 0)$, $z = 1^{p^2 - h_1 - h_2}$, and $h_2 \leq |xy| \leq p$ (Note: h_2 is length of y)
- By PL, $\forall i$, $xy^i z \in D$
- Choose $i = 2$. Consider $|xy^2z|$ $= h_1 + 2h_2 + (p^2 - h_1 - h_2) = h_2 + p^2$. A perfect square?
- Some algebra:
  $p^2 = 0 + p^2 < (h_2 + p^2) \leq p + p^2 < 1 + 2p + p^2 = (p+1)^2$
  $p^2 < |xy^2z| < (p+1)^2$. So $|xy^2z|$ is not a perfect square.
  Then $xy^2z \notin D$. A contradiction to PL. So D is non-R.

Example 5: Prove that $A = \{10^n1^n \mid n \geq 0\}$ is not regular.

- Assume A is RL. The PL applies. Let p be the constant.
- Select $s = 10^p1^p \in A$. $|s| = 2p + 1 > p$
- By PL, $s = 1\underbrace{0\ldots0}_{p}\underbrace{1\ldots1}_{p} = xyz$, $|y| > 0$, $|xy| \leq p$
- By PL, $\forall i$, $xy^iz \in A$. Consider two cases for y.
  - Case 1. y contains the first 1: $x = \varepsilon$, $y = 10^*$.
    Choose $i = 0$. $xy^0z = xz = 0^{p'}1^p \notin A$, for $p' \leq p$
  - Case 2. y does not contain the first 1: $x = 10^*$, $y = 0^+$.
    Choose $i = 0$. $xy^0z = xz = 10^{p'}1^p \notin A$, for $p' < p$
- For both cases, we have found contradiction to PL. So A is non-R.

Example 6: Prove that $A = \{(01)^a 0^b | a > b \geq 0\}$ is not regular.

- Assume A is RL. The PL applies. Let p be the constant.
- Select $s = (01)^p 0^{p-1} \in A$. $|s| = 3p - 1$.
- By PL, $s = \underbrace{01\ldots01}_{2p}\underbrace{0\ldots0}_{p-1} = xyz$, $|y| > 0$, $|xy| \leq p$
- Consider y which is entirely in $(01)^p$.
  - Case 1: Even length, i.e., $y = 01\ldots01$ or $y = 10\ldots10$. Choose $i = 0$ to remove at least a substring 01 or 10, violating $a > b$
  - Case 2: Odd length, i.e., $y = 01\ldots10$ or $y = 10\ldots01$ or $y = 0$ or $y = 1$. Choose $i = 2$ to create substrings of 00 or 11, which is not allowed .
- In each case listed, we can find an i such that $xy^iz \notin A$. A contradiction to the PL. So A is non-R.

4.2 Prove nonregularity by closure properties

To prove that A is non-regular, assume it is regular. Find a regular language B and a language operator that preserves regularity, and then apply the operator on A and B to get a regular language C. If C is known to be non-regular, a contradiction is found.

Example 1: Prove that
$C = \{w \in \{0,1\}^* | w \text{ has an equal \# of 0s and 1s}\}$ is not regular.

- Assume C is regular
- Let $B = \{0^*1^*\}$. B is known to be RL
- Let $D = C \cap B = \{0^n1^n\}$. D is known to be non-R
- But D is regular by CP under intersection
- A contradiction! So C is non-R

Example 2: Prove that $A = \{0^m1^n \mid m \neq n\}$ is not regular.

- Assume A is RL
- $\{0^*1^*\} = \{0^n1^n\} \cup A$ (two disjoint sets)
- $\{0^n1^n\} = \{0^*1^*\} - A$
- Since $\{0^*1^*\}$ and A are both RL, the difference of the two RLs is still RL by CP.
- So $\{0^n1^n\}$ is RL. A contradiction.
- A is non-R

****

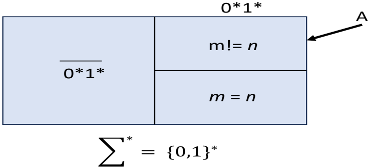


Figure 19: Second method: $\{0^*1^*\} - A = \{0^n1^n\}$

Example 3: Prove that $A = \{a^m b^n c^{m+n} \mid m, n \geq 0\}$ is non-R

About homomorphism:

$h : \Sigma \to (\Sigma_1)^*$, e.g., $h(a) = 01$, $h(b) = 0$, $h(c) = \varepsilon$

$h : \Sigma^* \to (\Sigma_1)^*$, e.g., $h(ab) = 010$

$h : h(A) = \{h(w) \mid \forall w \in A\}$

- Assume A is RL.
- Define homomorphism h such that $h(a) = 0$, $h(b) = 0$, $h(c) = 1$
- $h(A) = \{0^m 0^n 1^{m+n}\} = \{0^{m+n} 1^{m+n}\} = \{0^n 1^n\}$
- By CP under homomorphism, since A is RL, so is $\{0^n 1^n\}$
- A contradiction. So A is non-R