
Decision Properties of RLs & Automaton Minimization

Martin Fränzle

Informatics and Mathematical Modelling
The Technical University of Denmark

02140 Languages and Parsing, MF, Fall 2003 – p.1/16

What you'll learn

1. Decidable properties of DFA, NFA, ε -NFA, RE:

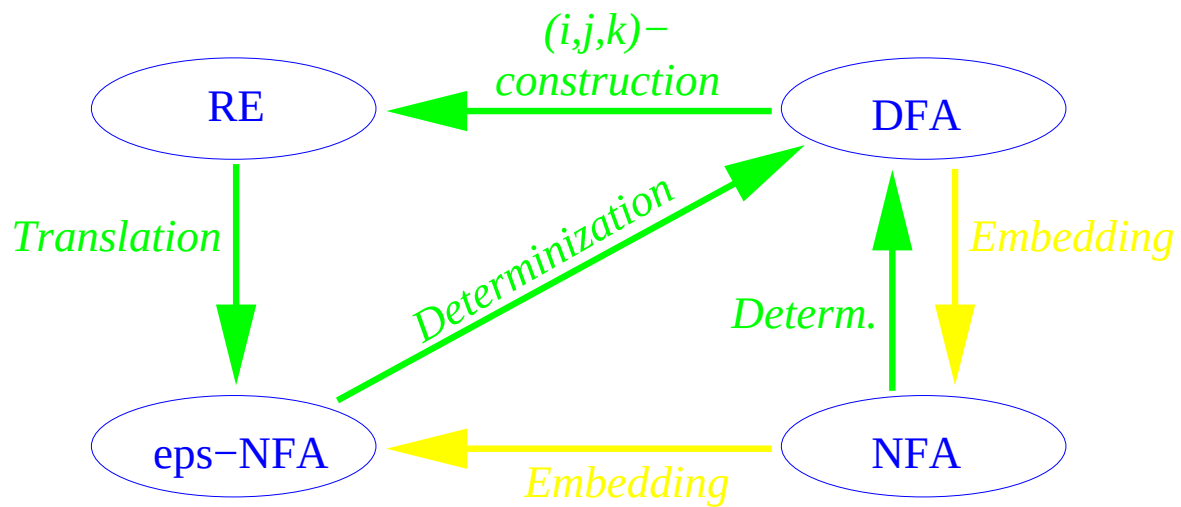
- Language emptiness, universality, inclusion, ...

2. Automaton minimization:

- An algorithm for reducing the state set of DFA
- Applicability to NFA

02140 Languages and Parsing, MF, Fall 2003 – p.2/16

Converting between descriptions



02140 Languages and Parsing, MF, Fall 2003 – p.3/16

Worst-case cost of conversion

- ϵ -NFA to NFA: $O(n^3)$, where n is the number of states
- NFA to DFA: $O(2^n)$
- DFA to NFA: $O(n)$
- DFA to RE: $O(n^3 4^n)$, since there are n^3 REs to be constructed and each of the rounds can quadruple the size,
- RE to ϵ -NFA: $O(n)$, where n is the length of the RE

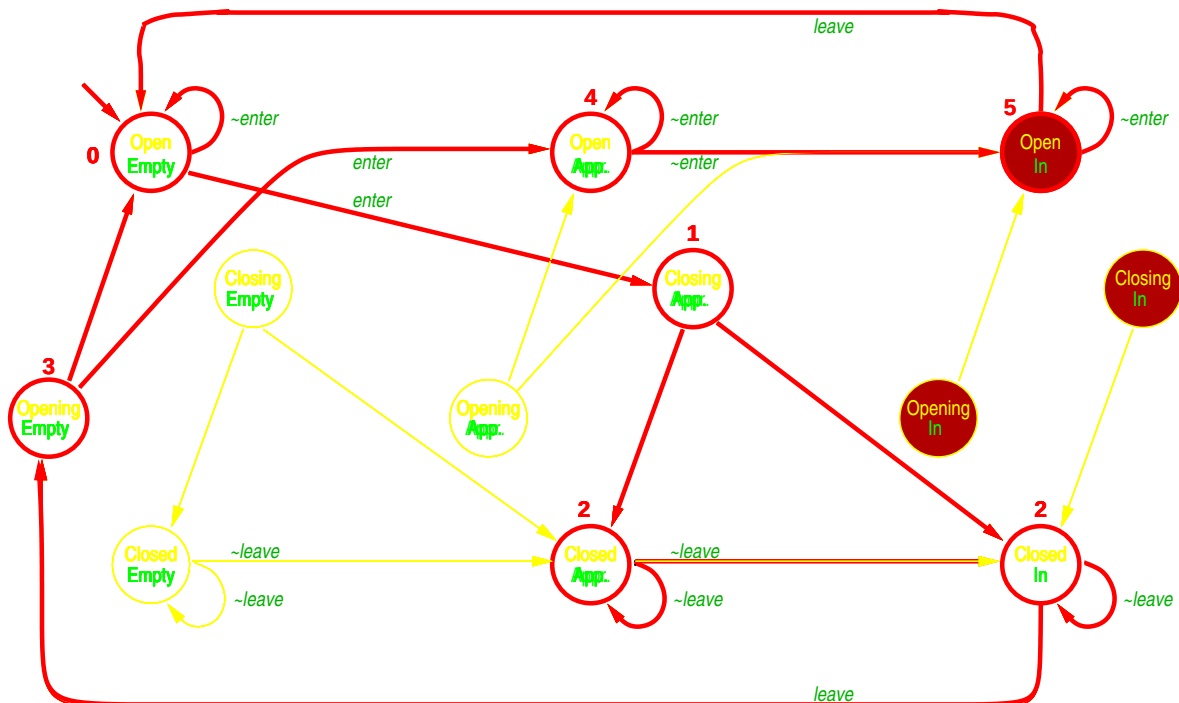
02140 Languages and Parsing, MF, Fall 2003 – p.4/16

Decision procedures:

Language emptiness

02140 Languages and Parsing, MF, Fall 2003 – p.5/16

Emptiness check



02140 Languages and Parsing, MF, Fall 2003 – p.6/16

The coloring algorithm

The set R of reachable states of a (D/N/ ϵ N)FA $(Q, \Sigma, \delta, q_0, F)$ is the *smallest* set satisfying

Base case: $q_0 \in R$,

Recursion: if $p \in R$ and there is $a \in \Sigma$ ($a \in \Sigma \cup \{\epsilon\}$, respectively) s.t. $q = \delta(p, a)$ (or $q \in \delta(p, a)$ respectively) then $q \in R$.

Adding some bookkeeping to avoid reexamination of already visited states, above rules yield an algorithm for computing the reachable states.

Complexity: $O(m)$, where m is the number of state pairs connected via transitions in the automaton; $m \leq n^2$ when n is the number of states.

02140 Languages and Parsing, MF, Fall 2003 – p.7/16

Direct emptiness test on REs

Base case: For atomic REs, we have

$$\text{empty}(\emptyset) = \text{true}$$

$$\text{empty}(\epsilon) = \text{false}$$

$$\text{empty}(a) = \text{false}$$

Recursion: For compound REs:

$$\text{empty}(R + S) = \text{empty}(R) \wedge \text{empty}(S)$$

$$\text{empty}(RS) = \text{empty}(R) \vee \text{empty}(S)$$

$$\text{empty}(R^*) = \text{false}$$

$$\text{empty}((R)) = \text{empty}(R)$$

02140 Languages and Parsing, MF, Fall 2003 – p.8/16

Membership test

Whether a string $w \in \Sigma^*$ is in the language $L(A)$ of some automaton A can be tested by “running the automaton on w ”.

Other decidable properties

Due to the closure properties of regular languages, the emptiness test can be used to decide

- **universality** — is $L(A) = \Sigma^*$?
- **Kleene-closedness** — is $L(A)^* = L(A)$?
- **inclusion** — is $L(A) \subseteq L(B)$?
- **disjointness** — is $L(A) \cap L(B) = \emptyset$?
- **image** — is $h(L(A)) = L(B)$ for some given homomorphism h ?

Furthermore, **(in-)finiteness of $L(A)$** is decidable, e.g. by analysing the RE — does it contain a Kleene star over a language different from $\emptyset$ and ε ?

Automaton Minimization

02140 Languages and Parsing, MF, Fall 2003 – p.11/16

State equivalence

Def: Two states p and q of a DFA are called **equivalent** iff

$$\hat{\delta}(p, w) \in F \text{ iff } \hat{\delta}(q, w) \in F$$

holds for all $w \in \Sigma^*$.

Deciding equivalence: The set of *distinguishable state pairs* is the smallest set $D \subseteq Q \times Q$ satisfying

Base case: If $p \in F$ and $q \notin F$ or vice versa then $(p, q) \in D$.

Recursion: If $(r, s) \in D$ and there is $a \in \Sigma$ s.t $\delta(p, a) = r$ and $\delta(q, a) = s$ then $(p, q) \in D$.

Theorem: p and q are equivalent iff $(p, q) \notin D$.

02140 Languages and Parsing, MF, Fall 2003 – p.12/16

Automaton minimization

Obviously, equivalent states can be merged into a single one, as no string distinguishes the two.

We will now see that the resulting automaton is **minimal** in the following sense:

1. There is no DFA with fewer states accepting the same language.
2. It is “isomorphic” (in the sense that the automata differ only in their state names) to all other DFAs with the same number of states that accept the same language.

02140 Languages and Parsing, MF, Fall 2003 – p.13/16

The minimal automaton

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, R its set of reachable states, and let $E = (Q \times Q) \setminus D$ be the set of its **equivalent state pairs**.

E induces **equivalence classes** on Q : $\llbracket p \rrbracket = \{q \in Q \mid (p, q) \in E\}$.

The automaton $M = (Q', \Sigma, \delta', F')$ defined by

- $Q' = \{\llbracket p \rrbracket \mid p \in R\}$,
- $\delta'(\llbracket p \rrbracket, a) = \llbracket \delta(p, a) \rrbracket$,
- $F' = \{\llbracket p \rrbracket \mid p \in F \cap R\}$,

is called the *minimization of A* .

M is well-defined as $(p, q) \in E$ implies $(\delta(p, a), \delta(q, a)) \in E$.

02140 Languages and Parsing, MF, Fall 2003 – p.14/16

Minimality

Lemma: If A is a DFA and M the DFA constructed from A by the minimization algorithm then

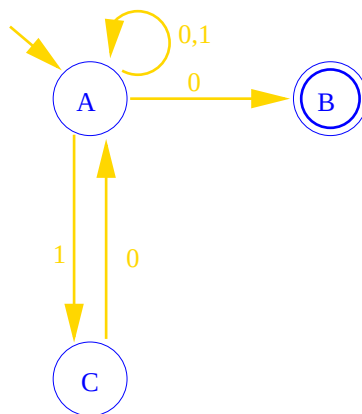
1. $L(A) = L(M)$,
2. $|Q_A| \geq |Q_M|$.

Theorem: If A is a DFA and M the DFA constructed from A by the minimization algorithm then

3. $|Q_B| \geq |Q_M|$ for each DFA B with $L(B) = L(M)$.

02140 Languages and Parsing, MF, Fall 2003 – p.15/16

The NFA



has no equivalent states.

Yet, it is non-minimal: state C is redundant.

Minimization algorithm based on state equiv. applies to DFA only!

02140 Languages and Parsing, MF, Fall 2003 – p.16/16