BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023

REGULATED LANGUAGE OPERATIONS AND THEIR USE

David Chocholatý

Supervisor: prof. RNDr. Alexander Meduna, CSc.

General Jumping Finite Automata

Definition 1.1. A *general jumping finite automaton* (a GJFA for short) is a quintuple

$$M = (Q, \Sigma, R, s, F),$$

where Q is a finite set of *states*, Σ is the *input alphabet*, $\Sigma \cap Q = \emptyset$, $R \subseteq Q \times \Sigma^* \times Q$ is a finite relation, $s \in Q$ is the *start state*, and $F \subseteq Q$ is a set of *final states*. Members of R are referred to as *rules* of M.

Definition 1.2. Let $M = (Q, \Sigma, R, s, F)$ be a GJFA. A *configuration* of M is any string in $\Sigma^* Q \Sigma^*$. The binary *jumping relation*, symbolically denoted by $\curvearrowright_M$, over $\Sigma^* Q \Sigma^*$, is defined as follows. Let $x, z, x', z' \in \Sigma^*$ such that $xz = x'z'$ and $py \rightarrow q \in R$; then, M makes a *jump* from $xpyz$ to $x'qz'$, written as

$$xpyz \curvearrowright_M x'qz'.$$

Definition 1.3. Let $M = (Q, \Sigma, R, s, F)$ be a GJFA. The *language accepted* by M, denoted by $L(M)$, is defined as

$$L(M) = \{uv \mid u, v \in \Sigma^*, usv \curvearrowright_M^* f, f \in F\}.$$

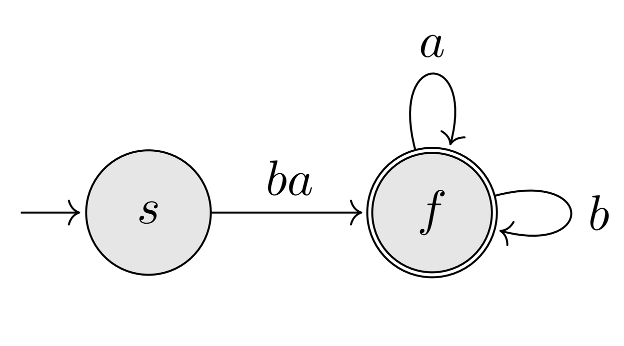

Figure 1.1: An example of a GJFA state control.

Erasing Systems

Definition 2.1. An *erasing system* (an ES for short) is a triplet

$$ES = (\Sigma, E, R),$$

where Σ is the *alphabet* that always contains a special symbol, $\#$, $E \subseteq (\Sigma - \{\#\})^*$ is a finite set of *erasing strings*, and $R \subseteq (\Sigma - \{\#\})^*$ is a *regular language*.

Definition 2.2. Let $ES = (\Sigma, E, R)$ be an ES. Set $\overline{\Sigma} = \Sigma - \{\#\}$. A *configuration* of ES is any pair $(u, v) \in \overline{\Sigma}^* \{\#\} \overline{\Sigma}^* \times \overline{\Sigma}^*$. Let for every string $u \in E$ and for every string $x, y, x', y', z \in \overline{\Sigma}^*$ such that $xy = x'y'$,

$$(x\#uy, z) \triangleright_{ES} (x'\#y', zu)$$

be an *erasing step*, where $\# \in \Sigma$.

Definition 2.3. Let $ES = (\Sigma, E, R)$ be an ES and set $\overline{\Sigma} = \Sigma - \{\#\}$. The *language accepted* by ES, denoted by $L(E, R)$, is defined as

$$L(E, R) = \{uv \mid u, v \in \overline{\Sigma}^*, (u\#v, \varepsilon) \triangleright_{ES}^* (\#, w), \# \in \Sigma, w \in R\}.$$

Example 2.1. Consider the ES, $ES = (\Sigma, E, R)$, where $\Sigma = \{a, b, c\}$, $E = \{a, b, c\}$, and $R = \{abc\}^*$. The language

$$L(E, R) = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$$

is the language accepted by ES. Then, for the input string, $w = ababcc$, it is possible to make the erasing steps

$$\begin{aligned}(ab\#abcc, \varepsilon) &\triangleright_{ES} (a\#bbcc, a) \triangleright_{ES} (abc\#c, ab) \triangleright_{ES} (\#abc, abc)\\ &\triangleright_{ES} (\#bc, abca) \triangleright_{ES} (\#c, abcab) \triangleright_{ES} (\#, abcabc),\end{aligned}$$

from which it follows $abcabc \in R$ which implies $w \in L(E, R)$.

Relations with Dyck Languages and SHUF Languages

Theorem 3.1. $\mathcal{D} \subset$ **ES**.

Theorem 3.2. $\mathcal{S} \subset$ **ES**.

Theorem 3.3. **ES** *and* **SHUF** *are incomparable.*

Relations with Well-Known Language Families

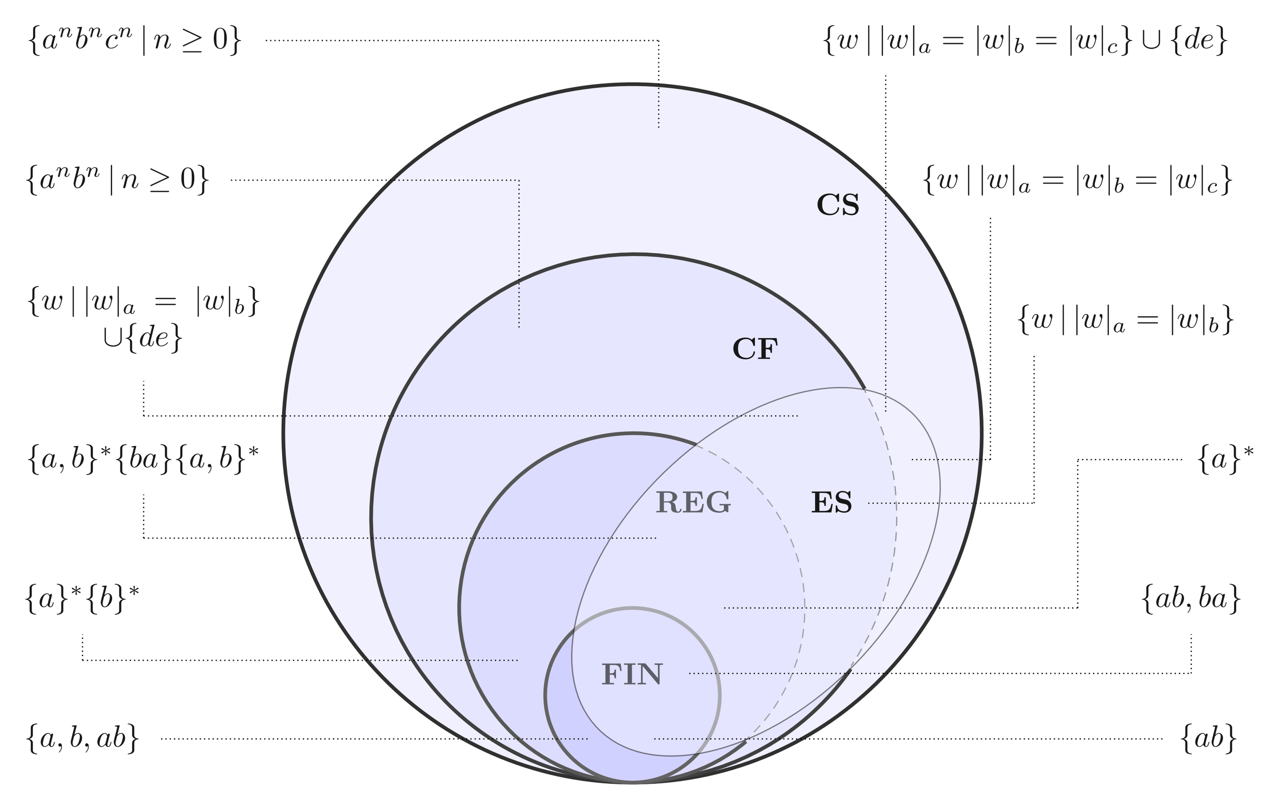

Figure 4.1: Relationship of the family of languages accepted by erasing systems (**ES**) with language families according to Chomsky's hierarchy.

Closure Properties

Table 5.1: Closure properties of families of languages accepted by erasing systems (**ES**) and general jumping finite automata (**GJFA**). REG stands for the family of regular languages.

	ES	**GJFA**	**REG**
Endmarking	–	–	+
Concatenation	–	–	+
Complement	–	–	+
Union	–	+	+
Intersection	–	–	+
Intersection with regular languages	–	–	+
Substitution	–	–	+
Regular substitution	–	–	+
Finite substitution	–	–	+
Homomorphism	–	–	+
ε-free homomorphism	–	–	+
Inverse homomorphism	–	–	+
Kleene star	–	–	+
Kleene plus	–	–	+
Shuffle	–	–	+
Reversal	–	+	+

Applications of the Erasing Systems

- Searching for sequences in DNA, RNA and RNA secondary structure
- Searching for proteins and amino acid sequences of proteins
- Properties of RNA secondary structure
- Balanced brackets in text editors
- Validation of the solution to problem n queens

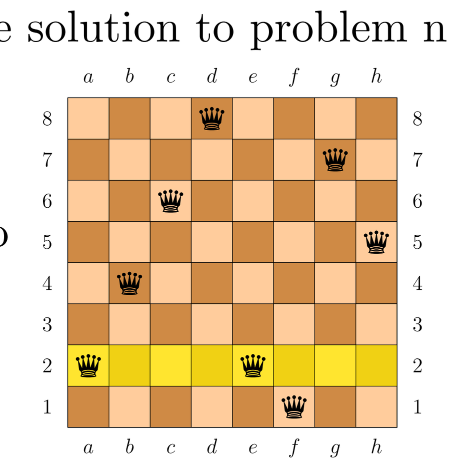

Figure 6.1: Solution with two queens in one row.

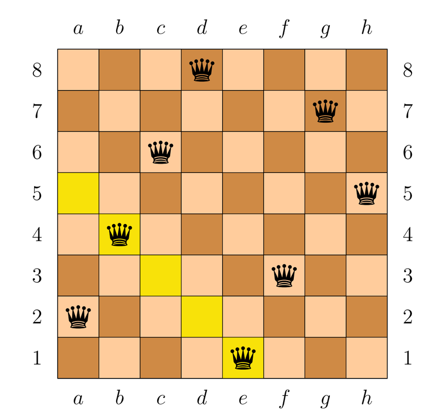

Figure 6.2: Solution with two queens in one diagonal.

Implementation of the Erasing System and its Applications

Input: input string $w \in \overline{\Sigma}^*$, $v = \varepsilon \in \overline{\Sigma}^*$, $w' = w \in \overline{\Sigma}^*$ and $v' = v \in \overline{\Sigma}^*$, where $\overline{\Sigma} = \Sigma - \{\#\}$, $test\_all\_positions \in \{True, False\}$.
Output: *True* iff $w \in L(E, R)$, *False* otherwise.

```
procedure IS_ACCEPTED(w', v')
    if w' = ε then
        if v' ∈ R then
            return True
        else
            return False
    U ← ∅
    for all u ∈ E do
        if w' = xuy and v'uz ∈ R and x, y, z ∈ (Σ − {#})* then
            U ← U ∪ {u}
    while U ≠ ∅ do
        u' ← random(U)
        P ← all_overlapping_positions(w', u')
        while P ≠ ∅ do
            p' ← random(P)
            w'' ← erase(w', p')
            v'' ← v'u'
            if IS_ACCEPTED(w'', v'') then
                return True
            else if test_all_positions = True then
                P ← P − {p'}
            else
                P ← ∅
        U ← U − {u'}
    return False
```

Algorithm 7.1: Algorithm of operation of erasing system accepting language $L(E, R)$.

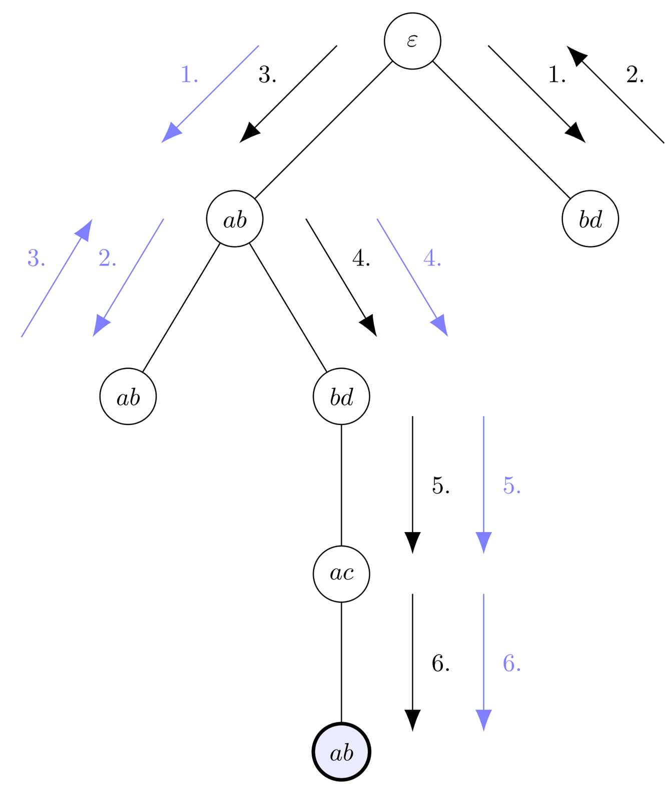

Figure 7.1: Accepting the string, $w = abdaabcb$, using two quantifier types (greedy — blue, lazy — black).