Universitetet i Oslo
Institutt for Informatikk
PMA, PSE
Krogdahl, Møller-Pedersen, Steffen

INF 5110: Compiler construction

Spring 2016 — **Handout 2** — 26. 01. 2016

Handout 2: Scanning etc.

Issued: 26. 01. 2016

Definition 1 (Alpabet Σ) Finite set of elements called "letters" or "symbols" or "characters".

Definition 2 (Words and languages) Given alphabet Σ, a *word* over Σ is a finite sequence of letters from Σ. A *language* over alphabet Σ is a *set* of finite *words* over Σ.

Definition 3 (Regular expressions) A *regular expression* is one of the following

1. a *basic* regular expression of the form $\mathbf{a}$ (with $a \in \Sigma$), or $\boldsymbol{\epsilon}$, or $\boldsymbol{\emptyset}$
2. an expression of the form $r \mid s$, where r and s are regular expressions.
3. an expression of the form $r\,s$, where r and s are regular expressions.
4. an expression of the form r^*, where r is a regular expression.
5. an expression of the form (r), where r is a regular expression.

Definition 4 (Regular expression) Given an alphabet Σ. The meaning of a regexp r (written $\mathcal{L}(r)$) over Σ is given by equation (1).

$$
\begin{array}{rcll}
\mathcal{L}(\boldsymbol{\emptyset}) & = & \{\} & \text{empty language} \\
\mathcal{L}(\boldsymbol{\epsilon}) & = & \epsilon & \text{empty word} \\
\mathcal{L}(\boldsymbol{a}) & = & \{a\} & \text{single "letter" from } \Sigma \\
\mathcal{L}(r \mid s) & = & \mathcal{L}(r) \cup \mathcal{L}(s) & \text{alternative} \\
\mathcal{L}(r^*) & = & \mathcal{L}(r)^* & \text{iteration}
\end{array} \tag{1}
$$

Definition 5 (FSA) A FSA $\mathcal{A}$ over an alphabet Σ is a tuple $(\Sigma, Q, I, F, \delta)$

- Q: finite set of states
- $I \subseteq Q$, $F \subseteq Q$: initial and final states.
- $\delta \subseteq Q \times \Sigma \times Q$ transition relation

Definition 6 (DFA) A *deterministic, finite automaton* $\mathcal{A}$ (DFA for short) over an alphabet Σ is a tuple $(\Sigma, Q, I, F, \delta)$

- Q: finite set of states
- $I = \{i\} \subseteq Q$, $F \subseteq Q$: initial and final states.
- $\delta : Q \times \Sigma \to Q$ transition function

Definition 7 (Accepting words and language of an automaton) A word $c_1 c_2 \ldots c_n$ with $c_i \in \Sigma$ is *accepted* by automaton $\mathcal{A}$ over Σ, if there exists states $q_0, q_2, \ldots q_n$ all from Q such that

$$q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} q_2 \xrightarrow{c_3} \quad \ldots \quad q_{n-1} \xrightarrow{c_n} q_n \ ,$$

and were $q_0 \in I$ and $q_n \in F$. The *language* of an FSA $\mathcal{A}$, written $\mathcal{L}(\mathcal{A})$, is the set of all words $\mathcal{A}$ accepts

Definition 8 (NFA (with ϵ transitions)) A *non-deterministic* finite-state automaton (NFA for short) $\mathcal{A}$ over an alphabet Σ is a tuple $(\Sigma, Q, I, F, \delta)$, where

- Q: finite set of states
- $I \subseteq Q$, $F \subseteq Q$: initial and final states.
- $\delta : Q \times \Sigma \to 2^Q$ transition function

In case, one uses the alphabet $\Sigma + \{\epsilon\}$, one speaks about an NFA with ϵ-transitions.

Definition 9 (Acceptance with ϵ-transitions) A word w over alphabet Σ is *accepted* by an NFA with ϵ-transitions, if there exists a word w' which is accepted by the NFA with alphabet $\Sigma + \{\epsilon\}$ according to Definition 7 and where w is w' with all occurrences of ϵ removed.

Definition 10 (ϵ-closure, a-successors) Given a state q, the ϵ-closure of q, written $close_\epsilon(a)$, is the set of states reachable via zero, one, or more ϵ-transitions. We write q_a for the set of states, reachable from q with one a-transition. Both definitions are used analogously for sets of states.

References

[Louden, 1997] Louden, K. (1997). *Compiler Construction, Principles and Practice.* PWS Publishing.