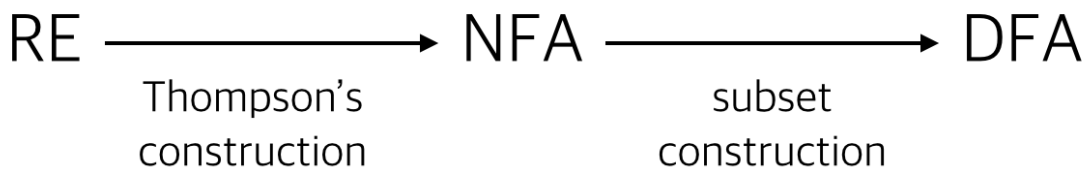
COSE312: Compilers

Lecture 4 — Lexical Analysis (3)

Hakjoo Oh
2015 Fall

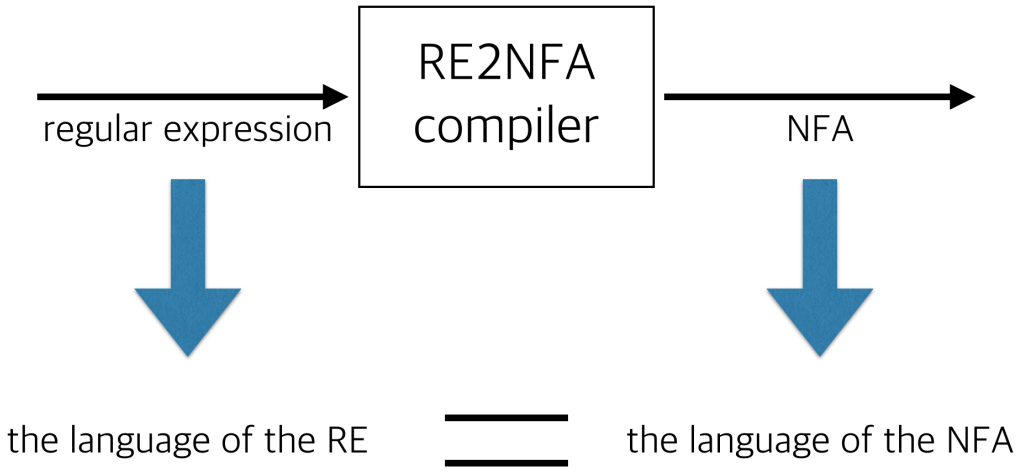
Part 3: Automation

Transform the lexical specification into an executable string recognizers:



From REs to NFAs

Transform a given regular expression into a semantically equivalent NFAs:



An instance of "compilation":

- The source language is regular expressions and the target language is NFAs.
- The correctness is defined by the equivalence of the denoted languages.

Principles of Compilation

Every automatic compilation

1. is done "compositionally", and
2. maintains some "invariants" during compilation.

ex) compilation of regular expressions, e.g., $\boldsymbol{R_1}|\boldsymbol{R_2}$:

1. The compilation of $\boldsymbol{R_1}|\boldsymbol{R_2}$ is defined in terms of the compilation of $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$.
2. Compiled NFAs for $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$ satisfy the invariants:
   - an NFA has exactly only one accepting state,
   - no arcs into the initial state, and
   - no arcs out of the accepting state.

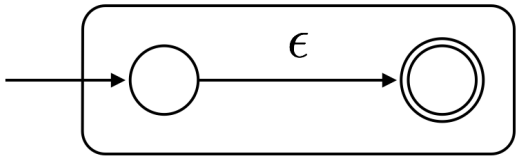
The Source Language

$$
\begin{array}{rcl}
R & \rightarrow & \emptyset \\
 & | & \epsilon \\
 & | & a \in \Sigma \\
 & | & R_1 \mid R_2 \\
 & | & R_1 \cdot R_2 \\
 & | & R_1^* \\
 & | & (R)
\end{array}
$$

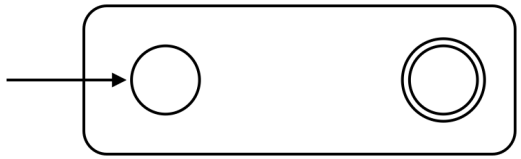
Compilation

Base cases:

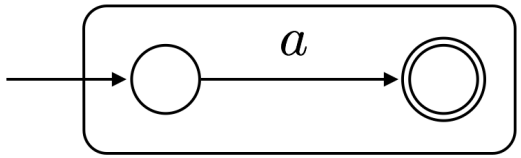
- $\boldsymbol{R = \epsilon}$:



- $\boldsymbol{R = \emptyset}$



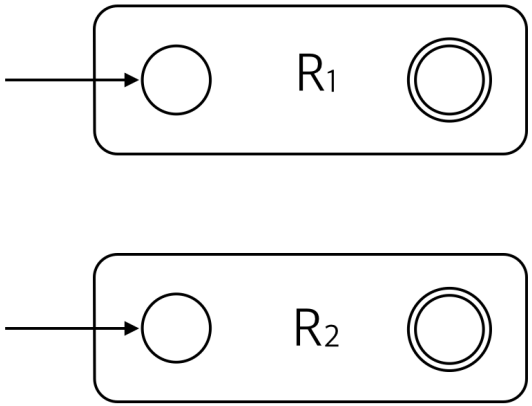
- $\boldsymbol{R = a}$ $(\in \boldsymbol{\Sigma})$



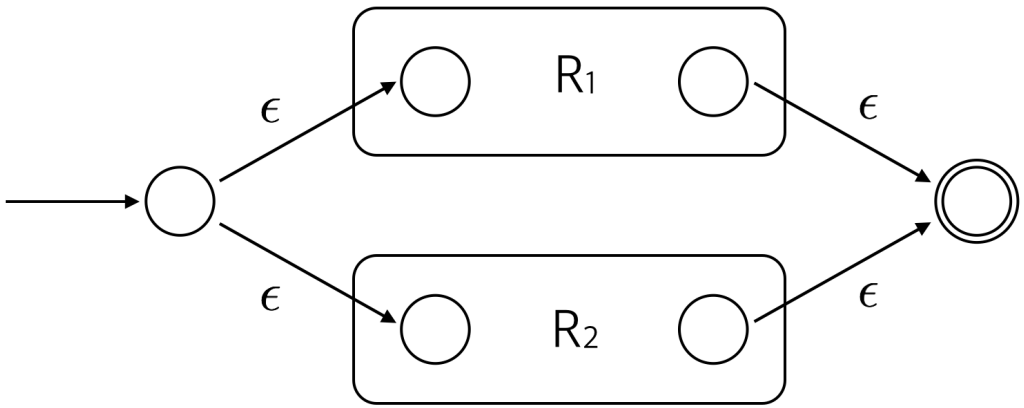
Compilation

Inductive cases:

- $R = R_1|R_2$:
  1. Compile R_1 and R_2:

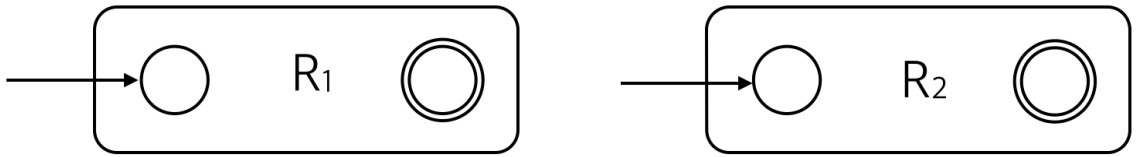


  2. Compile $R_1|R_2$ using the results:

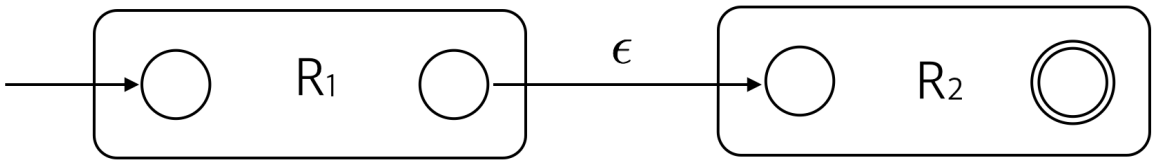


Compilation

- $\boldsymbol{R = R_1 \cdot R_2}$:
  1. Compile $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$:

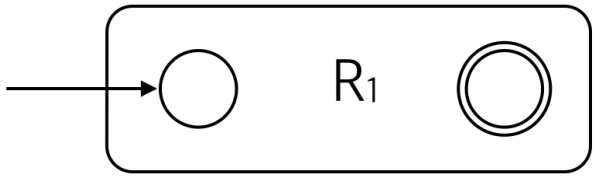


  2. Compile $\boldsymbol{R_1 \cdot R_2}$ using the results:

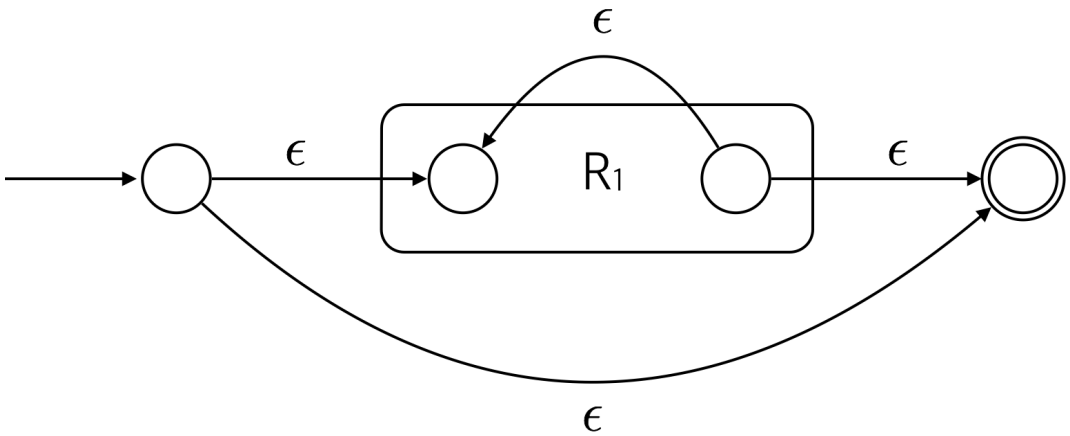


Compilation

- $\boldsymbol{R = R_1^*}$:
  1. Compile $\boldsymbol{R_1}$:



  2. Compile $\boldsymbol{R_1^*}$ using the results:



Examples

- $\mathbf{0 \cdot 1^*}$:
- $\mathbf{(0|1) \cdot 0 \cdot 1}$:
- $\mathbf{(0|1)^* \cdot 1 \cdot (0|1)}$:

Homework 1-1 (due: next class)

Prove that the compilation process is correct; for any regular expression $\boldsymbol{R}$, the compiler produces an NFA $\boldsymbol{N}$ such that

$$\boldsymbol{L(R) = L(N).}$$

Formally prove the equivalence by structural induction.