apl. Prof. Dr. Thomas Noll
Kevin Batz, Ira Fesefeldt

Compiler Construction 2022/23

— Solution -1 —

Credit: https://xkcd.com/208/

General Remarks

- If you have questions regarding the exercises, feel free to write us an email:

  cc22@i2.informatik.rwth-aachen.de

- Exercises are *optional*, i.e., not required for admission to exams. However, corrections to students' solutions are provided as annotations to the submissions.
- You can hand in your solution to the tasks digitaly in the Moodle room. Alternatively, you can hand in your solution to the tasks at our chair in the corresponding box or before the exercise class.
- Please hand in your solutions in *groups of four* and hand in only one solution per group. You can use the forum in this Moodle room to find group members.

Exercise 1 (10+10+15+15 Points)

(a) Describe the language $L((01)^+(10)^+)$ in words.

(b) Construct a regular expression over the alphabet $\Sigma = \{0, 1\}$ for the set of all strings with exactly three 0's,

(c) Construct a regular expression over the alphabet $\Sigma = \{0, 1\}$ for the set of all strings with equal number of 0's and 1's such that no prefix has two more 0's than 1's nor two more 1's than 0's.

(d) Show that there exists a language L such that $L = \{0w \mid w \in L\} \cup \{1\}$ and L is regular.

Solution:

(a) The set of all strings consisting of 0's and 1's starting and ending with 0 and containing exactly one consecutive 1's but no consecutive 0's.

(b) The regular expression is:

$$1^*01^*01^*01^*$$

(c) The regular expression is:

$$(01|10)^*$$

(d) Guess the solution to be 0^*1 (because it feels right).

$$L = \{0w \mid w \in L(0^*1)\} \cup \{1\}$$

By substituting L with $L(0^*1)$ in the equation above, we have:

$$L(0^*1) = \{0w \mid w \in L(0^*1)\} \cup \{1\}$$

We now prove the equivalence of both sets. That is, we assume $w \in L(0^*1)$ and show that $0w$ or 1 is in L and that w can be decomposed into either 1 or $0w'$ such that w' is in $L(0^*1)$.

For the first direction we have:

- $1 \in L(0^01) \subseteq L(0^*1)$,
- For $w \in L(0^*1)$ we assume that $w = 0^n1$. Then we have $0w = 0^{n+1}1$ and thus also $0w \in L(0^{n+1}1) \subseteq L(0^*1)$.

For the other direction we have: $w \in L(0^n1) \subseteq L(0^*1)$ for some n. We prove now that w can be decomposed into either 1 or $0w'$ such that $w' \in L(0^*1)$ by case distinction on n.

- For $n = 0$ we have that $w = 1$ and thus w can be decomposed into 1,
- For $n > 0$ we have that $w = 0^n1$ and thus $w = 0w'$ with $w' = 0^{n-1}1$. Remark that w' is well defined because $n > 0$. Finally, $w' \in L(0^{n-1}1) \subseteq L(0^*1)$.

Exercise 2 (15+15+10+10 Points)

Let Ω be the set of all relevant characters in some encoding, e.g. all UTF-8 characters. In particular $*, /, \backslash, ", ' \in \Omega$.

(a) Provide a regular expression for single-line comments (`//`) and multi-line comments (`/* ... */`) in *WHILE* (c.f. Programming Exercise 1). Please keep in mind that `*` as well as `/` may also occur inside of comments. Note that single-line comments are terminated by either a newline symbol `\n` (Linux), a carriage return `\r` (Mac OS) or both `\r\n` (Windows). Your regular expression should support comments on all three operating systems.

Examples of *valid comments* are:

- `// comment\n`
- `/* comment \r new line */`
- `/*foo * bar / */`
- `// /* test */ more comments\r\n`

Examples of *invalid comments* are:

- `// abc`
- `/*two*//*comments*/`

(b) Provide a regular expression capturing a string. Strings begin and end with either double quotation marks (") or single quotation marks ('). If a string is enclosed by double quotation marks, then single quotation marks are allowed inside the string (and vice versa). Lastly, an escaped quotation mark (\" or \') is also allowed inside a string.

Examples of *valid strings* are:

- `"string"`
- `'02a\a'`
- `"Let's go"`
- `'Let\'s go'`

Examples of invalid strings are:

- `No string`
- `"Two""Strings"`
- `"String`
- `"He said "No""`

(c) Derive *one* NFA A that accepts both the languages for a comment or a string as defined in the previous tasks.

(d) Prove that `"It\'s"` $\in L(A)$.

Solution:

In the following we use the set notation $\{a_1, \ldots, a_n\}$ as a shorthand for $a_1|a_2|\ldots|a_n$.

(a)

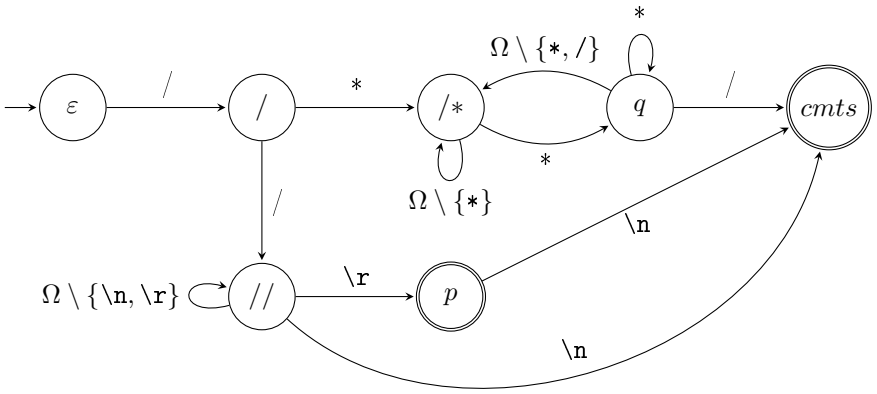
$$\begin{aligned}
\mathcal{L}_{single_cmt} &:= \texttt{//}(\Omega \setminus \{\texttt{\textbackslash r}, \texttt{\textbackslash n}\})^*(\texttt{\textbackslash n} \mid \texttt{\textbackslash r} \mid \texttt{\textbackslash r\textbackslash n}) \\
\mathcal{L}_{multi_cmt} &:= \texttt{/*}\big(\texttt{/} \mid \texttt{*}^*(\Omega \setminus \{\texttt{*}, \texttt{/}\})\big)^*\texttt{*}^*\texttt{*/} \\
\mathcal{L}_{comment} &:= \mathcal{L}_{single_cmt} \mid \mathcal{L}_{multi_cmt}
\end{aligned}$$

(b)

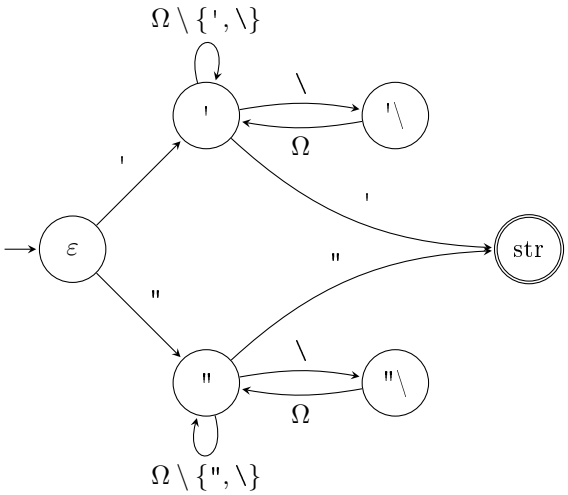
$$\begin{aligned}
\mathcal{L}_{single_string} &:= \texttt{'}\big((\Omega \setminus \{\texttt{'}, \texttt{\textbackslash}\}) \mid \texttt{\textbackslash}\Omega\big)^*\texttt{'} \\
\mathcal{L}_{multi_string} &:= \texttt{"}\big((\Omega \setminus \{\texttt{"}, \texttt{\textbackslash}\}) \mid \texttt{\textbackslash}\Omega\big)^*\texttt{"} \\
\mathcal{L}_{string} &:= \mathcal{L}_{single_string} \mid \mathcal{L}_{multi_string}
\end{aligned}$$

(c) We use sets as transition labels whenever there is a corresponding transition for every element of that set.

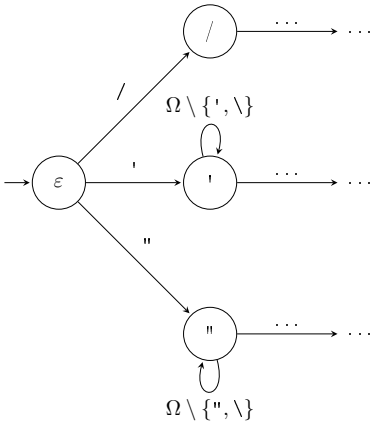
The NFA for a comment looks as follows:



The NFA for a string looks as follows:



Combining both NFAs yields the NFA we are looking for:



(d) The NFA method performs the powerset construction on the fly:

$$
\begin{aligned}
&(\{\varepsilon\}, \texttt{"It\textbackslash's"}) \\
\vdash &(\{\texttt{"}\}, \texttt{It\textbackslash's"}) \\
\vdash &(\{\texttt{"}\}, \texttt{t\textbackslash's"}) \\
\vdash &(\{\texttt{"}\}, \texttt{\textbackslash's"}) \\
\vdash &(\{\texttt{"\textbackslash}\}, \texttt{'s"}) \\
\vdash &(\{\texttt{"}\}, \texttt{s"}) \\
\vdash &(\{\texttt{"}\}, \texttt{"}) \\
\vdash &(\{\text{str}\}, \varepsilon)
\end{aligned}
$$