Answers to the exercises about regular expressions

Christian Rinderknecht

4 October 2005

Question 1. Identify the lexemes and their corresponding token in the following C program.

```
/* Return maximum of integers i and j */
int max (int i, int j) {
  return i>j? i : j;
}
```

Answer 1. The token may be different for different languages. Even if we do not know exactly their names for the C language, we still can find meaningful names based on what they denote. Also we may decide or not to make a token for each keyword. This is a creative work! Therefore several solutions may be acceptable as long as they enable a meaningful implementation. However, it is usual to call the identifiers "identifiers". One answer to the question is therefore

TOKEN	LEXEME
keyword	`int`
identifier	`max`
symbol	`(`
keyword	`int`
identifier	`i`
symbol	`,`
keyword	`int`
identifier	`j`
symbol	`)`
symbol	`{`

TOKEN	LEXEME
keyword	`return`
identifier	`i`
relation	`>`
identifier	`j`
symbol	`?`
identifier	`i`
symbol	`:`
identifier	`j`
terminator	`;`
symbol	`}`

Note that the comments are recognised by the lexer and then discarded, i.e., they are not transmitted to the parser. The same happens to spaces, i.e., blanks and tabulations, which are used for separating lexemes in the source but are not meaningful by themselves.

Question 2. Let the alphabet $\Sigma = \{a, b\}$ and the following regular expressions:

$$r = a(a|b)^\star ba$$
$$s = (ab)^\star | (ba)^\star | (a^\star | b^\star)$$

The language denoted by r is noted $L(r)$ and the language denoted by s is noted $L(s)$.

Find a word x such as

1. $x \in L(r)$ and $x \notin L(s)$
2. $x \notin L(r)$ and $x \in L(s)$
3. $x \in L(r)$ and $x \in L(s)$
4. $x \notin L(r)$ and $x \notin L(s)$

Answer 2. The method to answer these questions is simply to try small words by constructing them in order to satisfy the constraints.

1. The shortest word x belonging to L(r) is found by taking ε in place of $(a|b)^\star$. So $x = aba$. Let us check if $x \in L(s)$ or not. $L(s)$ is made of the union of four sub-languages (subsets). To make this clear, let us remove the useless parentheses on the right side:

   $$s = (ab)^\star \,|\, (ba)^\star \,|\, a^\star \,|\, b^\star$$

   Therefore, membership tests on $L(s)$ have to be split into four: one membership test on $(ab)^\star$, one on $(ba)^\star$, one on $a^\star$ and another one on $b^\star$. In other words:

   $$x \in L(s) \Leftrightarrow x \in L((ab)^\star) \; or \; x \in L((ba)^\star) \; or \; x \in L(a^\star) \; or \; x \in L(b^\star)$$

   Let us test the membership with $x = aba$:

   (a) The words in $L((ab)^\star)$ are ε, ab, $abab\ldots$ Thus $aba \notin L((ab)^\star)$.
   (b) The words in $L((ba)^\star)$ are ε, ba, $baba\ldots$ Hence $aba \notin L((ba)^\star)$.
   (c) The words in $L(a^\star)$ are ε, a, $aa\ldots$ Therefore $aba \notin L(a^\star)$.
   (d) The words in $L(b^\star)$ are ε, b, $bb\ldots$ So $aba \notin L(b^\star)$.

   Finally the conclusion is $aba \notin L(s)$, which is what we were looking for.

2. What is the shortest word belonging to $L(s)$? Since the four sub-languages composing $L(s)$ are starred, it means that $\varepsilon \in L(s)$. Since we showed at the item (1) that aba is the shortest word of $L(r)$, it means that $\varepsilon \notin L(r)$ because ε is of length 0.

3. This question is a bit more difficult. After a few tries, we cannot find any x such as $x \in L(r)$ and $x \in L(s)$. Then we may try to prove that $L(r) \cap L(s) = \varnothing$,

i.e., there is no such x. How should we proceed? The idea is to use the decomposition of $L(s)$ into for sub-languages and try to prove

$$L(r) \cap L((ab)^\star) = \varnothing$$
$$L(r) \cap L((ba)^\star) = \varnothing$$
$$L(r) \cap L(a^\star) = \varnothing$$
$$L(r) \cap L(b^\star) = \varnothing$$

Indeed, if all these four equations are true, they imply $L(r) \cap L(s) = \varnothing$.

(a) Any word in $L(r)$ ends with a whereas any word in $L((ab)^\star)$ finishes with b or is ε. Thus $L(r) \cap L((ab)^\star) = \varnothing$.

(b) For the same reason, $L(r) \cap L(b^\star) = \varnothing$.

(c) Any word in $L(r)$ contains both a and b whereas any word in $L(a^\star)$ contains only b or is ε. Therefore $L(r) \cap L(a^\star) = \varnothing$.

(d) Any word in $L(r)$ starts with a whereas any word in $L((ba)^\star)$ starts with b or is ε. Thus $L(r) \cap L((ba)^\star) = \varnothing$.

Finally, since all the four equations are false, they imply that

$$L(r) \cap L(s) = \varnothing$$

4. Let us construct letter by letter a word x which does not belong neither to $L(r)$ not $L(s)$. First, we note that all words in $L(r)$ start with a, so we can try to start x with b: this way $x \notin L(r)$. So we have $x = b\ldots$ and we have to fill the dots with some letters in such a way that $x \notin L(s)$.

   We use again the decomposition of $L(s)$ into four sub-languages and make sure that x does not belong to any of those sub-languages.

   First, because x starts with b, $x \notin L(a^\star)$ and $x \notin L((ab)^\star)$.

   Now, we have to add some more letters such that $x \notin L(b^\star)$ and $x \notin L((ba)^\star)$.

   Since any word in $L(b^\star)$ has a letter b as second letter or is ε, we can choose the second letter of x to be a. This way $x = ba\ldots \notin L(b^\star)$.

   Finally, we have to add more letters to make sure that

   $$x = ba\ldots \notin L((ba)^\star)$$

   Any word in $L((ba)^\star)$ is either ε or ba or $baba\ldots$, hence the third letter is b. Therefore, let us choose the letter a as the third letter of x and we thus have $x = baa \notin L((ba)^\star)$. Summary:

   $$baa \notin L(r),\ baa \notin L(b^\star),\ baa \notin L((ba)^\star),\ baa \notin L(a^\star),\ baa \notin L((ab)^\star)$$

   which is equivalent to

   $$baa \notin L(r) \text{ and } baa \notin L((ab)^\star) \cup L((ba)^\star) \cup L(a^\star) \cup L(b^\star) = L(s)$$

   Therefore $x = baa$ is one possible answer.

Question 3. Given the binary alphabet $\Sigma = \{a, b\}$ and the order on letters $a < b$, write regular definitions for the following languages.

1. All words starting and ending with a.
2. All non-empty words.
3. All words in which the third last letter is a.
4. All words containing exactly three a.
5. All words containing at least one a before a b.
6. All words in which the letters are in increasing order.
7. All words with no letter following the same one.

Answer 3. When answering these questions, it is important to keep in mind that the language of words made up on the alphabet Σ is Σ^* and that there are, in general, several regular expressions describing one language.

1. The constraint on the words is that they must be of the shape $a \ldots a$ where the dots stand for "any combination of a and b." In other words, one answer is $a(a \mid b)^\star a \mid a$.
2. This question is very simple since the language of all words is $(a \mid b)^\star$, we have to remove ε, i.e., one simple answer is $(a \mid b)^+$.
3. The question implies that the words we are looking for are of the form $\ldots a\_\_$ where the dots stand for "any sequence of a and b" and each $\_$ stands for a regular expression denoting any letter. Any letter is described by $(a \mid b)$; therefore one possible answer is $(a \mid b)^\star a\,(a \mid b)\,(a \mid b)$.
4. The words we search contain, at any place, exactly three a, so are of the form $\ldots a \ldots a \ldots a \ldots$, where the dots stand for "any letter except a", i.e., "any number of b." In other words: $b^\star a b^\star a b^\star a b^\star$.
5. Because the alphabet contains only two letters, the question is equivalent to: "All words containing the substring ab", i.e., the words are of the form $\ldots ab \ldots$ where the dots stand for "any sequence of a and b." It is then easy to understand that a short answer is $(a \mid b)^\star ab (a \mid b)^\star$.
6. Because the alphabet is made only of two letters, the answer is easy: we put first all the a and then all the b: $a^\star b^\star$.
7. Since the alphabet contains only two letters, the only way to not repeat a letter is to only have substrings ab or ba in the words we look for. In other words: $abab \ldots ab$ or $abab \ldots aba$ or $baba \ldots ba$ or $baba \ldots bab$. In short: $(ab)^\star a? \mid (ba)^\star b?$ or, even shorter: $a?(ba)^\star b?$.

Question 3bis. Let the alphabet be now the ASCII. Find regular definitions for

1. All comments on one line between braces, i.e., `{` and `}`.
2. All comments between `/*` and `*/` without `*/` inside, except between double-quotes, i.e `"`.

Hint: Use the complement of a letter in the alphabet, e.g., $\overline{\texttt{A}}$ denotes all letters except `A`. g

Answer 3bis.

1. Here, the key point is how to *exclude* some letters or combinations of letters from the comment. We must use regular expressions corresponding to the *complement* operation on languages of one letter in the alphabet, e.g. $\overline{a}$ denotes all the letters in the alphabet except a. Therefore, since we have to exclude the end-of-line character and the `}` symbol inside the comments, the regular expression for any letter different from `}` and `\n` is $\overline{(\} \mid \backslash n)}$. Therefore, we can write the following regular definitions:

$$\begin{aligned} \textsf{inside} &\rightarrow \overline{(\} \mid \backslash n)} \\ \textsf{comment} &\rightarrow \texttt{\{}\ \textsf{inside}^\star\ \texttt{\}} \end{aligned}$$

2. Here, the delimiters of the comments are `/*` and `*/`, so we can start writing

$$\begin{aligned} \textsf{inside} &\rightarrow \\ \textsf{comment} &\rightarrow \texttt{/*}\ \textsf{inside}^\star\ \texttt{*/} \end{aligned}$$

   We have three cases in **inside**:

   (a) it is a string between double-quotes,

   (b) it is `*` not followed by `/`,

   (c) it is a letter different from `*` or `"`

   These cases are described by the following regular expressions:

   (a) **string**

   (b) $\texttt{*}\ \overline{\texttt{/}}$

   (c) $\overline{\texttt{*} \mid \texttt{"}}$

   In other words:

$$\begin{aligned} \textsf{string} &\rightarrow \\ \textsf{inside} &\rightarrow \textsf{string} \mid \texttt{*}\overline{\texttt{/}} \mid \overline{\texttt{*} \mid \texttt{"}} \\ \textsf{comment} &\rightarrow \texttt{/*}\ \textsf{inside}^\star\ \texttt{*/} \end{aligned}$$

   The string cannot contain a double-quote, otherwise there is no way to determine where the string finishes.

$$\begin{aligned}
\textbf{string} &\to \texttt{"}(\overline{\texttt{"}})^\star\texttt{"}\\
\textbf{inside} &\to \textbf{string} \mid *\overline{/} \mid \overline{*\mid \texttt{"}}\\
\textbf{comment} &\to /* \ \textbf{inside}^\star */
\end{aligned}$$

Question 4. Simplify, if possible, the following regular expressions.

$$(\varepsilon \mid a^\star \mid b^\star \mid a \mid b)^\star$$
$$a(a \mid b)^\star b \mid (ab)^\star \mid (ba)^\star$$

Answer 4.

1. The first regular expression can be simplified in the following way:

$$\begin{aligned}
(\varepsilon \mid a^\star \mid b^\star \mid a \mid b)^\star &= (\varepsilon \mid a^\star \mid b^\star \mid b)^\star && \text{since } L(a) \subset L(a^\star)\\
&= (\varepsilon \mid a^\star \mid b^\star)^\star && \text{since } L(b) \subset L(b^\star)\\
&= (\varepsilon \mid a^+ \mid b^+)^\star && \text{since } \{\varepsilon\} \subset L(x^\star)\\
&= (a^+ \mid b^+)^\star && \text{since } (\varepsilon \mid x)^\star = x^\star
\end{aligned}$$

   Words in $L((a^+ \mid b^+)^\star)$ are of the form ε or $(a \dots a)(b \dots b)(a \dots a)(b \dots b) \dots$ where the dots stand for "repetition any number of time, including zero." So we recognise $(a \mid b)^\star$. Therefore $(\varepsilon \mid a^\star \mid b^\star \mid a \mid b)^\star = (a \mid b)^\star$.

2. The second regular expression can be simplified in the following way. We note first that the expression is made of the disjunction of three regular sub-expressions (i.e., it is a union of three sub-languages). The simplest idea is then to check whether one of these sub-languages is redundant, i.e., if one is included in another. If so, we can simply remove it from the expression.

$$\begin{aligned}
a(a \mid b)^\star b \mid (ab)^\star \mid (ba)^\star &= a(a \mid b)^\star b \mid \varepsilon \mid (ab)^+ \mid (ba)^\star && \text{since } (ab)^\star = \varepsilon \mid (ab)^+\\
&= a(a \mid b)^\star b \mid (ab)^+ \mid (ba)^\star && \text{since } \{\varepsilon\} \subset L((ba)^\star)
\end{aligned}$$

   We have:

$$\begin{aligned}
(ab)^+ &= (ab)(ab) \dots (ab)\\
&= a(ba)(ba) \dots (ba)b \mid ab\\
&= a(ba)^\star b
\end{aligned}$$

   Also $L((ba)) \subset L((a \mid b)^\star)$ and then $L((ba)^\star) \subset L((a \mid b)^\star)$, because $(a \mid b)^\star$ denotes all the words. Therefore

$$\begin{aligned}
L(a(ba)^\star b) &\subset L(a(a \mid b)^\star b)\\
L((ab)^+) &\subset L(a(a \mid b)^\star b)
\end{aligned}$$

   As a consequence, one possible answer is

$$a(a \mid b)^\star b \mid (ab)^\star \mid (ba)^\star = a(a \mid b)^\star b \mid (ba)^\star$$

The intersection between $L(a(a|b)^\star b)$ and $L((ba)^\star)$ is empty because all the words of the former start with a, while all the words of the other start with b (or is ε). Therefore we cannot simply further this way.