Homework 5

6CCS3CFL - Compilers & Formal Languages

Finley Warman

November 1, 2020

Contents

Question 1

Q: How would you show a property $P(r)$ holds for all regular expressions r, by structural induction?

A:

- As a basis, show $P(r)$ holds for all terminal cases, i.e. `rBase ::= 0 | 1 | c .`
- Let r1 and r2 be regexes, and suppose P(r1) and P(r2) hold [This is the inductive hypothesis, IH].
- Induction over the remaining cases (which are inductively defined) to show property holds on by applying IH:
  `rInd ::== r1+r2 | r1.r2 | r* .`
- Thus all cases are covered, which is sufficient to prove a property by structural induction.

Question 2

Q: Define a regular expression, ALL, that can match every string.
Write this in terms of `r ::= 0 | 1 | c | r1+r2 | r1.r2 | r* |` r
A: `ALL = ~(0)`

Question 3

Q: Define the expressions r^+, $r^?$, $r^{\{n\}}$, $r^{\{m,n\}}$ in terms of basic regular expressions.

A:

- $r^+ = \text{r . r*}$
- $r^? = \mathbf{1} + \text{r}$
- $r^{\{n\}} = r_1 \ . \ r_2 \ . \ ... \ . \ r_n$
- $r^{\{m,n\}} = (r_1 \ . \ r_2 \ . \ ... \ . \ r_m) \ . \ (\ 1 + (r_{m+1}) + (r_{m+1} \ . \ r_{m+2}) + (r_{m+1} \ . \ ... \ . \ r_n)\ )$
- (where $r_i = r$ for all i)

Question 4

Q: Given the regular expressions for lexing a language with identifiers, parenths, numbers (postive/negative), and operators `+, -, *`:

A:

```
DIGIT       = RANGE("0123456789")
START_DIGIT = RANGE("123456789")
NUMBER      = OPT(CHAR('-')) . (DIGIT + (START_DIGIT . DIGIT*))
                   [Numbers are implicitly positive, or explicitly negative]

LPAREN      = CHAR('(')
RPAREN      = CHAR(')')

OPERATOR    = CHAR('+') + CHAR('-') + CHAR('*')

LOWERCASE   = RANGE("abcdefghijklmnopqrstuvwxyz")
ID          = LOWERCASE . LOWERCASE*

LANG        = ("num":NUMBER) + ("op":OPERATOR) + ("lp":LPAREN)
                   + ("rp":RPAREN) + ("id":ID)
```

- `(a3+3)*b` = YES, `lp:(, id:a, op:+, num:3, rparen:), op:*, id:b`
- `)()++-33` = YES, `rp:), lp:(, rp:), op:+, op:+, num:-33`
- `(b42/3)*3` = NO, `/` is not an accepted token.

Question 5

Q: Suppose the context-free grammar G:

```
S ::= A.S.B | B.S.A | ϵ
A ::= a | ϵ
B ::= b
```

where the starting symbol is S, which of the following are in the language of G?:

A:

- a - No, a must be followed by b
- b - Yes
- ab - Yes
- ba - Yes
- bb - Yes (B . (B . S . A) . A)
- baa - No, a must be followed by b

Question 6

Q: Suppose the context-free grammar $S ::= a.S.a \mid b.S.b \mid \epsilon$
Describe the language generated by this grammar.

A: All even-length palindromes over the alphabet {a,b}.

Question 7 (Optional)

Q: Prove by induction on r that the property $L(der\ c\ r) = Der\ c\ (L(r))$ holds.

A: $Der(c, A) \equiv \{s \mid c :: s \in A\}$, i.e. the derivate of language A w.r.t character c is the language of all strings in A beginning with c, with the c 'chopped off' the front.

Basis:

- $R = 0$: $der(c, R) = 0$ and $L(0) = \{\}$. $Der(c, \{\}) = \{\}$ so holds.
- $R = 1$: $der(c, R) = 0$, $L(0)$ as above, so holds.
- $R = c_1$: $der(c, c_1) = 1\ if c = c_1\ else\ 0\}$.
  $L(1) = \{[]\}$ and $L(0) = \{\}$.
  $Der(c, \{c\}) = \{[]\}$ and $Der(c, \{d\}) = \{\}$ so holds.

Inductive Hypothesis:

Assume some expressions r_1, r_2 such that $L(der(c, r_i)) \equiv Der(c, L(r_i))$ holds for $i = 1, 2$. Show true for remaining regex cases.

Inductive Proof:

- $R = r_1 + r_2$
  The language accepted by $r_1 + r_2$ is $L(r_1) \cup L(r_2)$.
  The language accepted by $der(c, r_1 + r_2)$ is $L(der(c, r_1)) \cup L(der(c, r_2))$.
  Since by IH $L(der(c, r_i)) \equiv Der(c, L(r_i))$ holds,
  $Der(c, L(r_1 + r_2)) = L(der(c, r_1)) \cup L(der(c, r_2)) = L(der(c, r_1 + r_2))$ also holds.

- $R = r_1\ .\ r_2$
  The language accepted by $r_1.r_2$ is $L(r_1)@L(r_2)$.
  The language accepted by $der(c, r_1 + r_2)$ is
  $L(((der(c, r_1)).r_2) + der(c, r_2))$ if $\epsilon \in L(r_1)$, else $L((der(c, r_1)).r_2)$.

  The second case is equivalent to $L(der(c, r_1))@L(r_2)$ and the first case is equivalent to $(L(der(c, r_1))@L(r_2)) \cup L(der(c, r_2))$
  Since by IH $L(der(c, r_i)) \equiv Der(c, L(r_i))$ holds,
  when $\epsilon \in L(r_1)$:
  $Der(c, L(r_1.r_2)) = (L(der(c, r_1))@L(der(c, r_2))) \cup L(der(c, r_2)) = L(der(c, r_1 + r_2))$ also holds, as does $Der(c, L(r_1.r_2)) = L(der(c, r_1))@L(der(c, r_2)) = L(der(c, r_1 + r_2))$ otherwise.

- $R = r_1*$
  $D = der(c, R) = der(c, r_1*)\ .\ r*$
  $L(D) = L(der(c, r_1)) \cup L(R)$
  $Der(c, r*) = Der(c, L(r)) \cup L(r*) = L(D)$
  Holds, by all previous cases as r_i can only be of the forms:
  $r := 0|1|c|r_1 + r_2|r_1.r_2|r^*$.