

Principles of Programming Languages - Quiz I (Solutions)

K. Subramani
LDCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Problems

1. Categorize the errors that can occur in a program, with an example of each category.

Solution: The different types of errors that can occur in a program are as follows:

- (a) Lexical errors - For instance, a symbol may be used inappropriately, e.g., $x@ = 3$, in **C**, or a keyword may be misspelled.
- (b) Syntax errors - For instance, malformed expressions, missing tokens, etc.
- (c) Semantic errors - Using undeclared variables, making assignments, without respecting types and so on.
- (d) Logical errors - The program logic does not correspond to what the programmer intended.

□

2. Let $\Sigma = \{0, 1\}$ denote an alphabet. Let $L \subseteq \Sigma^*$ denote the language which consists of all strings that start with 0 and end with 1. Write a regular expression **and** a CFG corresponding to L .

Solution:

- (a) Regular Expression - $0 \cdot (0 + 1)^* \cdot 1$.
- (b) CFG -

$$\begin{aligned} S &\rightarrow 0 A 1 \\ A &\rightarrow 0 A \mid 1 A \mid \epsilon \end{aligned}$$

□

3. Consider the CFG $G = \langle V, T, P, S \rangle$, with $V = \{S\}$, $T = \{0, 1\}$ and P given by:

$$S \rightarrow 0S1 \mid 1S0 \mid \epsilon$$

Argue that every string in $L(G)$ has an equal number of 0s and 1s.

Solution: It is important to note that you cannot use induction in the usual manner, since the derived strings have even length. Once this fact has been established formally, we can use induction over the set of even integers!

Lemma 1.1 Let x be any terminal string derived from S ; $|x|$ is even.

Proof: We use induction on the number of steps used in the derivation of x from S .

BASIS: x was derived from S in one step. In this case, x must be ϵ and hence $|x| = 0$. It follows that the base case is proven.

Assume that the hypothesis is true for all terminal strings derived from S in exactly k steps.

Now consider a terminal string y derived from S in $(k + 1)$ steps. Without loss of generality, assume that the first step in the derivation of y is $S \Rightarrow 0S1$. It follows that y must be of the form: $y = 0w1$, where w is a terminal string which is derived from S . However, the derivation of w from S takes exactly k steps and hence $|w|$ must be even, as per the inductive hypothesis. It follows that $|y| = |w| + 2$ is also even.

Using the first principle of mathematical induction, we can conclude that all terminal strings derived from S have even length. \square

We are now ready to prove the main conjecture.

Theorem 1.1 *Let x be a terminal string derived from S ; x must have an equal number of 0s and 1s.*

Proof: We use induction on $|x|$; since $|x|$ is always even, the induction is on the set of even positive integers, i.e., $\{0, 2, 4, \dots\}$.

BASIS: $|x| = 0$. In this case, x must be ϵ and since ϵ has the same number of 0s as 1s, the basis is proven.

Assume that for all terminal strings x which are derived from S and which have length $2 \cdot k$, where k is a positive integer, x has an equal number of 0s and 1s.

Now consider a terminal string y of length $2 \cdot k + 2$ (the next even number!), which is derived from S . Without loss of generality, assume that the first step in the derivation is $S \Rightarrow 0S1$. It follows that y is of the form: $0w1$, where w is some terminal string derivable from S with length $2 \cdot k$. However, as per the inductive hypothesis, w must have an equal number of 0s and 1s. It follows that y also has the same number of 0s as 1s.

Using the first principle of mathematical induction, we conclude that every string derived from S has an equal number of 0s and 1s. \square

\square

4. Consider the CFG $G = \langle V, T, P, S \rangle$, with $V = \{S\}$, $T = \{a\}$ and P given by:

$$S \rightarrow S \cdot S \mid a$$

Write the set equation corresponding to this grammar and give a solution to this equation.

Solution: Let S denote the set of all strings derivable from S . Then the set equation for the above grammar is:

$$S = S \cdot S \cup \{a\}$$

A solution (indeed the least fixed-point solution) to the above set equation is:

$$S = \{a, aa, aaa, \dots\}$$

\square

5. Consider the following block of C code:

```
#include <stdio.h>

int a, b;

int p(void)
{
    int a, p;
```

```

    a =0; b=1; p=2;
    return p;
}

void print( void)
{
    printf( ``%d\n%d\n'',a,b);
}

void q(void)
{
    int b;

    a= 3; b=4;
    print();
}

main()
{
    a=p();
    q();
}

```

What values will be printed, when the program is parsed using (a) Lexical scope, and (b) Dynamic scope?

Solution:

- (a) Lexical Scope - Under lexical scoping, the values of *a* and *b* are 3 and 1 respectively, at the time of printing.
- (b) Dynamic Scope - Under dynamic scoping, the values of *a* and *b* are 3 and 4 respectively, at the time of printing.

□