

# Assignment 3

COS30023 - Languages in Software Development

Daniel Parker - 971328X

August 31, 2014

## 1. Problem 1

### 1.1. String specification

$$S ::= \epsilon \\ \quad | aS$$

### 1.2. String Length Specification

$$\begin{aligned} \text{length}(\epsilon) &= 0 \\ \text{length}(aS) &= 1 + \text{length}(S) \end{aligned}$$

### 1.3. String Concatenation Specification

$$\begin{array}{lll} s_1 = \epsilon : & \epsilon \oplus s_2 & = s_2 \\ s_1 = as'_1 : & as'_1 \oplus s_2 & = a(s'_1 \oplus s_2) \end{array}$$

#### 1.4. a)

Show that if  $s \in S$ , then  $s \oplus \epsilon = s$

**Base Case 1**

$$s \equiv \epsilon$$

$$\epsilon \oplus \epsilon = \epsilon$$

The first base case is when  $s$  is structurally equivalent to the empty string  $\epsilon$  and it's a fact that concatenating two empty strings will yield another empty string.

**Base Case 2**

if  $length(s) = 1$  then

$$s \oplus \epsilon = s$$

In this second base case we take  $s$  to be any string of length 1 and again it is a fact that any string concatenated with the empty string  $\epsilon$  will be the original first string. In this case the string will still be of length 1.

**Inductive Step**

We assume that the base case holds for all lengths of  $s$  greater than 1, therefore:

if  $length(s) > 1$

$$s \oplus \epsilon = s$$

**Q.E.D.**

#### 1.5. b)

Show that if  $s_1, s_2 \in S$  then  $length(s_1 \oplus s_2) = length(s_1) + length(s_2)$

**Base Case 1**

$$s_1, s_2 \equiv \epsilon$$

Given that  $length(\epsilon) = 0$  then it goes to say that  $length(\epsilon \oplus \epsilon) = 0$ .

**Base Case 2**

$s_1 \neq \epsilon$ , where  $length(s_1) = 1$  and  $s_2 \equiv \epsilon$ .

$$length(s_1 \oplus \epsilon) = length(s_1)$$

For the case where  $s_1$  is a string of length 1 and  $s_2$  is an empty string, the concatenated length will be the length of  $s_1$ .

### Base Case 3

$s_1, s_2 \neq \epsilon$ , where  $length(s_1) = 1$  and  $length(s_2) = 1$ .

$$length(s_1 \oplus s_2) = length(s_1) + length(s_2) = 2$$

For the case where both  $s_1$  and  $s_2$  are strings of length 1, the concatenated length will be the sum of the individual lengths, which is 2.

### Inductive Step

$length(s_1) > 1$  and  $length(s_2) > 1$

$$length(s_1 \oplus s_2) = length(s_1) + length(s_2)$$

We assume the base case 3 holds for strings  $s_1, s_2$  of any length greater than 1.

**Q.E.D.**

## 2. Problem 2

```
% List length predicate
list_length([], 0).
list_length(_|T, N) :- list_length(T, M), N is M+1.

% List concatenation predicate
list_concatenation([], Xs, Xs).
list_concatenation([H|Xs], Ys, [H|Zs]) :- list_concatenation(Xs, Ys, Zs).

% List equality predicate
equal([], []) :- true.
equal([Xhead|Xtail], [Yhead|Ytail]) :- Xhead == Yhead, equal(Xtail, Ytail).

% Check that concatenating the empty set to Xs equals Xs
checkA(Xs) :- list_concatenation([], Xs, R), equal(R, Xs).

% Check that concatenating two strings equals the length of the sum
% of individual lengths of the substrings
checkB(Xs, Ys) :- list_concatenation(Xs, Ys, Rs),
                  list_length(Rs, Rl),
                  list_length(Xs, Xl),
                  list_length(Ys, Yl),
                  Rl == Xl + Yl.
```