HW 1 - Solutions

Note: There are different possible solutions to some problems.

Part 1: Definition of languages

Exercise 1

Draw the tree representing $a \ast (b + c) \ast d$.
- is left-associative so there is only one way to draw the tree.
$a \ast (b + c) \ast d$ is $(a \ast (b + c)) \ast d$.

Postfix: visit the left subtree and right subtree completely before visiting the root node.
$abc + \ast d\ast$

Prefix: visit the root before visiting the left subtree and the right subtree.
$\ast \ast a + bcd$

Exercise 2

$7 7 \ast 4 2 \ast 3 \ast -$  
Here is the stack (in the left - top of the stack):

7  
7 7  
49  
49 4  
49 4 2  
49 8  
49 8 3  
49 24  
25  

$7 7 \ast 4 2 \ast 3 \ast - = 24$
Exercise 3 and 4

while expr do id := expr

Exercise 5

Terminals: A and 
Non-terminals: < goal >, < letter > and < next >.
The language is composed of A and A, A only.

Exercise 6

Us phone numbers.

<phone> ::= <acode> <x> <x> <x> - <x> <x> <x> <x>

<acode> ::= ((<x> <x> <x>) | ^ (empty word)

<x> ::= 0 | 1 | 2 | ... | 9

Exercise 7

Write a BNF grammar for identifiers that consist of an arbitrarily long string of letters and
digits, the first one of which must be a letter.

<identifier> ::= <letter><word>

<word> ::= <character><word> | ^

<character> ::= <letter> | <digit>

<letter> ::= a | b | ... | z | A | B | ... | Z
<digit> ::= 0 | 1 | ... | 9
Part 2: Recursion, functional language programming and SML

Exercise 1
See lecture notes.

Exercise 2
In SML with pattern matching:

```sml
fun pow(0) = 1  
| pow(n) = 2*pow(n-1);  
val pow = fn : int -> int
```

Exercise 3
In SML with pattern matching:

```sml
fun fib(0) = 1  
| fib(1) = 1  
| fib(n) = fib(n-1) + fib(n-2);  
val fib = fn : int -> int

fib(10) = 55  
fib(40) = 102334155
```

If you draw the tree of the evaluation of \(fib(5)\) by the previous definition of \(fib\) you will see that the tree grows very quickly. This is an exponential computation so it is sure that you will not be able to compute \(fib(800)\) with this definition. The question is now: How to improve the definition? (iteration, local variables...)

Exercise 4

\[
 f(n) = \begin{cases} 
 n - 1 & \text{if } n > 0 \\
 f(f(n + 2)) & \text{otherwise} 
\end{cases}
\]

\[
 f(10) = 10 - 1 = 9  
 f(0) = f(f(0 + 2)) = f(f(2)) = f(2 - 1) = f(1) = 1 - 1 = 0 \text{ (inner-most evaluation)}
```

Exercise 5
Types:

```sml
hd([6,1,2]);  
tl [7,2,3];
```
hd [1];
tl [6];
explode ("abcd");
implode ["abc","def","hi"];
"f" :: ["a", "c", "i", "l", "e"];
["m", "e", "t"] @ ["a", "n", "l", "a", "n", "g", "a", "g", "e"];

SOLUTION

hd([6,1,2]);
val it = 6 : int
tl [7,2,3];
val it = [2,3] : int list
hd [1];
val it = 1 : int
tl [6];
val it = [] : int list
explode ("abcd");
char list
implode ["abc","def","hi"];
val it = "abcdehi" : string
"f" :: ["a", "c", "i", "l", "e"];
val it = ["f", "a", "c", "i", "l", "e"] : string list
["m", "e", "t"] @ ["a", "n", "l", "a", "n", "g", "a", "g", "e"];
val it = ["m", "e", "t", "a", "n", "l", "a", "n", "g", "a", "g", "e"] : string list

Exercise 6

((1,2),3);
(1,(2,3));
(1,2,3);
(1.2,(["2",[4,5]]));
[[3,4],[1],[5]];
[([9,3,5),(1,2,1),(9,4,2)];
(["b","a"],[nil,[1,2,3]]);

Note: The constructor of type list has more priority than *.

SOLUTION

((1,2),3);
val it = ((1,2),3) : (int * int) * int
(1,(2,3));
val it = (1,(2,3)) : int * (int * int)
(1,2,3);
val it = (1,2,3) : int * int * int
(1,2, ["2", [4,5]]);
val it = (1,2, ["2", [4,5]]) : real * (string * int list)
[[3,4],[[]],[5]];
val it = [[3,4],[[]],[5]] : int list list
[(9,3,5), (1,2,1), (9,4,2)];
val it = [(9,3,5), (1,2,1), (9,4,2)] : (int * int * int) list
(['b"", "a"'], [nil, [1,2,3]]);
val it = (['b"", "a"'], [[1,2,3]]) : string list * int list list

Exercise 7

int list list list
int * string list
(int * string) list
string list * (int * (real * string)) * int
((int * int) * (string list) * real) * (real * string)

SOLUTION

int list list list
[[[3,4],[[]],[5],[6,7,8],[1],[[]],[[2]]]];
int * string list
(1,['a"", "b"']);
(int * string) list
[(1,"ab"), (2,"c"), (3,"fc")];
string list * (int * (real * string)) * int
(['toto"", "titi"'], (1, (1,2,"titi"), 3));
((int * int) * (string list) * real) * (real * string)
(((1,2), ['toto"", "titi"'], 1.5), (1.2,"tutu"));

Exercise 8

SOLUTION

fun max(x, y) =
if x>y
then x
else y;
val max = fn : int * int -> int
fun max4(x,y,z,t) = max(max(x,y),max(z,t));
val max4 = fn : int * int * int * int -> int
max4(1,2,3,4);
val it = 4 : int
max4(8,2,3,4);
val it = 8 : int
Exercise 9

Circumference and area.

SOLUTION

First solution:
val pi = 3.14159;
fun circumference r = 2.0 * pi * r;
fun area r = pi * r * r;
Other solution:
(the 2 functions are defined at the same time)
local val pi = 3.14159 in fun circumference r = 2.0 * pi * r fun area r = pi * r * r end;
val circumference = fn : real → real
val area = fn : real → real
circumference 10.0;
val it = 62.8318 : real
circumference 10;
std_in:6.1-6.16 Error: operator and operand don’t agree (tycon mismatch)
operator domain: real
operand: int
in expression:
circumference 10

Exercise 10

SOLUTION

Using pattern matching:
fun move([]) = []
| move (x::l) = l@[x];

val move = fn : 'a list → 'a list
move([]); warning
move([]);
move([1,2,3,4,5,6]);
move([1.2,4.5,6.4]);