Homework 3: Introduction to ML using list and tuple

Stephen Wagstaff
CS 431
September 127, 2018

---

1. Write a function plus that adds two complex numbers, where a complex number is written as a tuple of two integers.

- For example, `(1, 2)` is a complex number with real part `1` and imaginary part `2`. `plus ((1, 2), (3, 4))` should return `(4, 6)`.

```
fun plus((r1, i1), (r2, i2)) = ((r1+r2), (i1+i2));
```

2. Write a function times that multiples two complex numbers.

- For example, `times ((1,2), (3,4))` should return `(1 * 3 - 2 * 4, 1 * 4 + 2 * 3)`, which is `(~5, 10,)`.

```
fun times((r1, i1), (r2, i2)) = ((r1 * r2 - i1 * i2), (r1 * i2 + i1 * r2));
```

3. Write a function until that takes two integers `x` and `y` and return a list from `x` to `y - 1`. If `x  y`, it should return `nil`.

- For example, `until (1, 4)` should return `[1,2,3]`.

```
fun createUntil(x, y, list) = if x>= y
                              then list
                              else x::createUntil(x +1, y, list);

fun until(x, y) = if x >= y
                  then nil
                  else createUntil(x, y, nil);
```

4. Write a function append that takes an integer x and a list of integers and return a list of pairs where the left of each pair is x and right of each pair is a list element.

- For example, `append (1, [1, 2, 3])` should return `[(1,1), (1,2), (1, 3)]`.

```
fun append(_, nil) = nil
  | append(x, list) = (x, hd list)::append(x,tl list);
```

5. Write a function pair that takes two lists of integers and generates a list of pairs, where each pair is a combination of each element from each list.

- For example, `pair ([1,2], [3,4,5])` should return `[(1,3), (1,4), (1,5), (2,3), (2,4), (2,5)]`.

```
fun pair(nil, list2) = nil
  | pair(list1 , nil) = nil
  | pair(list1, list2) = append(hd list1, list2) @ pair(tl list1, list2);
```