

“Stand-alone topics” outline:

- ▶ Foldl, and how to model mathematical functions (Today)
- ▶ Fixed-point iteration (this coming Monday)
- ▶ The Huffman encoding (next Wednesday)
- ▶ Review for the final exam (next Friday)

Outline for today:

- ▶ Go over a couple test problems
- ▶ The foldr and foldl functions
- ▶ Modeling polynomials

Recall that a *full binary tree* is either a node by itself or a node together with two subtrees that are full binary trees. Prove that for any full binary tree T , $\text{nodes}(T) = 2 \cdot \text{internals}(T) + 1$.

The function below computes the quotient and remainder (mod) of x divided by d . Prove that $I(n)$ is an invariant for the loop of `divmod`.

$$I(n) = \text{ after } n \text{ iterations, } x = q \cdot d + r$$

```
fun divmod(x, d) =  
  let val q = ref 0;  
      val r = ref x;  
  in  
    (while !r > d do  
      (q := !q + 1;  
       r := !r - d);  
    (!q, !r))  
  end;
```

Write a function `funSwitch` that takes three functions as parameter. The first parameter has type `int → bool` and the other two parameters have type `int → int`. The function `funSwitch` returns a function that takes an `int` and tests its parameter using the first function parameter to `funSwitch`. If the result is `true` then it applies the second parameter to `funSwitch`, otherwise it applies the third parameter, and either way returns the result. Thus `funSwitch(c, a, b)` returns a function that acts like either `a` or `b`, using `c` to determine which.

For example,

```
funSwitch(fn(x) => x mod 2 = 0, fn (x) => x div 2, fn(x) => x - 1)(12)
```

will return 6 and

```
funSwitch(fn(x) => x mod 2 = 0, fn (x) => x div 2, fn(x) => x - 1)(11)
```

will return 10.

For next time:

Ex “7.14.(a–d)” on Schoology

Read 7.12 carefully Skim 7.13

If you have not taken calculus, ask a friend to give you the five-minute explanation of what a derivative is.