

CMSC 330: Organization of Programming Languages

Project 4

Regular Expression

Signature

```
type regexp =  
  Empty_String  
  | Char of char  
  | Union of regexp * regexp  
  | Concat of regexp * regexp  
  | Star of regexp
```

```
val regexp_to_string : regexp -> string  
val string_to_regexp : string -> regexp
```

CMSC 330

3

NFAs & Regular Expressions

Goal

- Implement regular expressions
 - > Print regular expression in postfix notation
 - > Parse string to regular expression
- Implement NFAs
 - > Calculate 1 step for NFA
 - > Determine whether string accepted by NFA
- Convert regular expression to NFA

CMSC 330

2

NFA

Signature

```
type nfa  
type transition = int * char option * int  
  (* (s0, Some c, s1) or (s0, None, s1)  *)
```

```
val make_nfa : int -> int list -> transition list -> nfa  
val step : nfa -> int list -> char -> int list  
val accept : nfa -> string -> bool
```

CMSC 330

4

Simultaneous Declarations

- ▶ You can use `let rec ... and ... and ... in e` to declare multiple functions simultaneously
 - All functions known in bodies of functions at same level
 - Useful for mutual recursion

```
let rec f1 x = print_int x ; f2 (x-1)  
    and f2 y = print_int y ; f3 (y-1)  
    and f3 z = if z > 0 then f1 z  
in f1 5 (* prints 5 4 3 2 1 0 *)
```

CMSC 330

5

Concatenation Operators

▶ @

- Type: 'a list -> 'a list -> 'a list
- Example: [1;2;3] @ [4;5]
 - > Returns the concatenated list [1;2;3;4;5]

▶ ^

- Type: string -> string -> string
- Example: "abc" ^ "de"
 - > Returns the concatenated string "abcde"

▶ Other functions in Pervasives module at

- <http://caml.inria.fr/pub/docs/manual-ocaml/libref/Pervasives.html>

CMSC 330

6

String Operations

- ▶ **String.get**
 - Type: `string -> int -> char`
 - Example: `String.get s x`
 - > Returns char in string `s` at index `x`
- ▶ **String.concat**
 - Type: `string -> string list -> string`
 - Example: `String.concat s0 [s1;s2;s3]`
 - > Returns string `s1^s0^s2^s0^s3`
- ▶ **Other functions in String module at**
 - <http://caml.inria.fr/pub/docs/manual-ocaml/libref/String.html>

CMSC 330

7

OCaml Examples

- ▶ Look at examples from discussion section
 - 13-dfs.ml
 - 14-parseArith.ml
 - 15-parseArith_fact.ml

CMSC 330

8