CSE215
Foundations of Computer Science

Instructor: Zhoulai Fu

State University of New York, Korea

Plan

- Today:
    - Homework 04
    - Review of lambda calculus
    - Mini-mock
- Midterm 1 on Thursday 04/03 12h30-13h50 pm at B103 (this room)
- Covering everything until last lecture. Yes, Ocaml is covered in midterm 1
- Any amount of physical notes are allowed.
- E-devices are not allowed.

Homework 04

Review on lambda calculus

Small-1

- (λ x.x) a
- (λ x.y) a
- (λ x.xy) a
- (λ x. yx) a
- (λ x. xx) a
- (λ x. yy) a

Small-2

- (λ x.x) a b
- (λ x.y) a b
- (λ x.xy) a b
- (λ x. yx) a b
- (λ x. xx) a b
- (λ x. yy) a b

Small-3

- (λ x.x) λ a. b
- (λ x.y) λ a. b
- (λ x.xy) λ a. b
- (λ x. yx) λ a. b
- (λ x. xx) λ a. b
- (λ x. yy) λ a. b

Small-4

- (λ x.x) x
- (λ x.y) x
- (λ x.xy) x
- (λ x. yx) x
- (λ x. xx) x
- (λ x. yy) x

Small-5

- (λ x.x) x y
- (λ x.y) x y
- (λ x.xy) x y
- (λ x. yx) x y
- (λ x. xx) x y
- (λ x. yy) x y

Exercise: beta reduction

- (λz.z) (λz.z z) (λz.z q)

Exercise: beta reduction

- (λs.λq.s q q) (λa.a) b

Exercise: beta reduction

- (λs.λq.s q q) (λq.q) q

Exercise: beta reduction

- ((λs.s s) (λq.q)) (λq.q)

Exercise: beta reduction

- (λ x.λy.x) x y

Exercise: beta reduction

- (λ x. λ y. λ z. y (w y x)) λ s. λ z. z

Mini-Mock Midterm1

- Regular expressions
- Context-free grammar
- lambda calculus
- Ocaml

Regular expression

In our class, we have studied the core regular expressions and some abbreviations based on those core regular expressions.

r	Meaning	Language $\mathcal{L}(r)$
`a`	Character `a`	$\{\text{"a"}\}$
ε	Empty string	$\{\text{""}\}$
$r_1 r_2$	r_1 followed by r_2	$\{s_1 s_2 \mid s_1 \in \mathcal{L}(r_1), s_2 \in \mathcal{L}(r_2)\}$
r^*	Zero or more r	$\{s_1 \ldots s_n \mid s_i \in \mathcal{L}(r), n \geq 0\}$
$r_1 \mid r_2$	Either r_1 or r_2	$\mathcal{L}(r_1) \cup \mathcal{L}(r_2)$

Abbrev.	Meaning	Expansion
`[aeiuo]`	Set	`a|e|i|o|u`
`[0–9]`	Range	`0|1|...|8|9`
`[0-9a-z]`	Ranges	`0|1|...|8|9|a|b|...|y|z`
r?	Zero or one r	$r \mid \varepsilon$
r^+	One or more r	$r\, r^*$

Give a regular expression over {a, b} that has aab as a substring

Regular expression

(1) Write a regular expression pattern to match valid music notes according to the criteria below: A music note is represented by a capital letter A to G (inclusive) followed by an optional symbol: sharp (#), flat (b), or natural (n).

```
Example valid inputs: C, D#, Fb, Gn

Example invalid inputs: H, C##, Fm, C#b
```

Note. The sharp symbol ("#") is not a special character in regular expressions. So you do *not* need to escape it with a backslash.

Context-free grammar

(2)** What is the language generated by the following grammar? Select one answer from the four choices.

```
S -> aSbb | ε
```

A. The set of all strings that start with 'a' and end with two 'b'.

B. The set of all strings that contain twice as many 'b's as 'a's.

C. The set of all strings that contain an odd number of 'a's followed by an even number of 'b's.

D. The set of all strings that contain n 'a's followed by m 'b's, where m = 2n >= 0

lambda calculus

- reduce the lambda term to the normal form. No need to write the reduction in details.

  - (λx.(x y))(λz.z)

lambda calculus

- reduce the lambda term to the normal form. No need to write the reduction in details.

  - (λx.(λy.x y)x)(λz.y)

Ocaml

- Give the type of the following OCaml expressions:
  - print_string
  - Let f = fun x y -> x+y in f 3
  - let f x = x *. 3.14 in f 5.6