CS415 — Discussion Section Notes 4

Pieter Hooimeijer

2007-02-19

Outline

1. Housekeeping
2. Written Assignment Recap
3. PA3 Preview

Housekeeping

- *Written Assignment 3* is due this Thursday at 1pm. Please put your name and university user id on there somewhere, and hand in at the cs415 dropbox or before class.
- *Programming Assignment 3* is due this Friday at 11:59pm. Use the Toolkit, start early.

Homework Recap

- Let L_1 be the language consisting of all non-empty palindromes over $\Sigma = \{a, b\}$. Let L_2 be the language over $\{a, b\}$ denoted by the regular expression $a(a|b)^*$.

  Question: What is wrong with the following answer?

  $$S \rightarrow aCa$$
  $$C \rightarrow aCa \mid bCb \mid \epsilon$$

- Consider the grammar:

  $$S \rightarrow aSb \mid Sb \mid \epsilon$$

- "This is the language of all strings that start with any number of a's followed by at least as many b's."
  –The TA
- Consider the following two solutions:

$$S \rightarrow aTa \mid Tb$$
$$T \rightarrow aTb \mid Tb \mid \epsilon$$

$$S \rightarrow aSb \mid T \mid \epsilon$$
$$T \rightarrow Tb \mid \epsilon$$

  What is wrong with these?

- Let L_1, L_2 be context-free languages, and let L_3, L_4 be regular languages. What can you say about the following languages?
  - $L_3 \cap L_4$
  - $L_1 \cap L_2$
  - $L_1 \cap L_3$

  How would you prove that context-free languages are closed under regular intersection?

  How would you prove that context-free languages are *not* closed under intersection in general?

Programming Assignment 3 Preview

Consider the following piece of code:

Listing 1: test.cl

```
1  (* test... *)
2
3  Class Main inherits IO {
4    main() : Object {
5      let done : Bool <- true in
6      if (done) then
7        let blah : String <- "Blah.\n" in
8        out_string(blah)
9      else
10       done <- false
11     fi
```

```
12    };
13 };
```

The .cl-lex file for this is exactly 99 lines long, so I won't be showing it here. The .cl-ast file is just 55 lines, starting as follows:

Listing 2: test.cl-ast

```
1
3
Main
inherits
3
IO
1
method
5
main
0
5
Object
6
let
1
let_binding_init
6
done
6
Bool
6
true
```