

CSE 401

Section 2

WINTER 2017

SLIDES BY ERIN PEACH



Administrivia

Hello! I'm Erin!

OH – M thru F – 4:30 to 5:30

Homework 1 due tonight – Returned before Scanner due date

cse401-staff@cs.washington.edu

Section slides get posted +/- 6 hours from section time



Today's Agenda

Everything you need to know about GitLab in 3 slides

Getting started on the Project

Ambiguity Practice



Everything you need to know about GitLab in 3 slides

(at least for CSE401)

SSH Keys

Matches a specific computer to a Git(Lab/Hub) account

Do I need to do this? Yes, if you are asked for a git@gitlab... Password.

Command line:

- `ssh-keygen -t rsa -b 4096 -C your_email@example.com`

Copy and paste into Git(Lab/Hub) account

Easy, step-by-step instructions:

- <https://help.github.com/articles/generating-an-ssh-key/>



Git 101

Clone a repo to get a personal copy

- Master lives in cloud

Make changes to your personal repo

Commit (Local “save”) and Push (Cloud “save”)

Pull any Cloud changes you want to your local repo

With 2 people on a project

- Merge conflicts
- Branches – you can if you want, but don’t get in over your head

Docs: <https://docs.gitlab.com/>



In This Class

No starter code in the repo (different from CSE331, CSE333)

Empty repo

One team member should download the starter code from the web and do an initial push to the repo



Getting Started on the Project



Set-up

Starter code is on the website

(You might need to install Ant)



Scanner Demo



Ambiguity Practice



2015 Midterm – Question 4

Question 4. Context-free grammars (14 points) Consider the following syntax for expressions involving addition and field selection:

$expr ::= expr + field$
 $expr ::= field$
 $field ::= expr . id$
 $field ::= id$

(a) (8 points) Show that this grammar is ambiguous.

2015 Midterm – Question 4

Question 4. Context-free grammars (14 points) Consider the following syntax for expressions involving addition and field selection:

$expr ::= expr + field$

$expr ::= field$

$field ::= expr . id$

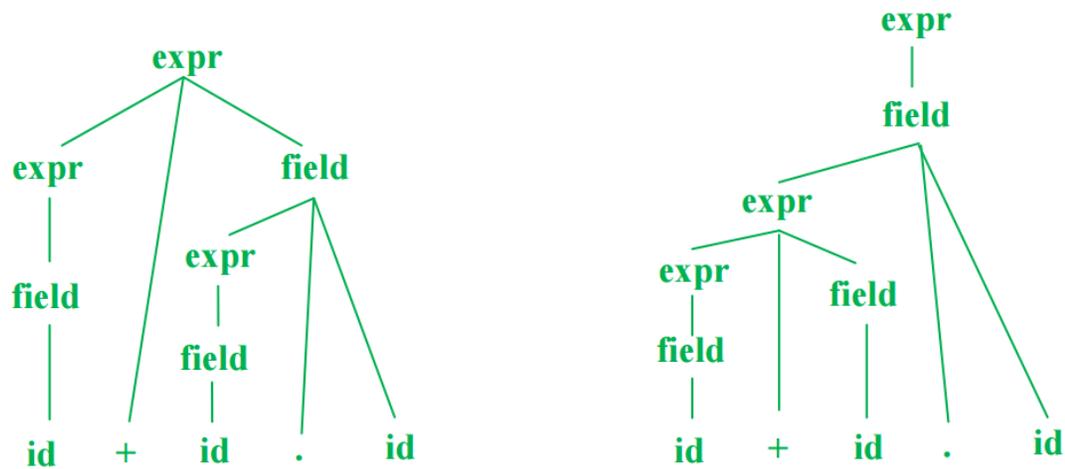
$field ::= id$

(b) (6 points) Give an unambiguous context-free grammar that fixes the problem(s) with the grammar in part (a) and generates expressions with `id`, field selection and addition. As in Java, field selection should have higher precedence than addition and both field selection and addition should be left-associative (i.e., `a+b+c` means `(a+b)+c`).

Solution (Not exhaustive)

(a) (8 points) Show that this grammar is ambiguous.

Here are two derivations of `id+id.id`:



Solution (Not exhaustive)

(b) (6 points) Give an unambiguous context-free grammar that fixes the problem(s) with the grammar in part (a) and generates expressions with `id`, field selection and addition. As in Java, field selection should have higher precedence than addition and both field selection and addition should be left-associative (i.e., `a+b+c` means `(a+b)+c`).

The problem is in the first rule for *field*, which creates an ambiguous precedence. Here is a reasonably simple fix.

```
expr ::= expr + field  
expr ::= field  
field ::= field . id  
field ::= id
```