University of Kansas | Drew Davidson

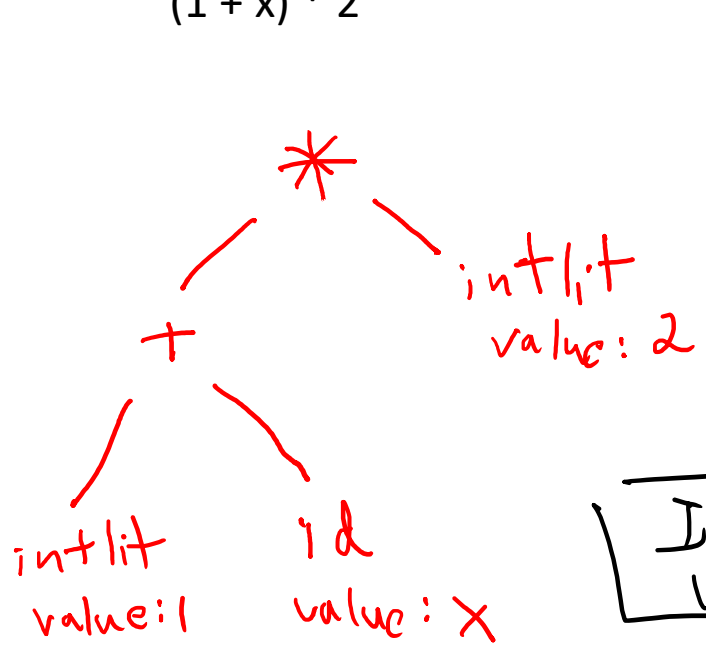
Flipped Wednesday!

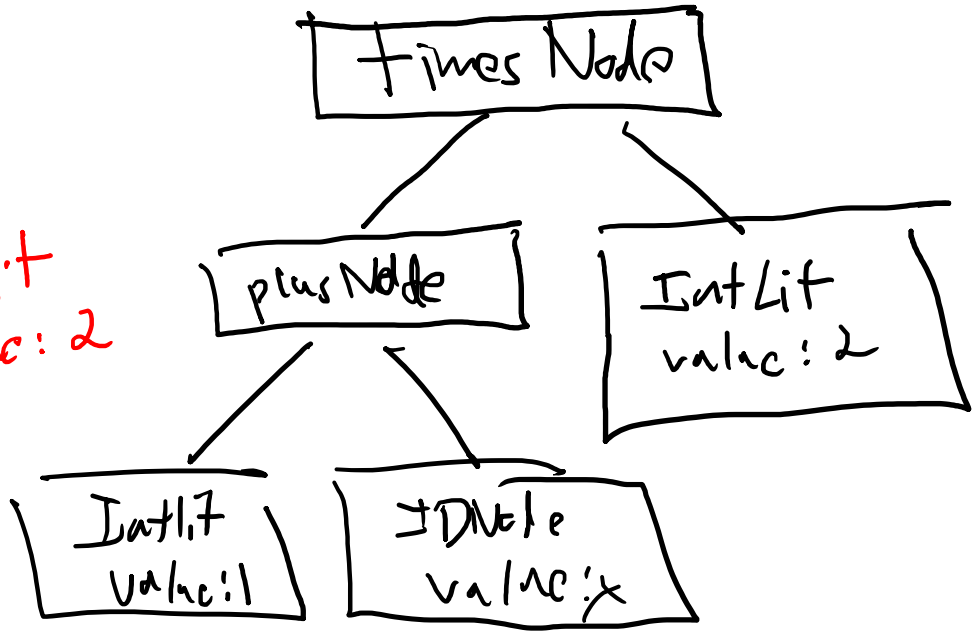
Check-In #7

Warm-up

Show the AST for the expression

(1 + x) * 2





Agenda

Wednesday

- ✓ Check-in
- Announcements
- Written Work #2
- Open Discussion / Questions

Announcements

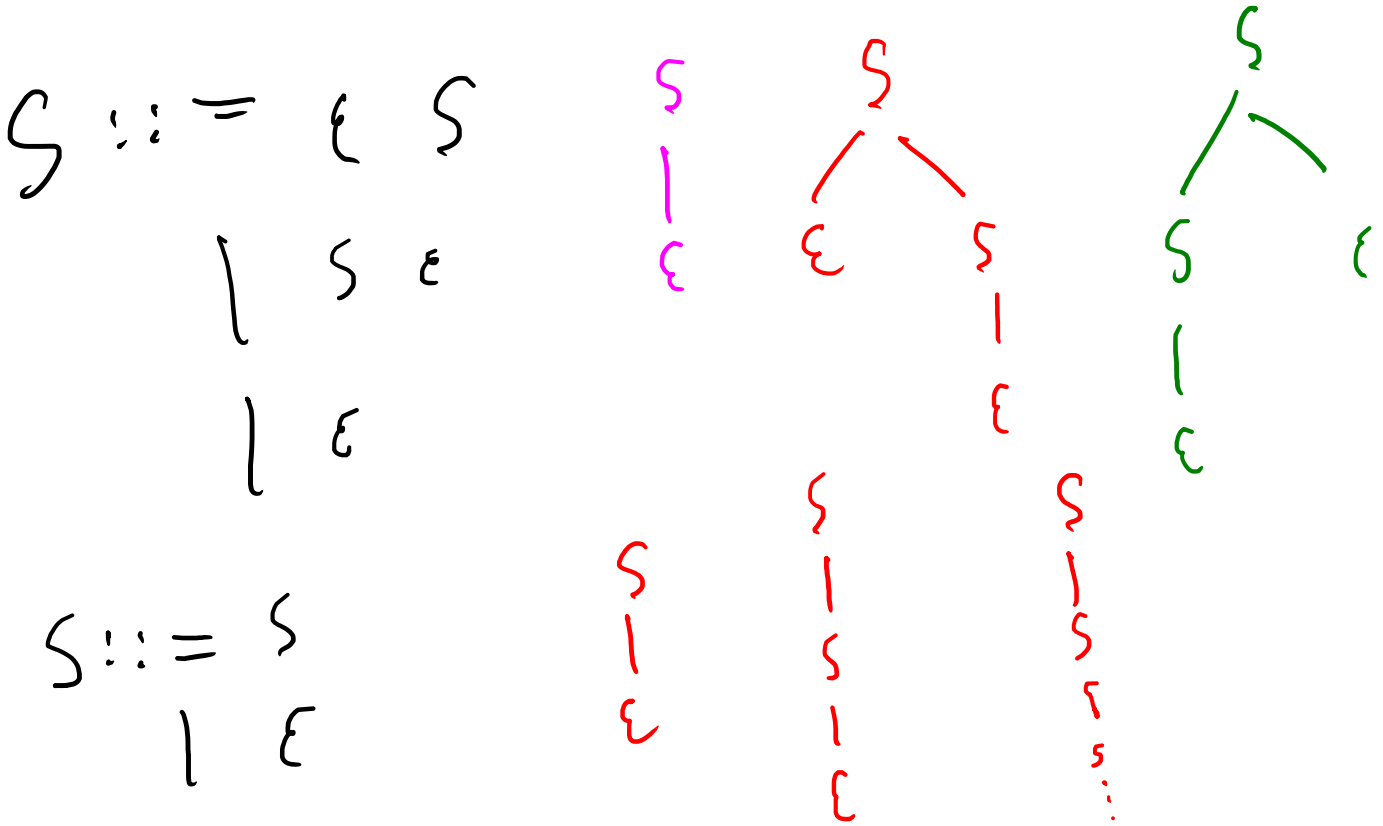
Wednesday

- Quiz 1 is Friday, **in class**
- Quiz 1 review is tonight at 5:00 at my Office hours link: https://kansas.zoom.us/j/6894602792
- Grades for Written Work were placed incorrectly, now fixed (??)
- Reminder: laptops/naps are for the back row
- Reminder: “drop date” is Monday night

Question #1

Written Work #2

- Create a Context Free Grammar that only accepts the empty string, but is still ambiguous



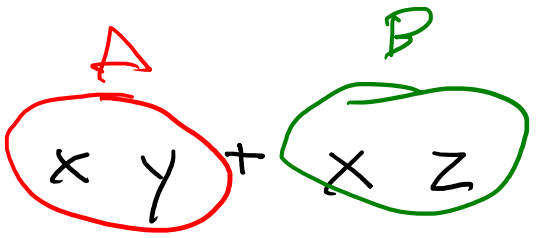
Question #2

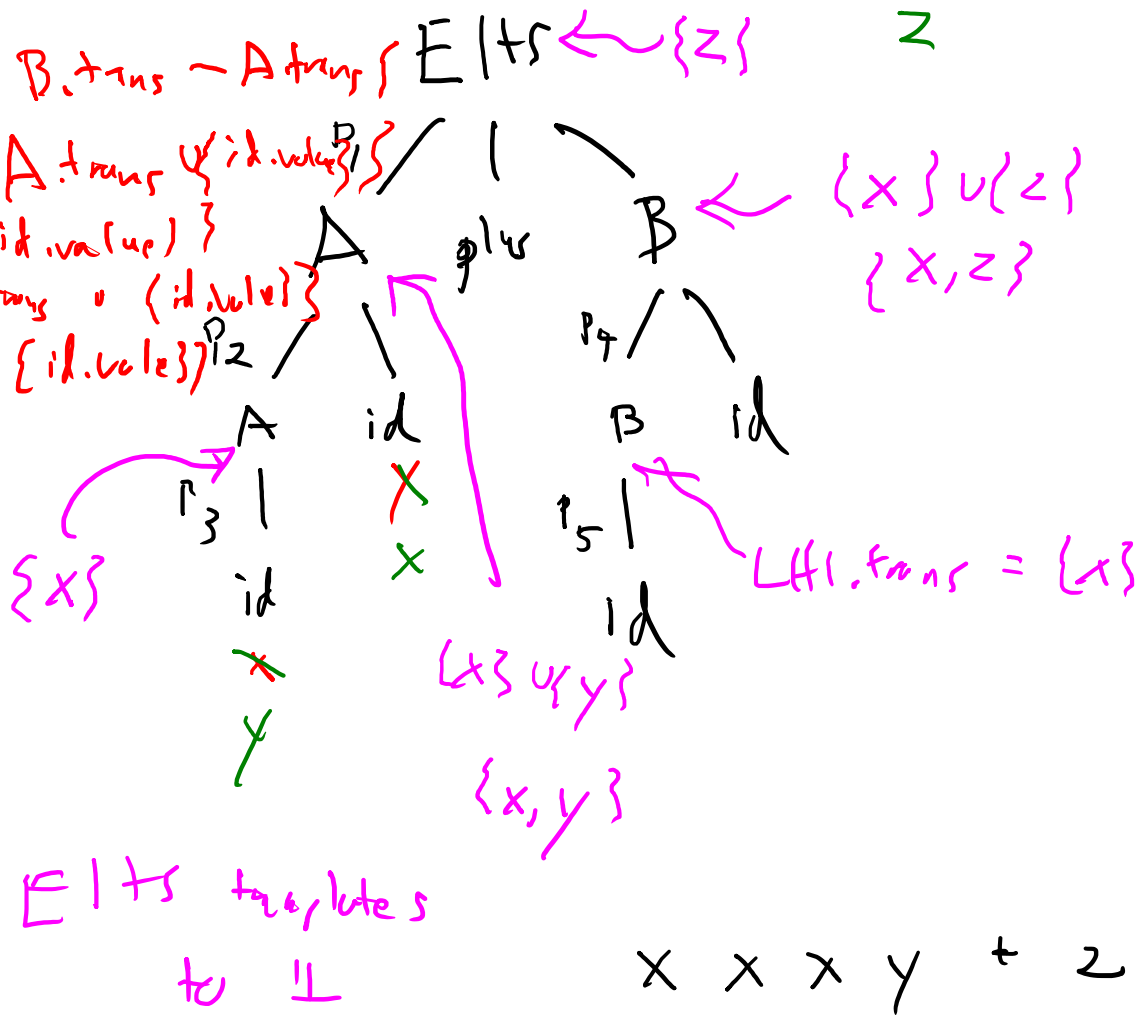
Written Work #2

Consider the following grammar:

p_1	*Elts*	::=	*A* **plus** *B*
p_2	*A*	::=	*A* **id**
p_3		|	**id**
p_4	*B*	::=	*B* **id**
p_5		|	**id**

Assume that a tokenizer has already imbued each **id** with a value field that contains the name of the identifier. Write an SDD that sets the *Elts* translation to be set of identifier names mentioned under a *B* nonterminal, but never mentioned under an *A* nonterminal.





Question #3

Written Work #2

DL ::= ε
| dot A
A ::= comma dot A

Let *DotList* be a language such that:

- The empty string is in the language
- The single terminal **dot** is in the language
- Sequences of more than 1 **dot** terminal separated by the **comma** terminal are in the language. e.g.:
  - **dot comma dot**
  - **dot comma dot comma dot**

No other strings are in the language

Write an unambiguous grammar that recognizes *DotList*

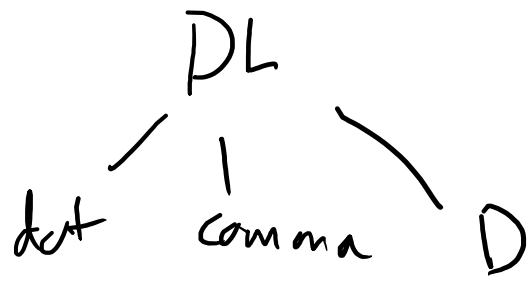
P1 DL ::= ε | ε

P2 ~~| dot~~

P3 | ~~dot comma D~~ D

D ::= dot
| dot comma D

DL
|
D



DL
|
dot

DL
|
D
|
dot

Question #4

Written Work #2

Alter the expression grammar to recognize expressions with explicit parentheses, e.g. allows the expression 1 - (2 - 3)

```
E ::= E - T
    | T
T ::= T * F
    | F
F ::= intlit
```