Syntax Directed Translation

Syntax directed translation

- Yacc can do a simple kind of syntax directed translation from an input sentence to C code
- We can also think of it as compilation
- Each node in a parse tree produces a value
  - That value depends on the type of the node
  - And on the values produced by its children
- The value is usually produced by a rule associated with the node
- This is just the rules in Yacc, e.g.:
  - \( \text{( } S\text{ = } S1 \text{ + } S3; \text{ )} \)

Why is \( a == 200 \) after this?

```
pcalc> calc
331 Calculator
(type ? for help and . to exit)
>> a = 1
1
>> if 1 (a = 100) (a = 200)
100
>> a
200
>>
```

Example

Grammar + input string => parse tree

```
1 * 2 + a
```

Example

Grammar + input string => parse tree + annotations

```
1 * 2 + a
```

Example

Do a post order traversal of the annotated parse tree to determine the execution order of the nodes.
Conditionals

- If evaluates all three args and selects the value to return
- Evaluation is bottom up, left to right
- Watch out for side effects!

```
e  = a1

if ($2) $5 = $3; else $5 = $4;
$1 = value = $3; $5 = $3;
```

```
e e

if

$1 = value = $3; $5 = $3;
```
```
a = 1
e = 2
```

```
e

if ($2) $5 = $3; else $5 = $4;
$1 = value = $3; $5 = $3;
```
```
e
```
```
```