

KU | EECS | Drew Davidson

EECS 665

COMPILER

CONSTRUCTION

Written Work #5

Administrivia

- A very special double-flipped Wednesday
- Remember – no class today
- Also remember – Quiz 2 on Friday!

Flipped Wednesday



Written Work Topics

- Saying farewell to parsers (LR)
- Name Analysis
- Type Analysis

Question 1

Recall that C++ is a statically-scoped language. Assume a new variant of C++, called C++', that uses the exact same syntax as C++ but is dynamically-scoped. Provide an input file that would be valid in both C++ and C++', but prints a different result based on which scoping system is used. Show what the output would be either under scheme.

Question 1 Solution

Recall that C++ is a statically-scoped language. Assume a new variant of C++, called C++', that uses the exact same syntax as C++ but is dynamically-scoped. Provide an input file that would be valid in both C++ and C++', but prints a different result based on which scoping system is used. Show what the output would be either under scheme.

Example Program

```
1  int g;  
2  void foo() {  
3      g = 4;  
4  }  
5  void main() {  
6      int g = 1;  
7      foo();  
8      std::cout << g;  
9  }
```

Static bindings

g at line 3 binds to global decl (line 1)
g at line 6 binds to local decl (line 6)
g at line 8 binds to local decl (line 6)

Static Output

1

Dynamic bindings

g at line 6 binds to local decl (line 6)
g at line 3 binds to local decl (line 6)
g at line 8 binds to local decl (line 6)

Dynamic Output

4

Question 2

Provide an input file that would be valid in C++', but **not** in C++.

Question 2 Solution

Provide an input file that would be valid in C++', but **not** in C++.

~~Same Example Program~~

```
1 int g;
2 void foo() {
3     g = 4;
4 }
5 void main() {
6     int g = 1;
7     foo();
8     std::cout << g;
9 }
```

Question 3

Explain the difference between an SLR and an LR(0) parser. Provide an example grammar that is SLR-parseable but not LR(0) parseable.

Question 3 Solution Outline / Intuition

Explain the difference between an SLR and an LR(0) parser. Provide an example grammar that is SLR-parseable but not LR(0) parseable.

Parsing Table Sketch

	t_1	t_2	eof	NT_1	NT_2
I_0					
I_1					
I_2	I_k		red $\odot$		
I_3					
I_4					

Question 3 Solution

Explain the difference between an SLR and an LR(0) parser. Provide an example grammar that is SLR-parseable but not LR(0) parseable.

Grammar

P1 $X ::= Y$

P2 $X ::= Y a$

P3 $Y ::= b$

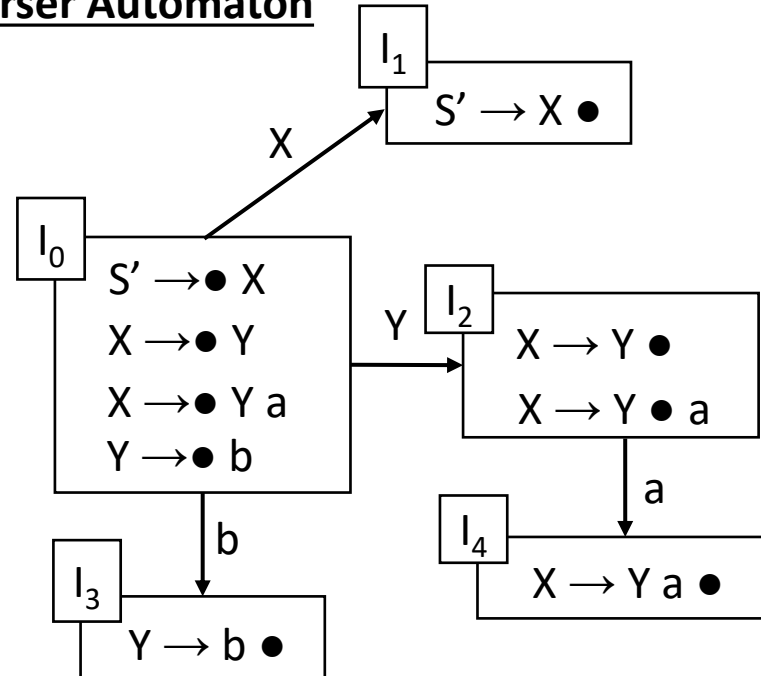
Question 3 Solution

Explain the difference between an SLR and an LR(0) parser. Provide an example grammar that is SLR-parseable but not LR(0) parseable.

Grammar

- P0 $S' ::= X$
- P1 $X ::= Y$
- P2 $X ::= Y a$
- P3 $Y ::= b$

Parser Automaton



Parsing Table

	a	b	eof	X	Y
I_0		I_3		I_1	I_2
I_1			accept		
I_2	I_4		red P1		
I_3	red P3		red P3		
I_4			red P2		

Question 4

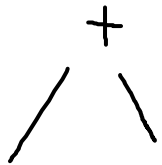
Explain how a type analysis that propagates an error type reduces cascading error reports.

Give an example of a program where propagating an error type might reduce the number of reports

Give an example of a program where propagating the error type still results in multiple reports within the same expression

Example Program

```
int main() {  
    bool c;  
    int a;  
    int b;  
    a = b = c = 4; ++ ++;  
}
```



Expression AST

