Introduction to Flexc++ and Bisonc++

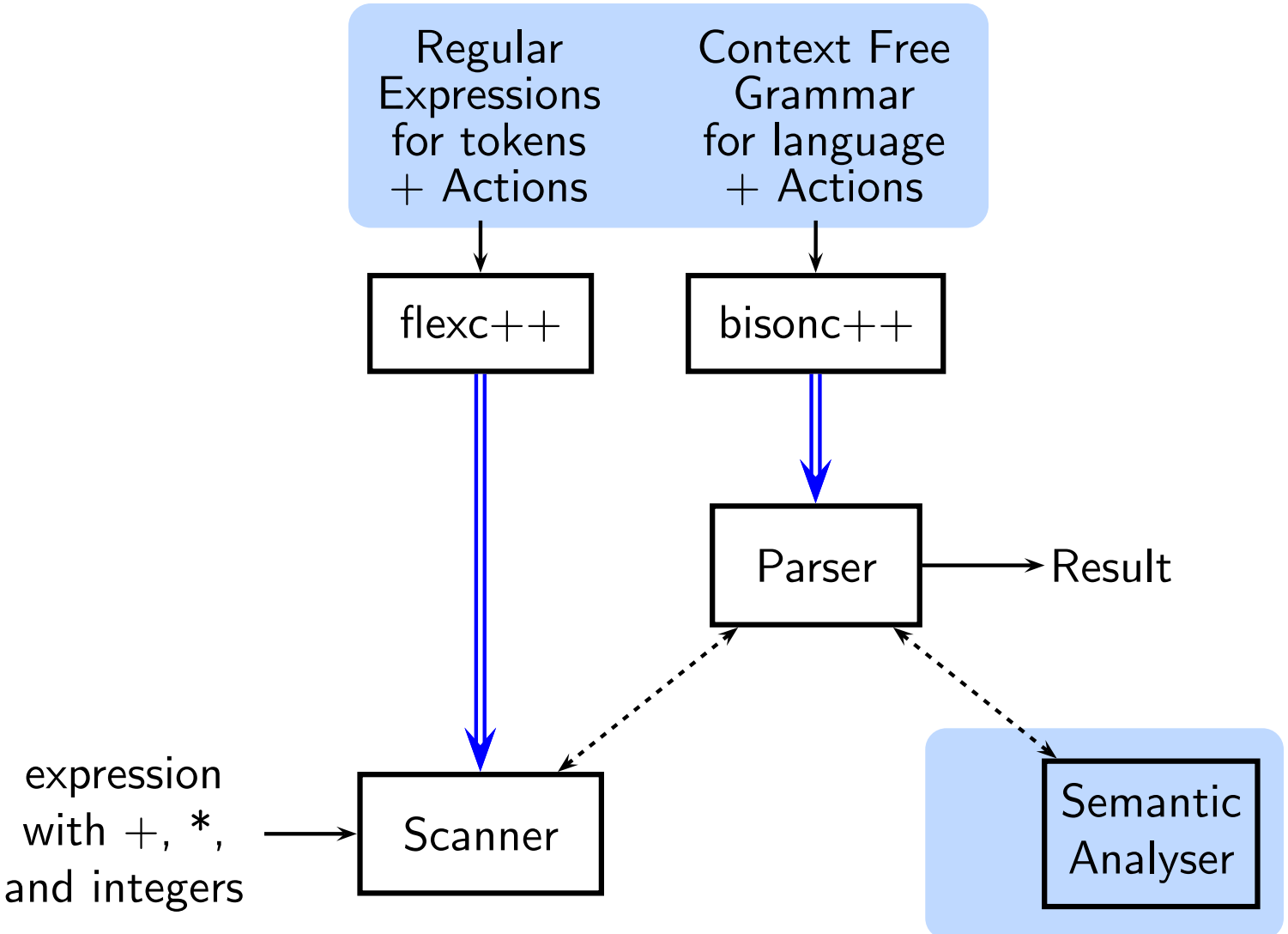
Uday Khedker

(www.cse.iitb.ac.in/~uday)

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay

Jan. 2014

simCalc: A Simple Calculator



Introduction to Lex and Yacc

- lex script num.l to show token identification
- exp1.y to show a simple expression. Use the lex script exp.l
- exp2.y to show construction of a PLUS expression
  Interesting input: `1 + 2 + 3 + 4 + 5 + 6`
- exp3.y fixes the problem by making + left associative
- exp4.y includes + and *
  Interesting inputs: 1 + 2 * 3 and 1 * 2 + 3
- exp5.y fixes the above problem

The Interaction Between Scanner and Parser

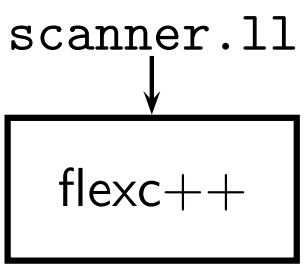
- Grammar

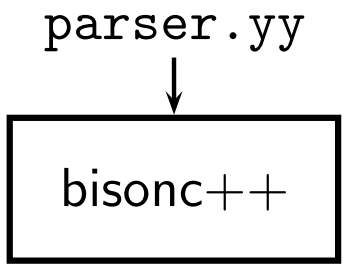
```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- Terminals and non-terminals get defined by the grammar
- Scanner identifies the tokens and communicates the details to the parser

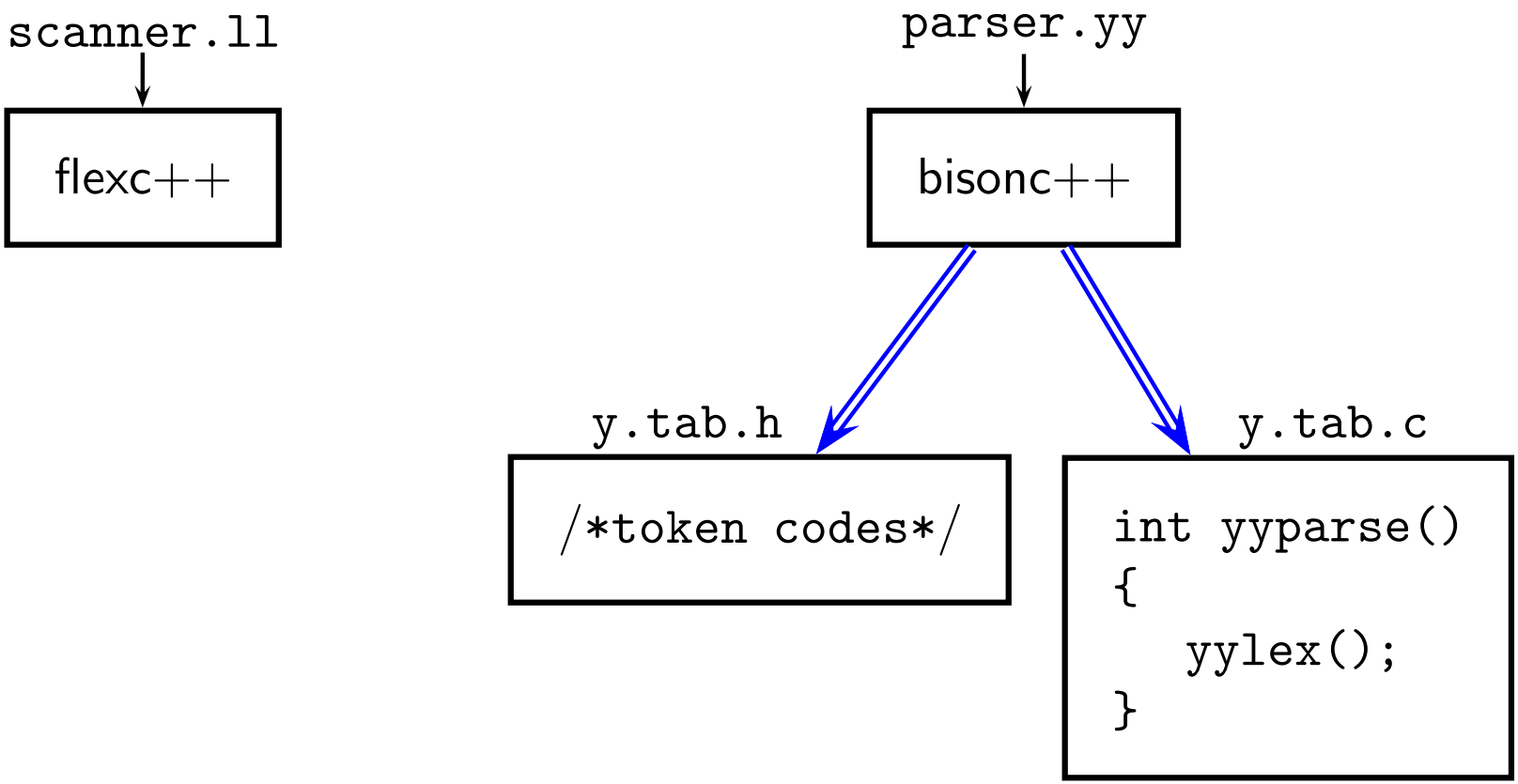
Token Name	Token Lexemes	Token Value	Token Code
Number	"10" "345" "03"	10 345 3	NUM
+ operator	"+"		'+'
* operator	"*"		'*'

The Interaction Between Scanner and Parser

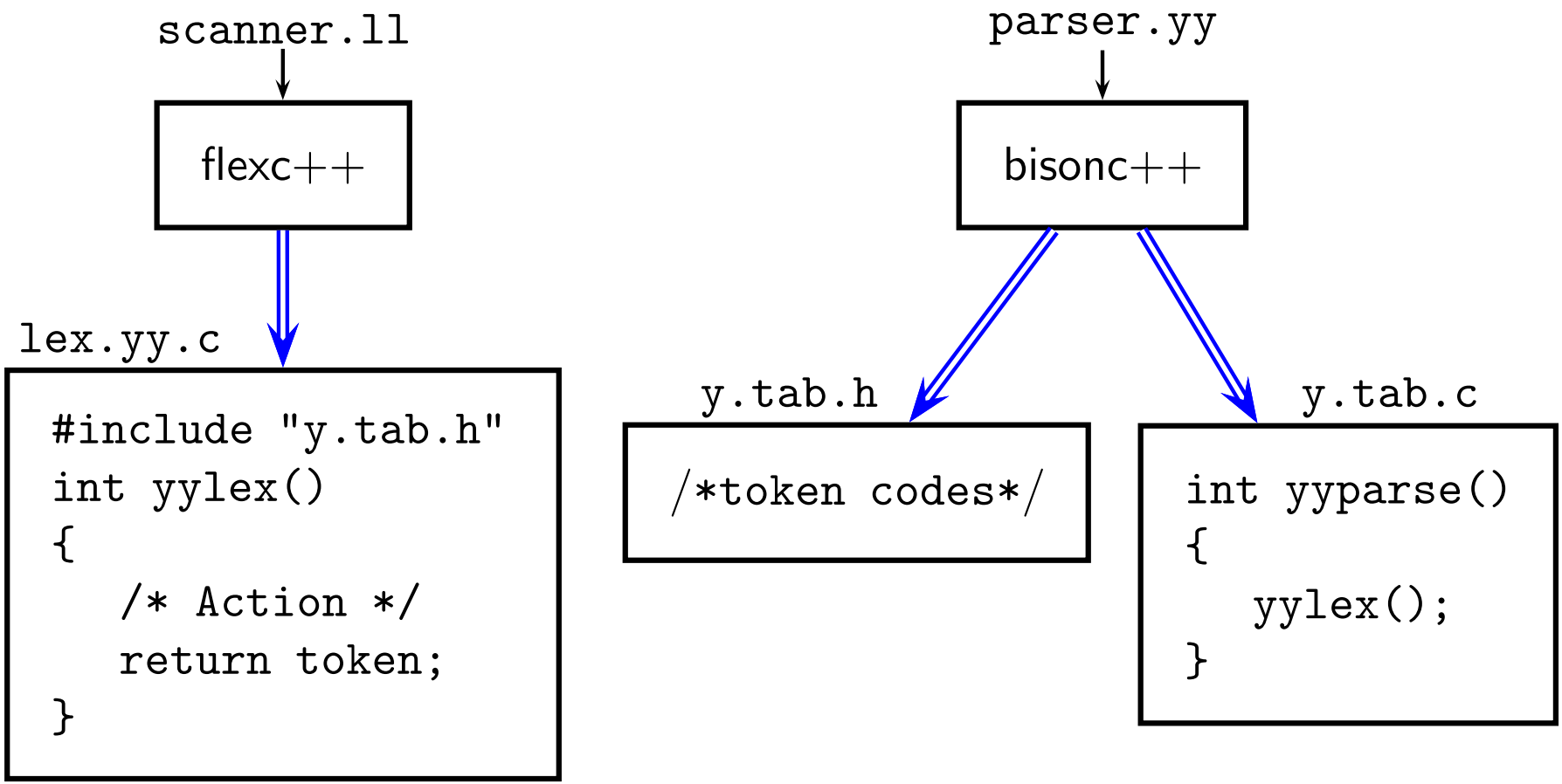




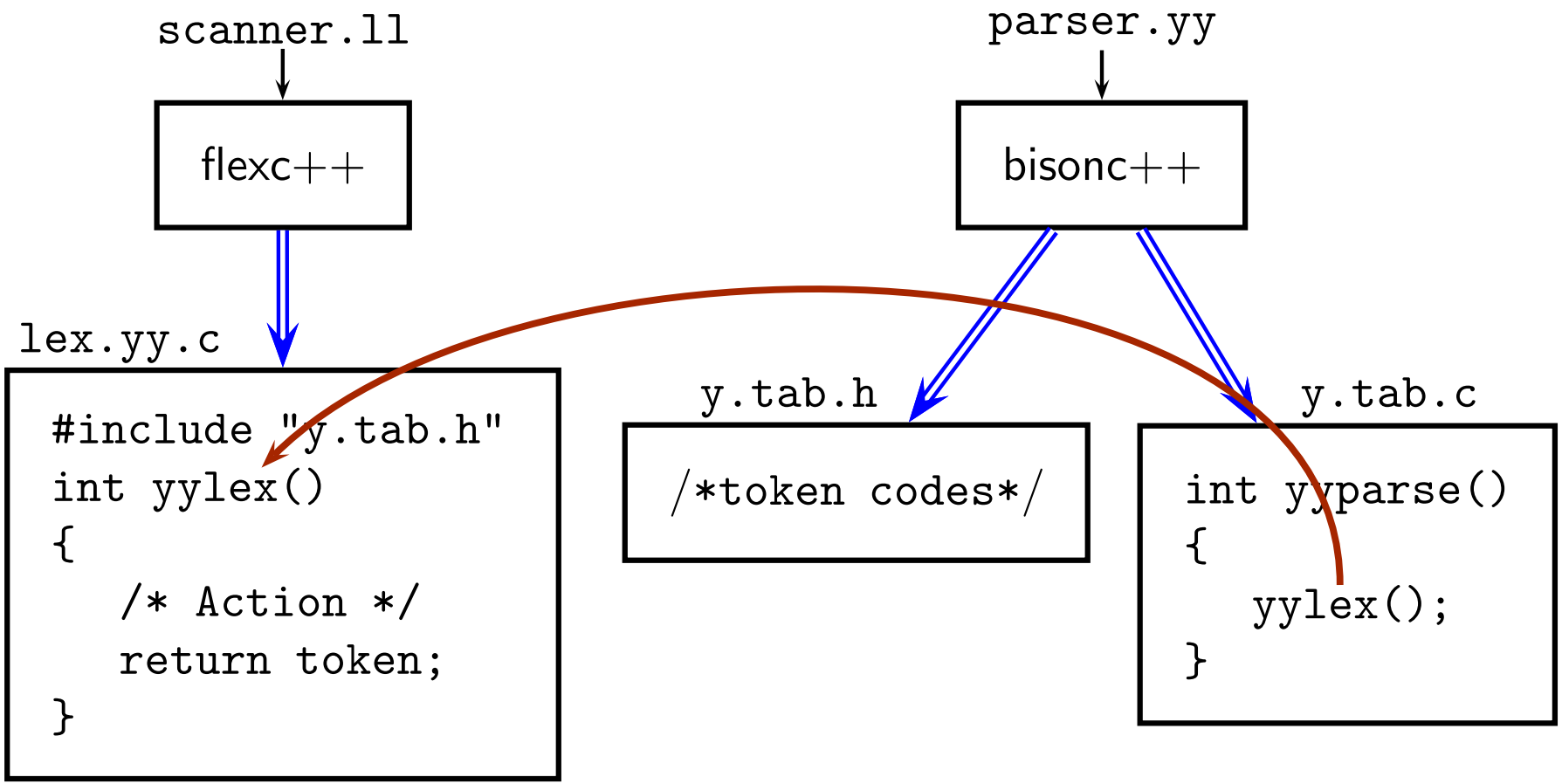
The Interaction Between Scanner and Parser



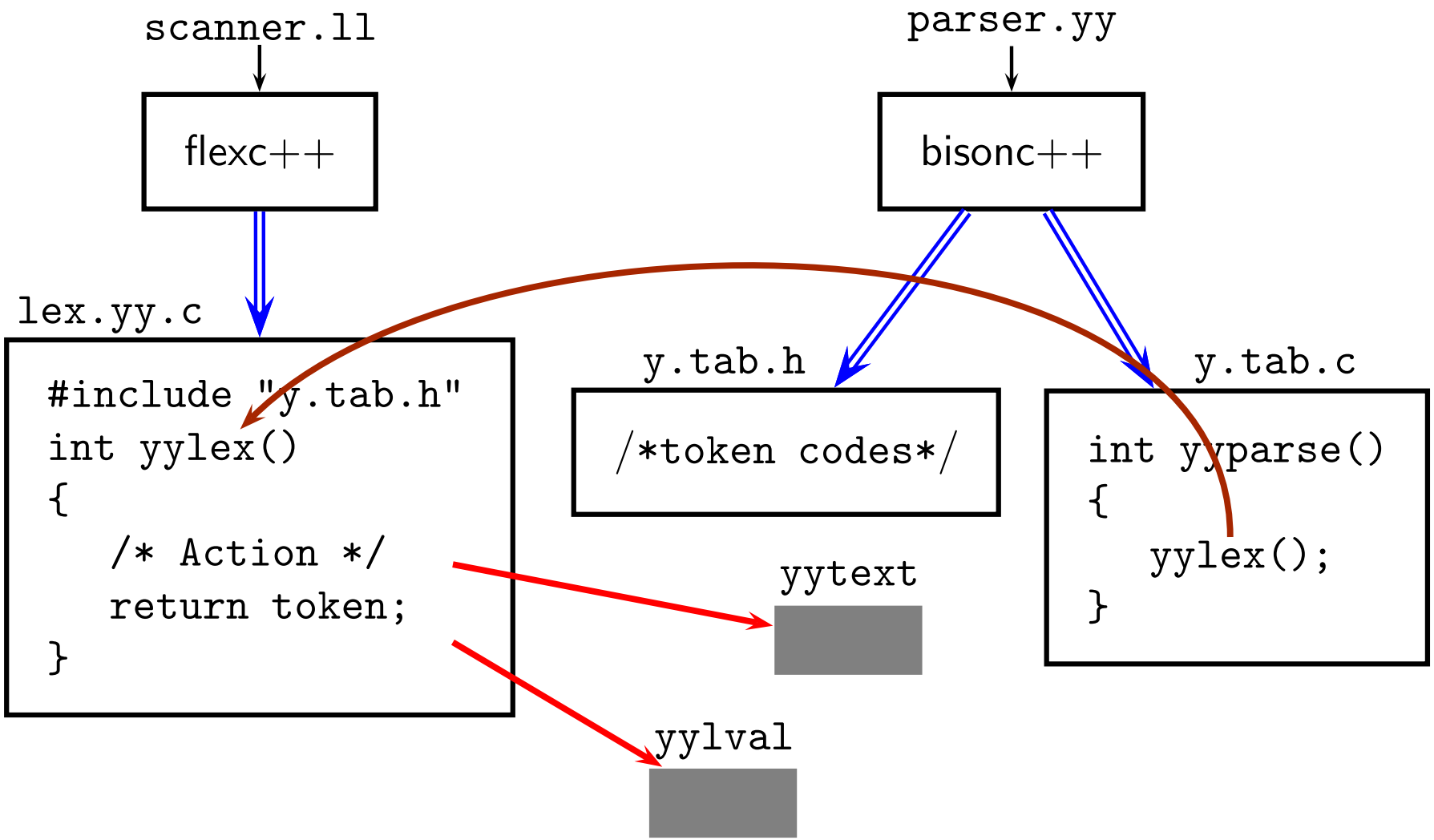
The Interaction Between Scanner and Parser



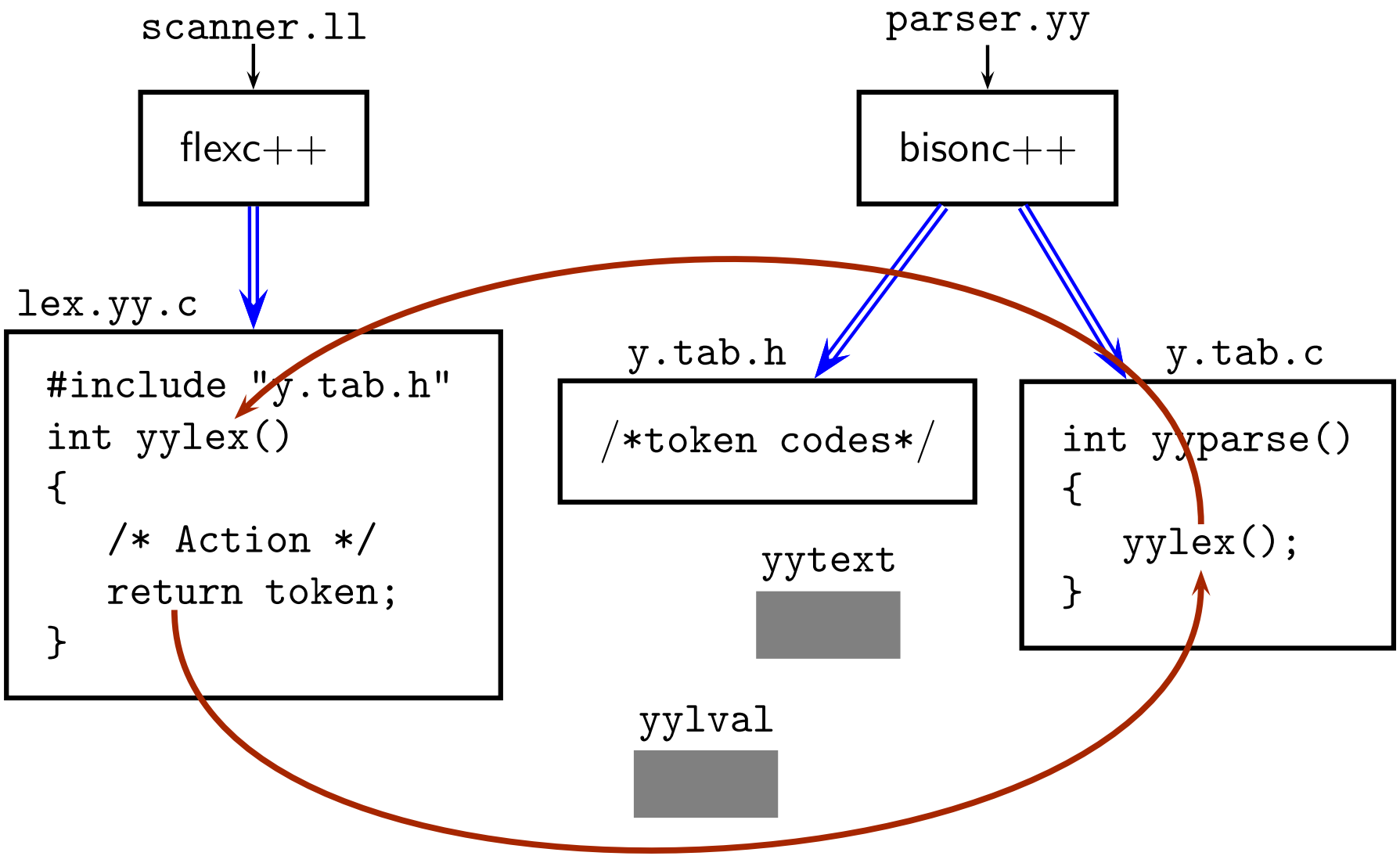
The Interaction Between Scanner and Parser



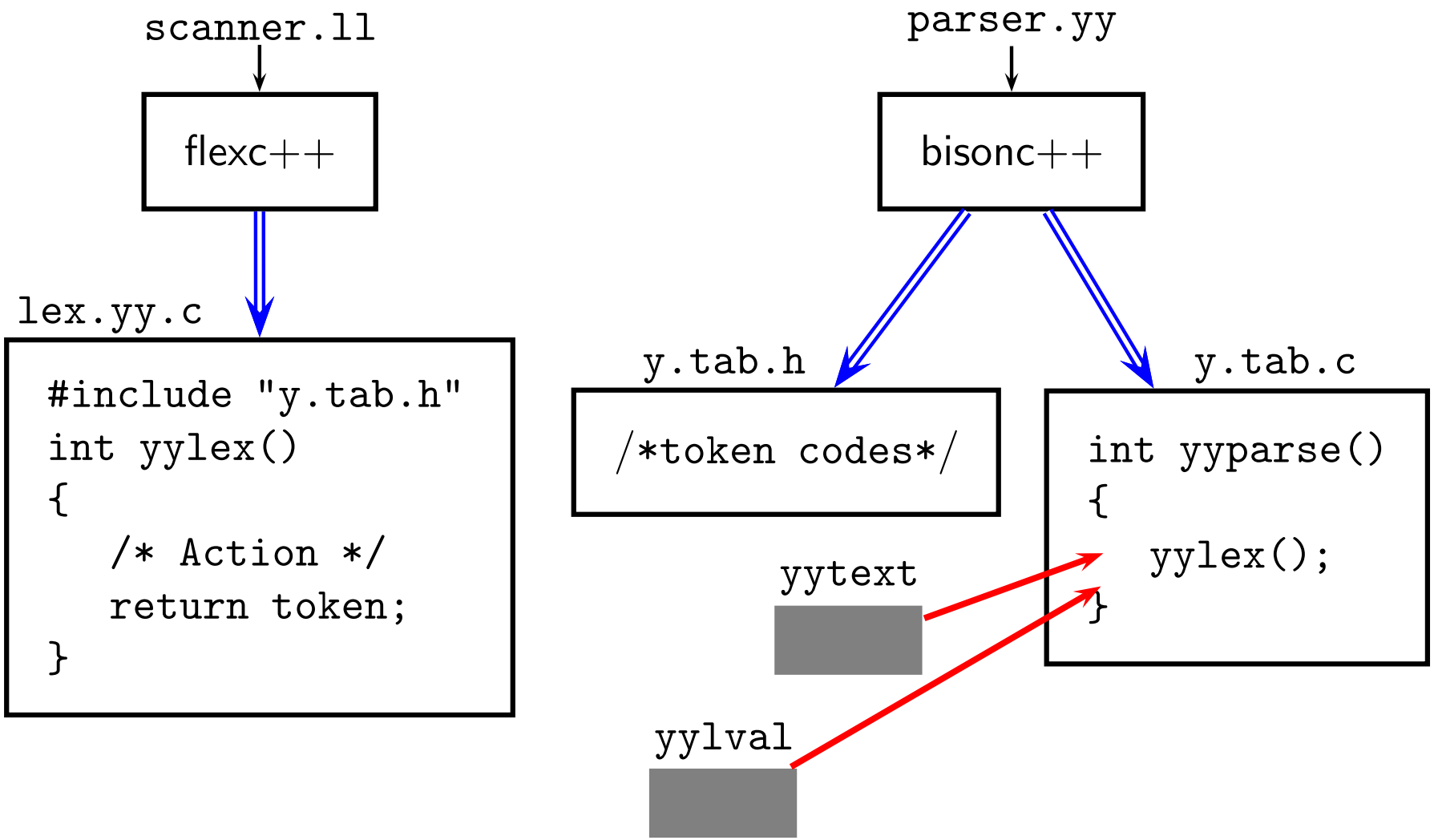
The Interaction Between Scanner and Parser



The Interaction Between Scanner and Parser



The Interaction Between Scanner and Parser



Some Issues in the Simple Calculator

- Syntax: Precedences and associativities
- Semantics: Values of
  - Terminals (from scanner)
  - Non-terminals (in the parser)
- Unit testing:

  Use of `#if SCANTEST` to test the scanner

Notes

- Sample inputs

  a 20 + b 3

- Use of "."
  - If a string can match two patterns, the first is chosen.
  - If a string can match a pattern and its prefix can also match a pattern, the prefix matching is ignored in favour of the longer string.

- The Lex and Yacc Page

  `http://dinosaur.compilertools.net/`

Passing Values From Scanner to the Parser

parser.h

```
class Parser:
      public ParserBase
{
  Scanner d_scanner;
  public:




  private:
    int parse();



};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();




  private:
    ...

};
```


Passing Values From Scanner to the Parser

parser.h

```
class Parser:
     public ParserBase
{
  Scanner d_scanner;
  public:




  private:
    int parse();

  protected: /*inherited*/
   STYPE__    d_val__;
};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();




  private:
    ...

};
```


Passing Values From Scanner to the Parser

parser.h

```
class Parser:
     public ParserBase
{
  Scanner d_scanner;
  public:




  private:
    int parse();

  protected: /*inherited*/
   STYPE__     d_val__;
};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();




  private:
    ...
    ParserBase::STYPE_ * dval;
};
```

Passing Values From Scanner to the Parser

parser.h

```
class Parser:
     public ParserBase
{
  Scanner d_scanner;
  public:
    Parser()
    { d_scanner.setSval(
                  &d_val__);
    }

  private:
    int parse();

  protected: /*inherited*/
   STYPE__    d_val__;
};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();
    void setSval(
          ParserBase::STYPE_ * val);
    ParserBase::STYPE_  * getSval();

  private:
    ...
    ParserBase::STYPE_ * dval;
};
```

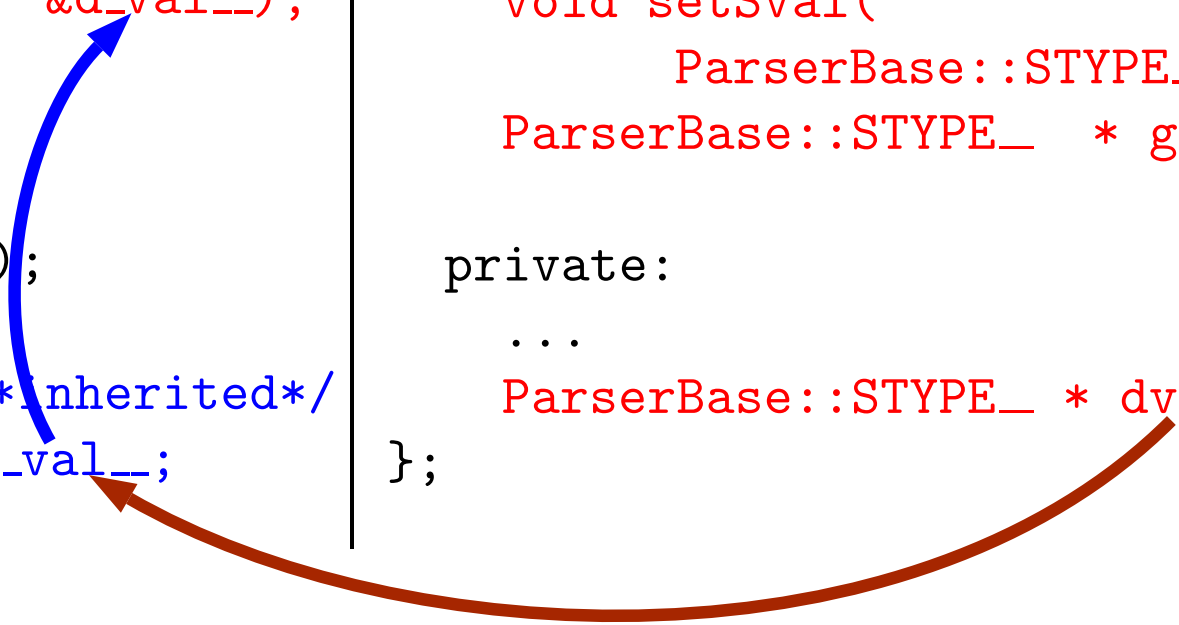
Passing Values From Scanner to the Parser

parser.h

```
class Parser:
     public ParserBase
{
  Scanner d_scanner;
  public:
    Parser()
    { d_scanner.setSval(
                  &d_val__);
    }

  private:
    int parse();

  protected: /*Inherited*/
   STYPE__    d_val__;
};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();
    void setSval(
          ParserBase::STYPE_ * val);
    ParserBase::STYPE_  * getSval();

  private:
    ...
    ParserBase::STYPE_ * dval;
};
```



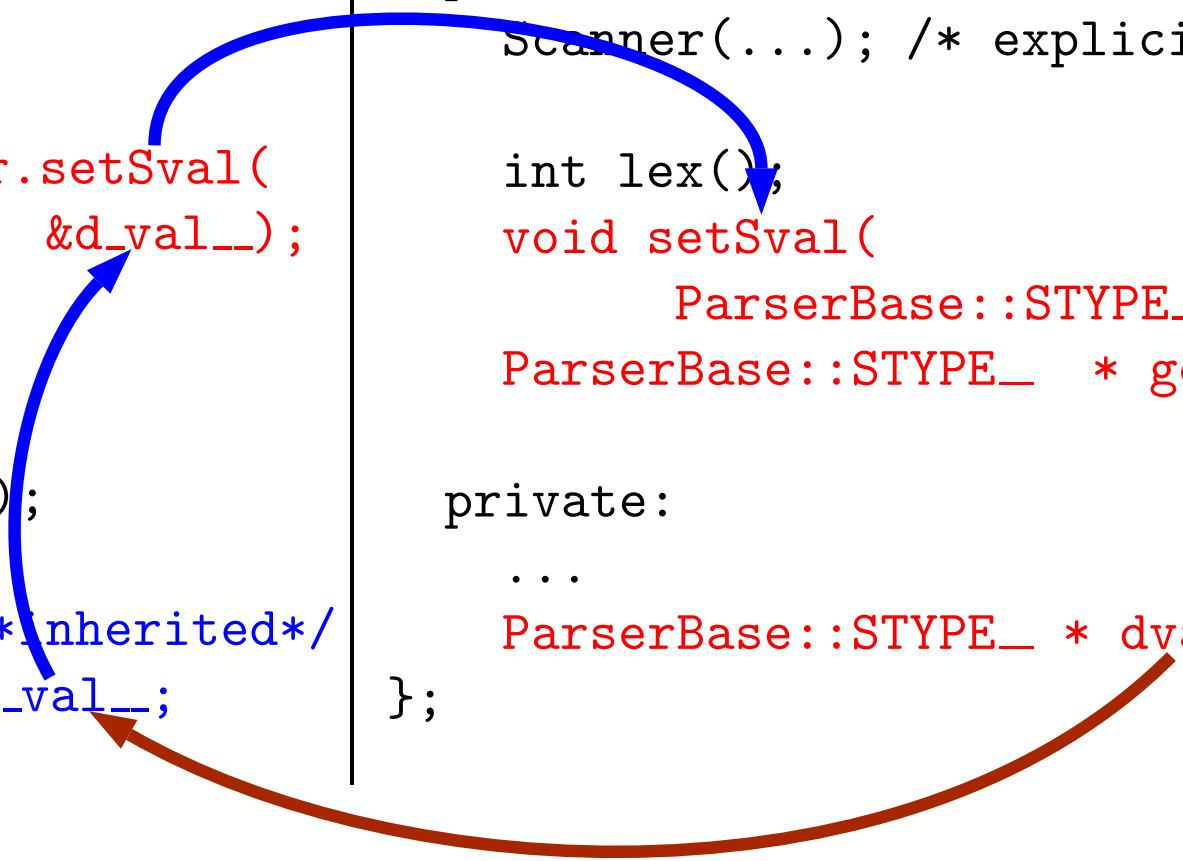
Passing Values From Scanner to the Parser

parser.h

```
class Parser:
     public ParserBase
{
  Scanner d_scanner;
  public:
    Parser()
    { d_scanner.setSval(
                  &d_val__);
    }

  private:
    int parse();

  protected: /*Inherited*/
   STYPE__    d_val__;
};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();
    void setSval(
          ParserBase::STYPE_ * val);
    ParserBase::STYPE_  * getSval();

  private:
    ...
    ParserBase::STYPE_ * dval;
};
```



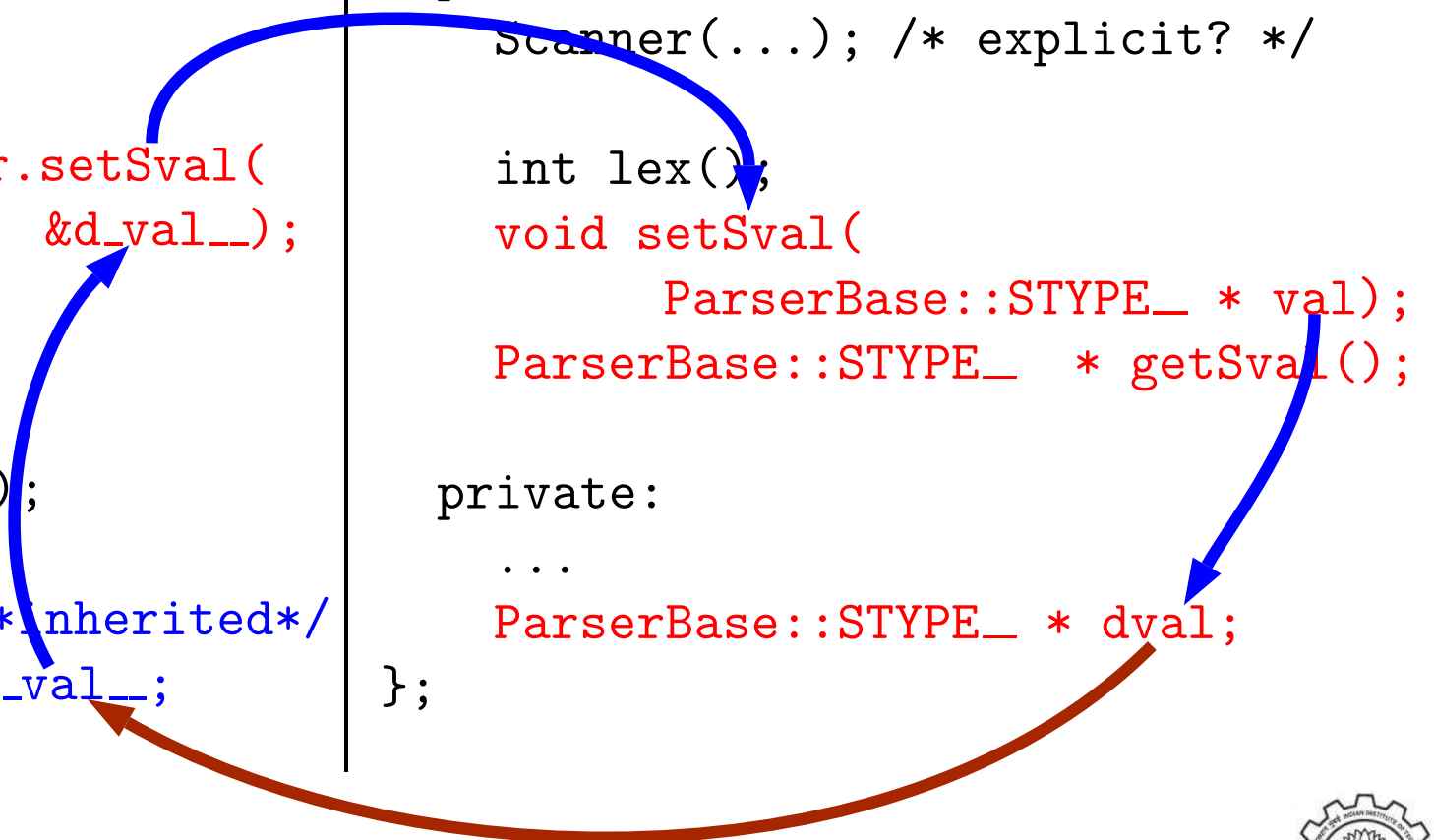
Passing Values From Scanner to the Parser

parser.h

```
class Parser:
     public ParserBase
{
  Scanner d_scanner;
  public:
    Parser()
    { d_scanner.setSval(
                  &d_val__);
    }

  private:
    int parse();

  protected: /*Inherited*/
   STYPE__    d_val__;
};
```

scanner.h

```
class Scanner:
    public ScannerBase
{
  public:
    Scanner(...); /* explicit? */

    int lex();
    void setSval(
          ParserBase::STYPE_ * val);
    ParserBase::STYPE_  * getSval();

  private:
    ...
    ParserBase::STYPE_ * dval;
};
```



An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10	+	20	*	3	eof
NUM					

Parsing Stack

Action: Shift NUM

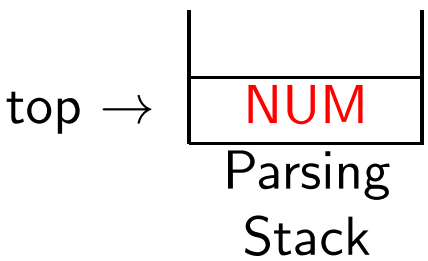
An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10	+	20	*	3	eof
	+				



Action: Reduce by "Expr: NUM"

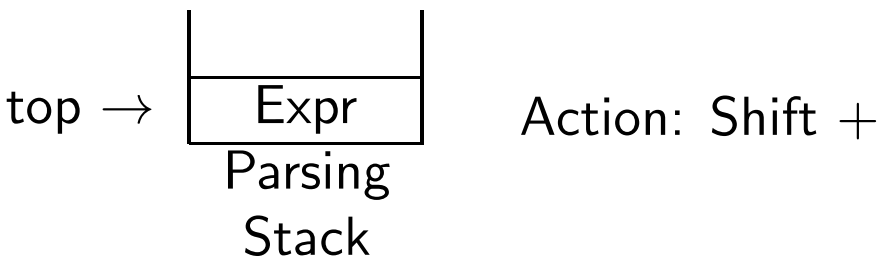
An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10	+	20	*	3	eof
	+				



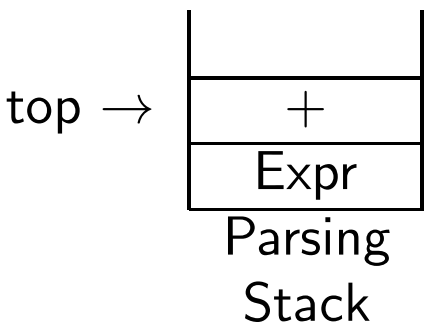
An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10	+	20	*	3	eof
		NUM			



Action: Shift NUM

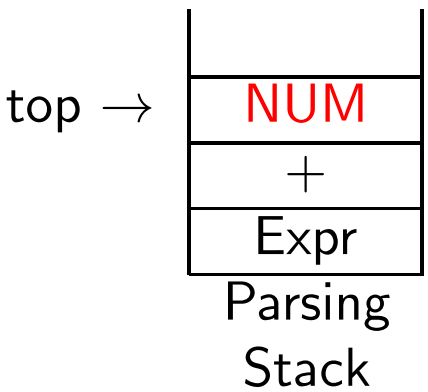
An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10	+	20	*	3	eof
			*		



Action: Reduce by "Expr: NUM"

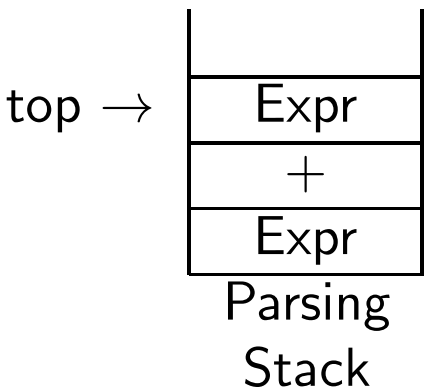
An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10	+	20	*	3	eof
			*		



Action: Shift $*$ or Reduce by "Expr: Expr + Expr"?

An Overview of Shift Reduce Parsing

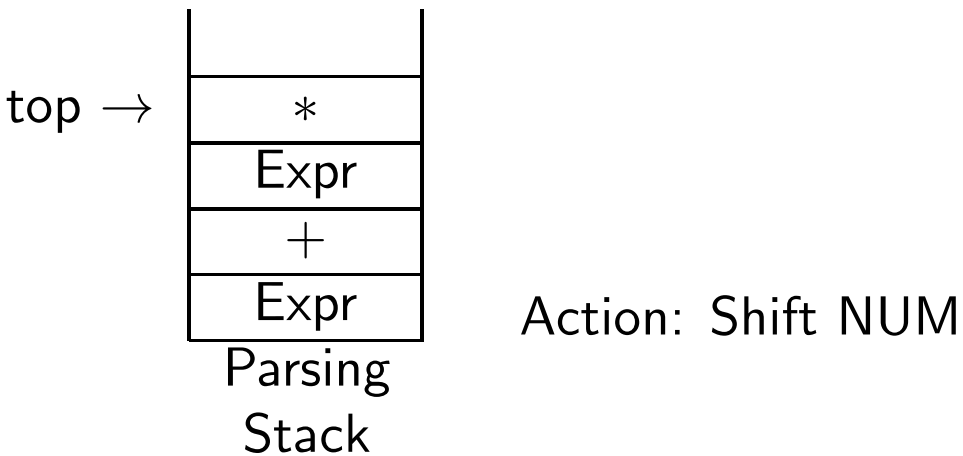
- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10 + 20 * 3 eof

NUM



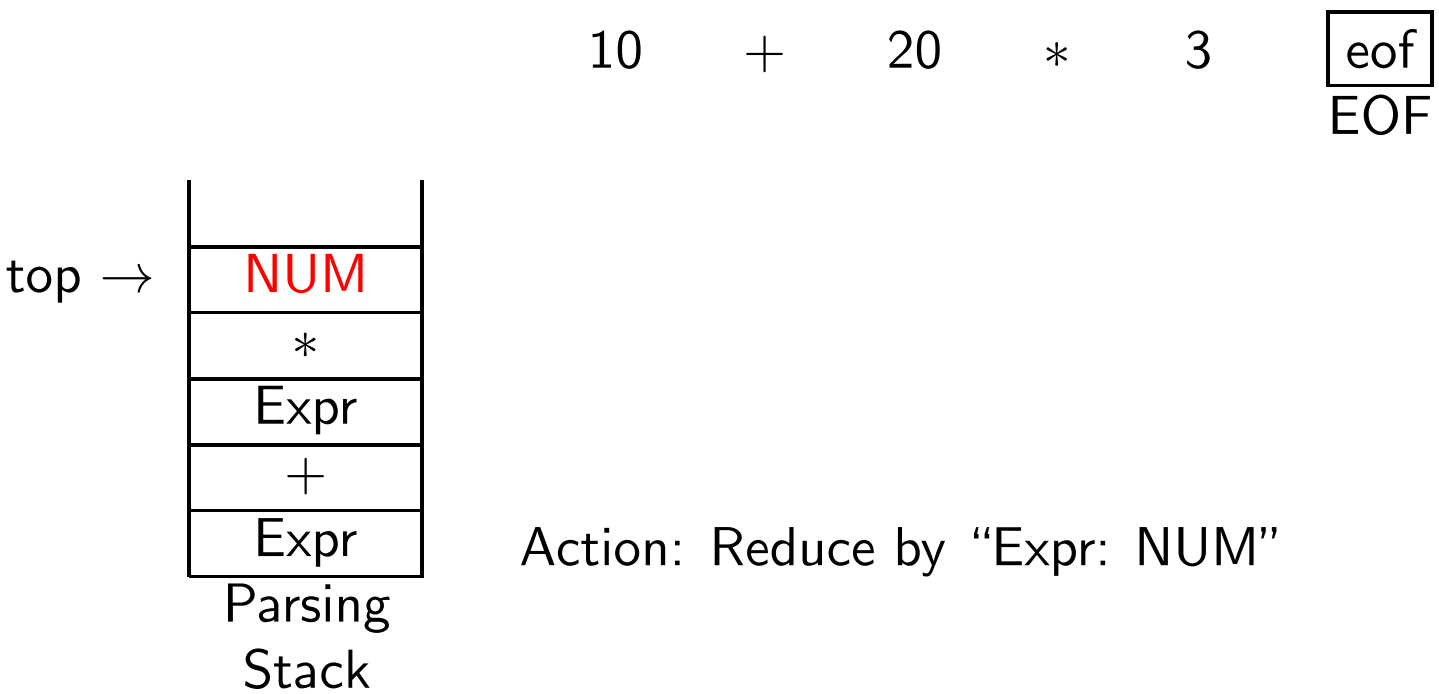
Action: Shift NUM

An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing



An Overview of Shift Reduce Parsing

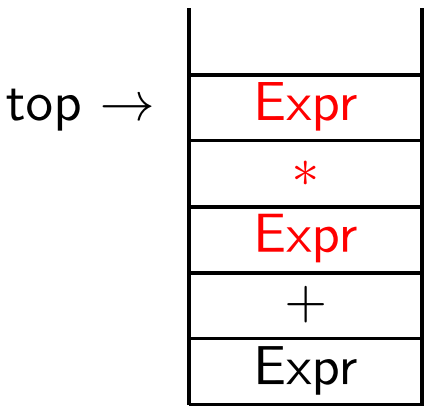
- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

10 + 20 * 3 | eof

EOF



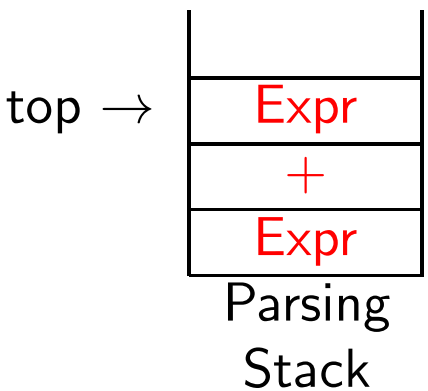
Action: Reduce by "Expr: Expr * Expr"

An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing



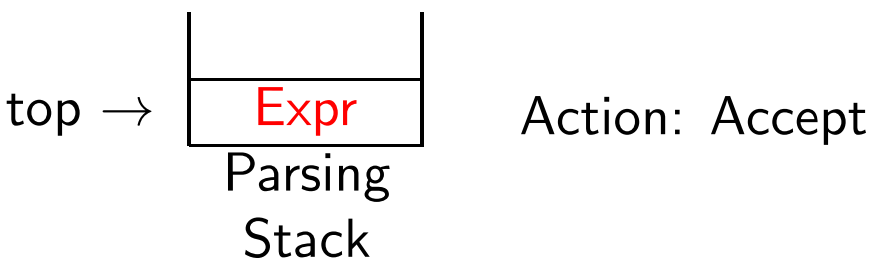
Action: Reduce by "Expr: Expr + Expr"

An Overview of Shift Reduce Parsing

- Grammar

```
Expr   :   Expr '+' Expr
       |   Expr '*' Expr
       |   NUM
```

- The process of parsing

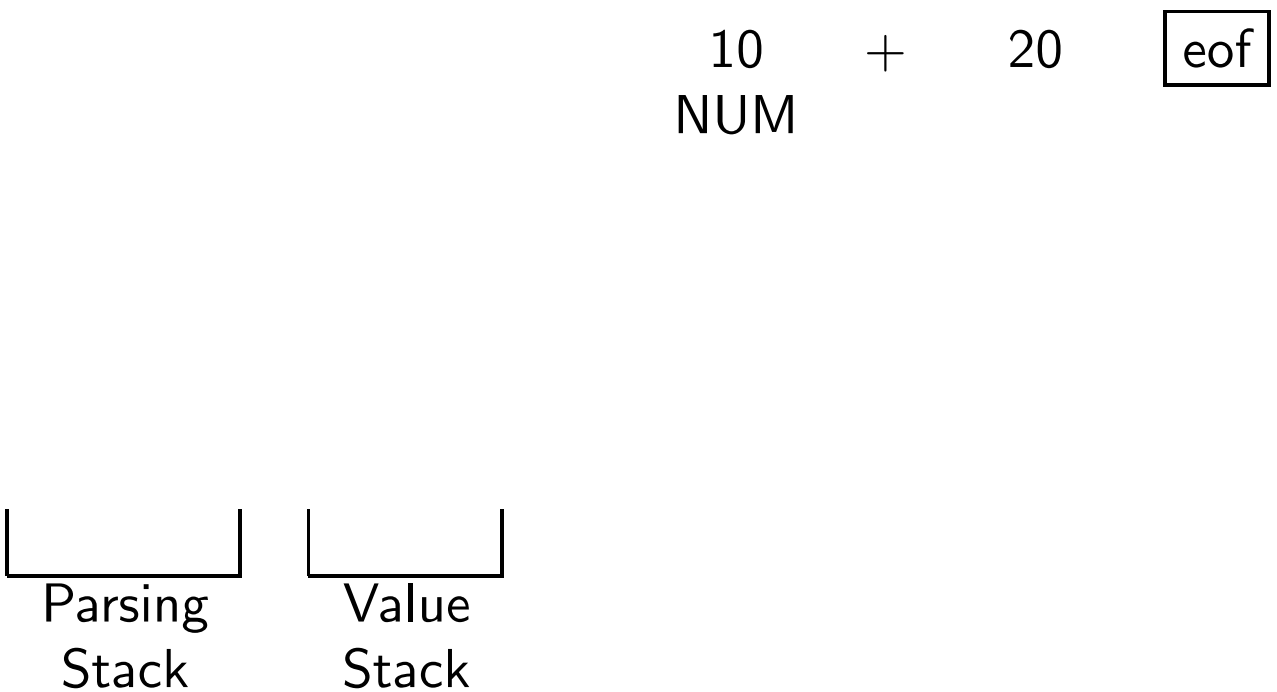


An Overview of Attribute Evaluation

- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation



An Overview of Attribute Evaluation

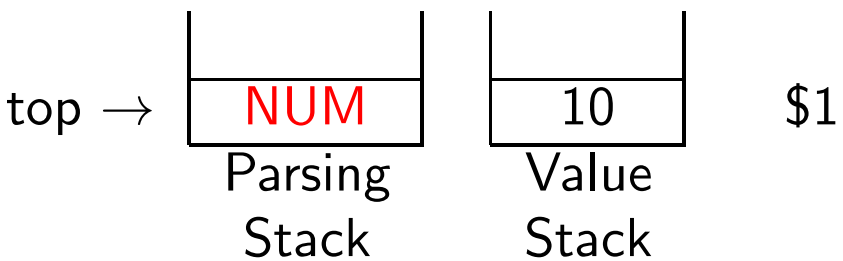
- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation

10 + 20 eof

+

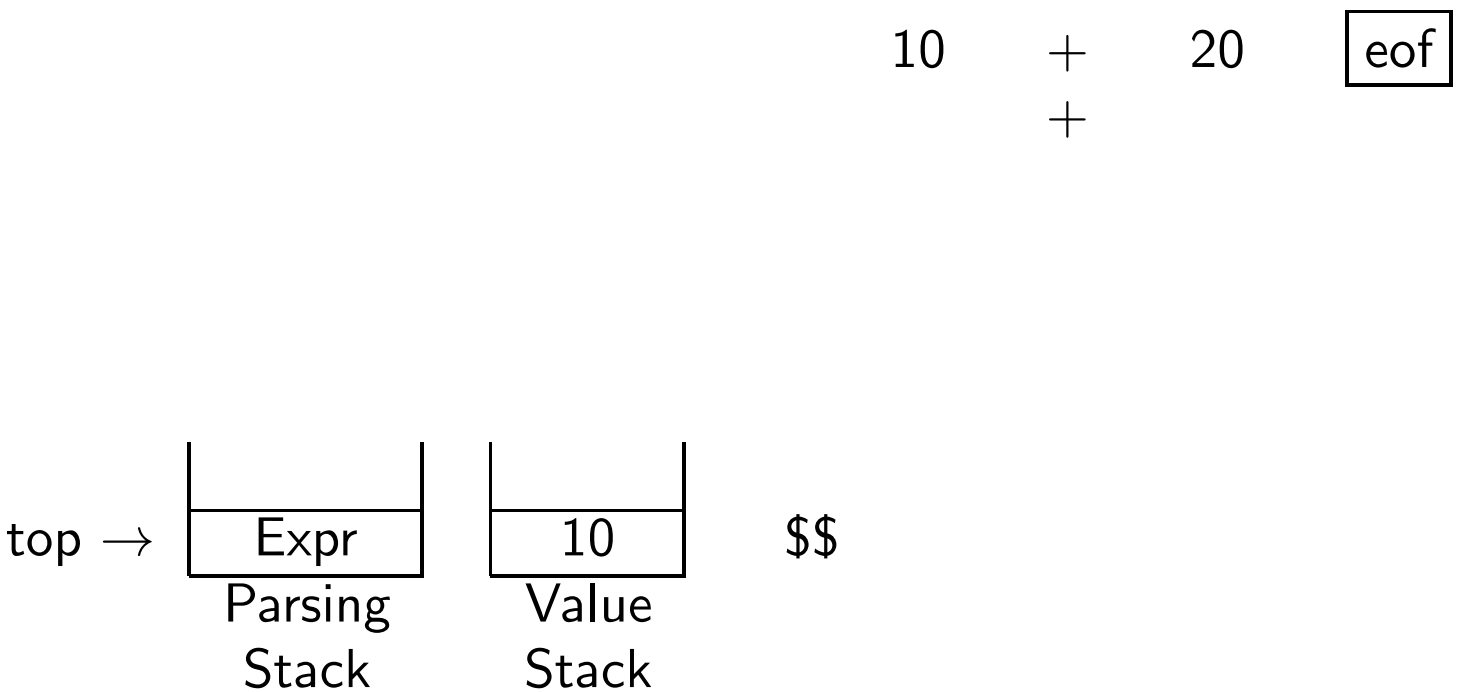


An Overview of Attribute Evaluation

- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation



An Overview of Attribute Evaluation

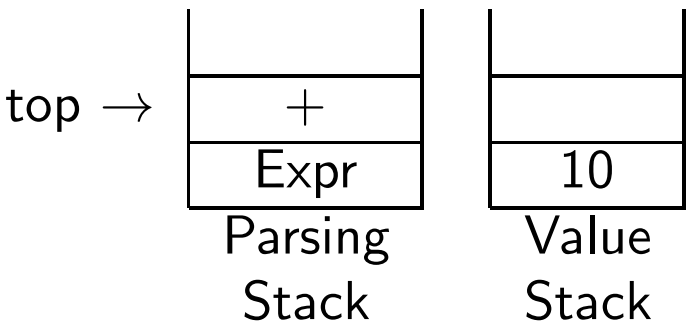
- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation

10 + 20 eof

NUM

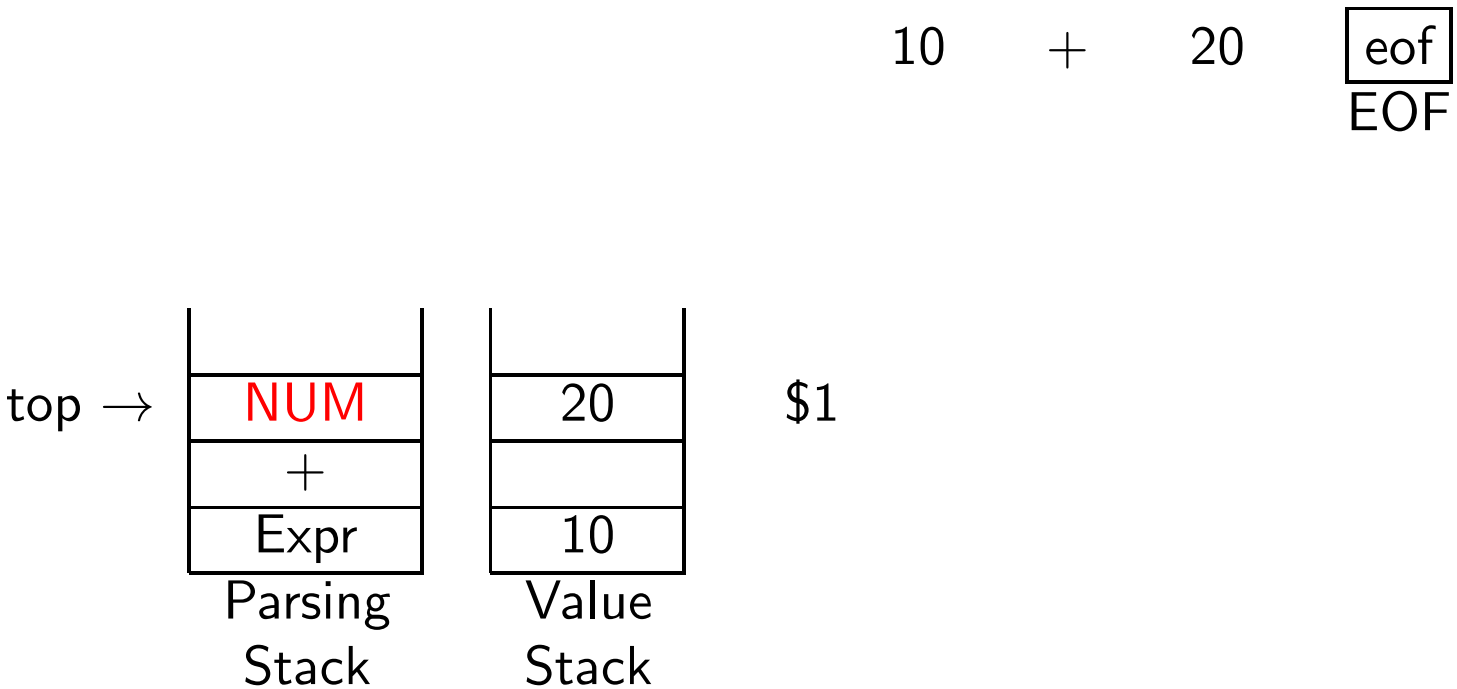


An Overview of Attribute Evaluation

- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation

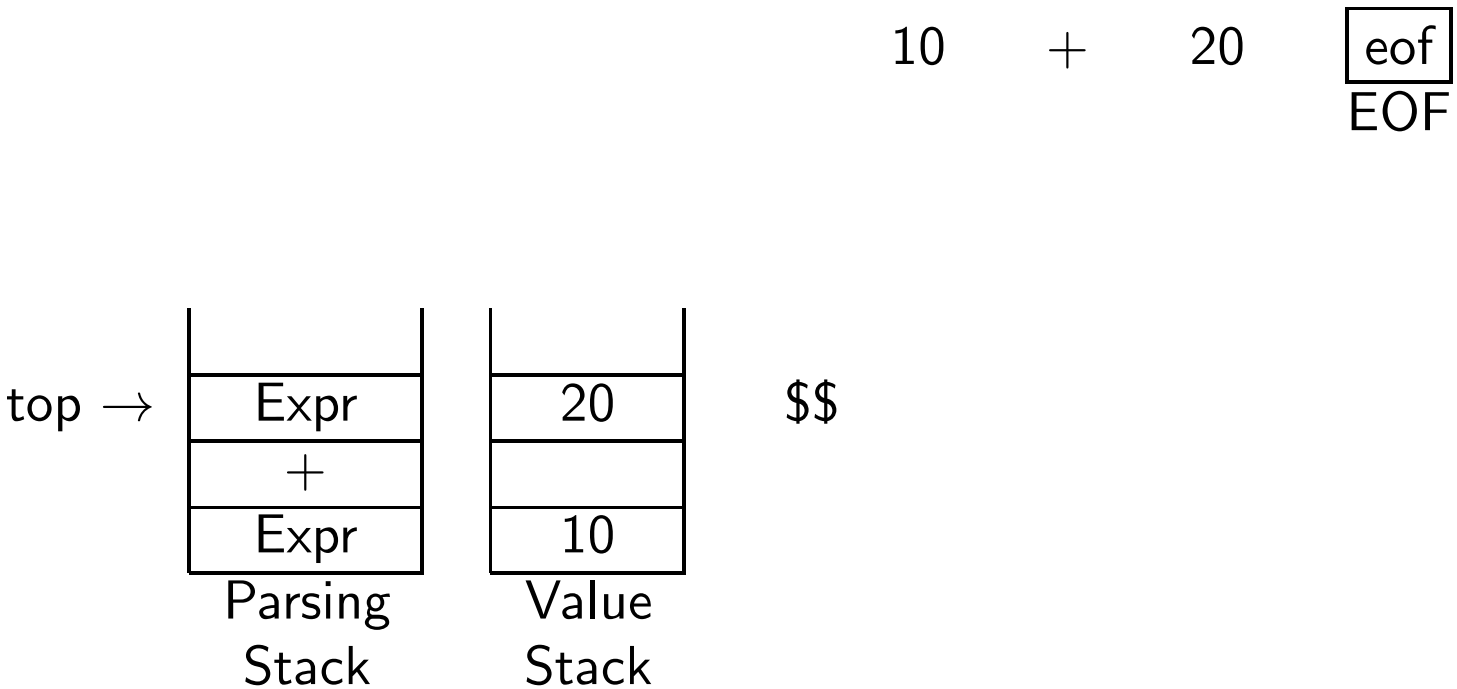


An Overview of Attribute Evaluation

- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation

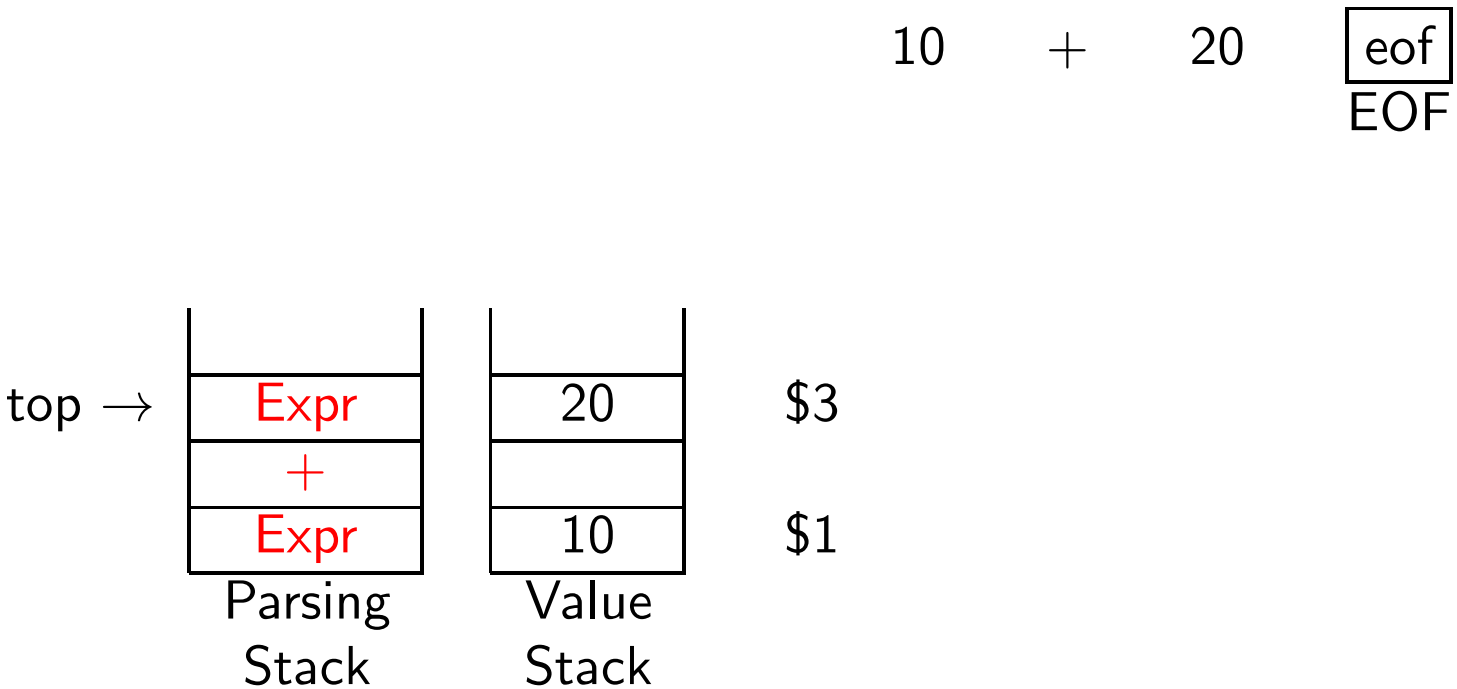


An Overview of Attribute Evaluation

- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation



An Overview of Attribute Evaluation

- Grammar and the associated actions

```
Expr   :   Expr '+' Expr     { $$ = $1 + $3;}
       |   NUM
```

- The process of parsing and attribute evaluation