

IN4305 Engineering project

Compiler construction



Koen Langendoen

Delft University of Technology
The Netherlands

Course organization



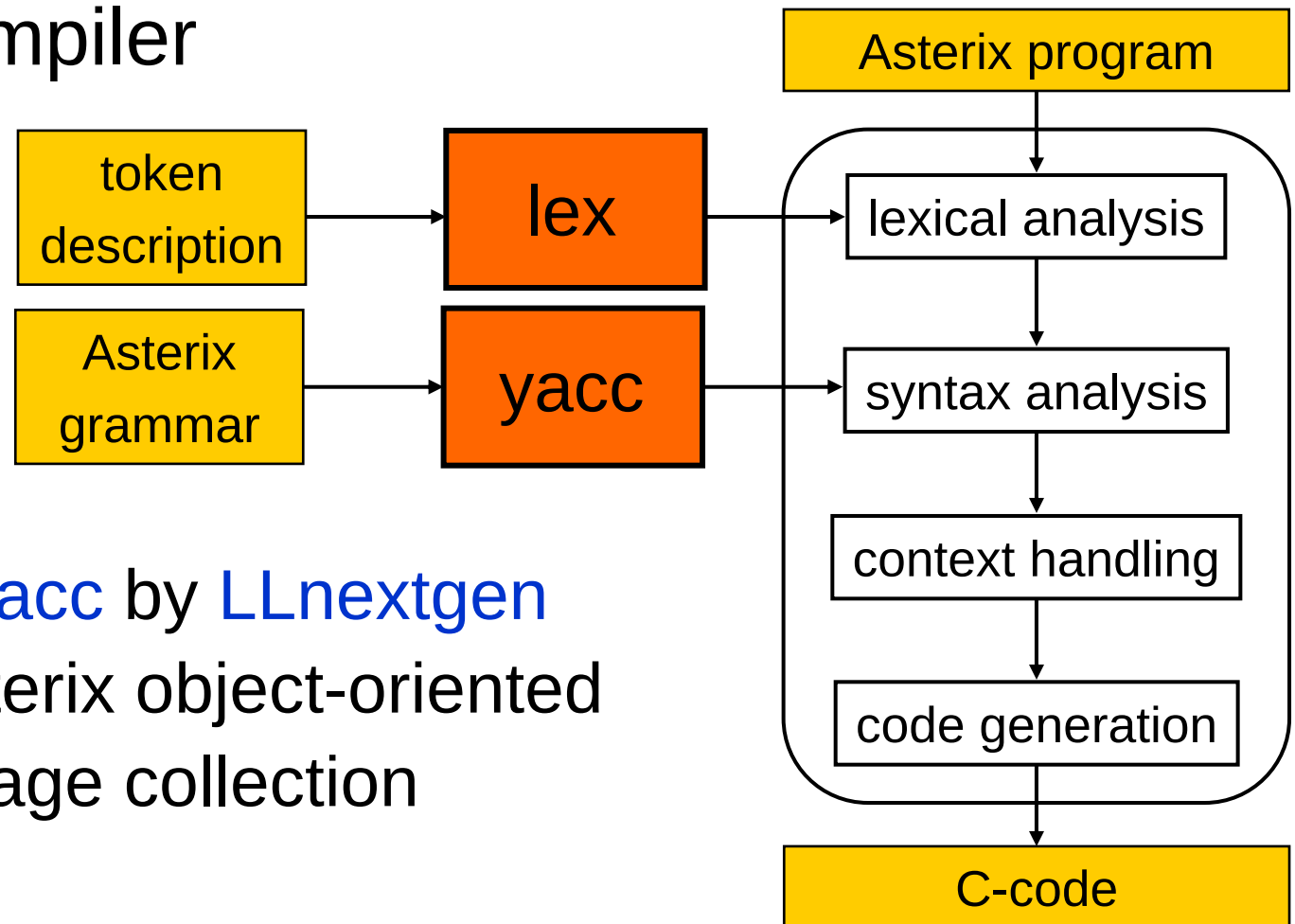
- kick/off
- lectures (2x)
- lab work (16x)

practice makes perfect

- NO exam, just programming!

Assignments

Asterix compiler



- 1) replace **yacc** by **LLnextgen**
- 2) make Asterix object-oriented
- 3) add garbage collection

Practicalities



- groups of 2 students
 - register with **Venkat Iyer** <V.G.Iyer@tudelft.nl>
 - CPM submission tool <cpm.ewi.tudelft.nl>
- lab rooms Drebbelweg
 - Thursday 08:45 – 12:30 DW_1-180
- instructors: [Venkat](#) + Alexander

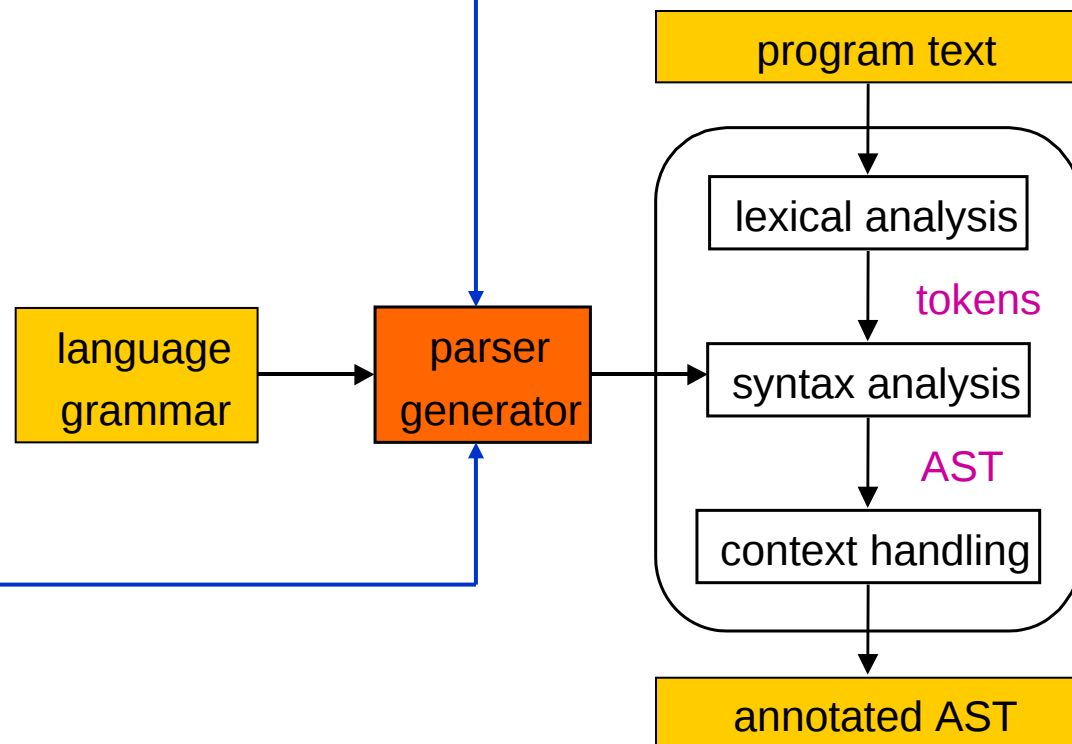
Assignment 1: Parser generators

Yacc

- bottom-up
- LALR(1)

LLnextgen

- top-down
- LL(1)



LLgen: LL(1) parser generator



- LLgen is part of the Amsterdam Compiler Kit
 - **LLnextgen** is improvement by Gertjan Halkes
- takes LL(1) grammar + semantic actions in C and generates a recursive descent parser
- LL**next**gen features:
 - repetition operators
 - advanced error handling
 - parameter passing / **return values**
 - control over semantic actions
 - dynamic conflict resolvers

LLnextgen example: expression interpreter

- start from LR(1) grammar
- make grammar LL(1)
 - left recursion
 - operator precedence
- use repetition operators
- add semantic actions
 - add types (and params) to grammar rules
 - insert C-code between the symbols

```
%token DIGIT;

main    : line+
        ;
line    : expr '\n'
        ;
expr    : term [ '+' term ]*
        ;
term    : factor [ '*' factor ]*
        ;
factor  : '(' expr ')'
        | DIGIT
        ;
```

LLnextgen

```

main      : line+
          ;
line
          : expr '\n'          { printf("%d\n", expr); }
          ;
expr<int>
          : term                { LLretval = term; }
          [ '+' term<right>      { LLretval += right; }
          ]*
          ;
term<int>
          : factor              { LLretval = factor; }
          [ '*' factor          { LLretval *= factor; }
          ]*
          ;
factor<int>
          : '(' expr ')'        { LLretval = expr; }
          | DIGIT               { LLretval = yylval; }
          ;

```


grammar


semantics

LLnextgen interface to lexical analyzer

- by default LLnextgen invokes `yylex()` to get the next token
- the “value” of a token can be stored in **any** global variable (`yylval`) of **any** type (`int`)

```
yylex()
{
    int c;

    c = getchar();

    if (isdigit(c)) {
        yyval = c - '0';
        return DIGIT;
    }
    return c;
}
```

LLnextgen interface to back-end

- LLnextgen generates a user-named function (**parse**)
- LLnextgen handles syntax errors by inserting missing tokens and deleting unexpected tokens
- LLmessage() is invoked to notify the lexical analyzer

```
%start parse, main;

void LLmessage(int class)
{
    switch (class) {
        case -1:
            printf("expecting EOF, ");
        case 0:
            printf("deleting token (%d)\n",LLsymb);
            break;
        default:
            /* push back token LLsymb */
            printf("inserting token (%d)\n",class);
            break;
    }
}
```

Exercise (5 min.)

Extend the
desk calculator
with registers
named a – z.

Example input:
u = v = 3*(w+4)

```
main      : line+
           ;

line       : expr '\n'           { printf("%d\n", expr); }
           ;

expr<int>  : term                 { LLretval = term; }
           [ '+' term<right>      { LLretval += right; }
           ]*
           ;

term<int>  : factor               { LLretval = factor; }
           [ '*' factor          { LLretval *= factor; }
           ]*
           ;

factor<int>: '(' expr ')'         { LLretval = expr; }
           | DIGIT               { LLretval = yylval; }
           ;
```

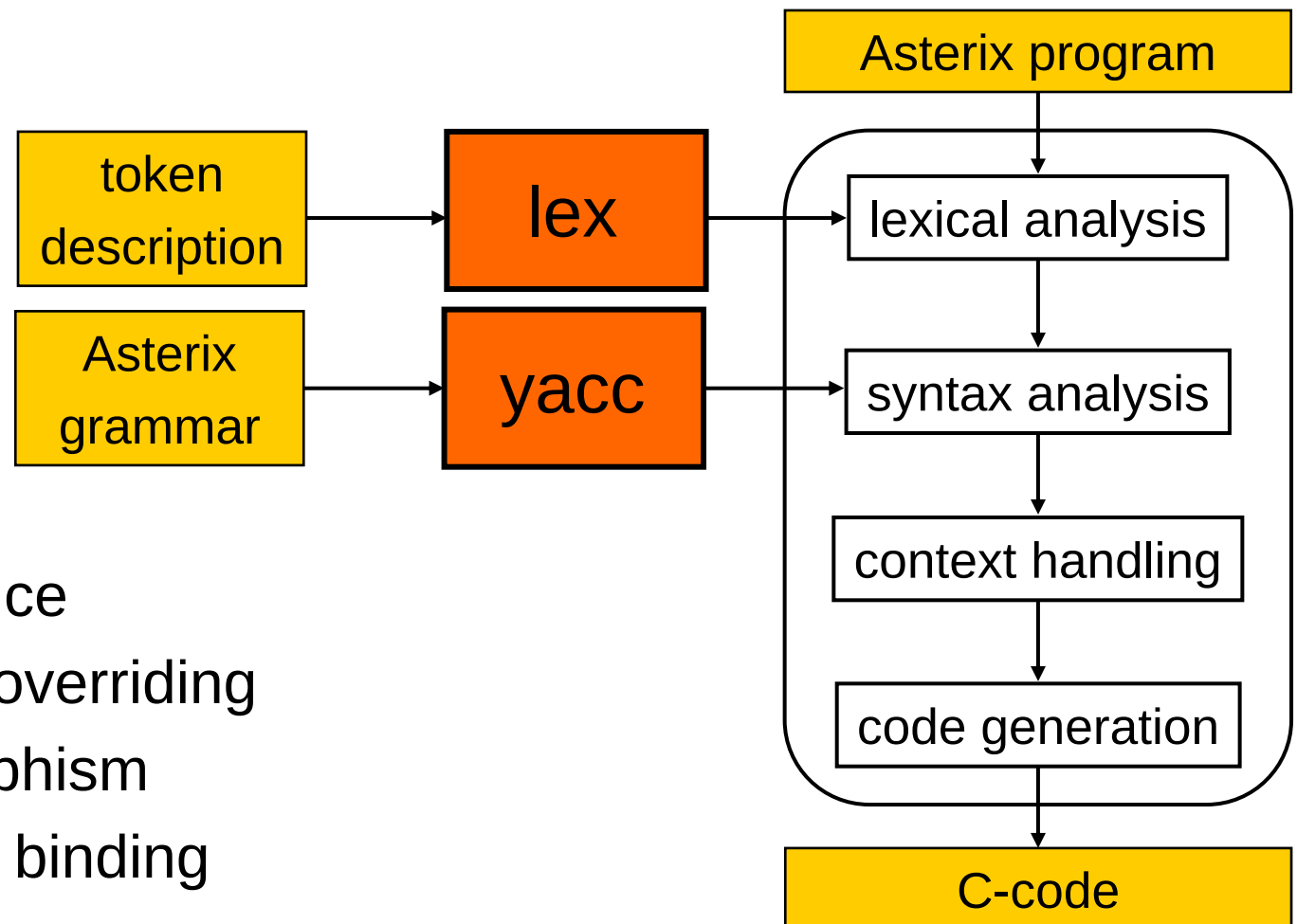
Answers



Assignment 2: Object orientation

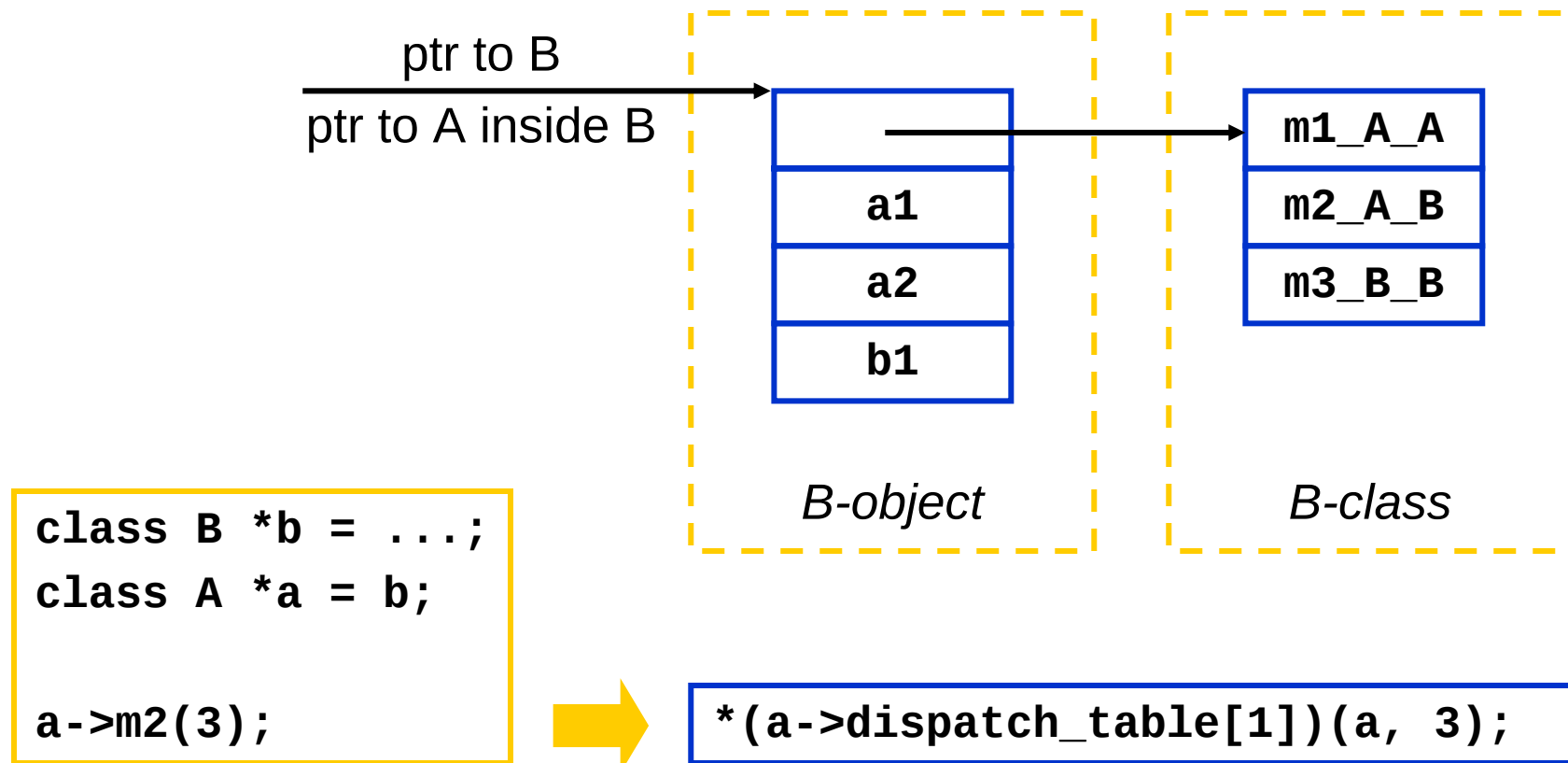
- features

- inheritance
- method overriding
- polymorphism
- dynamic binding

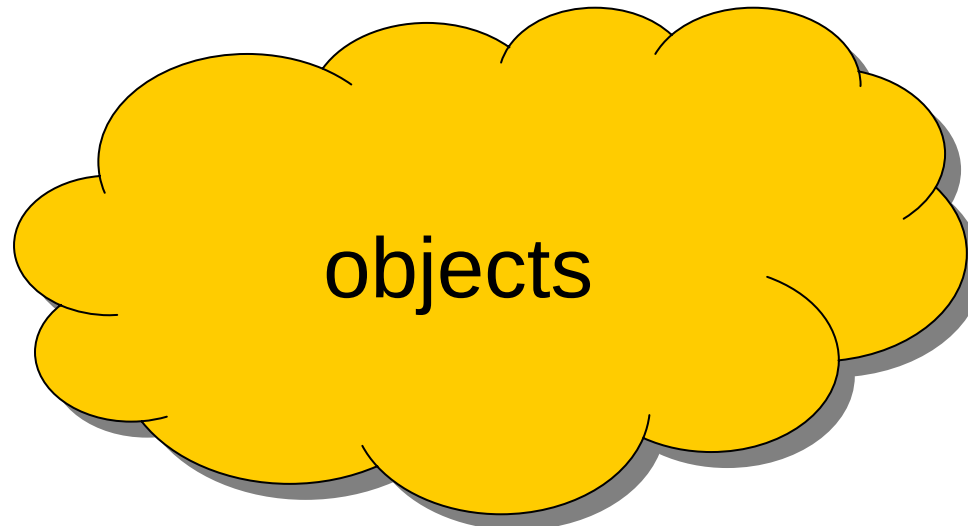


Dynamic typing

- implementation: dispatch tables



Online assistance: Panoramix



Candy?

Name or group number:

Code to compile:

Browse...

Go!

Online assistance: Diagnostix



erroneous compilers				
1	2	3	4	5
X			X	

Candy?

Name or group number:

Assignment:

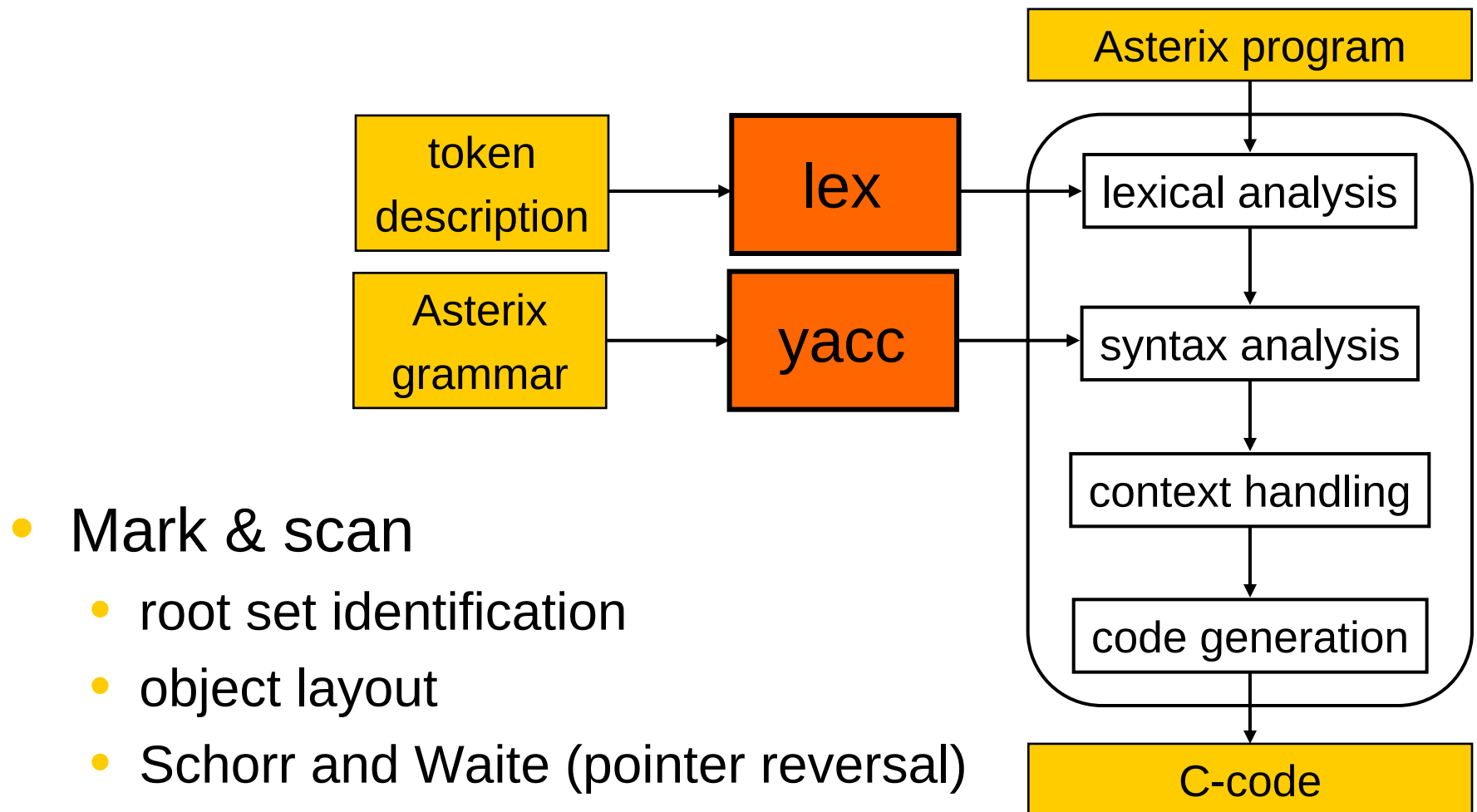
Tar file with test set:

[Browse...](#)

[Go!](#)

Assignment 3:

Garbage collection



Grading



Assignment	Points	Deadline
1	30	08:59 March 22
2	40	08:59 May 17
3	30	08:59 June 6

Home work



- Register for the lab work
 - teams of 2 students
 - send email to Venkat Iyer <V.Iyer@tudelft.nl>
- Lab starts
 - Next week (Thursday Feb 11)
- Next lectures
 - Thursday Feb 11, 08:45 – 10:30 EWI-D
 - Thursday **Mar 4**, 08:45 – 10:30 EWI-D