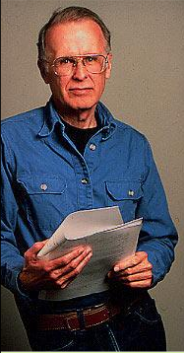


## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics




**Jerzy Nawrocki**  
Faculty of Computing and Inf. Sci.  
Poznan University of Technology  
jerzy.nawrocki@put.poznan.pl

# Compilers

John Backus

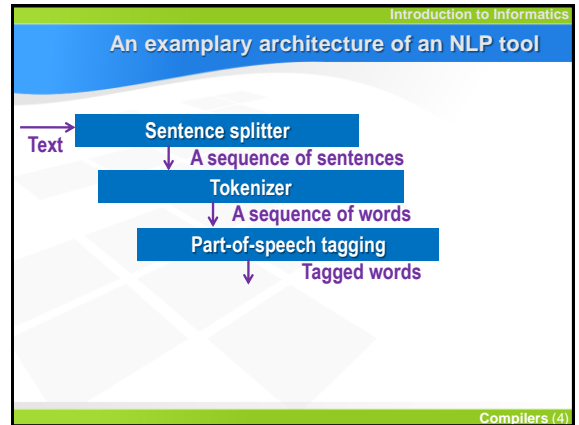
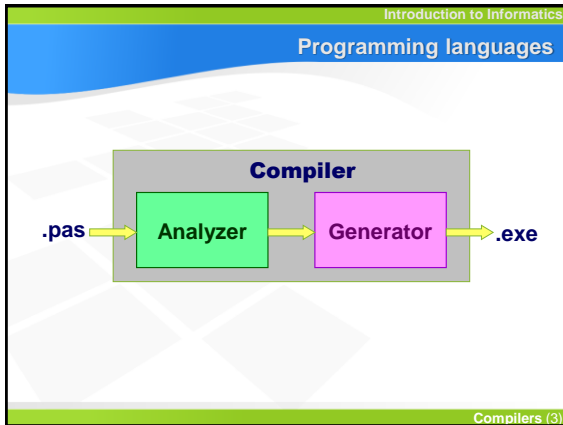
Introduction to Informatics

## Aim



**To present basics of compiler construction and syntax-directed translation.**


Compilers (2)



Introduction to Informatics

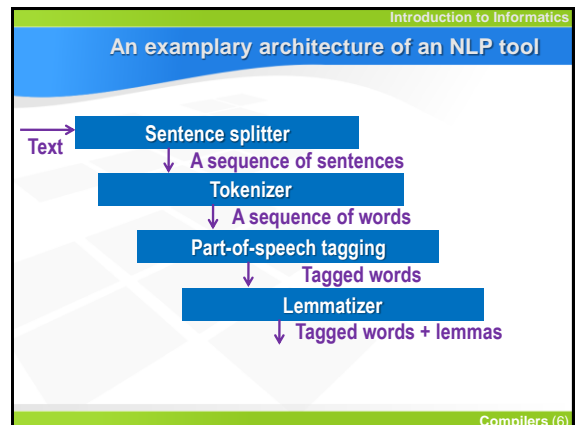
## Part-of-speech tagging

The grand jury commented on a number of other topics.

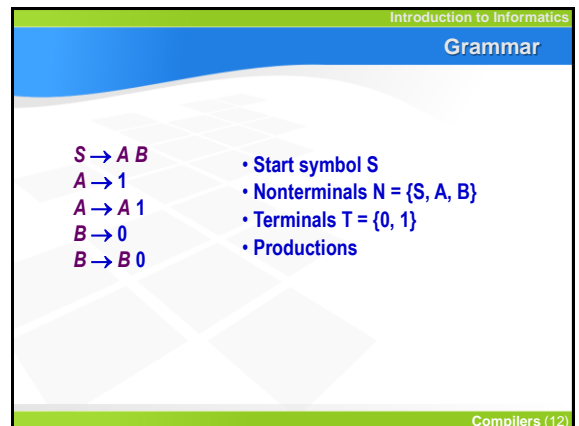
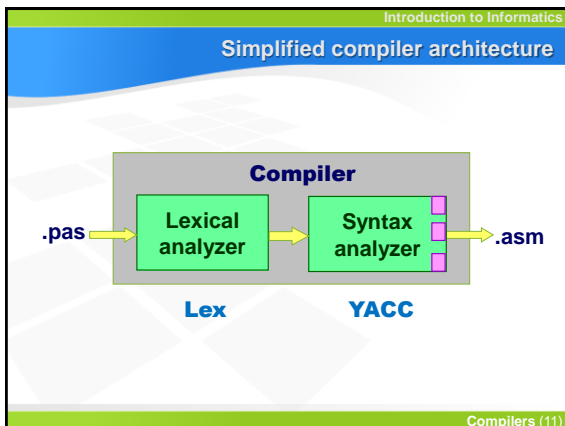
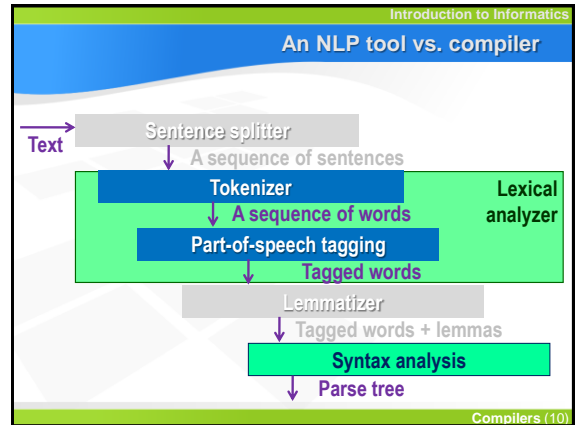
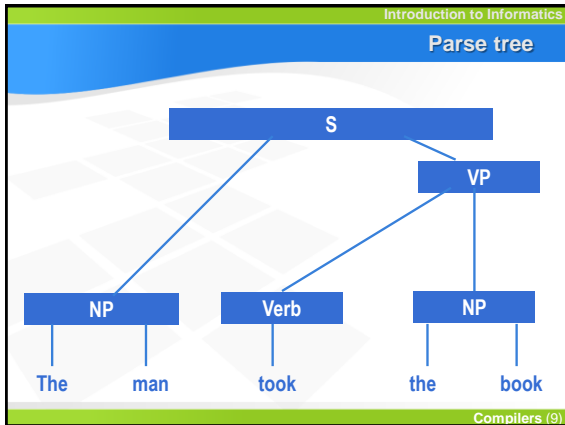
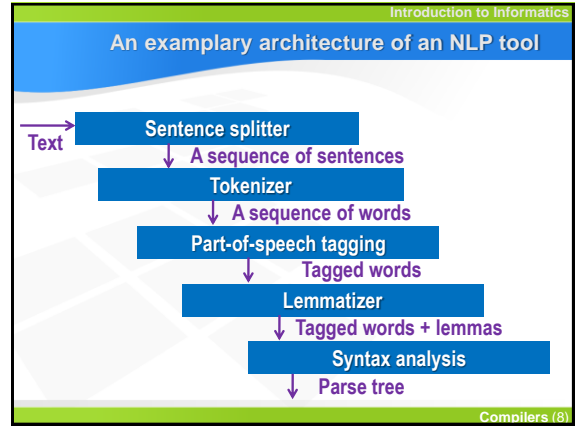
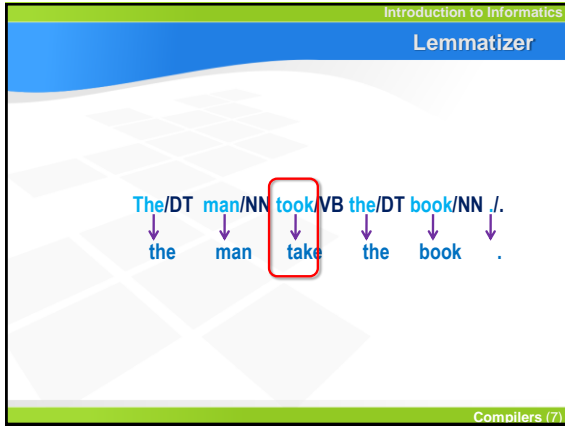


The/*DT* grand/*JJ* jury/*NN* commented/*VBD* on/*IN*  
a/*DT* number/*NN* of/*IN* other/*JJ* topics/*NNS* ./

Compilers (5)



(c) Jerzy Nawrocki, Introduction to Informatics



(c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

Context-free grammar

1.  $E \rightarrow C$
2.  $E \rightarrow E + C$
3.  $C \rightarrow F$
4.  $C \rightarrow C * F$
5.  $F \rightarrow N$
6.  $F \rightarrow (E)$
7.  $N \rightarrow 1$
8.  $N \rightarrow 2$
9.  $N \rightarrow 3$

One nonterminal

Compilers (13)

Introduction to Informatics

Grammar

```

S → A B
A → 1
A → A 1
B → 0
B → B 0
    
```

YACC	
S:	A B
	;
A:	'1'
	A '1'
	;
B:	'0'
	B '0'
	;

Compilers (14)

Introduction to Informatics

Grammar

```

S → A B
A → 1
A → A 1
B → 0
B → B 0
    
```

YACC	
S:	A B
	;
A:	'1'
	A '1'
	;
B:	'0'
	B '0'
	;

Compilers (15)

Introduction to Informatics

Grammar

```

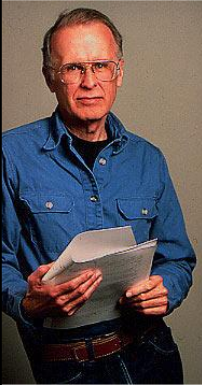
S → A B
A → 1
A → A 1
B → 0
B → B 0
    
```

YACC	
S:	A B
	;
A:	'1'
	A '1'
	;
B:	'0'
	B '0'
	;

Compilers (16)

Introduction to Informatics

Agenda



- The Pas language
- Exemplary lexical analysis
- Exemplary syntax analysis
- Pas compiler

Compilers (17)

Introduction to Informatics

Pas

```

Prog: Dekla
    "begin" Instr
        "end" '.'
    ;
    
```

**Not the best solution!**

Compilers (18)

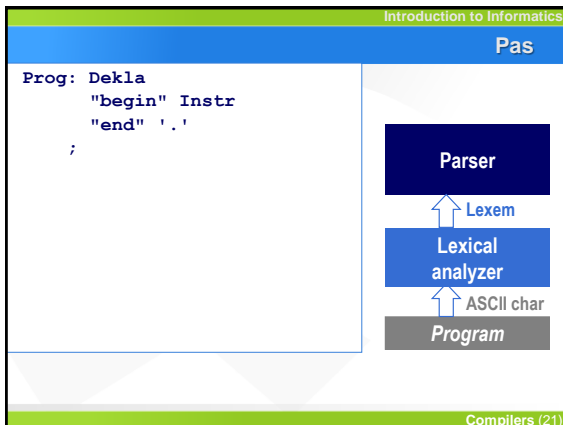
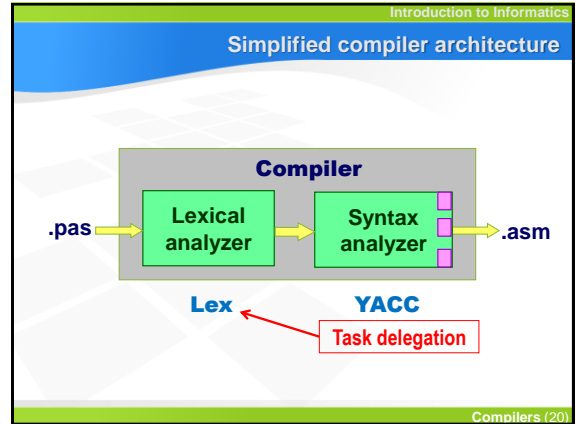
## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Context-free grammar

1.  $E \rightarrow C$
2.  $E \rightarrow E + C$
3.  $C \rightarrow F$
4.  $C \rightarrow C * F$
5.  $F \rightarrow N$
6.  $F \rightarrow ( E )$
7.  $N \rightarrow 1$
8.  $N \rightarrow 2$
9.  $N \rightarrow 3$

Compilers (19)



Introduction to Informatics

### Interface between Lex and YACC

```

%token Begin End
%%
Prog: Dekla
  Begin Instr
  End '.'
;
    
```

Lexical analyzer (Lex)

```

%%
"begin" {return Begin;}
"end"   {return End;}
"."     {return '.';}
    
```

Compilers (22)

Introduction to Informatics

### Pas

```

%token Begin End Var
%%
Prog: Dekla
  Begin Instr
  End '.'
;
Dekla: Var Nazwy ':' Typ ';'
;
    
```

Nazwy = Names  
Nazwa = Name

```

%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"."     {return '.';}
":"     {return ':';}
";"    {return ';'}
    
```

Compilers (23)

Introduction to Informatics

### Pas

```

%token Begin End Var Nazwa
%%
Prog: Dekla
  Begin Instr
  End '.'
;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa Nazwy
;
    
```

Nazwy = Names  
Nazwa = Name

```

%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
[a-zA-Z]+ {return Nazwa;}
"."     {return '.';}
":"     {return ':';}
";"    {return ';'}
    
```

Compilers (24)

## (c) Jerzy Nawrocki, Introduction to Informatics

```

Introduction to Informatics
Pas
%token Begin End Var Nazwa Int Rea
%%
Prog: Dekla
    Begin Instr
    End '.'
;
Dekla: Var Nazwy ':' Typ
;
Nazwy: Nazwa
    | Nazwa Nazwy
;
Typ: Int
    | Rea
;
%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"."      {return '.';}
":"      {return ':';}
";"     {return ';'}

```

Compilers (25)

```

Introduction to Informatics
Pas
%token Begin End Var Nazwa Ir Dekla
%%
Prog: Dekla
    Begin Instr
    End '.'
;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
    | Nazwa Nazwy
;
Typ: Int
    | Rea
;
%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"."      {return '.';}
":"      {return ':';}
";"     {return ';'}

```

```

Introduction to Informatics
Pas
%token Begin End Var Nazwa Ir Dekla
%%
Prog: Dekla
    Begin Instr
    End '.'
;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
    | Nazwa ',' Nazwy
;
Typ: Int
    | Rea
;
%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"."      {return '.';}
":"      {return ':';}
";"     {return ';'}
","      {return ',';}

```

```

Introduction to Informatics
Pas
%token Begin End Var Nazwa Int Rea As
%%
Prog: Dekla
    Begin Instr
    End '.'
;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
    | Nazwa ',' Nazwy
;
Typ: Int
    | Rea
;
Instr: Podst
;
Podst: Nazwa As Wyr
;
%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"."      {return '.';}
":"      {return ':';}
";"     {return ';'}
"="      {return '=';}

```

**Podstawienie = Assignment**  
**Wyrażenie = Expression**

```

Introduction to Informatics
Pas
%token Begin End Var Nazwa Int Rea As Calk
%%
Prog: Dekla
    Begin Instr
    End '.'
;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
    | Nazwa ',' Nazwy
;
Typ: Int
    | Rea
;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
    | Nazwa
    | Calk '+' Wyr
    | Nazwa '+' Wyr ;
;
%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"."      {return '.';}
":"      {return ':';}
";"     {return ';'}
"="      {return '=';}
"+"     {return '+';}
[0-9]+  {return Calk;}

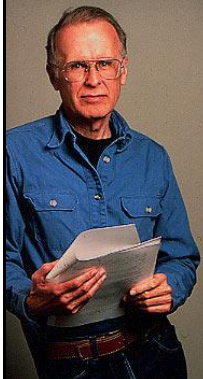
```

**Podstawienie = Assignment**  
**Wyrażenie = Expression**  
**Calkowita = Integer**

Introduction to Informatics

### Agenda

- The Pas language
- Exemplary lexical analysis
- Exemplary syntax analysis
- Pas compiler



Compilers (30)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

```
var x, y: integer;
begin
y:= x + 10
end.
```

Lexical analysis

Var

```
%%
"begin" {return Begin;}
"end" {return End;}
"var" {return Var;}
"integer" {return Int;}
"real" {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"." {return '.';}
":" {return ':';}
";" {return ';';}
"," {return ',';}
":=" {return As;}
"+" {return '+';}
[0-9]+ {return Calk;}

```

Compilers (31)

Introduction to Informatics

```
var x y: integer;
begin
y:= x + 10
end.
```

Lexical analysis

Var  
Nazwa

```
%%
"begin" {return Begin;}
"end" {return End;}
"var" {return Var;}
"integer" {return Int;}
"real" {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"." {return '.';}
":" {return ':';}
";" {return ';';}
"," {return ',';}
":=" {return As;}
"+" {return '+';}
[0-9]+ {return Calk;}

```

Compilers (32)

Introduction to Informatics

```
var x, y: integer;
begin
y:= x + 10
end.
```

Lexical analysis

Var  
Nazwa  
,

```
%%
"begin" {return Begin;}
"end" {return End;}
"var" {return Var;}
"integer" {return Int;}
"real" {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"." {return '.';}
":" {return ':';}
";" {return ';';}
"," {return ',';}
":=" {return As;}
"+" {return '+';}
[0-9]+ {return Calk;}

```

Compilers (33)

Introduction to Informatics

```
var x, y integer;
begin
y:= x + 10
end.
```

Lexical analysis

Var  
Nazwa  
,  
Nazwa

```
%%
"begin" {return Begin;}
"end" {return End;}
"var" {return Var;}
"integer" {return Int;}
"real" {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"." {return '.';}
":" {return ':';}
";" {return ';';}
"," {return ',';}
":=" {return As;}
"+" {return '+';}
[0-9]+ {return Calk;}

```

Compilers (34)

Introduction to Informatics

```
var x, y integer;
begin
y:= x + 10
end.
```

Lexical analysis

Var  
Nazwa  
,  
Nazwa  
:

```
%%
"begin" {return Begin;}
"end" {return End;}
"var" {return Var;}
"integer" {return Int;}
"real" {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"." {return '.';}
":" {return ':';}
";" {return ';';}
"," {return ',';}
":=" {return As;}
"+" {return '+';}
[0-9]+ {return Calk;}

```

Compilers (35)

Introduction to Informatics

```
var x, y: integer;
begin
y:= x + 10
end.
```

Lexical analysis

Var  
Nazwa  
,  
Nazwa  
:  
Int  
;  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End  
.

```
%%
"begin" {return Begin;}
"end" {return End;}
"var" {return Var;}
"integer" {return Int;}
"real" {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"." {return '.';}
":" {return ':';}
";" {return ';';}
"," {return ',';}
":=" {return As;}
"+" {return '+';}
[0-9]+ {return Calk;}

```

Compilers (36)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Agenda

- The Pas language
- Exemplary lexical analysis
- Exemplary syntax analysis
- Pas compiler

Compilers (37)

Introduction to Informatics

### Syntax analysis – parser actions

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

- Reduce the stack
- Shift on the stack

Stack

```

Var
Nazwa
:
Nazwa
:
Int
:
Begin
Nazwa
As
Nazwa
+
Calk
End
.
    
```

Compilers (38)

Introduction to Informatics

### Shift on the stack

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

```

Nazwa
:
Int
:
Begin
Nazwa
As
Nazwa
+
Calk
End
.
    
```

Compilers (39)

Introduction to Informatics

### Shift on the stack

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

```

Nazwa
:
Int
:
Begin
Nazwa
As
Nazwa
+
Calk
End
.
    
```

Compilers (40)

Introduction to Informatics

### Reduce the stack

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

```

Nazwy
:
Nazwa
:
Var
    
```

Compilers (41)

Introduction to Informatics

### Reduce the stack

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

```

Nazwy
:
Var
    
```

Compilers (42)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

Syntax analysis		Var
Prog: Dekla Begin Instr End '.' ;	Reduce   Shift ?	Nazwa ; Nazwa ; Int ; Begin Nazwa As Nazwa + Calk End
Dekla: Var Nazwy ':' Typ ';' ;	<b>Stack</b>	
Nazwy: Nazwa   Nazwa ',' Nazwy ;		
Typ: Int   Rea ;		
Instr: Podst ;		
Podst: Nazwa As Wyr ;		
Wyr: Calk   Nazwa   Calk '+' Wyr   Nazwa '+' Wyr ;		

Compilers (43)

Introduction to Informatics

Syntax analysis		Nazwa
Prog: Dekla Begin Instr End '.' ;	Reduce   Shift	Nazwa ; Nazwa ; Int ; Begin Nazwa As Nazwa + Calk End
Dekla: Var Nazwy ':' Typ ';' ;	<b>Stack</b>	
Nazwy: Nazwa   Nazwa ',' Nazwy ;		
Typ: Int   Rea ;		
Instr: Podst ;		
Podst: Nazwa As Wyr ;		
Wyr: Calk   Nazwa   Calk '+' Wyr   Nazwa '+' Wyr ;		Var

Compilers (44)

Introduction to Informatics

Syntax analysis		Nazwa
Prog: Dekla Begin Instr End '.' ;	Redukcja   przesunięcie ?	Nazwa ; Nazwa ; Int ; Begin Nazwa As Nazwa + Calk End
Dekla: Var Nazwy ':' Typ ';' ;	<b>Stack</b>	
Nazwy: Nazwa   Nazwa ',' Nazwy ;		
Typ: Int   Rea ;		
Instr: Podst ;		
Podst: Nazwa As Wyr ;		
Wyr: Calk   Nazwa   Calk '+' Wyr   Nazwa '+' Wyr ;		Var

Compilers (45)

Introduction to Informatics

Syntax analysis		Nazwa
Prog: Dekla Begin Instr End '.' ;	Reduce   Shift	Nazwa ; Nazwa ; Int ; Begin Nazwa As Nazwa + Calk End
Dekla: Var Nazwy ':' Typ ';' ;	<b>Stack</b>	
Nazwy: Nazwa   Nazwa ',' Nazwy ;		
Typ: Int   Rea ;		
Instr: Podst ;		
Podst: Nazwa As Wyr ;		
Wyr: Calk   Nazwa   Calk '+' Wyr   Nazwa '+' Wyr ;		Nazwa Var

Compilers (46)

Introduction to Informatics

Syntax analysis		Nazwa
Prog: Dekla Begin Instr End '.' ;	Reduce   Shift?	Nazwa ; Int ; Begin Nazwa As Nazwa + Calk End
Dekla: Var Nazwy ':' Typ ';' ;	<b>Stack</b>	
Nazwy: Nazwa   Nazwa ',' Nazwy ;		
Typ: Int   Rea ;		
Instr: Podst ;		
Podst: Nazwa As Wyr ;		
Wyr: Calk   Nazwa   Calk '+' Wyr   Nazwa '+' Wyr ;		Nazwa Var

Compilers (47)

Introduction to Informatics

Syntax analysis		Nazwa
Prog: Dekla Begin Instr End '.' ;	Reduce   Shift?	Nazwa ; Int ; Begin Nazwa As Nazwa + Calk End
Dekla: Var Nazwy ':' Typ ';' ;	<b>Stack</b>	
Nazwy: Nazwa   Nazwa ',' Nazwy ;		
Typ: Int   Rea ;		
Instr: Podst ;		
Podst: Nazwa As Wyr ;		
Wyr: Calk   Nazwa   Calk '+' Wyr   Nazwa '+' Wyr ;		Nazwa Var

If we reduced to Wyr?

Compilers (48)



(c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Wyr  
Var

Nazwa  
Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

**Does not fit!**

Compilers (49)

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
Var

Nazwa  
Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

**If we reduced to Nazwy?**

Compilers (50)

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwy  
Var

Nazwa  
Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

**Does not fit!**

Compilers (51)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
Var

Nazwa  
Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (52)

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
Var

Nazwa  
Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (53)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
Var

Nazwa  
Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (54)

(c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
Nazwa  
Var

Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (55)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

R1

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
Nazwa  
Var

Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (56)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwy  
Nazwa  
Var

Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (57)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

R2

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwy  
Nazwa  
Var

Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (58)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwy  
Var

Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (59)

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
;
Typ: Int
     | Rea ;
Instr: Podst
;
Podst: Nazwa As Wyr
;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Nazwy  
Var

Int  
Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Compilers (60)

(c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Int  
; Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Nazwy  
Var

Compilers (61)

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Int  
; Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Nazwy  
Var

Compilers (62)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Int  
; Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Nazwy  
Var

Compilers (63)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Int  
; Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Nazwy  
Var

Compilers (64)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Int  
; Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Nazwy  
Var

Compilers (65)

Introduction to Informatics

### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Int  
; Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Nazwy  
Var

Compilers (66)

### (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Typ  
Nazwy  
Var

Compilers (67)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Typ  
Nazwy  
Var

Compilers (68)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Begin  
Nazwa  
As  
Nazwa  
+  
Calk  
End

Dekla

Compilers (69)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
As  
Nazwa  
+  
Calk  
End

Begin  
Dekla

Compilers (70)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

As  
Nazwa  
+  
Calk  
End

Nazwa  
Begin  
Dekla

Compilers (71)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

As  
Nazwa  
+  
Calk  
End

Nazwa  
Begin  
Dekla

Compilers (72)

### (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

As  
Nazwa  
Begin  
Dekla

Nazwa  
+  
Calk  
End

Compilers (73)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Nazwa  
As  
Nazwa  
Begin  
Dekla

+  
Calk  
End

Compilers (74)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift ?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Nazwa ... +

As  
Nazwa  
Begin  
Dekla

+  
Calk  
End

Compilers (75)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

+  
Nazwa  
As  
Nazwa  
Begin  
Dekla

Calk  
End

Compilers (76)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Calk  
+  
Nazwa  
As  
Nazwa  
Begin  
Dekla

End

Compilers (77)

Introduction to Informatics

#### Syntax analysis

Reduce | Shift

R5

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Calk  
+  
Nazwa  
As  
Nazwa  
Begin  
Dekla

End

Compilers (78)

(c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
    | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Wyr  
+  
Nazwa  
As  
Nazwa  
Begin  
Dekla

End

Compilers (79)

Introduction to Informatics

Syntax analysis

Reduce | Shift

R6

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
    | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Wyr  
+  
Nazwa  
As  
Nazwa  
Begin  
Dekla

End

Compilers (80)

Introduction to Informatics

Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
    | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Wyr  
As  
Nazwa  
Begin  
Dekla

End

Compilers (81)

Introduction to Informatics

Syntax analysis

Reduce | Shift

R7

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
    | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Wyr  
As  
Nazwa  
Begin  
Dekla

End

Compilers (82)

Introduction to Informatics

Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
    | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Podst  
Begin  
Dekla

End

Compilers (83)

Introduction to Informatics

Syntax analysis

Reduce | Shift

R8

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
    | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Podst  
Begin  
Dekla

End

Compilers (84)

(c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
     | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Instr  
Begin  
Dekla

End

Compilers (85)

Introduction to Informatics

### Syntax analysis

Reduce | Shift ?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
     | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

Instr  
Begin  
Dekla

End

Compilers (86)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
     | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

End  
Instr  
Begin  
Dekla

Compilers (87)

Introduction to Informatics

### Syntax analysis

Reduce | Shift ?

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
     | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

End  
Instr  
Begin  
Dekla

Compilers (88)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
     | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

End  
Instr  
Begin  
Dekla

Compilers (89)

Introduction to Informatics

### Syntax analysis

Reduce | Shift

R9

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ: Int
     | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr: Calk
     | Nazwa
     | Calk '+' Wyr
     | Nazwa '+' Wyr ;
    
```

Stack

End  
Instr  
Begin  
Dekla

Compilers (90)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Syntax analysis

Reduce | Shift

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Prog

Compilers (91)

Introduction to Informatics

### Syntax analysis

Accept:

Input = empty

Stack = Start symbol

```

Prog: Dekla
      Begin Instr
      End '.' ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

Stack

Prog

Compilers (92)

Introduction to Informatics

### List of reductions

```

Nazwy: Nazwa
Nazwy: Nazwa ',' Nazwy
Typ: Int
Dekla: Var Nazwy ':' Typ ';'
Wyr: Calk
Wyr: Nazwa '+' Wyr
Podst: Nazwa As Wyr
Instr: Podst
Prog: Dekla
      Begin Instr End '.'
    
```

R1 Var Nazwa , Nazwa ← Stack

R2 Var Nazwa , Nazwy

R3 Var Nazwy : Int

R4 Var Nazwy : Typ ;

R5 Dekla Begin Nazwa As Nazwa + Calk

R6 Dekla Begin Nazwa As Nazwa + Wyr

R7 Dekla Begin Nazwa As Wyr

R8 Dekla Begin Podst

R9 Dekla Begin Instr End .

Compilers (93)

Introduction to Informatics

### Agenda

- The Pas language
- Exemplary lexical analysis
- Exemplary syntax analysis
- Pas compiler

Compilers (94)

```

%token Begin End Var Nazwa Int Rea As Calk
%%
Prog: Dekla
      Begin Instr
      End '.'
      ;
Dekla: Var Nazwy ':' Typ ';'
      ;
Nazwy: Nazwa
      | Nazwa ',' Nazwy
      ;
Typ:   Int
      | Rea
      ;
Instr: Podst
      ;
Podst: Nazwa As Wyr
      ;
Wyr:  Calk
      | Nazwa
      | Calk '+' Wyr
      | Nazwa '+' Wyr ;
    
```

```

%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {return Nazwa;}
"."     {return '.';}
":"     {return ':';}
"="     {return '=';}
"+"     {return '+';}
[0-9]+  {return Calk;}
    
```

Introduction to Informatics

### Lexem and their attributes

```

graph TD
    Program[program] -- ASCII car --> Lexical[Lexical analyzer]
    Lexical -- yytext[..] --> Parser[Parser]
    Lexical -- yyval --> Attributes[Lexem's attribute]
    
```

Compilers (96)



## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Attributes

```

%%
"begin" {return Begin;}
"end"   {return End;}
"var"   {return Var;}
"integer" {return Int;}
"real"  {return Rea;}
[a-zA-Z]+ {yyval= yytext[0]; return Nazwa;}
"."      {return '.';}
":"     {return ':';}
";"     {return ';'}
"|"     {return '|'}
"|"     {return '|'}
";="    {return '=';}
"|"     {return '+';}
[0-9]+  {yyval= atoi(yytext); return Calk;}
    
```

Compilers (97)

Introduction to Informatics

### Reductions

Nazwy: Nazwa	<b>R1</b> Var Nazwa , Nazwa	← Stack
Nazwy: Nazwa ',' Nazwy	<b>R2</b> Var Nazwa , Nazwy	
Typ: Int	<b>R3</b> Var Nazwy : Int	
Dekla: Var Nazwy ':' Typ ';' ;	<b>R4</b> Var Nazwy : Typ ;	
Wyr: Calk	<b>R5</b> Dekla Begin Nazwa As Nazwa + Calk	
Wyr: Nazwa '+' Wyr	<b>R6</b> Dekla Begin Nazwa As Nazwa + Wyr	
Podst: Nazwa As Wyr	<b>R7</b> Dekla Begin Nazwa As Wyr	
Instr: Podst	<b>R8</b> Dekla Begin Podst	
Prog: Dekla Begin Instr End '.'	<b>R9</b> Dekla Begin Instr End .	

Compilers (98)

Introduction to Informatics

### Code generation

```

Dekla: Var Nazwy ':' Typ ';' {if $4=1 printf(" DW ?\n");}
;
Nazwy: Nazwa R1{printf("%c", $1);}
      | Nazwa ',' Nazwy {printf(", %c", $1);}
;
Typ: Int {$$= 1;}
     | Rea {$$= 2;}
;
    
```

**Before**

Stack	Int Begin Nazwa (y) As Nazwa (x) + Calk (10) End Var
-------	--

Compilers (99)

Introduction to Informatics

### Code generation

```

Dekla: Var Nazwy ':' Typ ';' {if $4=1 printf(" DW ?\n");}
;
Nazwy: Nazwa R1{printf("%c", $1);}
      | Nazwa ',' Nazwy {printf(", %c", $1);}
;
Typ: Int {$$= 1;}
     | Rea {$$= 2;}
;
    
```

**After**

Stack	Int Begin Nazwa (y) As Nazwa (x) + Calk (10) End Var
-------	--

Compilers (100)

Introduction to Informatics

### Code generation

```

Dekla: Var Nazwy ':' Typ ';' {if $4=1 printf(" DW ?\n");}
;
Nazwy: Nazwa {printf("%c", $1);}
      | Nazwa ',' Nazwy R2{printf(", %c", $1);}
;
Typ: Int {$$= 1;}
     | Rea {$$= 2;}
;
    
```

**Before**

Stack	Int Begin Nazwa (y) As Nazwa (x) + Calk (10) End Var
-------	--

Compilers (101)

Introduction to Informatics

### Code generation

```

Dekla: Var Nazwy ':' Typ ';' {if $4=1 printf(" DW ?\n");}
;
Nazwy: Nazwa {printf("%c", $1);}
      | Nazwa ',' Nazwy R2{printf(", %c", $1);}
;
Typ: Int {$$= 1;}
     | Rea {$$= 2;}
;
    
```

**After**

Stack	Int Begin Nazwa (y) As Nazwa (x) + Calk (10) End Var
-------	--

Compilers (102)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics		
List of reductions		
Nazwy: Nazwa	<b>R1</b>	Var Nazwa , Nazwa ← Stack
Nazwy: Nazwa ',' Nazwy	<b>R2</b>	Var Nazwa , Nazwy
Typ: Int	<b>R3</b>	Var Nazwy : Int
Dekla: Var Nazwy ':' Typ ':'	<b>R4</b>	Var Nazwy : Typ ;
Wyr: Calk	<b>R5</b>	Dekla Begin Nazwa As Nazwa + Calk
Wyr: Nazwa '+' Wyr	<b>R6</b>	Dekla Begin Nazwa As Nazwa + Wyr
Podst: Nazwa As Wyr	<b>R7</b>	Dekla Begin Nazwa As Wyr
Instr: Podst	<b>R8</b>	Dekla Begin Podst
Prog: Dekla Begin Instr End '.'	<b>R9</b>	Dekla Begin Instr End .

Compilers (103)

Introduction to Informatics		
Code generation		
Dekla: Var Nazwy ':' Typ ':'	<b>R4</b>	if \$4=1 printf(" DW ?\n");
;		
Nazwy: Nazwa		{printf("%c", \$1);}
Nazwa ',' Nazwy		{printf(", %c", \$1);}
;		
Typ: Int	<b>R3</b>	(\$\$= 1;)
Rea		(\$\$= 2;)
;		

Before

y, x

Stack

; Begin  
Nazwa (y)  
As  
Nazwa (x)  
+  
Calk (10)  
End  
; Nazwy  
Var  
.

Compilers (104)

Introduction to Informatics		
Code generation		
Dekla: Var Nazwy ':' Typ ':'	<b>R4</b>	if \$4=1 printf(" DW ?\n");
;		
Nazwy: Nazwa		{printf("%c", \$1);}
Nazwa ',' Nazwy		{printf(", %c", \$1);}
;		
Typ: Int	<b>R3</b>	(\$\$= 1;)
Rea		(\$\$= 2;)
;		

After

y, x

Stack

; Begin  
Nazwa (y)  
As  
Nazwa (x)  
+  
Calk (10)  
End  
; Nazwy  
Var  
.

Compilers (105)

Introduction to Informatics		
Code generation		
Dekla: Var Nazwy ':' Typ ':'	<b>R4</b>	if \$4=1 printf(" DW ?\n");
;		
Nazwy: Nazwa		{printf("%c", \$1);}
Nazwa ',' Nazwy		{printf(", %c", \$1);}
;		
Typ: Int		(\$\$= 1;)
Rea		(\$\$= 2;)
;		

Before

y, x

Stack

; Begin  
Nazwa (y)  
As  
Nazwa (x)  
+  
Calk (10)  
End  
; Typ (1)  
; Nazwy  
Var  
.

Compilers (106)

Introduction to Informatics		
Code generation		
Dekla: Var Nazwy ':' Typ ':'	<b>R4</b>	if \$4=1 printf(" DW ?\n");
;		
Nazwy: Nazwa		{printf("%c", \$1);}
Nazwa ',' Nazwy		{printf(", %c", \$1);}
;		
Typ: Int		(\$\$= 1;)
Rea		(\$\$= 2;)
;		

After

y, x DW ?

Stack

; Begin  
Nazwa (y)  
As  
Nazwa (x)  
+  
Calk (10)  
End  
; Dekla  
.

Compilers (107)

Introduction to Informatics		
List of reductions		
Nazwy: Nazwa	<b>R1</b>	Var Nazwa , Nazwa
Nazwy: Nazwa ',' Nazwy	<b>R2</b>	Var Nazwa , Nazwy
Typ: Int	<b>R3</b>	Var Nazwy : Int
Dekla: Var Nazwy ':' Typ ':'	<b>R4</b>	Var Nazwy : Typ ;
Wyr: Calk	<b>R5</b>	Dekla Begin Nazwa As Nazwa + Calk
Wyr: Nazwa '+' Wyr	<b>R6</b>	Dekla Begin Nazwa As Nazwa + Wyr
Podst: Nazwa As Wyr	<b>R7</b>	Dekla Begin Nazwa As Wyr
Instr: Podst	<b>R8</b>	Dekla Begin Podst
Prog: Dekla Begin Instr End '.'	<b>R9</b>	Dekla Begin Instr End .

Compilers (108)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Code generation

```
Podst: Nazwa As Wyr {printf("pop %c\n", $1);}
;
Wyr: Calk R5 {printf("push %d\n", $1);}
| Nazwa {printf("push %c\n", $1);}
| Calk '+' Wyr {printf("pop si\n pop di\n");
printf("add si, di\n push si\n");}
| Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
printf("push si\n");}
```

**Before**

y, x DW ?
+
Nazwa (x)
As
Nazwa (y)
Begin
Dekla
Stack
Calk (10)
End

Compilers (109)

Introduction to Informatics

### Code generation

```
Podst: Nazwa As Wyr {printf("pop %c\n", $1);}
;
Wyr: Calk R5 {printf("push %d\n", $1);}
| Nazwa {printf("push %c\n", $1);}
| Calk '+' Wyr {printf("pop si\n pop di\n");
printf("add si, di\n push si\n");}
| Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
printf("push si\n");}
```

**Before**

y, x DW ?
push 10
+
Nazwa (x)
As
Nazwa (y)
Begin
Dekla
Stack
Wyr
End

Compilers (110)

Introduction to Informatics

### List of Reductions

```
Nazwy: Nazwa R1 Var Nazwa , Nazwa
Nazwy: Nazwa ',' Nazwy R2 Var Nazwa , Nazwy
Typ: Int R3 Var Nazwy : Int
Dekla: Var Nazwy ':' Typ ':' R4 Var Nazwy : Typ ;
Wyr: Calk R5 Dekla Begin Nazwa As Nazwa + Calk
Wyr: Nazwa '+' Wyr R6 Dekla Begin Nazwa As Nazwa + Wyr
Podst: Nazwa As Wyr R7 Dekla Begin Nazwa As Wyr
Instr: Podst R8 Dekla Begin Podst
Prog: Dekla R9 Dekla Begin Instr End .
Begin Instr End '.'
```

Compilers (111)

Introduction to Informatics

### Code generation

```
Podst: Nazwa As Wyr {printf("pop %c\n", $1);}
;
Wyr: Calk {printf("push %d\n", $1);}
| Nazwa {printf("push %c\n", $1);}
| Calk '+' Wyr {printf("pop si\n pop di\n");
printf("add si, di\n push si\n");}
R6 | Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
printf("push si\n");}
```

**Before**

y, x DW ?
push 10
+
Nazwa (x)
As
Nazwa (y)
Begin
Dekla
Stack
Wyr
End

Compilers (112)

Introduction to Informatics

### Code generation

```
Podst: Nazwa As Wyr {printf("pop %c\n", $1);}
;
Wyr: Calk {printf("push %d\n", $1);}
| Nazwa {printf("push %c\n", $1);}
| Calk '+' Wyr {printf("pop si\n pop di\n");
printf("add si, di\n push si\n");}
R6 | Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
printf("push si\n");}
```

**After**

y, x DW ?
push 10
pop si
add si, x
push si
+
Nazwa (y)
Begin
Dekla
Stack
Wyr
End

Compilers (113)

Introduction to Informatics

### Code generation

```
Podst: Nazwa As Wyr {printf("pop %c\n", $1);} R7
;
Wyr: Calk {printf("push %d\n", $1);}
| Nazwa {printf("push %c\n", $1);}
| Calk '+' Wyr {printf("pop si\n pop di\n");
printf("add si, di\n push si\n");}
| Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
printf("push si\n");}
```

**Before**

y, x DW ?
push 10
pop si
add si, x
push si
+
Nazwa (y)
Begin
Dekla
Stack
Wyr
End

Compilers (114)

## (c) Jerzy Nawrocki, Introduction to Informatics

Introduction to Informatics

### Code generation

```

Podst: Nazwa As Wyr {printf("pop %c\n", $1);}
;
Wyr: Calk          {printf("push %d\n", $1);}
    | Nazwa        {printf("push %c\n", $1);}
    | Calk '+' Wyr {printf("pop si\n pop di\n");
    | Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
                    printf("push si\n");}

```

**Stack**

**After**

```

y, x DW ?
push 10
pop si
add si, x
push si
pop y

```

Podst  
Begin  
Dekla

End

Compilers (115)

Introduction to Informatics

### Code generation

```

...
Dekla: Var Nazwy ':' Typ ';'
;
Nazwy: Nazwa
    | Nazwa ',' Nazwy
;
Typ: Int
    | Rea
;
Instr: Podst
;
Podst: Nazwa As Wyr {printf("pop %d\n", $1);}
;
Wyr: Calk          {printf("push %d\n", $1);}
    | Nazwa        {printf("push %c\n", $1);}
    | Calk '+' Wyr {printf("pop si\n pop di\n");
    | Nazwa '+' Wyr {printf("pop si\n add si, %c\n", $1);
                    printf("push si\n");}

```

Introduction to Informatics

### Weaknesses

Should be:

```

y DW ?
x DW ?

```

y, x DW ?  
push 10  
pop si  
add si, x  
push si  
pop y

Compilers (117)

Introduction to Informatics

### Weaknesses

Lack of SEGMENT, ENDS itp.  
One could add int 3

y, x DW ?  
push 10  
pop si  
add si, x  
push si  
pop y

Compilers (118)

Introduction to Informatics

### Summary

At last!

- YACC and Lex
- Pas definition
- Reduction-based syntax analysis
- Simple compiler

Compilers (119)

Introduction to Informatics

### Bibliography

- Aho, Sethi, Ullman, **Compilers: Principles, Techniques, and Tools**, Addison-Wesley, 1986.

Compilers (120)