

## FORMELLA SPRÅK

### FORMELLA SPRÅK OCH SYNTAXANALYS

#### BARA FÖR BEHÖVANDE!

- ALFABET, STRÄNGAR, FORMELLA SPRÅK
- REGULJÄRA UTTRYCK
- KONTEXTFRI GRAMMATIK, BNF
- SYNTAXTRÄD
- LEXIKAL ANALYS
- SYNTAXANALYS MED REKURSIV MEDÄKNING

ETT FORMELLT SPRÅK ÄR EN MÄNGD STRÄNGAR

EXEMPEL 1:  $\{xy, yxx, xyzy, zxy\}$

EXEMPEL 2:  $\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, \dots\}$

EXEMPEL 3:  $\{\text{SYNTAXTISKT KORREKTA C-PROGRAM}\}$

EXEMPEL 4:  $\{10, 1010, 101010, 10101010, \dots\}$

EXEMPEL 5:  $\{\text{SATISFIERBARA BOOLESKA FORMLER}\}$

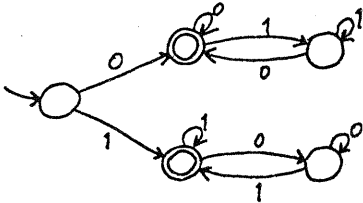
DET FINNS FLERA SÄTT ATT BESKRIVA FORMELLA SPRÅK:

- RÄKNA UPP STRÄNGARNA I SPRÅKET
- ANGE REGLER FÖR HUR STRÄNGAR SOM TILLHÖR SPRÅKET MÅSTE SE UT (GRAMMATIK)
- KONSTRUERA EN ALGORITM SOM KAN AVGÖRA HURVIDA EN STRÄNG TILLHÖR SPRÅKET ELLER INTE (BESLUTSPROBLEM)

## DETERMINISTISK ÄNDLIG AUTOMAT

### DFA

EXEMPEL: KOLLA OM FÖRSTA OCH SISTA SIFFRAN I INMÄTTNINGEN ÄR SAMMA.



- REGLER:
- BÖRJA I STARTTILLSTÄNDET
  - EN SYMBOL LÄSES I VARJE ÖVERGÅNG
  - AUTOMATEN ACCEPTERAR INMÄTTNINGEN OM DEN NÄR INMÄTTNINGEN ÄR SLUT BEFINNER SEJ I ETT ACCEPTERANDE TILLSTÄND.

EN FORMELL DEFINITION AV EN DFA ANGER:

- $\Sigma$  ALFABET (ÄNDLIG MÄNGD SYMBOLER)
- $Q$  TILLSTÄND (ÄNDLIG MÄNGD)
- $q_0$  STARTTILLSTÄND ( $q_0 \in Q$ )
- $F$  ACCEPTERANDE TILLSTÄND ( $F \subseteq Q$ )
- $\delta$  ÖVERGÅNGSFUNKTION ( $\delta: Q \times \Sigma \rightarrow Q$ )



# GRAMMATIK

$\langle S \rangle \rightarrow \langle NP \rangle \langle VP \rangle \mid \langle NP \rangle \langle VP \rangle \langle ADVP \rangle$

$\langle VP \rangle \rightarrow \langle V \rangle \mid \langle V \rangle \langle ADVP \rangle$

$\langle ADVP \rangle \rightarrow \langle PP \rangle \mid \langle ADV \rangle$

$\langle PP \rangle \rightarrow \langle PREP \rangle \langle NP \rangle$

$\langle NP \rangle \rightarrow \langle N \rangle \mid \langle ADJ \rangle \langle N \rangle$

OMSKRIVNINGSGREGLER ELLER PRODUKTION

$\langle V \rangle \rightarrow$  BÖRJADE | STARTADE | HÖPPAR | HÖPPADE | BESTÄMMER

$\langle N \rangle \rightarrow$  BILEN | SALE 3 | TÄNKEN | FÖRELÄSNINGEN

$\langle ADJ \rangle \rightarrow$  GULA | FÖRSTÅNIGA | ÖTVETYDIGA

$\langle ADV \rangle \rightarrow$  VÅL | GULT | NYLIGEN | HEMSKT | GENAST

$\langle PREP \rangle \rightarrow$  PÅ | I | EFTER | UTANFÖR

STARTSYMBOL:  $\langle S \rangle$

SLUTSYMBOL  
ICKESLUTSYMBOL

KALLAS PRODUKTIONSGRAMMATIK

ELLER FRASSTRUKTURGRAMMATIK

# HÄRLEDNING



OMSKRIVNINGSGREGLER:

$X \rightarrow Y$  VARJE FÖREKOMST AV X KAN BYTAS MOT Y

$X \rightarrow Y \mid Z$  X KAN BYTAS MOT Y ELLER Z

EXEMPEL:

HÄRLEDNING AV FRASEN "GULA BILEN STARTADE"

STARTSYMBOLER:  $\langle S \rangle$

$\langle S \rangle \rightarrow \langle NP \rangle \langle VP \rangle$

$\langle NP \rangle \langle VP \rangle$

$\langle NP \rangle \rightarrow \langle ADJ \rangle \langle N \rangle$

$\langle ADJ \rangle \langle N \rangle \langle VP \rangle$

$\langle ADJ \rangle \rightarrow$  GULA

GULA  $\langle N \rangle \langle VP \rangle$

$\langle N \rangle \rightarrow$  BILEN

GULA BILEN  $\langle VP \rangle$

$\langle VP \rangle \rightarrow \langle V \rangle$

GULA BILEN  $\langle V \rangle$

$\langle V \rangle \rightarrow$  STARTADE

GULA BILEN STARTADE

ALLA FRASER SOM KAN HÄRLEDAS FRÅN STARTSYMBOLER BILDAR SPRÅKET SOM GENERERAS AV GRAMMATIKEN

# VÄNSTER/HÖGERHÄRLEDNING

SAMMA FRAS KAN HÄRLEDAS PÅ OLIKA SÄTT.

OM MAN VÄLJER ATT ALLTID SKRIVA OM DEN HÖGRASTE ICKESLUTSYMBOLER FÅR MAN EN HÖGERHÄRLEDNING.

EXEMPEL:

HÖGERHÄRLEDNING AV FRASEN "GULA BILEN STARTADE"

STARTSYMBOLER:  $\langle S \rangle$

$\langle S \rangle \rightarrow \langle NP \rangle \langle VP \rangle$   $\langle NP \rangle \langle VP \rangle$

$\langle VP \rangle \rightarrow \langle V \rangle$   $\langle NP \rangle \langle V \rangle$

$\langle V \rangle \rightarrow$  STARTADE  $\langle NP \rangle$  STARTADE

$\langle NP \rangle \rightarrow \langle ADJ \rangle \langle N \rangle$   $\langle ADJ \rangle \langle N \rangle$  STARTADE

$\langle N \rangle \rightarrow$  BILEN  $\langle ADJ \rangle$  BILEN STARTADE

$\langle ADJ \rangle \rightarrow$  GULA GULA BILEN STARTADE

↓ UPPFRÅN-NER-ANALYSATORER ANVÄNDER VÄNSTERHÄRLEDNING.

↑ NERIFRÅN-UPP-ANALYSATORER ANVÄNDER HÖGERHÄRLEDNING.

# EBNF - UTVIDGAD BNF

## NYA OPERATIONER:

( ) GRUPPERAR

$E ::= E ('+' | '-') T | T$

[ ] NOLL ELLER EN FÖREKOMST

$E ::= [ E ('+' | '-') ] T$

{ } NOLL ELLER FLERA FÖREKOMSTER

LISTA ::=  $E \{ ',' E \}$

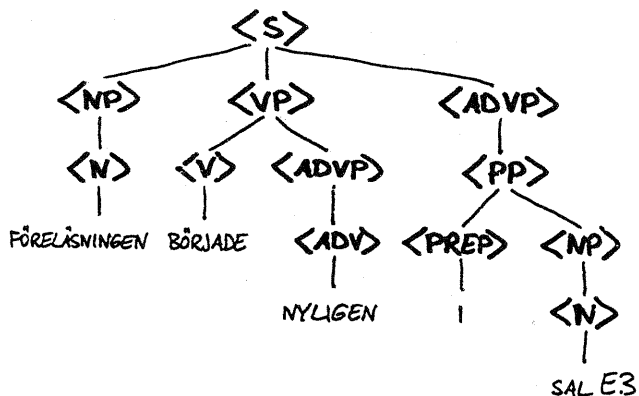
## EBNF GER INGEN ÖKAD

UTTRYCKSFULLHET MEN

ÖKAD BEKVÄMLIGHET.

# SYNTAXTRÄD

EN HÄRLEDNING FÖR EN KONTEXTFRI GRAMMATIK BESKRIVS ENKLAST MED ETT SYNTAXTRÄD.



LÖVEN FINNS SLUTSYMBOLER.

INRE NÖDER FINNS IKKESLUTSYMBOLER.

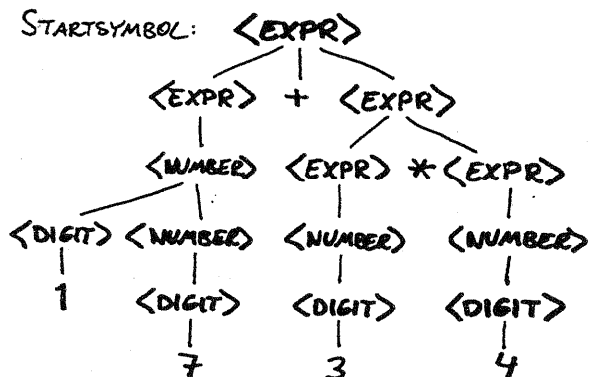
## EXEMPEL: TALUTTRYCK

$\langle \text{EXPR} \rangle ::= \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$   
|  $\langle \text{EXPR} \rangle - \langle \text{EXPR} \rangle$   
|  $\langle \text{EXPR} \rangle * \langle \text{EXPR} \rangle$   
|  $\langle \text{EXPR} \rangle / \langle \text{EXPR} \rangle$   
|  $\langle \text{EXPR} \rangle ^ \langle \text{EXPR} \rangle$   
|  $- \langle \text{EXPR} \rangle$   
|  $( \langle \text{EXPR} \rangle )$   
|  $\langle \text{NUMBER} \rangle$

$\langle \text{NUMBER} \rangle ::= \langle \text{DIGIT} \rangle$   
|  $\langle \text{DIGIT} \rangle \langle \text{NUMBER} \rangle$

$\langle \text{DIGIT} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

STARTSYMBOL:  $\langle \text{EXPR} \rangle$



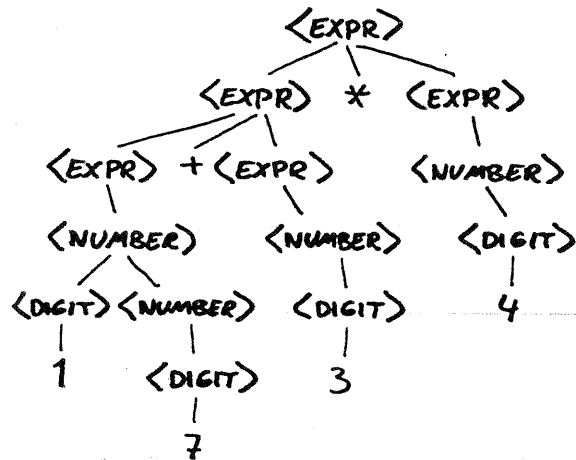
# FLERTYDIGHET

EN GRAMMATIK KAN VARA ENTYDIG ELLER FLERTYDIG (AMBIGUOUS).

DÅ KAN SAMMA STRÄNG HÄRLEDAS MED FLERA SYNTAXTRÄD.

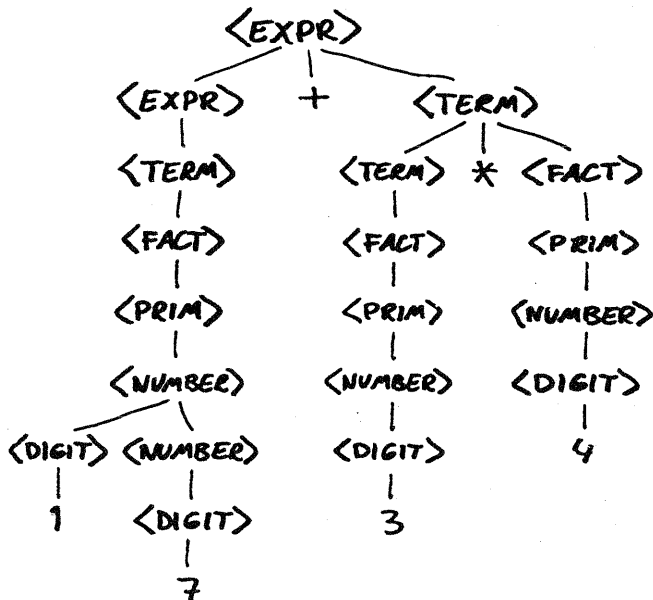
EXEMPEL: TALUTTRYCKSGRAMMATIKEN ÄR FLERTYDIG.

STRÄNGEN  $17+3*4$  HAR OCKSÅ SYNTAXTRÄDET



DETTA SYNTAXTRÄD SPEGLAR INTE SEMANTIKEN. VI VILL HA TOLKNINGEN  $17+(3*4)$ .

EXEMPEL: SYNTAXTRÄD FÖR  $17+3*4$



# OMSKRIVNING AV GRAMMATIK

EN GRAMMATIK KAN IBLAND SKRIVAS OM SÅ DEN BLIR ENTYDIG. OM DETTA INTE GÅR ÄR GRAMMATIKEN OHJÄLPUGT FLERTYDIG (INHERENTLY AMBIGUOUS).

EXEMPEL: ENTYDIG GRAMMATIK FÖR TALUTTRYCK.

$$\langle \text{EXPR} \rangle ::= \langle \text{TERM} \rangle$$
  

$$| \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle$$
  

$$| \langle \text{EXPR} \rangle - \langle \text{TERM} \rangle$$

$$\langle \text{TERM} \rangle ::= \langle \text{FACT} \rangle$$
  

$$| \langle \text{TERM} \rangle * \langle \text{FACT} \rangle$$
  

$$| \langle \text{TERM} \rangle / \langle \text{FACT} \rangle$$

$$\langle \text{FACT} \rangle ::= \langle \text{PRIM} \rangle$$
  

$$| \langle \text{PRIM} \rangle \wedge \langle \text{FACT} \rangle$$

$$\langle \text{PRIM} \rangle ::= \langle \text{NUMBER} \rangle$$
  

$$| - \langle \text{PRIM} \rangle$$
  

$$| ( \langle \text{EXPR} \rangle )$$

$$\langle \text{NUMBER} \rangle ::= \langle \text{DIGIT} \rangle$$
  

$$| \langle \text{DIGIT} \rangle \langle \text{NUMBER} \rangle$$

$$\langle \text{DIGIT} \rangle ::= 0|1|2|3|4|5|6|7|8|9$$

# FÖRUTSÄGANDE SYNTAXANALYSATOR

(PREDICTIVE PARSER)

IDÉ:

- HÄRLED FRASEN UPPIFRÅN (FRÅN START-SYMBOLEN) OCH NED (MOT SLUTSYMBOLERNA).
- VÄLJ VILKEN OMSKRIVNINGSGREGLER SOM SKA TILLÄMPAS GENOM ATT KIKKA PÅ NÄSTA SYMBOL I INMÄTNINGEN.

EXEMPEL:

REGLER: 
$$\langle \text{PRIM} \rangle ::= \text{NUMBER}$$
  

$$| - \langle \text{PRIM} \rangle$$
  

$$| ( \langle \text{EXPR} \rangle )$$

HÄRLEDNINGUTTRYCK:  $\langle \text{PRIM} \rangle$

INMÄTNING | TECKEN: - 4711  
 -" - | SLUTSYMBOLER: - NUMBER

NÄSTA SYMBOL I INMÄTNINGEN ÄR -  
 VÄLJ DÄRFÖR HÖGERLEDET -  $\langle \text{PRIM} \rangle$

# PROBLEM 1: HÖGERLED SOM BÖRJAR MED SAMMA SYMBOL

EXEMPEL:  $\langle \text{FACT} \rangle ::= \langle \text{PRIM} \rangle$   
 $\quad \quad \quad | \langle \text{PRIM} \rangle \wedge \langle \text{FACT} \rangle$

LÖSNING: • FAKTORISERA UT DET GEMENSAMMA PREFIXET  
 • SKAPA NY REGEL MED RESTEN

EXEMPEL:  $\langle \text{FACT} \rangle ::= \langle \text{PRIM} \rangle \langle \text{REST} \rangle$   
 $\langle \text{REST} \rangle ::= \epsilon \quad | \quad \wedge \langle \text{FACT} \rangle$

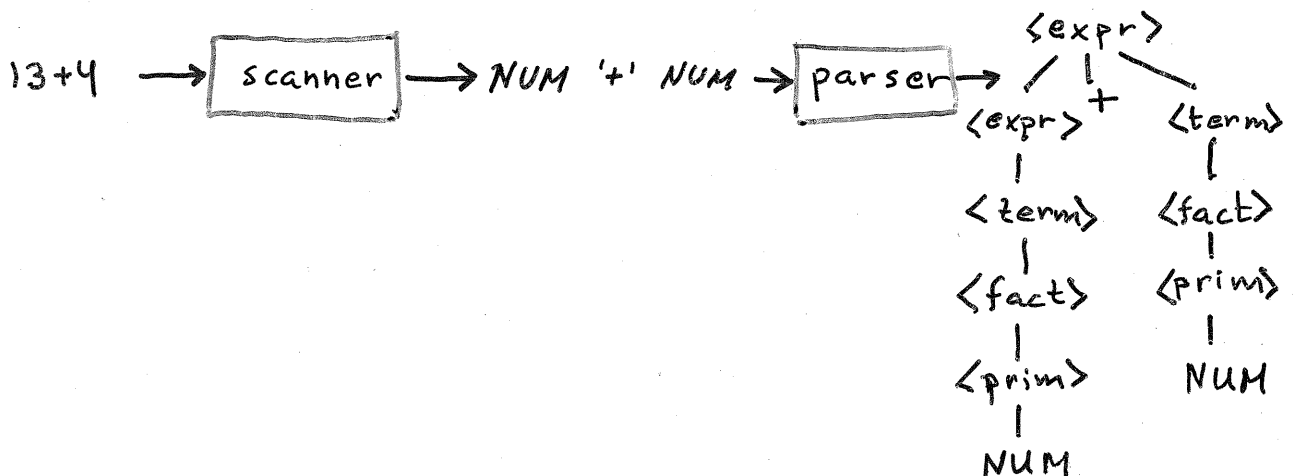
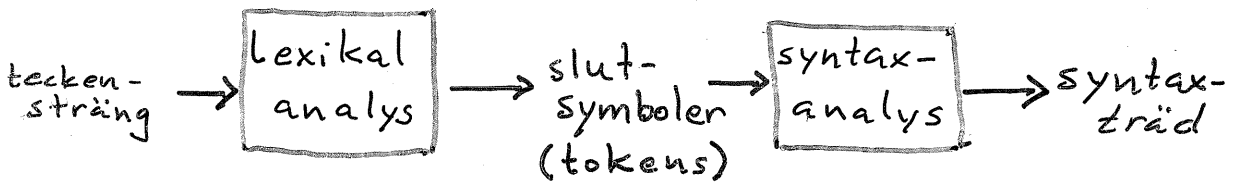
# PROBLEM 2: VÄNSTERREKURSION

EXEMPEL:  $\langle \text{EXPR} \rangle ::= \langle \text{TERM} \rangle$   
 $\quad \quad \quad | \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle$   
 $\quad \quad \quad | \langle \text{EXPR} \rangle - \langle \text{TERM} \rangle$

LÖSNING: • SEPARERA VÄNSTERREKURSIVA HÖGERLED FRÅN ÖVRIGA  
 • SKAPA EN NY ICKESLUTSYMBOL  $\langle H \rangle$   
 • LÄGG TILL  $\langle H \rangle$  PÅ SLUTET AV ALLA ICKE VÄNSTERREKURSIVA HÖGERLED  
 • LÅT HÖGERLEDEN TILL  $\langle H \rangle$  VARA DEN VÄNSTERREKURSIVA HÖGERLEDEN OMGJORDA TILL HÖGERREKURSIVA

EXEMPEL:  $\langle \text{EXPR} \rangle ::= \langle \text{TERM} \rangle \langle H \rangle$

$\langle H \rangle ::= \epsilon$   
 $\quad \quad \quad | + \langle \text{TERM} \rangle \langle H \rangle$   
 $\quad \quad \quad | - \langle \text{TERM} \rangle \langle H \rangle$



# REKURSIV MEDÅKNING

(RECURSIVE-DESCENT)

UPPIFRÅN OCH NERSYNTAXANALYS IMPLEMENTERAD MED HJÄLP AV PROCEDURER UPPBYGGDA FRÅN GRAMMATIKEN.

## 1. EN LEXIKALANALYSPROCEDUR MATCH(SYMBOL)

KOLLAR ATT NÄSTA SYMBOL I INMATNINGEN ÄR JUST SYMBOL OCH LÄSER IN EFTERFÖLJANDE SYMBOL I NEXTTOKEN.

## 2. EN PROCEDUR FÖR VARJE REGEL (ICKE-SLUTSYMBOL) SOM ANALYSERAR GENOM ATT ANROPA PROCEDURER I DEN ORDNING

MOTSVARANDE SYMBOLER FÖREKOMMER I GRAMMATIKEN.

EXEMPEL:  $fact() = prim(); rest()$   
 $rest() = \text{IF } nextToken = '^' \text{ THEN } \left. \begin{array}{l} MATCH('^'); fact() \end{array} \right\} \begin{array}{l} \text{KAN SLÅS} \\ \text{IHOP} \end{array}$

```
expr(): term();  
while nextToken ∈ {'+', '-'}  
    match(nextToken); term();
```

```
term(): fact();  
while nextToken ∈ {'*', '/'}  
    match(nextToken); fact();
```

```
fact(): prim();  
if nextToken = '^' then  
    match('^'); fact();
```

```
prim(): case nextToken of  
    NUM: match(NUM)  
    '-' : match('-'); prim();  
    '(' : match('(');  
        expr();  
        match(')');
```

```
/* Struktur för rekursiv medåkning */
```

```
typedef enum /* slutsymboler: */  
{  
    START, ERROR, NUMBERSYM, ENDSYM,  
    ADDSYM = '+', MINUSSYM = '-',  
    MULSYM = '*', DIVSYM = '/',  
    EXPSYM = '^', PRINTSYM = ';',  
    LEFTPARSYM = '(', RIGHTPARSYM = ')' }  
TokenValue;
```

```
/* nästa slutsymbol i inmatningen: */  
TokenValue nextToken;
```

```
/* semantiskt värde för NUMBERSYM: */  
double numberValue;
```

```
/* Match kollar att token matchar  
inmatningen och läser ny nextToken: */  
TokenValue Match(TokenValue token);
```

```
/* procedurer för ickeslutsymboler: */  
double prim(void);  
double factor(void);  
double term(void);  
double expr(void);  
void start(void);
```

```
main(void) /* huvudprogram */  
{ Match(START); start(); }
```

