

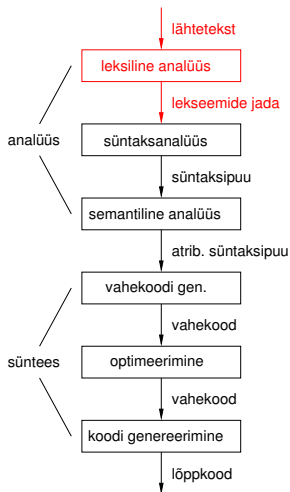
# Leksiline analüüs

Sissejuhatus

Regulaaravaldised

# Leksiline analüüs

- **Leksiline analüüs** kontrollib programmi sõnade (literaalsümbolite) vastavust leksilistele reeglitele ning teisendab programmi sümbolite (**tokens**) jadaks:
  - eemaldab tühisümbolid ja kommentaarid;
  - identifitseerib võtmesõnad, identifikaatorid ja konstandid;
  - konstrueerib sümbolite tabeli;
  - leiab sümbolite rea- ja veerunumbrid;
  - teavitab vajadusel leksilistest vigadest.
- Leksilist analüüsi kutsutakse **skaneerimiseks** (**scanning**) ning vastavat analüsaatorit nimetatakse **skanneriks** (**scanner**).



## Käsitsi kodeeritud skanner

Näide: võtmesõna if äratundmine:

```
c = readchar();
if (c != 'i')
    error();
else {
    c = readchar();
    if (c != 'f')
        error();
    else
        return IF_TOKEN;
}
```

# Käsitsi kodeeritud skanner

## Probleemid:

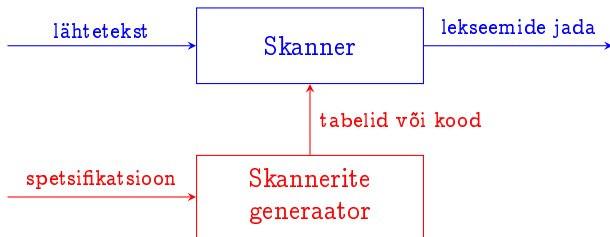
- Vaja kombineerida erinevat liiki lekseemide skannereid
  - Fikseeritud sõnad ja märgijadad (võtmesõnad, operaatorid)
  - Reeglite abil defineeritud lekseemid (identifikaatorid, arvud)
- Lekseemid võivad kattuda
  - = vs. ==
  - if vs. if0

## Käsitsi kodeeritud skanner

Näide: `if` on võtmesõna, kuid `if0` identifikaator:

```
c = readchar();
if (c != 'i') { /* teised lekseemid... */ }
else {
    c = readchar();
    if (c != 'f') { /* teised lekseemid... */ }
    else {
        c = readchar();
        if (c not alpha-numeric) {
            putback(c);
            return IF_TOKEN;
        }
        while (c alpha-numeric) {
            /* ehitada identifikaator */
        }
    }
}
```

# Skannerite generaator



- Kuna käsitsi kodeerimine on tülikas ja vigade aldis, siis tavaliselt genereeritakse automaatselt **skannerite generaatori** abil.
- Programmi sõnade leksilised reeglid esitatakse tavaliselt **regulaaravaldiste** abil.

Regulaaravaldised

# Regulaaravaldised

- Regulaaravaldised üle (lõpliku) tähestiku  $\Sigma$

$$E ::= \varepsilon \mid a \mid (E E) \mid (E \mid E) \mid E^*$$

kus  $a \in \Sigma$ .

- Regulaaravaldis  $E$  defineerib keele  $L(E) \subseteq \Sigma^*$

$$L(\varepsilon) = \{""\}$$

$$L(a) = \{"a"\}$$

$$L(E_1 E_2) = \{vw \mid v \in L(E_1), w \in L(E_2)\}$$

$$L(E_1 \mid E_2) = L(E_1) \cup L(E_2)$$

$$L(E^*) = \{""\} \cup \{vw \mid v \in L(E), w \in L(E^*)\}$$



# Regulaaravaldised

Näiteid:

Regulaaravaldis

$a \mid b$

abba

$ab^*a$

$(ab)^*$

$(a \mid b)^*$

Defineeritav keel

$\{ "a", "b" \}$

$\{ "abba" \}$

$\{ "aa", "aba", "abba", "abbba", \dots \}$

$\{ "", "ab", "abab", "ababab", \dots \}$

$\{ "", "a", "b", "aa", "ab", "ba", \dots \}$

## Regulaaravaldised

- Regulaaravaldistes esinevate sulgude vähendamiseks on operaatoritele määratud prioriteedid:
  - sulundioperaator  $(\cdot)^*$  seob kõige tugevamalt;
  - valikuoperaator  $(\cdot \mid \cdot)$  seob kõige nõrgemalt.
- Lühendavaid tähistusi regulaaravaldiste esitamiseks:
  - *mittetühi sulund*:  $E^+ = EE^*$ ;
  - *optsoon*:  $E? = \varepsilon \mid E$ ;
  - *märgihulgad*:  $[abc] = a \mid b \mid c$ ;
  - *märgivahemikud*:  $[a - z] = a \mid \dots \mid z$ .

# Regulaaravaldised

Regulaarne kirjeldus tähestikus  $\Sigma$  on reeglite hulk

$$d_1 \rightarrow E_1$$

$$d_2 \rightarrow E_2$$

$\dots$

$$d_n \rightarrow E_n$$

kus  $d_i$  on (unikaalne) nimi ja  $E_i$  on regulaaravaldis tähestikus  $\Sigma \cup \{d_1, \dots, d_{i-1}\}$ .

# Regulaaravaldised

Näiteid regulaarsetest kirjeldustest:

## Identifikaatorid:

Letter → [a - z A - Z]  
Digit → [0 - 9]  
Identifier → Letter (Letter | Digit)\*

## Arvkonstandid:

Sign → (+ | -)?  
Integer → 0 | Sign [1 - 9] Digit\*  
Decimal → Integer . Digit+  
Real → (Integer | Decimal) E Integer