Süntaksanalüüs

Ennustav parsimine
Rekursiivse laskumise meetod
Tabeljuhitav LL(1) parser
Vasakfaktoriseerimine

Ennustav parsimine

- Vale produktsioonireegli valimine põhjustab tagasipöördumise.
- Tihti on võimalik valitava reegli õigsuse üle otsustada mingi arvu sisendsümbolite ettevaatamise teel.
- Üldjuhul on tarvis ette vaadata piiramata arv sümboleid.
  - Näit.: Cocke-Younger-Kasami või Earley algoritmid.
- Ennustav parsimine (predictive parsing) on ülalt-alla parsimine, kus alati on võimalik ige reegel valida nii, et pole vaja tagasipöörduda.
  - Grammatika peab olema selline, et järgmine sisendsümbol (või mingi fikseeritud arv sümboleid) määrab unikaalselt ära valitava reegli.

Ennustav parsimine

Iga lausevormi $\alpha \in (N \cup T)^\star$ jaoks defineerime:

$$
\begin{aligned}
Nullable(\alpha) &= \begin{cases} true & \text{if } \alpha \Longrightarrow^\star \varepsilon \\ false & \text{otherwise} \end{cases} \\
First(\alpha) &= \{c \in T \mid \alpha \Longrightarrow^\star c\,\beta\}
\end{aligned}
$$

kus $\beta \in (N \cup T)^\star$.

Ennustav parsimine

Iga lausevormi $\alpha \in (N \cup T)^\star$ jaoks defineerime:

$$
\begin{aligned}
Nullable(\alpha) &= \begin{cases} true & \text{if } \alpha \Longrightarrow^\star \varepsilon \\ false & \text{otherwise} \end{cases} \\
First(\alpha) &= \{c \in T \mid \alpha \Longrightarrow^\star c\,\beta\}
\end{aligned}
$$

kus $\beta \in (N \cup T)^\star$.

Nullable võrrandid:

$$
\begin{aligned}
Nullable(\varepsilon) &= true \\
Nullable(c) &= false \\
Nullable(\alpha\beta) &= Nullable(\alpha) \wedge Nullable(\beta) \\
Nullable(A) &= Nullable(\alpha_1) \vee \ldots \vee Nullable(\alpha_n) \\
&\quad \text{kasutades kõiki produktsioone } A \to \alpha_i
\end{aligned}
$$

Ennustav parsimine

Iga lausevormi $\alpha \in (N \cup T)^\star$ jaoks defineerime:

$$\begin{aligned} Nullable(\alpha) &= \begin{cases} true & \text{if } \alpha \Longrightarrow^\star \varepsilon \\ false & \text{otherwise} \end{cases} \\ First(\alpha) &= \{c \in T \mid \alpha \Longrightarrow^\star c\,\beta\} \end{aligned}$$

kus $\beta \in (N \cup T)^\star$.

$First$ võrrandid:

$$\begin{aligned} First(\varepsilon) &= \emptyset \\ First(c) &= \{c\} \\ First(\alpha\beta) &= \begin{cases} First(\alpha) \cup First(\beta) & \text{if } Nullable(\alpha) \\ First(\alpha) & \text{otherwise} \end{cases} \\ First(A) &= First(\alpha_1) \cup \ldots \cup First(\alpha_n) \\ & \quad \text{kasutades kõiki produktsioone } A \to \alpha_i \end{aligned}$$

Ennustav parsimine

- Kui grammatikas on reeglid $A \to \alpha$ ja $A \to \beta$ sellised, et $First(\alpha) \cap First(\beta) = \emptyset$, siis on esimese sisendsümboli põhjal võimalik otsustada kumba reeglit valida.
- Kehtib ainult juhul, kui $Nullable(\alpha) \vee Nullable(\beta) = false$.
- Vastasel korral vaja vaadata "järgnevat" sümbolit ...

Ennustav parsimine

Lisame uue algsümboli S' ja reegli $S' \to S\$$, kus \$ on spetsiaalne sisendilõpu marker.

Iga mitteterminali $A \in N$ jaoks defineerime hulga:

$$Follow(A) \quad = \quad \{c \in T \mid S' \Longrightarrow^{\star} \alpha\, A\, c\, \beta\}$$

kus $\alpha, \beta \in (N \cup T)^{\star}$.

Ennustav parsimine

Lisame uue algsümboli S' ja reegli $S' \to S\$$, kus \$ on spetsiaalne sisendilõpu marker.

Iga mitteterminali $A \in N$ jaoks defineerime hulga:

$$Follow(A) \quad = \quad \{c \in T \mid S' \Longrightarrow^{\star} \alpha\, A\, c\, \beta\}$$

kus $\alpha, \beta \in (N \cup T)^{\star}$.

Follow võrratused:

- iga $A \in N$ jaoks vaatleme kõiki reegleid $B \to \alpha A \beta$, kus A on parempooles;
- $First(\beta) \subseteq Follow(A)$;
- $Follow(B) \subseteq Follow(A)$, kui $Nullable(\beta)$.

Ennustav parsimine

Näide:

$$
\begin{array}{lcl}
S & \to & A\,B\,C \\
A & \to & a\,A \mid \varepsilon
\end{array}
\qquad
\begin{array}{lcl}
B & \to & b \mid \varepsilon \\
C & \to & c \mid d
\end{array}
$$

$$
\begin{array}{lcl}
First(C) & = & \{c, d\} \\
First(B) & = & \{b\} \\
First(A) & = & \{a\} \\
First(S) & = & First(A\,B\,C) \\
 & = & First(A) \\
 & & \cup\ First(B) \\
 & & \cup\ First(C) \\
 & = & \{a, b, c, d\}
\end{array}
\qquad
\begin{array}{lcl}
Follow(S) & = & \{\$\} \\
Follow(C) & = & \{\$\} \\
Follow(B) & = & First(C) \\
 & = & \{c, d\} \\
Follow(A) & = & First(B) \\
 & & \cup\ First(C) \\
 & = & \{b, c, d\}
\end{array}
$$

Ennustav parsimine

Iga reegli $A \to \alpha$ jaoks defineerime ettevaatamishulga (look-ahead set):

$$la(A \to \alpha) = \begin{cases} First(\alpha) \cup Follow(A), & \text{if } Nullable(\alpha) \\ First(\alpha), & \text{otherwise} \end{cases}$$

Grammatika on LL(1), kui iga (paarikaupa erineva) produktsioonireegli $A \to \alpha$ ja $A \to \beta$ korral

$$la(A \to \alpha) \cap la(A \to \beta) = \emptyset$$

Ennustav parsimine

- Rekursiivselt laskuv (recursive descent) parsimine on ennustava parsimise meetod, kus:
  - iga mitteterminali kohta realiseeritakse üks protseduur, mis tunneb ära sellele mitteterminalile vastavad lausevormid;
  - iga protseduur valib sõltuvalt sisendist reegli ning kutsub (rekursiivselt) välja reegli paremas pooles asuvatele mitteterminalidele vastavad protseduurid.
- Rekursiivselt laskuv parsimine on levinuim meetod (lihtsate keelte) parserite käsitsi kirjutamiseks.

Ennustav parsimine

Näide: olgu A produktsioonireeglid $A \to a\,B \mid b\,C\,A \mid D\,E$
ja $la(A \to D\,E) = \{c\}$

```
parseA() {
  if token = a then {
    token := nextWord(); parseB();

  } else if token = b then {
    token := nextword(); parseC(); parseA();

  } else if token = c then {
    parseD(); parseE();

  } else {
    error();
  }
}
```

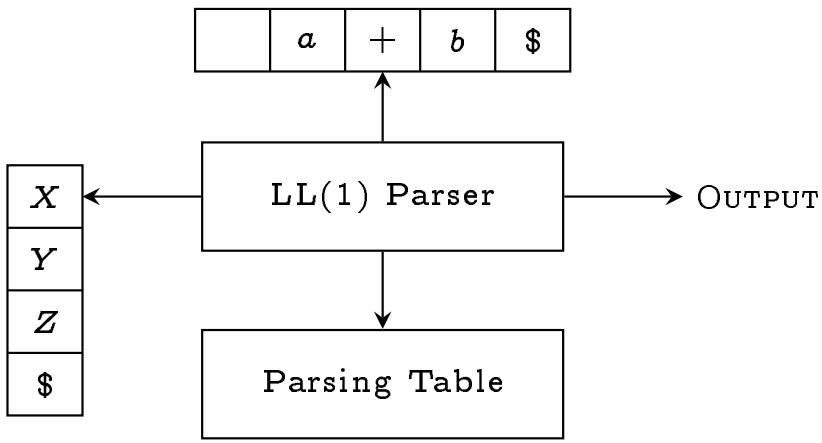
Ennustav parsimine

Näide: olgu A produktsioonireeglid $A \to a\,B \mid b\,C\,A \mid D\,E$
ja $la(A \to D\,E) = \{c\}$

$parseA()$ {
  if $token = a$ **then** {
    $token := nextWord();\ t_1 := parseB();$
    return$(mkTreeRule1(t_1));$
  } **else if** $token = b$ **then** {
    $token := nextword();\ t_1 := parseC();\ t_2 := parseA();$
    return$(mkTreeRule2(t_1, t_2));$
  } **else if** $token = c$ **then** {
    $t_1 := parseD();\ t_2 := parseE();$
    return$(mkTreeRule3(t_1, t_2));$
  } **else** {
    $error();$
  }
}

Ennustav parsimine

- Automaatselt genereeritavate LL(1) parserite korral kasutatakse tavaliselt tabeljuhitavat LL(1) parsimist:
  - moodustatakse maatriks M, kus read ja veerud on vastavalt indekseeritud mitteterminalide ja terminalidega;
  - tabeli pesades on produktsioonireeglid, mida antud mitteterminali ja sisendsümboli korral valida.
- Tabeljuhitava LL(1) parseri struktuur:



Ennustav parsimine

LL(1) parsimise algoritm:

$push(\$);\ push(S);$
$token := nextWord();$
while $stack \neq empty$ **do** {
  $A := pop();$
  if $A \in N$ **then** {
    if $M[A, token] = B_1 \ldots B_n$ **then** {
      $push(B_n); \ldots; push(B_1);$
    } **else** $error();$
  } **else if** $A = token$ **then** {
    $token := nextWord();$
  } **else** $error();$
}

Ennustav parsimine

- Paljusid grammatikaid on võimalik teisendada LL(1) kujule kasutades:
  - vasakrekursiooni elimineerimist;
  - vasakfaktoriseerimist;
  - halvimal juhul üldistame natuke grammatikat (ning kontrollime eemaldatud kitsendusi pärast parsimist).
- Vasakfaktoriseerimine (left-factoring) asendab ühise prefiksiga reeglid uute reeglitega, kus ühine prefiks on ainult ühes parempooles.
- Näide:

$$\begin{array}{lcl} A & \to & B\,a\,C\,D \\ & | & B\,a\,C\,E \end{array} \longrightarrow \begin{array}{lcl} A & \to & B\,a\,C\,Z \\ Z & \to & D \\ & | & E \end{array}$$

Ennustav parsimine

Vasakfaktoriseerimise algoritm:

1. Iga mitteterminali $A \in N$ korral leiame pikima prefixi α, mis esineb kahes või rohkemas A produktsioonireegli parempooles.
2. Kui $\alpha \neq \varepsilon$, siis asendame kõik A produktsioonid

$$A \rightarrow \alpha\,\beta_1 \mid \alpha\,\beta_2 \mid \ldots \mid \alpha\,\beta_n \mid \gamma$$

produktsioonidega

$$\begin{aligned} A &\rightarrow \alpha\,Z \mid \gamma \\ Z &\rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n \end{aligned}$$

kus $Z \in N$ on uus mitteterminal.

3. Kordame protsessi, kuni ükski parempool ei oma ühist prefiksit.