

Übungsblatt 06

Übung

Abgabe bis Di., 06.06., 8 Uhr

Allgemein

Ihre Lösungen werden nur korrigiert, wenn Sie einer Übungsgruppe angehören, d.h. Teilnehmer einer Stud.IP-Veranstaltung *GdPI - Übung - (\langle Termin \rangle)* sind.

Nur wer in Ilias etwas abgibt kann Punkte bekommen. Siehe dazu die Anmerkungen auf Übungsblatt 02.

Benutzen Sie Markdown mit AsciiMath um die Lösung der Aufgaben zu strukturieren und zu kodieren. Fassen Sie die **Lösungen aller Aufgaben in einer Datei** zusammen.

Formatieren Sie Ihre Abgabe so, dass **die einzelnen Aufgaben klar durch passende Überschriften voneinander** getrennt sind.

Geben Sie die Datei über das Lernmodul *GdPI 06 - Markdown+AsciiMath* ab.

Aufgabe 1 – 48 Punkte

Nicht-rekursiver Parser I

Grammatik $G = (N, T, S, P)$.

Nichtterminale $N = \{program, stmtList, stmt, expr, termTail, term, factorTail, factor, addOp, multOp\}$

Terminale $T = \{id, (,), :=, +, -, *, /\}$

Startsymbol $S = program$

Produktionen

$$P = \left\{ \begin{array}{ll} program & \rightarrow stmtList \\ stmtList & \rightarrow stmt \, stmtList \mid \varepsilon \\ stmt & \rightarrow id \, := \, expr \\ expr & \rightarrow term \, termTail \\ termTail & \rightarrow addOp \, term \, termTail \mid \varepsilon \\ term & \rightarrow factor \, factorTail \\ factorTail & \rightarrow multOp \, factor \, factorTail \mid \varepsilon \\ factor & \rightarrow (\, expr \,) \mid id \\ addOp & \rightarrow + \mid - \\ multOp & \rightarrow * \mid / \end{array} \right\}$$

1. Folgende FIRST-Mengen sind gegeben.

$$\text{FIRST}(\text{multOp factor factorTail}) = \{*, /\}$$

$$\text{FIRST}(\epsilon) = \{\epsilon\}$$

$$\text{FIRST}((\text{expr})) = \text{FIRST}(()) = \{($$

$$\text{FIRST}(\text{id}) = \{\text{id}\}$$

$$\text{FIRST}(+) = \{+\}$$

$$\text{FIRST}(-) = \{-\}$$

$$\text{FIRST}(*) = \{*\}$$

$$\text{FIRST}(/) = \{/ \}$$

Berechnen Sie für jede noch fehlende rechte Seite α der Produktion aus P jeweils die Menge $\text{FIRST}(\alpha)$.

(18 Punkte)

2. Folgende FOLLOW-Mengen sind gegeben.

$$\text{FOLLOW}(\text{program}) = \{\$\}$$

$$\text{FOLLOW}(\text{stmtList}) = \{\$\}$$

$$\text{FOLLOW}(\text{stmt}) = \{\text{id}, \$\}$$

$$\text{FOLLOW}(\text{expr}) = \{\text{id},), \$\}$$

$$\text{FOLLOW}(\text{multOp}) = \{(, \text{id}\}$$

Berechnen Sie für die übrigen Nichtterminale A die Menge $\text{FOLLOW}(A)$.

(18 Punkte)

3. Konstruieren Sie eine Parse-Tabelle mit Hilfe von FIRST und FOLLOW.

(12 Punkte)

Hinweis. Es reicht aus in die Zellen die rechten Seiten der Produktionen einzutragen.

Hinweis

Für die Markierung des Endes der Eingabe wird in dieser Aufgabe $\$$ verwendet, weil $\$$ ein Steuerzeichen von AsciiMath ist und sich $\$$ deshalb nur mit etwas Aufwand darstellen lässt.

Aufgabe 2 – 19 Punkte

Nicht-rekursiver Parser II

Betrachten Sie folgende Parse-Tabelle, $\langle 1 \rangle$ ist das Startsymbol der zugehörigen Grammatik.

	0	1	\$neg\$	\$vv\$	\$^~\$	\$\\$	
:---	:---	:---	:---	:---	:---	:---	
<1>	<2>	<2>	<2>				
<2>	<3><4>	<3><4>	<3><4>				
<3>	0	1	\$neg\$<2>				
<4>				\$vv\$<3><4>	\$^~\$<3><4>	\$\epsilon\$	

Hinweis

In den Zellen sind nur die rechten Seiten der Produktionen notiert. Z.B. Zelle [$\langle 3 \rangle$, \neg] enthält $\neg \langle 2 \rangle$, d.h. der Inhalt dieser Zelle repräsentiert die Produktion $\langle 3 \rangle \rightarrow \neg \langle 2 \rangle$.

1. Geben Sie die zugehörige Grammatik $G = (N, T, S, P)$ an. Orientieren Sie sich an der Darstellung in Aufgabenteil .
(7 Punkte)
2. Stellen Sie dar wie ein nicht-rekursiver Parser unter Benutzung der Parse-Tabelle folgende Eingabe abarbeitet (§§ markiert das Ende der Eingabe).

$$\neg 0 \vee 0 \wedge 1 \vee \neg 1 \quad §§$$

Vervollständigen Sie dazu nachfolgende Tabelle.
(12 Punkte)

Hinweis

In einer Tabellenzeile ist der Zustand des Stapels und die noch nicht verarbeitete Eingaben, inklusive des aktuellen Eingabensymbols, dargestellt. Weiterhin die Aktion, die aus Stapel und Eingabe folgt. Mögliche Aktionen sind z.B. Folgende.

- Anwendung einer Produktion, dann wird diese notiert.
- Entfernen des obersten Stapelsymbols und des aktuellen Eingabensymbols, notiere *match*.
- Stapel γ , Eingabe §§, notiere *stop*.
- Ansonsten *error*.

Aus der Aktion ergeben sich Stapel und Eingabe der nächsten Zeile.

Stapel	Eingabe	Aktion
---	---	---
$\langle 1 \rangle \gamma$	$\neg 0 \vee 0 \wedge 1 \vee \neg 1 §§$	$\langle 1 \rangle \rightarrow \neg \langle 2 \rangle$
$\langle 2 \rangle \gamma$	$\neg 0 \vee 0 \wedge 1 \vee \neg 1 §§$	$\langle 2 \rangle \rightarrow \neg \langle 3 \rangle \langle 4 \rangle$
$\langle 3 \rangle \langle 4 \rangle \gamma$	$\neg 0 \vee 0 \wedge 1 \vee \neg 1 §§$	$\langle 3 \rangle \rightarrow \neg \neg \langle 2 \rangle$
$\neg \neg \langle 2 \rangle \langle 4 \rangle \gamma$	$\neg 0 \vee 0 \wedge 1 \vee \neg 1 §§$	match
$\langle 2 \rangle \langle 4 \rangle \gamma$	$0 \vee 0 \wedge 1 \vee \neg 1 §§$	

Hinweis

Beide Tabellen finden Sie in der Stud.IP-Veranstaltung *Grundlagen der Praktischen Informatik (Informatik II)* unter *Übung → uebung06-data*.

Aufgabe 3 – 8 Punkte

LL-Syntaxanalyse

$$P' = \{$$

$product$	$\rightarrow factor$
$factor$	$\rightarrow factor * id \mid id$
sum	$\rightarrow id \ sumList$
$sumList$	$\rightarrow + \ sum \mid + \ (\ product \)$

$$\}$$

Schreiben Sie die Produktionen in P' so um, dass keine Linkrekursionen und keine gemeinsamen Präfixe mehr vorkommen.

(8 Punkte)

Praktische Übung

Dieses Übungsblatt hat keine Praktische Übung.