Grundlagen der Praktischen Informatik **Sommersemester 2023**
Dr. Henrik Brosenne Georg-August-Universität Göttingen
Institut für Informatik

Übungsblatt 07

Übung

Abgabe bis Di., 13.06., 8 Uhr

Allgemein

Ihre Lösungen werden nur korrigiert, wenn Sie einer Übungsgruppe angehören, d.h. Teilnehmer einer Stud.IP-Veranstaltung *GdPI - Übung - ⟨⟨Termin⟩⟩* 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 07 - Markdown+AsciiMath* ab.

Aufgabe 1 – 8 Punkte

KNF

Gegeben sei die folgende aussagenlogische Formel F mit den atomaren Aussagen A, B, C, D.

$$\big((C \wedge \neg D) \wedge \neg(\neg D \implies B)\big) \vee \big((\neg A \vee \neg B) \wedge (\neg A \implies \neg C)\big)$$

Formen Sie die Formel in Konjunktive Normalform (KNF) um und geben Sie die resultierende Klauselmenge an.
(8 Punkte)

Aufgabe 2 – 13 Punkte

Resolutionskalkül

Betrachten Sie die folgende aussagenlogische Formel.

$$F = (X \vee \neg Y) \wedge (\neg X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Z) \wedge (X \vee Y \vee Z) \wedge (Y \vee \neg Z)$$

1. Bestimmen Sie die Klauselmenge $\mathcal{K}_0$ der Formel F.
   (2 Punkte)

2. F ist nicht erfüllbar. Zeigen Sie das mit dem Resolutionskalkül der Aussagenlogik.

   Geben Sie für jedem Schritt i folgendes an.

   $$\mathcal{K}_i = \text{Res}(\mathcal{K}_{i-1}) = \mathcal{K}_{i-1} \cup \{R \mid R \text{ ist Resolvente von } K', K'' \in \mathcal{K}_{i-1}\}$$

   (11 Punkte)

   Hinweis

   Der letzte Schritt ist die Bildung der Resolvente der Klauseln $\{Z\}$ und $\{\neg Z\}$.

   Beispiel aus dem Skript

   $$\mathcal{K}_0 = \Big\{\{P, S\}, \{\neg P, \neg S\}, \{\neg S, A\}, \{\neg A, P\}, \{P\}, \{S\}\Big\}$$
   $$\mathcal{K}_1 = \text{Res}(\mathcal{K}_0) = \mathcal{K}_0 \cup \Big\{\{\neg S\} \mid \text{Resolvente von } \{\neg P, \neg S\}, \{P\}\Big\}$$
   $$\mathcal{K}_2 = \text{Res}(\mathcal{K}_1) = \mathcal{K}_1 \cup \Big\{\quad \emptyset \quad \mid \text{Resolvente von } \{\neg S\}, \{S\}\Big\}$$

Aufgabe 3 – 18 Punkte

Formeln und Terme

Betrachten Sie eine einfache arithmetische Sprache über der Individuenmenge der ganzen Zahlen $\mathbb{Z}$ mit folgender Signatur. Die arithmetischen Operationen und die Größenrelationen über $\mathbb{Z}$ werden als bekannt vorausgesetzt.

- Prädikate $\mathcal{P} = \{<, >\}$.
  - Alle Prädikate habe die Stelligkeit 2.
  - Es gibt $X < Y$ ist genau dann *wahr*, wenn X echt kleiner als Y ist.
  - Es gibt $X > Y$ ist genau dann *wahr*, wenn X echt größer als Y ist.
- Funktoren $\mathcal{F} = \{+, -\}$.
  - Alle Funktoren habe die Stelligkeit 2.
  - Der Funktor $+$ entspricht der Addition in $\mathbb{Z}$.
  - Der Funktor $-$ entspricht der Subtraktion in $\mathbb{Z}$.

Welche der folgenden Ausdrücke sind Formeln, welche Terme und welche gehören nicht zur Spache? Begründen Sie jeweils Ihre Entscheidung.

Kann für einen Formel eine Wahrheitswert angegeben werden, gegeben Sie diesen an und begründen Sie Ihre Antwort.

Kann für eine Term das bezeichnete Objekt angegeben werden, geben Sie dieses an und begründen Sie Ihre Antwort.

1. $X + (-Y)$
2. $1 + 2 - X - Y$
3. $2 < \big((2 > 3) \vee (3 \geq 2)\big)$
4. $1 > \big(\exists X : (X < 0)\big)$
5. $2 + 3 = 5$
6. $(X > 2) \wedge (X + 2)$
7. $(Z > 2) \vee \big((Z - 1) < 0\big)$
8. $\forall X : \big(X \Rightarrow \big(X \cdot (2 + 3)\big)\big)$
9. $\exists X : \Big(\exists Y : \big((X > 2) \wedge \big(\neg(Y < 3)\big)\big)\Big)$
10. $6 + 1 - 8$
11. $\forall X : \big(\big(X \wedge (2 > 3)\big) \Rightarrow X\big)$
12. $\exists : \big((Z > 2) \vee \big((Z - 1) < 0\big)\big)$

(18 Punkte)

Aufgabe 4 – 36 Punkte

Sprachen und Formeln

1. Gegeben ist eine Sprache über der Individuenmenge der rationalen Zahlen $\mathbb{Q}$ mit folgender Signatur.

   - Prädikate $\mathcal{P} = \{=\}$.

     Das Prädikat $=$ hat die Stelligkeit 2 und es gilt $X = Y$ ist genau dann *wahr*, wenn X und Y das gleiche Objekt aus $\mathbb{Q}$ bezeichnen.
   - Funktoren $\mathcal{F} = \emptyset$.

   Geben Sie den Wahrheitswert der folgenden Formeln an und begründen Sie jeweils ihre Antwort.

   a) $\forall X : \big(\exists Y : (X = Y)\big)$
   (4 Punkte)

b) $\forall X : \big(\forall Y : (X = Y)\big)$
(4 Punkte)

c) $\exists X : \big(\forall Y : (X = Y)\big)$
(4 Punkte)

d) $\exists X : \big(\exists Y : (X = Y)\big)$
(4 Punkte)

2. Gegeben ist eine Sprache über der Individuenmenge der natürlichen Zahlen (inklusive 0) $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ mit folgender Signatur. Die arithmetischen Operationen über $\mathbb{N}_0$ sind wohlbekannt.

   - Prädikate $\mathcal{P} = \{=\}$.

     Das Prädikat = hat die Stelligkeit 2 und es gilt $X = Y$ ist genau dann *wahr*, wenn X und Y das gleiche Objekt aus $\mathbb{N}_0$ bezeichnen.

   - Funktoren $\mathcal{F} = \{+, \text{dreieck}, \text{quadrat}\}$.

     Der Funktor + hat die Stelligkeit 2 und entspricht der Addition in $\mathbb{N}_0$.

     Der Funktor dreieck hat die Stelligkeit 1. Es gilt $\text{dreieck}(0) = 0$ und sonst $\text{dreieck}(X) = \sum_{i=1}^{X} i$.

     Der Funktor quadrat hat die Stelligkeit 1. Es gilt $\text{quadrat}(0) = 0$ und sonst $\text{quadrat}(X) = \sum_{i=1}^{X}(2i - 1)$.

   Geben Sie den Wahrheitswert der folgenden Formeln an. Mit Begründung.

   a) $\exists X : \big(\exists Y : \big((\neg(X = Y)) \wedge (\text{quadrat}(X) = \text{quadrat}(Y))\big)\big)$
   (5 Punkte)

   b) $\forall X : \big(\exists Y : \big((\neg(X = Y)) \wedge (\text{quadrat}(X) = \text{quadrat}(Y))\big)\big)$
   (5 Punkte)

   c) $\forall X : \big(\exists Y : \big((\text{dreieck}(X) + \text{dreieck}(Y)) = \text{quadrat}(X)\big)\big)$
   (5 Punkte)

   d) $\exists X \not\otimes : \Big(\forall Y : \Big(\neg\big((\text{dreieck}(X) + \text{dreieck}(Y)) = \text{quadrat}(X)\big)\Big)\Big)$
   (5 Punkte)

Praktische Übung

Abgabe der Prüfsumme bis Di., 13.06., 8 Uhr
Testat Do., 15.06. bis Do., 22.06.

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen.

Am **Mo., 12.06.**, 10-12 Uhr online (3 Tutor*innen) und 18-20 Uhr in Präsenz (2 Tutor*innen), finden **keine Testate** statt. Diese Rechnerübungen sind ausschließlich **für Fragen** reserviert.

Abgabe der Prüfsumme

- Siehe vorherige Übungen.
- Übermitteln Sie die Prüfsumme mit dem Test *GdPI 07 - Testat*.

Vorbereitung

1. Für diese praktische Übung ist das Kapitel *7. Python* der Vorlesung erforderlich.
2. Lesen Sie in der *Python documentation* im *Tutorial*, Kapitel *4. More Control Flow Tools* (`https://docs.python.org/3/tutorial/controlflow.html`), Abschnitt *4.4. break and continue Statements, and else Clauses on Loops*.

Aufgabe 1 – 25 Punkte

Nicht-rekursiver Parser

Betrachten Sie folgende Parse-Tabelle, das Startsymbol ist 0.

	`id`	`(`	`)`	`:=`	`+`	`-`	`*`	`/`	`$$`
0	1								1
1	2 1								ε
2	`id := 3`								
3	5 4	5 4							
4	ε		ε		8 5 4	8 5 4			ε
5	7 6	7 6							
6	ε		ε		ε	ε	9 7 6	9 7 6	ε
7	`id`	`( 3 )`							
8					`+`	`-`			
9							`*`	`/`	

Erstellen Sie ein Python-Skript für einen nicht-rekursiver Parser, der die vorstehende Parse-Tabelle benutzt.

Allgemeine Anforderungen

1. Die Terminale werden als Strings, die Nichtterminale als ganze Zahlen codiert.
2. Kommentieren Sie Ihre Code.
3. Bei einem Fehler wird das Skript mit einer Fehlermeldung abgebrochen.

```
print("ERROR: ...")
exit()
```

4. Obwohl Python weitreichende Möglichkeiten zur Strukturierung und Fehlerbehandlung bietet, reicht für diese Übung ein einfaches Skript, das die Aufgabenstellung (gradlinig) abarbeitet.
5. Ein interaktives Einlesen der Eingabe ist nicht nötig.
6. Testen Sie Ihr Skript mit der Eingabe **`id:=id*(id+id)`** und einer Eingabe, die nicht zur Sprache gehört.

(5 Punkte)

Scannen der Eingabe

1. Die Eingabe ist ein String.
2. Die Menge der Terminale ist ein Tupel.
3. Mit Hilfe der Menge der Terminale wird aus der Eingabe eine Liste von Terminalen erzeugt, an die als letztes Element die Markierung für das Ende der Eingabe angehängt wird.

(5 Punkte)

Parsen der Terminalliste

1. Die Parse-Tabelle ist ein Tupel, das für jede Zeile der Parse-Tabelle ein Dictionary enthält.

   Die Dictionaries haben nur Einträge für die Zellen der Parse-Tabelle, in denen die rechte Seite einer Produktion steht.

   Die Schlüssel der Dictionaries sind Terminale, die Werte sind Tupel von Nichterminalen und Terminalen. Ein leeres Tupel `()` repräsentiert ε.

2. Der Stapel des Parsers ist eine Liste von Terminalen und Nichtterminalen, die während des Parsens verändert, nicht kopiert, wird.

   Sie können auf ein Zeichen verzichten, das den leeren Stapel anzeigt.

3. Für das Parsen bietet sich eine **`while`**-Schleife an, die läuft, solange der Stapel nicht leer ist.

4. Eine Liste, die die Elemente einer Liste/eines Tupels **`xs`** in umgekehrter Reihenfolge enthält, kann man mit **`xs[::-1]`** erzeugen.

(15 Punkte)