

Übungsblatt 2 (24.10.2013)

Formale Konzepte, FC++

In dieser Übung:

- ✓ Wenden Sie die EBNF an
- ✓ Arbeiten Sie mit Alphabeten
- ✓ Führen Sie einen Induktionsbeweis durch
- ✓ Planen, implementieren und testen Sie ein Programm mit funktionaler Programmierung

Hinweis

Bitte geben Sie Ihre Lösung als pdf oder Textdokument (mit separaten Bilddateien) in Moodle ab. Außerdem ist es erforderlich, in der Lösung Ihre 1-2 Teammitglieder sowie das Aufgabenblatt und die jeweilige Aufgabe anzugeben. Bitte beachten Sie, dass sich **alle** Mitglieder eines Teams **in derselben Übungsgruppe befinden** müssen.

Eine Abgabe im Team ist **nicht** möglich bei Studierenden, die nur eine Bestätigung über die Übungsteilnahme benötigen und nicht an der Klausur teilnehmen.

Speichern Sie bitte Ihre Lösung bis **Freitag, den 01.11.2013, um 12.00 Uhr** in Moodle.

Aufgabe 1: EBNF zur Wohnungssuche

Punkte: 8

Team: Ja

Stellen Sie sich vor, Sie befinden sich gerade auf Wohnungssuche. Sie erhalten von Maklerin Klara, die auch Informatik studiert hat, eine SMS mit Angeboten. Diese Angebote sind in EBNF ausgedrückt. Definieren Sie eine Sprache, die die korrekten Mietwohnungsangebote beschreibt. Ihnen ist die Reihenfolge der Zimmer wichtig: mindestens ein Bad, ein Arbeitszimmer und kein, eines oder mehrere Wohnzimmer. Idealerweise hat die Wohnung einen Balkon, der allerdings kein Muss ist. Bad, Arbeitszimmer und Wohnzimmer müssen als Räume mindestens ein Fenster besitzen und klassifizierbar sein in klein ($<10\text{m}^2$), mittelgroß (10 bis 50m^2) und luxuriös ($>50\text{m}^2$). Außerdem kann in einem Raum öffentliches WLAN der Stadt verfügbar sein. Alle drei Räume haben zu guter Letzt natürlich individuelle Elemente. Das Bad hat entweder eine Dusche oder Badewanne. Das Arbeitszimmer könnte Telefonleitung oder LAN-Verkabelung bieten. Das Wohnzimmer könnte einen offenen Kamin und Parkett bieten.

Ein Beispiel für ein Angebot gemäß der Aufgabe ist etwa:

Bad Badewanne Fenster klein Arbeitszimmer Fenster mittelgroß Wohnzimmer Fenster klein Balkon

1. Geben Sie die Regeln für Ihre korrekte Mietwohnung in der erweiterten Backus-Naur-Form (EBNF) an. (4 Punkte)

Hinweis: Beenden Sie die Beschreibung aller Räume mit den Eigenschaften, die alle Räume haben (also ::= ... <Raum>).

Ergebnis: Definition in der EBNF (pdf)

2. Sie haben drei SMS erhalten. Wird Ihr Wunsch erfüllt, d.h. sind folgende Mietwohnungsangebote korrekt? Wenn ein Angebot nicht korrekt ist, nennen Sie alle Gründe. (3 Punkte)
 - Bad Dusche Fenster klein Arbeitszimmer Fenster mittelgroß Wohnzimmer offener Kamin Fenster Fenster luxuriös WLAN Balkon
 - Bad WLAN Badewanne Fenster mittelgroß Arbeitszimmer Telefonleitung gute Aussicht Fenster mittelgroß Wohnzimmer Parkett mittelgroß
 - Bad Badewanne Fenster klein Arbeitszimmer LAN-Verkabelung Fenster Fenster Wohnzimmer Parkett Fenster Wohnzimmer mittelgroß Fenster Balkon

Ergebnis: Beurteilungen (korrekt/nicht korrekt) und ggf. die jeweilige Begründung (pdf)

3. Wie sieht die kleinste korrekte Mietwohnung aus? Entscheiden Sie sich bei mehreren Optionen immer für die erste Option. (1 Punkt)


Ergebnis: Beschreibung der Mietwohnung in der Form von 1.2 (pdf)

Aufgabe 2: Alphabet und Wörter	
Punkte: 8	Team: Ja

Es sei $A = \{\$, x, y\}$ ein Alphabet. Wir betrachten nun eine Teilmenge $T \subseteq A^*$. Beispiele für Wörter in T sind:

$\$ \$ \$ x \$ y \$$ $\$ \$ \$ \$ \$ \$ \$ x \$ \$ \$ y \$ \$$ $\$ \$ \$ \$ \$ \$ x \$ \$ y \$ \$ \$$

Im Allgemeinen haben die Wörter die Gestalt:

$\$ \dots \$ x \$ \dots \$ y \$ \dots \$$

 Anzahl der $\$$ -Zeichen: $2n+m$ n m

Dabei ist die Anzahl der $\$$ -Zeichen vor dem x gleich der Anzahl der $\$$ -Zeichen zwischen x und y multipliziert mit 2, plus die Anzahl der $\$$ -Zeichen nach dem y . Sie können x als „gleich 2 mal“ und y als „plus“ interpretieren und die Ketten aus $\$$ -Zeichen als Kodierung für natürliche Zahlen.

1. Entwerfen Sie, analog zum WAS-System aus der Vorlesung, einen Satz von (möglichst wenigen) Regeln, die die obigen Worte des Systems erzeugen. (8 Punkte)

Ergebnis: Das „Basis“-Wort des Systems, die Regeln des Systems, sowie die Ableitungen für die 3 gegebenen Beispielworte (pdf)

Aufgabe 3: Vollständige Induktion

Punkte: 8

Team: Ja

Wir betrachten die Sprache WAS. Allerdings stehen jetzt nur die ersten drei Regeln zur Verfügung. Wir nennen die Sprache deshalb WAS*:

- wa ist ein gültiges Wort der Sprache WAS*.
- Wenn Xa ein gültiges Wort ist, dann auch Xas.
- Wenn wX ein gültiges Wort ist, dann auch wXX.

1. In dieser Aufgabe sollen Sie beweisen, dass in allen Worten der Sprache WAS* mehr oder gleich viele 'a' als 's' vorkommen. Dies geht mit einer kleinen Hilfsaussage durch vollständige Induktion.

Dabei sei für ein Wort Y aus WAS* und ein Zeichen z Anzahl(Y, z) die Anzahl der Vorkommen des Zeichens z in Y.

Beweisen Sie mit vollständiger Induktion: Für alle $X \in \text{WAS}^*$ gilt $\text{Anzahl}(X, a) \geq \text{Anzahl}(X, s)$ **und** für alle $Xa \in \text{WAS}^*$ gilt $\text{Anzahl}(Xa, s) = 0$.

Ergebnis: Beweis (pdf)

Aufgabe 4: Funktionales Programm zur Preisberechnung

Punkte: 16

Team: Ja

Beschreibung

Sie haben die günstige Gelegenheit erhalten, Ihr Budget aufzubessern: Sie können für einen Mobilfunkkartenhändler ein FC++-Programm entwickeln und testen. Der Händler würde gerne mit diesem Programm den Preis für die von ihm ausgegebenen Prepaid-Karten berechnen. Er bietet einen (sehr einfachen) Tarif CHEAP an und verwendet zur Preisberechnung die gewünschte Anzahl von Gesprächsminuten und SMS im Monat.

Eingaben

Ihr Programm enthält als Eingabe die Anzahl Gesprächsminuten und die Anzahl SMS.

Ausgaben

Ihr Programm liefert als Ausgabe den Preis bei gültigen Eingaben oder Fehlermeldungen bei ungültigen Eingaben.

Vorbedingung

Die Anzahl Gesprächsminuten und die Anzahl SMS sind größer gleich 0 und beides sind ganze Zahlen.

Nachbedingung

Der errechnete Preis hängt primär von der Anzahl der SMS ab, die verbrauchten Minuten werden dann pauschal abgerechnet.

1. Die Nutzung bis 249 SMS und
 - a) 0-99min kostet 10 Euro,
 - b) 100-299min kostet 15 Euro und
 - c) ab 300min kostet 20 Euro.

2. Die Nutzung ab 250 SMS und

- a) 0-199min kostet 25 Euro,
- b) 200-399min kostet 35 Euro und
- c) ab 400min kostet 45 Euro.

1. Erstellen Sie für den Black-Box-Test die gültigen und ungültigen Äquivalenzklassen für die Eingabe. Geben Sie dann an, welche Kombinationen getestet werden müssen, um alle wichtigen Fälle abzudecken. Wählen Sie für die jeweiligen Kombinationen entsprechende Repräsentanten aus und geben Sie die erwarteten Ergebnisse an. (8 Punkte)

Ergebnis: Äquivalenzklassen und Repräsentanten (pdf)

2. Überlegen Sie sich den Algorithmus Ihres Programms und schreiben Sie es in FC++. Dafür benötigen Sie Funktionen zum Einlesen der Werte, zum Berechnen der Daten sowie zur Ausgabe. (6 Punkte)

Ergebnis: gezipptes Code::Blocks-Projekt (zip mit komplettem Code::Blocks-Projektordner inklusive der Quellcode-Dateien)

3. Testen Sie Ihr Programm lokal auf Ihrem System, indem Sie die Repräsentanten aus Aufgabe a) als Eingaben anwenden und die Programmausgabe mit den erwarteten Ergebnissen vergleichen. (2 Punkte)

Ergebnis: Ausgabe der Testläufe (jeweils Ein- und Ausgabe) Ihres Programms (pdf oder Bilddatei)

Hinweis

Bitte geben Sie Ihren Quellcode wie folgt ab: Packen Sie den zugehörigen Projektordner aus Code::Blocks in ein Zip-Archiv und laden Sie dieses in Moodle hoch.

Sollten dabei Probleme oder Fragen auftreten, nutzen Sie bitte die Diskussionsforen in Moodle oder wenden sich in der Übung an Ihren Tutor. Für dringende Fragen finden Sie die Kontaktdaten ab dem 24.10. im Diskussionsforum zur Übungsgruppe.