

Informatik 1
WS 2006/07

Übungsblatt 1: Eigenschaften von Algorithmen, Formatierung, Syntax, Dateiformat

Besprechung: 30.10.–03.11.2006

Abgabe aller mit **Hausaufgabe** markierten Aufgaben bis Freitag, 27.10.2006, 18:00 Uhr

Vorbereitung: Sie werden im Lauf des Semesters für diese Vorlesung (und vielleicht auch für andere) viele Dateien herunterladen. Wir empfehlen Ihnen, in Ihrem Home-Verzeichnis von Anfang an geeignet strukturierte Verzeichnisse (= Ordner, *directory*, *folder*) dafür vorzubereiten, zum Beispiel ein Verzeichnis `Informatik1` anzulegen und darin ein Unterverzeichnis `Übungen` und in diesem wiederum Unterverzeichnisse für die einzelnen Übungen (z.B. `Übungsblatt01` bis `Übungsblatt12`). Sie können natürlich auch kürzere Namen dafür wählen (z.B. `Info1/Übungen/u01`). Für jedes Übungsblatt legen Sie bitte ein Verzeichnis an, in dem Sie die Lösungs-Dateien zu den einzelnen Aufgaben ablegen. Haben Sie alle Lösungen, die Sie abgeben möchten, fertig, dann erstellen Sie ein `zip`-Archiv aus diesem Verzeichnis. Wenn Sie mit einer `shell` im Verzeichnis des aktuellen Übungsblattes sind (z.B. `Info1/Übungen/u01`), erzeugen Sie mit dem Befehl `zip loesung01.zip loesung01/*` ein `zip`-Archiv namens `loesung01.zip`, das alle Dateien aus dem Lösungsverzeichnis `loesung01` enthält.

Unter `http://www.pst.fh.uni-wuerzburg/` können Sie jeweils das `zip`-Archiv mit Ihren gesammelten Lösungen für jedes Übungsblatt abgeben. Damit die Korrektoren Ihre Lösungen der richtigen Aufgabe problemlos zuordnen können, achten Sie bitte darauf, dass Ihre Dateien die richtigen Namen haben.

Achten Sie bitte in Ihrem eigenen Interesse darauf, die Zugriffsberechtigungen für ihre Verzeichnisse so zu setzen, dass nur der Eigentümer (also Sie selbst) darin lesen oder schreiben darf, aber weder die `Unix-Gruppe` (das wäre in diesem Fall praktisch jeder, der eine CIP-Kennung hat) noch „Sonstige“. Wenn Sie Ihre Verzeichnisse mit Konqueror bearbeiten, finden Sie im Menü „Bearbeiten“ einen Menüpunkt „Eigenschaften“, der Ihnen unter anderem die Möglichkeit gibt, die Berechtigungen zu überprüfen und zu ändern.

Aufgabe 1-1 Eigenschaften von Algorithmen (Hausaufgabe)

Stellen Sie sich einen Karteikasten vor, in dem Sie nach allen Karteikarten mit einem bestimmten Titel suchen.

Suchverfahren 1:

- (1) Sie greifen zufällig eine Karte heraus.
- (2) Hat diese den gewünschten Titel, so sind Sie fertig, ansonsten legen Sie sie zurück und wiederholen das Verfahren ab Schritt 1.

Suchverfahren 2:

- (1) Sie greifen zufällig eine Karte heraus.
- (2) Trägt diese den gewünschten Titel, so legen Sie die Karte auf einen Ergebnisstapel, ansonsten legen Sie sie auf die Seite.
- (3) Ist der Kasten leer, so beenden Sie die Suche.
- (4) Ansonsten wiederholen Sie das Verfahren ab Schritt 1.

Suchverfahren 3:

- (1) Sie greifen die Karte heraus, die in der Mitte liegt.
- (2) Trägt diese den gewünschten Titel, so sind Sie fertig, ansonsten rufen Sie nach Unterstützung (es stehen mehr Helfer als Karteikarten zur Verfügung).
- (3) In der ersten Hälfte des Kastens suchen Sie weiter beginnend mit Schritt 1, in der zweiten Hälfte sucht ihre Unterstützung nach dem selben Verfahren auch beginnend mit Schritt 1.
- (4) Das Verfahren ist beendet, wenn jeder entweder eine Karte gefunden hat oder nur mehr leere Kartenabschnitte hat.

Suchverfahren 4:

- (1) Sie greifen die erste Karte.
- (2) Trägt diese den gewünschten Titel, so legen Sie diese auf den Ergebnisstapel.
- (3) Ist der Kasten noch nicht leer, nehmen Sie die nächste Karte und setzen die Suche bei Schritt (2) fort.

Suchverfahren 5: (Vorbedingung: Karteikasten ist lexikographisch sortiert)

- (1) Sie greifen die Karte, die in der Mitte liegt.
- (2) Trägt diese den gewünschten Titel, legen Sie die Karte auf den Ergebnisstapel und gehen zu Schritt (6), ansonsten:
 - (3) Ist der gesuchte Titel alphabetisch vor dem Titel auf der ausgewählten Karte, so suchen Sie nur noch in der ersten Hälfte nach dem gleichen Verfahren ab Schritt 1 weiter.
 - (4) Ist der gesuchte Titel alphabetisch hinter dem Titel auf der ausgewählten Karte, so suchen Sie nur noch in der zweiten Hälfte nach dem gleichen Verfahren ab Schritt 1 weiter.
 - (5) Das Verfahren ist erfolglos beendet, wenn der letzte Kartenabschnitt leer ist.
- (6) Untersuchen Sie die nachfolgende Karte auf den gesuchten Titel. Falls die Karte ebenfalls diesen Titel trägt, legen Sie sie auf den Ergebnisstapel. Und wiederholen Schritt (6).
- (7) Untersuchen Sie die vorhergehende Karte auf den gesuchten Titel. Falls die Karte ebenfalls diesen Titel trägt, legen Sie sie auf den Ergebnisstapel. Und wiederholen Schritt (7).
- (8) Das Verfahren ist mit allen Ergebnissen beendet, wenn keine angrenzende Karte mehr auftritt, die den gesuchten Titel trägt.

Vergleichen Sie die oben angeführten Suchverfahren. Welches Verfahren erfüllt welche der folgenden Eigenschaften?

- (a) terminierend
- (b) deterministisch
- (c) determiniert
- (d) rekursiv
- (e) partiell korrekt
- (f) total korrekt

Begründen Sie Ihre Entscheidung kurz und geben Sie Ihre Lösung in einer Text-Datei 1-1.txt ab.

Aufgabe 1-2 *Formatierung (Hausaufgabe)*

```
fun vorzeichen (
  x
) = if
  0 then ~ x
| else if
  x =
  0 then 0 else
  1; fun
  betrag
  = if
  (x)
  > 0 then x else if x
< 0 then let
  val minusx
  = ~x
  in
  end
  end
  end
  minusx
```

In der WWW-Seite der Vorlesung finden Sie zu diesem Übungsblatt unter anderem die Datei 1-2-.sml mit obigem Inhalt. Kopieren Sie sich die Datei in Ihr Unterverzeichnis für dieses Übungsblatt.

Laden Sie diese Datei in SML. SML wird kein Problem mit der Datei haben, aber sie ist für Menschen völlig unleserlich.

Verändern Sie die Leerzeichen und Zeilenumbrüche, so dass die Definitionen gut lesbar eingetrickt sind, und geben Sie die modifizierte Datei 1-2-.sml als Lösung ab.

Aufgabe 1-3 *Syntaxfehler (Hausaufgabe)*

```
fun quadrat (x)
  = x * x;
fun summe_quadratrate (x,y)
  : quadrat (x) + quadrat (y);
fun pythagorasrelation (a,b,c)
  = ( summe_quadratrate (a,b) = quadrat (c) );
fun pythagoraszahl (z)
  = Pythagorasrelation (z-1, z, z+1);
val ist_4_pythagoraszahl
  = pythagoraszahl (4);
```

Kopieren Sie die Datei 1-3-.sml in Ihr Unterverzeichnis für dieses Übungsblatt. Sie hat den obigen Inhalt. Laden Sie diese Datei in SML. SML wird mit Fehlermeldungen reagieren.

Hinweis: Wenn SML Sie mit Fehlermeldungen konfrontiert, verfallen Sie nicht in Panik! Fehlermeldungen sind Ihre Freunde. Sie enthalten meistens sinnvolle Hinweise, wo in Ihrem Code etwas nicht korrekt ist.

Beheben Sie die drei Tippfehler und geben Sie die korrigierte Datei 1-3-.sml als Lösung ab.

Bitte beachten Sie: Wenn Sie eine sml-Datei als Lösung abgeben, sollte diese Datei von SML ohne Fehler geladen werden können, damit Ihre Korrekturen nicht unnötig viel Aufwand haben. (Dann haben sie nämlich mehr Zeit, Ihnen wertvolle Tipps zu geben.) Sollen Sie sich einmal bei einer Abgabe nicht sicher sein, wie Sie Syntaxfehler vermeiden, dann kommentieren Sie eine unfertige Definition einfach aus.

Aufgabe 1-4 *Dateiformate (Hausaufgabe)*

```
fun gegenteil ("Weich") = "Hart"
| gegenteil ("Hart") = "Weich"
| gegenteil ("Wort") = "Satz"
| gegenteil ("Satz") = "Wort"
| gegenteil ("Winzig") = "Riesig"
| gegenteil ("Riesig") = "Winzig"
| gegenteil (x) = "gegenteil (" ^ x ^ ")";

fun englisch ("Weich") = "Soft"
| englisch ("Hart") = "Hard"
| englisch ("Wort") = "Word"
| englisch ("Satz") = "Sentence"
| englisch ("Winzig") = "Micro"
| englisch ("Riesig") = "Macro"
| englisch (x) = "englisch (" ^ x ^ ")";

fun deutsch (x) = x

fun f (sprache, x, y, z) =
  sprache (gegenteil (x) ^ sprache (gegenteil (y)) ^ sprache (gegenteil (z)))

val nicht_akzeptiert = f (deutsch, "Riesig", "Hart", "Satz")
val not_accepted = f (englisch, "Riesig", "Hart", "Satz")
```

(a) Die Datei 1-4a wird auf manchen Plattformen so angezeigt, als hätte sie den obigen Inhalt. Kopieren Sie sie in Ihr Verzeichnis für dieses Übungsblatt. Wenn Sie versuchen, die Datei in SML zu laden, wird SML mit Fehlermeldungen reagieren.

Um den Grund zu verstehen, lohnt es sich, die Datei einmal mit irgend einem Editor unter Linux anzuschauen: wahrscheinlich werden Sie dann keinen Text erkennen, sondern eine Ansammlung ungewöhnlicher Zeichen. Das liegt daran, dass der Inhalt der Datei kein Text ist, sondern eine (proprietäre) Binärcodierung. Das System, mit dem die Datei erstellt wurde, wandelt den Binärcode in eine graphische Darstellung um, die menschlichen Lesern Text mit verschiedenen Schriftarten, Schriftgraden, Schriftschnitten usw. anzeigt, aber der eigentliche Inhalt der Datei ist eben Binärcode.

Geben Sie eine neue Datei 1-4a-.sml ab, die den obigen Inhalt hat, und die vom Abgabesystem akzeptiert wird.

(b) Welche Werte haben die Variablen nicht_akzeptiert und not_accepted? Schreiben Sie die Antwort in eine neue Datei 1-4b.txt und geben Sie diese Datei als Lösung ab.