LEHRSTUHL FÜR INFORMATIK III
RWTH Aachen
Univ.-Prof. Dr.-Ing. Manfred Nagl,
Christian Fuß, Ulrike Ranger

Die SWT-Programmiersprache Ada 95
Übung zur Vorlesung
SS 2005
28.04.2005

Übungsblatt 03

Aufgabe 7 Ableitungsbaum (6 Punkte)

Machen Sie sich mit der Ada EBNF vertraut und geben Sie für folgenden Teil des CAN-Treibers aus der letzten Übung den Ableitungsbaum gemäß der Ada EBNF an.

```
WITH ada.text_io;

PACKAGE BODY can_driver IS

  FUNCTION transmit (
        msg : IN     can_msg_transmit_struct)
    RETURN integer IS
  BEGIN
    dbg_message(msg);
    RETURN k_can_tx_ok;
  END transmit;

END can_driver;
```

Verwenden Sie dazu die **XML-Notation**. Folgend ist der Anfang der Lösung dargestellt, an dem die gewünschte Struktur zu erkennen ist. Alle Nicht-Terminalsymbole sind XML-Tags, alle Terminale werden als Inhalt der Tags dargestellt. Die Syntax findet sich im Anhang 6 des Ada-Buchs oder unter http://www.seas.gwu.edu/~adagroup/ada95-syntax/. Ich empfehle die Website, weil dort die Nicht-Terminale per Hyperlink verbunden sind.

```
<compilation_unit>
  <context_clause>
    <with_clause>
      WITH
      <library_unit_name>ada.text_io</library_unit_name>
      ;
    </with_clause>
  </context_clause>
  <library_item>

        ...

  </library_item>
</compilation_unit>
```

Um etwas Arbeit zu sparen, hören Sie mit dem Parsen auf, wenn das Nicht-Terminalsymbol auf **`name`**, **`identifier`** oder **`expression`** endet (z.B. für **`<library_unit_name>`**).

Aufgabe 8 EBNF Varianten (3 Punkte)

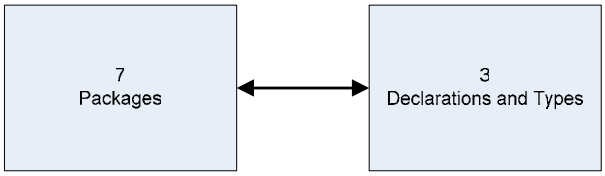
Die Ada EBNF enthält keine Meta-Zeichen für Gruppierung (wie z.B. die EBNF von Modula). Wir führen die Meta-Symbole **(** und **)** für die Gruppierung ein. Suchen Sie in der Ada EBNF eine Regel, die sich dann kompakter schreiben lässt und formulieren Sie sie um. Geben Sie die Regel vorher/nachher an und diskutieren Sie, was sich geändert hat.

Nachfolgend ein Beispiel für die Gruppierung aus der Modula-3 Syntax (Regeln werden mit . abgeschlossen, linke und rechte Seite durch = getrennt, Terminalsymbole mit "T" gekennzeichnet):

```
Exponent = ("E"|"D"|"X"|"e"|"d"|"x") ["+"|"-"] Digit {Digit}.
```

Aufgabe 9 Beziehungen in der Ada EBNF (6 Punkte)

Die Ada EBNF ist in Abschnitte aufgeteilt, die zusammengehörige syntaktische Konstrukte beschreiben. Zwischen den Konstrukten innerhalb eines Abschnitts bestehen zahlreiche Verbindungen, aber auch zwischen den Abschnitten bestehen Beziehungen. Tragen Sie in einem Diagramm, ähnlich dem folgenden Beispiel, alle direkten Referenzen aus Syntaxregeln zwischen den Kapiteln 2 bis 13 auf.



Ein Pfeil geht von Kapitel 3: *Declarations and Types* nach Kapitel 7: *Packages*, weil z.B. Regel 3.11 das Nicht-Terminal **package_body** aus Regel 7.2 referenziert:

```
3.11: proper_body ::=
        subprogram_body
      | package_body                    ← aus 7.2: Package Bodies
      | task_body
      | protected_body
```

Umgekehrt geht ein Pfeil von Kapitel 7 nach Kapitel 3, weil z.B. Regel 7.1 *Package Specifications and Declarations* **basic_daclarative_item** aus Regel 3.11 referenziert:

```
7.2: package_specification ::=
       PACKAGE defining_program_unit_name IS
       {basic_declarative_item}          ← aus 3.11: Declarative Parts
  [private {basic_declarative_item}]
  END [[parent_unit_name.]identifier]
```

***Abgabe**: Fr 6.05.2005 (Donnerstag ist Christi Himmelfahrt)*

Sie können Ihre Lösung zum obigen Termin in elektronischer Form per eMail an ada@i3.informatik.rwth-aachen.de abgeben.

Bitte vermerken Sie in jedem Fall die Namen und Matrikelnummern aller beteiligten Personen (maximal 3), beim Versenden per eMail auch alle eMail-Adressen. Abgaben in elektronischer Form können ausschließlich in den Dateiformaten *Plain-Text oder PDF* erfolgen.