Das Babel-System
Ein HPSG-Fragment für das Deutsche, ein Parser und eine Dialogkomponente
Benutzerhandbuch

Stefan Müller*
DFKI GmbH
Fachbereich Sprachtechnologie
Stulsatzenhausweg 3
66123 Saarbrücken

3. Juni 1998

Zusammenfassung

Dieses Dokument enthält eine Kurzbeschreibung der Befehle und Schalter, die im Babel-System (Müller(1996)) verwendet werden können.

*e-mail: `Stefan.Mueller@dfki.de`

Inhaltsverzeichnis

1 Installation und Start des Programms

Die Laufzeitversion des Babel-Systems findet man unter `http://www.dfki.de/~stefan/Babel/Runtime/` in der Datei `swi-babel-runtime.tgz`. Das File wird normalerweise vom WWW-Client ausgepackt und kann dann in ein Verzeichnis Ihrer Wahl gespeichert werden. Mit

```
tar -xf swi-babel-runtime.tar
```

wird das tar-File ausgepackt.

Wird das File als `.tgz`-File gespeichert, kann man es mit

```
tar -xzf swi-babel-runtime.tgz
```

auspacken, wenn GNU-tar installiert ist.

Das Programm setzt eine installierte Version von SWI-Prolog voraus (größer Version 2.9.9). SWI-Prolog ist eine frei verfügbare Prolog-Version, die von Jan Wielemaker entwickelt und gewartet wird. Die Quellen sind unter `http://www.swi.psy.uva.nl/usr/jan/SWI-Prolog.html` zu bekommen.

Mit Hilfe von `chpl`, das in der SWI-Distribution enhalten ist, muß der Pfad zu SWI-Prolog im Babel-Programm geändert werden. Das geschieht mit dem folgenden Aufruf.

```
chpl -e /der/pfad/zu/bin/pl babel
```

Sie können das Babel-System starten, indem Sie `babel` aufrufen. Das Programm benötigt einen Moment, um alle Bibliotheken zu laden. Es meldet sich wie folgt, wenn es bereit ist:

```
Babel 1.5

Autor:   Stefan Mueller  (Stefan.Mueller@dfki.de)

Version: 05.11.97

Geben Sie 'hilfe.' ein, um eine Auflistung aller Befehle
zu erhalten.

?-
```

Zur besseren Ausgabe von Merkmalstrukturen wird die Installation von Fegramed empfohlen. Fegramed findet man unter `ftp://lt-ftp.dfki.uni-sb.de/pub/fegramed/`.

2 Verlassen des Programms

Das Programm kann mit Control-D verlassen werden. Alternativ kann man `halt.` eingeben.

3 Aufruf von Beispielen

3.1 Anzeigen aller Syntax-Beispielsätze

Mit dem Aufruf von

```
saetze.
```

kann man sich alle vorhandenen Beispielsätze anzeigen lassen. Die Beispielsätze sind nach Phänomengruppen sortiert. Testsätze zum Testen von Relativsätzen gehören der Gruppe `rs` an. Innerhalb einer Gruppe sind den Testsätzen eindeutige Nummern zugeordnet.

```
rs 1 Der Mann, der Maria liebt, verachtet sich.
rs 2 Der Mann, der sich verachtet, liebt Maria.
rs 3 Der Mann, der Maria liebt, kuesst sie.
rs 4 Der Mann, der sie kuesst, liebt Maria.
rs 5 Den Mann, den Maria liebt, verachtet Karl.
rs 6 Maria verachtet den Mann, der sie liebt.
rs 7 Die Jacke, die er liebt, faerbt Karl um.
```

Ein '`*`' vor einem Satz bedeutet, daß er ungrammatisch ist.

3.2 Syntaxtest eines einzelnen Beispiels

Den Beispielsatz 5 aus der Gruppe der Relativsatz-Beispiele kann man mit

```
t(rs,5).
```

aufrufen.

Pro gefundener Lösung wird ein Syntaxbaum, eine logische Form oder ein `yes` ausgegeben.

3.3 Syntaxtest aller Beispiele einer Gruppe

Mit dem Aufruf von

```
t(rs,_).
```

bekommt man alle Analysen zu den Beispielen aus der Gruppe `rs`.

Die Ausgabe kann mit Control-C abgebrochen werden.

3.4 Syntaxtest aller Beispiele

Ruft man

```
test.
```

auf, so werden alle Testbeispiele ausgeführt. Pro gefundener Lösung wird ein `yes` ausgegeben.

Die Ausgabe kann mit Control-C abgebrochen werden. Nach einem Abbruch muß bei Prolog-Versionen unter 2.9.9

```
reset.
```

eingegeben werden.

4 Anzeigen einzelner Wörter und aller dem System bekannten Vollformen

Mit

```
wort(test).
```

werden alle Lexikoneinträge für das Wort *Test* ausgegeben.

Gibt man

```
woerter.
```

ein, so wird eine Liste der im Lexikon enthaltenen Vollformen ausgegeben.

Die Ausgabe kann mit Control-C abgebrochen werden.

5 Interaktive Eingabe von Sätzen

Nachdem

```
go.
```

eingegeben wurde, erscheint ein Prompt, und es können Sätze, die aus dem System bekannten Wörtern bestehen, eingegeben werden.

```
?- go.
>>>   notree.
>>>   Stefan hat in der Vorlesung geschlafen, die ich
      gehalten habe.
..........
yes

>>>   ende.

Yes
?-
```

Groß-/Kleinschreibung und Kommata werden ignoriert.

Eine Übersicht der durch die Grammatik beschriebenen Phänomene findet man in (Müller(1996)).

Je nach Einstellung der Ausgabeschalter (siehe Abschnitt 7) werden Merkmalstrukturen, logische Repräsentationen, Syntaxbäume oder beliebige Kombinationen dieser drei Ausgaben ausgegeben.

6 Interaktive Eingabe von Sätzen mit Semantikkomponente

Nachdem

```
disc.
```

eingegeben wurde, erscheint ein Prompt, und es können Sätze, die aus dem System bekannten Wörtern bestehen, eingegeben werden.

```
>>>   Karl schlaeft.
..
Verstanden!
>>>   Wer schlaeft?
..
schlafen(some_time,5)   karl

>>>   Er liebt Maria.
...
Verstanden!
>>>   Wen liebt Karl?
...
lieben(some_time,5,6) schlafen(some_time,5)   maria

>>>   Karl kennt Anna.
..
Verstanden!
>>>   Ist seine Frau schoen?
```

```
....
Ja, das ist richtig!

>>>
```

Die Semantikkomponente ist rudimentär. Die Bindungstheorie funktioniert, wie alle Bindungstheorien nur eingeschränkt.

Es gibt viele Bugs. Wenn ich mal Zeit finde und Lust habe, dann bringe ich's in Ordnung.

7 Ausgabeschalter

Sowohl in interaktiven Modi als auch vom Toplevel aus kann man die folgenden Ausgabeschalter benutzen:

```
ms.
noms.           (default)

tree.           (default)
notree.

cont.
nocont.         (default)

dom.
nodom.          (default)

chart_display.
nochart_display.

fegramed.
nofegramed.
```

`ms`	Ausgabe von Merkmalstrukturen
`tree`	Ausgabe von Syntaxbäumen
`cont`	Ausgabe logischer Formen
`dom`	Bei der Ausgabe von Merkmalstrukturen werden auch die Elemente der Wortstellungsdomänen (Töchter) mit ausgegeben.
`fegramed`	Die Ausgabe von Zeichen erfolgt mittels Fegramed (siehe Abschnitt 8). Für den Wert gibt es keinen Default. Beim Start des Systems wird nach Fegramed gesucht. Wird Fegramed im Pfad gefunden, so erfolgen die Ausgaben über Fegramed, ansonsten werden Merkmalstrukturen als ASCII ausgegeben.
`chart_display`	Beim Parsen wird ein Chart-Display (siehe Abschnitt 9 geöffnet. Für den Wert gibt es keinen Default. Beim Start des Systems wird im Pfad nach wish und nach den TCL/TK-Quellen für das Display gesucht. Werden sie gefunden, so wird ein Chart-Display geöffnet und entsprechende Ausgaben werden erzeugt.

Diese Schalter können in der Datei `.babelrc` voreingestellt werden.

8 Fegramed

Fegramed (Kiefer und Fettig(1995)) ist ein leistungsstarkes Programm zum Anzeigen (und Editieren) von Merkmalstrukturen. Die Reihenfolge der Anzeige der Merkmale kann vom Nutzer selbst bestimmt werden, Merkmale können ausgeblendet werden, Merkmalsstrukturen können temporär kollabiert werden.

Wenn der Fegramed installiert ist, wird das von Babel automatisch erkannt. Wenn dann die Option `ms` eingeschaltet ist, werden Fegramed-Merkmalstrukturen erzeugt, Fegramed wird gestartet und die Strukturen werden angezeigt.

Die Erzeugung von Fegramed-Strukturen ist nicht besonders schnell. Wenn Sie die Geschwindigkeit des Systems testen wollen, können Sie Fegramed abstellen. Mit `nofegramed.` werden statt Fegramed-Strukturen Textstrukturen ausgegeben. Mit `fegramed.` kann man Fegramed wieder einschalten.

9 Das Chart-Display

Ein am DFKI[1] in TCL/TK entwickeltes Chart-Display ist in einer angepaßten Version auch für Babel verfügbar. Mit der Option `chart_display` läßt sich die Ausgabe von Kanten wärend des Parsens einschalten.

Für Kanten und für das gesamte Chart-Display sind Aktionen definiert. Die Aktionen können nach dem Anklicken der Kanten bzw. nach dem Klicken auf eine leere Stelle im Display aus Pop-Up-Menüs ausgewählt werden.

9.1 Generelle Funktionalität

Das Chart-Display hat folgende Funktionalität:

- Ausgabe aller Zeichen, die beim Parsen erzeugt werden
- Ausgabe aller Töchter eines Zeichens
- Ausgabe aller Eltern eines Zeichens
- Ausgabe aller Zeichen, die an einem bestimmten Knoten beginnen bzw. enden
- Playback einer Analyse. Dabei werden die Kanten in derselben Reihenfolge, in der sie vom Parser gekommen sind, verzögert neugezeichnet. Der Prozeß ist unterbrechbar, so daß man jeden Parser-Zustand einzeln inspizieren kann.

9.2 Babel und das Chart-Display

Babel stellt im Zusammenhang mit dem Chart-Display noch folgende Funktionalität bereit:

- Ausgabe der Kanten, die zu einer erfolgreichen Analyse gehören
- Ausgabe einer Regel
- Ausgabe aller kontinuierlichen Kanten
- Ausgabe aller Kanten, die durch Anwendung einer bestimmten Regel entstanden sind
- Ausgabe der unmittelbaren Töchter einer Kante (nicht rekursiv, wie bei der vom Display zur verfügung gestellten Funktionalität)
- Debugging (siehe Abschnitt 9.3)

Es ist auch möglich nach einer Analyse mit abgeschaltetem Display die Chart durch den Aufruf von `drawchart` ausgeben zu lassen.

[1] http://www.dfki.de/

9.3 Debugging

Man kann eine bestimmte Kante als linke Tochter für eine Grammatikregel auswählen. Wählt man dann eine Grammatikregel aus, so wird versucht, die linke Tochter mit der Grammatikregel zu unifizieren. Gelingt die Unifikation, entsteht eine aktive Kante, die man sich ansehen kann. Schlägt die Unifikation fehl, so wird der Pfad zu der Merkmalstruktur ausgegeben, deren Unifikation fehlgeschlagen ist. Wenn Fegramed installiert ist, werden Fegramed-Fenster mit der Tochter und der Regel geöffnet. Die Stelle, an der die Unifikation fehlgeschlagen ist, ist in beiden Fenstern markiert.

Wird nach erfolgreicher Unifikation eine rechte Tochter ausgewählt, so wird diese mit der aktiven Kante unifiziert. Schlägt die Unifikation fehl, werden wieder der Pfad und die entsprechend markierten Merkmalstrukturen ausgegeben. Bei erfolgreicher Unifikation werden nacheinander für die Regel spezifizierte Constraints ausgeführt. Schlägt ein Constraint fehl, wird sein Name ausgegeben. Sind alle Constraints erfolgreich, wird das durch Regelanwendung entstehende Zeichen ausgegeben.

10 Ansehen von Teilanalysen

Nach der Analyse eines Satzes kann man sich mit `signs.` eine Liste aller erzeugten Zeichen ausgeben lassen.

```
go.
>>>   karl schlaeft.
..
yes

SPARC Ultra 2 / 300 MHz Die Zeit ohne Ausgaben: Zeit =
0,06 sec

>>>   signs.
[karl, schlaeft] , v , v
[schlaeft] , v , v
[schlaeft] , v , n
[karl, schlaeft] , v , v
[karl, schlaeft] , v , none
[schlaeft] , v , none
[schlaeft] , v , v
[karl, schlaeft] , v , none
[schlaeft] , v , n
[schlaeft] , v , none
[karl] , n , none
[karl] , n , none
>>>
```

Dabei steht an erster stelle der PHON-Wert eines Zeichens, dann kommt der MAJ-Wert (die syntaktische Kategorie) und zum Schluß der MAJ-Wert des SLASH-Elements bzw. *none*, wenn die SLASH-Liste leer ist.

Mit `psign(P)` kann man sich alle Zeichen mit dem PHON-Wert `P` ausgeben lassen.

```
>>>  psign([karl,schlaeft]).
yes
>>>
```

Bei Aufruf von `psigne(P)` werden nur die Zeichen mit leerem SLASH-Wert ausgegeben.

Literatur

Kiefer, Bernd und Fettig, Thomas. 1995. FEGRAMED—An Interactive Graphics Editor for Feature Structures. Research Report RR-95-06, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany.

Müller, Stefan. 1996. The Babel-System—An HPSG Prolog Implementation. In *Proceedings of the Fourth International Conference on the Practical Application of Prolog*, Seiten 263–277, London, http://www.dfki.de/˜stefan/Pub/babel.html. 03.06.1998.