Übungen 11: Merkmalstrukturen

Programmiertechniken in der Computerlinguistik II · Sommersemester 2005

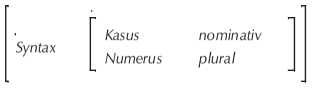
Programmtexte finden sich auf Homepage: http://www.cl.unizh.ch/siclemat/lehre/ss05/pcl2/

1. Einfache Merkmalstrukturen in Prolog

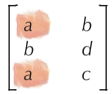
Die konkrete Syntax von einfachen Merkmalstrukturen in Prolog sei folgendermassen definiert im Backus-Naur-Format:

Merkmalstruktur	::=	`"[]"`
	|	`"["` *Merkmal* `":"` *Wert* `"|"` *Merkmalstruktur* `"]"`
Merkmal	::=	*Prolog-Atom*
Wert	::=	*Atomarer Prolog-Term (aussser `[]`)*
	|	*Merkmalstruktur*

a) Notiere den Informationsgehalt von nebenstehender Merkmalstruktur in allen möglichen Varianten der obigen Termsyntax. (D.H. schreibe sie als komplexe Liste.)



b) Definiere das Prädikat is_feature_structure/1, das genau dann gelingt, wenn das Argument von der Termsyntax her eine Merkmalstruktur ist. D.H. es akzeptiert "Merkmalstrukturen" wie nebenstehende, die semantisch nicht in Ordnung sind.

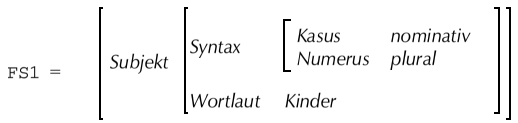


c) Definiere das Prädikat is_proper_feature_structure/1, das zusätzlich überprüft, dass kein Merkmal zwei unterschiedliche Werte hat. Teste dein Prädikat!

2. Pfade und ihre Werte

Schreibe ein Prädikat value_of_path(+*FS*, +*Path*, ?*Value*), das eine Merkmalstruktur *FS* sowie den Pfad *Path* entgegen nimmt und den Wert des Pfades im Term *Value* instantiiert. Verwende zum Zusammensetzen von Pfaden den rechts-assoziativen Infix-Operator :/2.

Beispiel:



Falls *FS1* obige Merkmalstruktur repräsentiert, lautet die Anfrage für den Numerus des Subjekts folgendermassen:

```
?- value_of_path(FS1, subjekt:syntax:numerus, Value).
Value = plural
```

3. Unifikation einfacher Merkmalstrukturen

Implementiere die Unifikation (widerspruchsfreie Vereinigung) von zwei Merkmalstrukturen, welche im Format von Aufgabe 1 vorliegen. Dazu soll das Prädikat

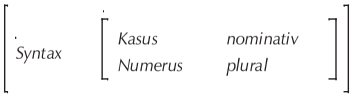
unify_feature_structure(+*FS1*, +*FS2*, ?*FS3*)

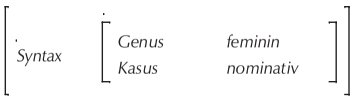
definiert werden, das die Unifikation aus den instantiierten Merkmalstrukturen *FS1* und *FS2* an die Variable *FS3* bindet.
Orientiere dich an der Beschreibung auf Folie 10. Als Hilfe für die rekursive Dekomposition des Problems, kannst du folgendes Rezept verwenden:

Um aus den Merkmalstrukturen *FS1* und *FS2* die Unifikation *FS3* herzustellen, mache folgendes:

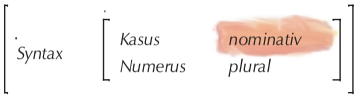
1. Füge der Reihe nach alle Merkmale von *FS1* in *FS3* ein:
   a) Falls ein Merkmal aus *FS1* nicht in *FS2* vorkommt, füge das Merkmal mit seinem Wert direkt in *FS3* ein.
   b) Falls ein Merkmal aus *FS1* in *FS2* vorkommt, unifiziere die Werte dieses Merkmals aus *FS1* und *FS2*, füge dann das resultierende Merkmal-Wert-Paar in *FS3* ein. Lösche dieses Merkmal für die Weiterverarbeitung aus *FS2* (d.h. arbeite mit einer Kopie von FS2 weiter, die dieses Merkmal nicht mehr enthält).
2. Wenn alle Merkmale von *FS1* verarbeitet sind, füge alle (restlichen) Merkmale aus *FS2* direkt in *FS3* ein (*FS2* enthält dann garantiert nur Merkmal-Wert-Paare, die nicht in *FS1* vorgekommen sind).

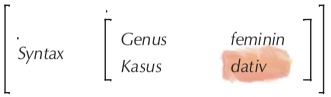
Wie kannst du das Fehlschlagen der Unifikation in Prolog modellieren? Teste dein Prädikat sowohl mit Strukturen, die unifizieren





als auch mit Strukturen, die nicht unifizieren:





Hinweis: Programmiere zuerst ein vereinfachtes Prädikat (und die nötigen Hilfsprädikate), das nur Merkmalstrukturen verarbeitet, deren Merkmalswerte alles Atome sind: