Wissensrepräsentation

Nebel/Rintanen/Wölfl — Universität Freiburg
WS 2004/2005 — Institut für Informatik

Projekt P3

Abgabe: Freitag, 11. Februar 2005

Abgabe per E-Mail an `helmert@informatik.uni-freiburg.de`

Dieses Projekt widmet sich strikten Vererbungsnetzen. Es umfasst zwei Teile:

- Im ersten Teil wird ein Programm zum Schlussfolgern in einfachen Vererbungsnetzen mit negierten Konzepten implementiert.
- Im zweiten Teil wird dieses Programm auf Vererbungsnetze mit konjunktiven Konzepten erweitert.

Hinweis: Bei den Projektaufgaben geht es um die Implementierung, nicht so sehr um die theoretischen Aspekte. Wenn Sie an bestimmten Stellen nicht genau wissen, wie die Algorithmen und Definitionen aus der Vorlesung in die Praxis umgesetzt werden, können Sie daher gerne Fragen stellen — entweder per E-Mail oder in der Übungsgruppe.

Aufgabe P3.1 (Vererbungsnetze mit Negation, 1 Punkt)

Implementieren Sie ein Programm zum Schlussfolgern in einfachen Vererbungsnetzen mit Negation, aber ohne Konjunktion. Ihr Programm sollte das in der Vorlesung vorgestellte graphenbasierte Verfahren verwenden.

Eingabe

Das zu implementierende Programm soll wie folgt aufgerufen werden:[1]

```
# ./inheritance-simple <knowledge base> <query>
```

Dabei ist `<knowledge base>` der Name einer Datei, die die Wissensbasis enthält und `<query>` die Anfrage an die Wissensbasis, eine **isa**-Aussage.[2] Jede Zeile der Wissensbasisdatei sollte eine **isa**-Aussage beinhalten oder eine Kommentarzeile sein, d. h. komplett aus Leerraum bestehen oder als erstes nicht-leeres Zeichen ein `#` enthalten. Kommentarzeilen werden ignoriert.
Eine **isa**-Aussage wird wie folgt notiert:

```
concept1 isa concept2
```

Dabei sind `concept1` und `concept2` Konzeptbeschreibungen in einer der folgenden beiden Formen:

- *Atomare Konzepte* (Konzeptnamen) sind Wörter bestehend aus Kleinbuchstaben, Ziffern und Bindestrichen, beginnend mit einem Kleinbuchstaben. Als Konzeptnamen nicht erlaubt sind die Wörter `isa`, `not` und `and`.
- *Negierte Konzepte* werden als `not concept` notiert, wobei `concept` eine Konzeptbeschreibung ist.

[1] Passen Sie das Kommando entsprechend an, wenn Sie eine interpretierte Programmiersprache wie Java verwenden.

[2] Die Aussage muss üblicherweise in Anführungszeichen eingeschlossen werden, da sie Leerzeichen enthält, aber von der Shell als einzelnes Argument betrachtet werden soll.

Ausgabe

Je nach dem, ob die Anfrage aus der Wissensbasis folgt, sollte die Ausgabe entweder

```
Yes.
```

oder

```
No.
```

lauten. Ansonsten sollten keine Ausgaben gemacht werden.

Aufgabe P3.2 (Negation und Konjunktion, 1 Punkt)

Erweitern Sie das Programm aus dem vorherigen Teil, so dass es auch konjunktive Konzepte der Form

```
concept1 and concept2 and ... and conceptk
```

akzeptiert.

Konzeptbeschreibungen dürfen von Klammern umschlossen sein, damit Mehrdeutigkeiten durch Kombinationen von `and` und `not` aufgelöst werden können. Sind keine Klammern gesetzt, sollte `not` stärker binden als `and`, so dass `not a and b` äquivalent zu `(not a) and b` ist, nicht aber zu `not (a and b)`.

Verwenden Sie für diesen Aufgabenteil den SAT-Solver `zchaff` (siehe Projekt P1), um die Anfrage zu beantworten. Ihr Programm sollte `inheritance` heißen und abgesehen vom verallgemeinerten Eingabeformat genauso verwendet werden und dieselben Ausgaben erzeugen wie das Programm aus dem vorigen Teil.

Beispiele

Beispieleingaben sind im Internet auf der Übungsseite zur Vorlesung verfübar.