

Logik für Informatiker (Diplom)

Prof. Dr. B. Nebel, Prof. Dr. W. Burgard
Wintersemester 2007/2008

Universität Freiburg
Institut für Informatik

Übungsblatt 10

Abgabe: Dienstag, 15. Januar 2008

Aufgabe 10.1 (Resolutionsrestriktionen)

Betrachten Sie die folgende Menge von Horn-Klauseln:

$$F = \{A \rightarrow B, A \wedge B \rightarrow C, A \wedge C \rightarrow D, B \wedge C \wedge D \rightarrow 0, 1 \rightarrow A\}.$$

Gegen Sie je eine Resolution von \square aus F mit folgenden Restriktionen an:

- (a) eine P-Resolution,
- (b) eine lineare Resolution beginnend mit $K = (1 \rightarrow A)$,
- (c) eine Stützmengenresolution mit Stützmenge $T = \{1 \rightarrow A\}$,
- (d) eine Inputresolution und
- (e) eine Einheitsresolution.

Beachten Sie, dass eine Resolution Lösung für mehrere Teilaufgaben sein kann.

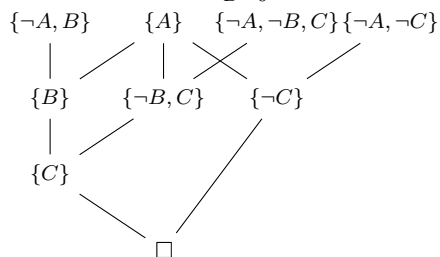
Aufgabe 10.2 (Widerlegungsvollständigkeit der P-Resolution)

Betrachten Sie die Klauselmeng F aus der vorigen Aufgabe. Vollziehen Sie anhand von F den Induktionsschritt im Beweis der Widerlegungsvollständigkeit der P-Resolution nach. Genauer: Es ist

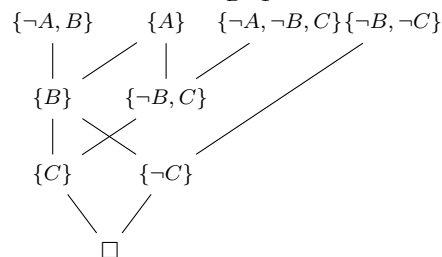
$$F_{D=0} = \{\{\neg A, B\}, \{\neg A, \neg B, C\}, \{\neg A, \neg C\}, \{A\}\} \quad \text{und} \\ F_{D=1} = \{\{\neg A, B\}, \{\neg A, \neg B, C\}, \{\neg B, \neg C\}, \{A\}\}.$$

Es gibt P-Resolutionen von \square aus $F_{D=0}$ bzw. $F_{D=1}$:

P-Resolution aus $F_{D=0}$:



P-Resolution aus $F_{D=1}$:



Konstruieren Sie aus diesen beiden P-Resolutionen eine P-Resolution von \square aus F wie im Beweis der Widerlegungsvollständigkeit der P-Resolution (vgl. Vorlesung bzw. Schöning).

Aufgabe 10.3 (SLD-Resolution)

Eine SLD-Resolution ist eine Input-Resolution, die mit einer negativen Klausel (also einer Klausel, die nur negative Literale enthält) beginnt, und bei der in jedem Resolutionsschritt eine Elternklausel eine nicht-negative Inputklausel ist.

Geben Sie eine SLD-Resolution von \square aus der Horn-Klauselmengemenge F aus den vorigen Aufgaben an. Beginnen Sie mit der negativen Klausel $(B \wedge C \wedge D \rightarrow 0)$.

Aufgabe 10.4 (Prolog)

Verschaffen Sie sich Zugriff auf einen Prolog-Interpreter, z. B.

- SICStus-Prolog (<http://www.sics.se/sicstus/>, kommerziell, sollte aber im Pool auf den Sun-Rechnern nach Aufruf von `setup lang/sicstus` verfügbar sein und mit `sicstus` gestartet werden können) oder
- SWI-Prolog (<http://www.swi-prolog.org/>, frei).

Daneben existieren weitere Prolog-Interpreter, etwa GNU-Prolog und YAP-Prolog (<http://www.gprolog.org/> bzw. <http://www.ncc.up.pt/yap/>).

Machen Sie sich mit der Bedienung des Interpreters vertraut: Nach dessen Start erscheint der Prolog-Prompt `?-`, hinter dem eine mit einem Punkt abgeschlossene Anfrage eingegeben werden kann. Prolog-Programme sollten in einer Datei gespeichert und dann mittels der Anfrage `?- [file].` in den Prolog-Interpreter eingelesen werden, wenn der Dateiname `file.pl` lautet.

Laden Sie von der Vorlesungswebsite die Datei `underground.pl` herunter.¹ Das Prädikat `reachable` listet in seinem dritten Argument die Zwischenstationen vom Start zum Ziel auf. Erweitern Sie das Prädikat um ein viertes Argument, das die Liste der auf dem Weg benutzten Linien zurückliefert. Beispielsweise sollte die Anfrage

```
?- reachable(bond_street,piccadilly_circus,Stations,Lines).
```

die folgende Ausgabe erzeugen:

```
Stations = [oxford_circus],
Lines = [central,bakerloo] ? ;

Stations = [green_park],
Lines = [jubilee,piccadilly] ? ;

Stations = [green_park,oxford_circus],
Lines = [jubilee,victoria,bakerloo] ? ;

no
```

Beachten Sie, dass Sie die jeweils nächste Antwort durch Eingabe eines Semikolons erhalten.

Die Übungsblätter dürfen und sollten in Gruppen von zwei Studenten bearbeitet werden. Bitte schreiben Sie beide Namen auf Ihre Lösung.

¹<http://www.informatik.uni-freiburg.de/~ki/teaching/ws0708/logic/underground.pl>