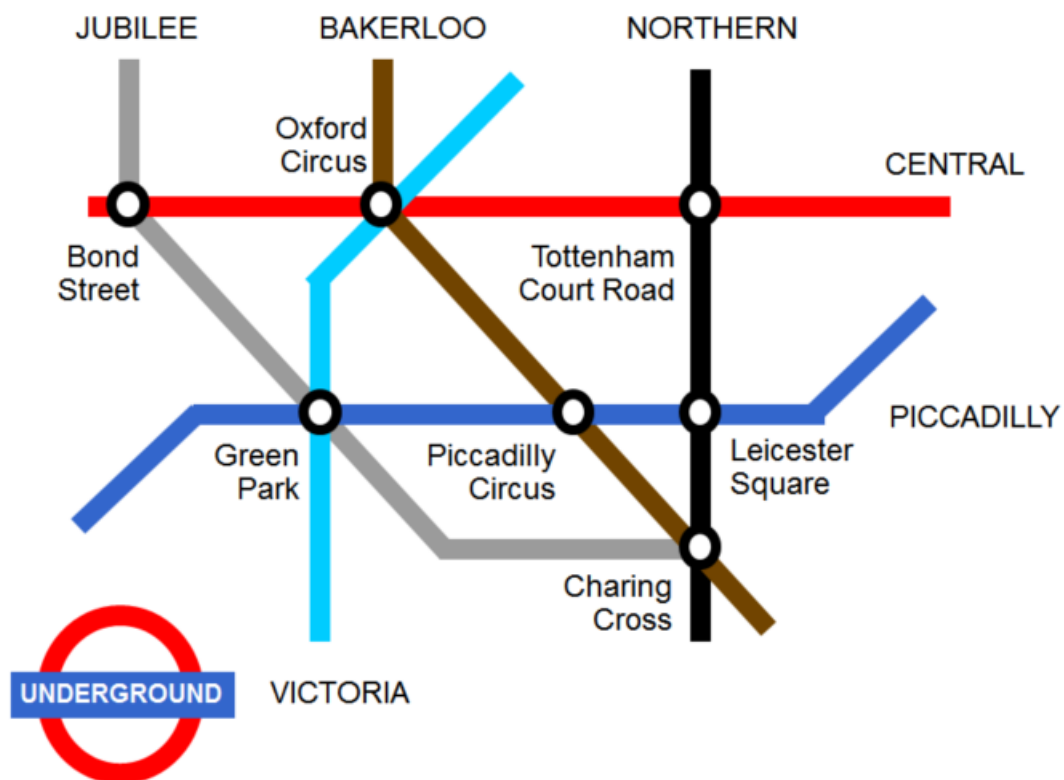


Einführung in die logische Programmierung

Aufgabenblock II

Die nachfolgenden Aufgaben in diesem Block stammen ursprünglich von Peter Flach (ehem. Universität Tilburg, NL)

Die Aufgaben befassen sich mit einem kleinen Abschnitt der Realität: ein Teil des Netzes der Londoner Underground:



Eine Abbildung dieses Netzes finden Sie in der Datei **Aufgaben_2016-02.p1**. Laden Sie diese Datei in Ihrem Prolog und probieren Sie die vorhandenen Prädikate aus.

Aufgabe 1

Beim Laden gibt Prolog eine Warnung (kein Error) aus. Was bedeuten diese? Ändern Sie den Code so, dass er funktional gleichbleibt bzw. sinngemäß funktioniert, die Warnungen aber nicht mehr erscheinen.

Aufgabe 2

Das Prädikat `reachable/3` zeigt in seinem letzten Argument³ eine Liste der Zwischenstationen, die auf der Route zwischen Argument¹ und Argument² liegen. Erweitern Sie es zu `reachable/4`, in dem das 4. Argument eine Liste aufzeigt, in dem alle befahrenen Linien aufgeführt sind. Z.B.:

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

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

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

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

false.
```

Aufgabe 3

Übersetze folgende Fragen in Prolog-Abfragen:

- 'Finde eine Station, die von Picadilly Circus erreichbar ist unter Benutzung von exakt 2 Linien.'
- 'Finde eine Station die von Bond Street erreichbar ist, wo man startet auf der Central Linie, wobei mindestens 2 Weitere Stations dazwischen sind.'
- 'Finde alle Stations die 4 Stations zwischen sich haben auf 5 Linien.'

Geben Sie jeweils Ihre Abfrage und Prologs Antwort an.

Aufgabe 4

Das Prädikat `same_line/2` sollte eigentlich symmetrisch sein, ist es aber nicht. Z.B., es gilt `same_line(bond_street, tottenham_court_road)` aber es gilt nicht `same_line(tottenham_court_road, bond_street)` . Das Gleiche gilt für das Prädikat `connected/3`.

Beheben Sie das Problem (der Nicht-Symmetrie). Aber machen Sie es so, dass die Information in der Definition des jetzigen `connected` nicht dupliziert werden muss.

Schreiben Sie eine neue Version von `reachable/4`, die Ihr symmetrisches `connected` benutzt. Spätestens dabei sollten Sie merken, dass Symmetrie zu einem neuen Problem führt -- welches? Und wie kann man das evtl. beheben? Machen!

Aufgabe 5

Schreiben Sie ein Prädikat, das die Stations auf einer Linie auflistet:

```
?- stations(Line, Stations) .

Line = victoria,
Stations = [green_park, oxford_circus] ? ;
```

```

Line = central,
Stations = [bond_street,oxford_circus,tottenham_court_road] ? ;

Line = jubilee,
Stations = [bond_street,green_park,charing_cross] ? ;

Line = piccadilly,
Stations = [green_park,piccadilly_circus,leicester_square] ? ;

Line = bakerloo,
Stations = [oxford_circus,piccadilly_circus,charing_cross] ? ;

Line = northern,
Stations = [tottenham_court_road,leicester_square,charing_cross] ? ;

No

```

Hint1: Sie können eine Lösung mit Akkumulator für die besuchten Stationen schreiben. Dann braucht jedoch die Victoria Linie eine Sonderbehandlung, weil sie nur 2 Stations hat.

Hint2: Sie schaffen eine alternative Lösung, wenn Sie sich schlau machen zum library Prädikat `setof/3`.

Zum Nachdenken: Können Sie ein Prädikat schreiben, das ausgibt, auf welchen Linien eine bestimmte Station liegt? Wenn ja, dann zeigen Sie es

Aufgabe 6

Wählen Sie eine neue Repräsentation für die Linien und Stations:

```

line(central,[bond_street,oxford_circus,tottenham_court_road]).
line(jubilee,[bond_street,green_park,charing_cross]).
line(piccadilly,[green_park,piccadilly_circus,leicester_square]).
line(victoria,[green_park,oxford_circus]).
line(bakerloo,[oxford_circus,piccadilly_circus,charing_cross]).
line(northern,[tottenham_court_road,leicester_square,charing_cross]).

```

Formulieren Sie eine neue Definition für `connected` die diese Repräsentation verwendet. (Sollte die gleiche Resultate liefern als die ursprüngliche asymmetrische Version).