Answer Set Solving in Practice

Prof. Torsten Schaub, Javier Romero

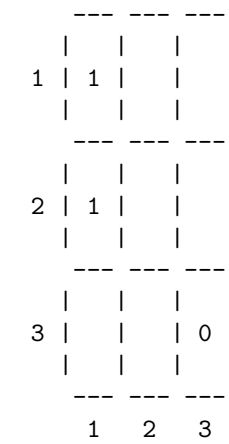
University of Potsdam — Winter Semester 2014/2015

Project 1 (Treffpunkt)

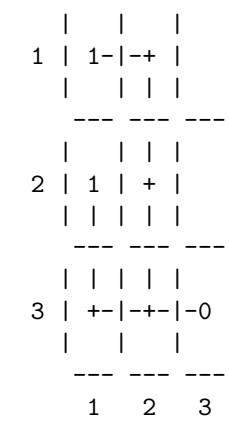
Problembeschreibung. In dieser Praktikumsaufgabe sollen Sie ein Logikpuzzle, bei dem innerhalb eines quadratischen Gitters Pfade zu einem gemeinsamen Treffpunkt gesucht sind, mit einem ASP-Encoding lösen. Initial sind zwischen zwei und vier Zellen des Gitters mit nicht-negativen ganzen Zahlen belegt. Diese vorgegebenen Zellen bilden die Startpunkte der gesuchten Pfade, die auf einer freien Zelle des Gitters, dem Treffpunkt, zusammenlaufen sollen. Die Pfade können horizontal oder vertikal aneinandergrenzende Gitterzellen miteinander verbinden. Sie dürfen sich weder verzweigen, überschneiden, noch eine der initial belegten Zellen einfach oder eine freie Zelle mehrfach erreichen. Wie bereits erwähnt sollen alle Pfade an einem gemeinsamen Treffpunkt enden. Unterwegs muss jeder Pfad so oft abbiegen, wie es die Zahl auf seinem jeweiligen Startpunkt verlangt.

Die folgende Darstellung illustriert Problemstellungen und -lösungen an einem Beispiel:

```
Instance:
number(1,1,1). number(2,1,1). number(3,3,0). cell(1..3,1..3).
```




```
Answer 1:
link(3,1,3,2) link(2,1,3,1) link(2,2,3,2) link(1,2,2,2) link(1,1,1,2) link(3,3,3,2)
```



Das obere der abgebildeten Gitter gibt eine Problemstellung wieder. Dabei ist die Seitenlänge des quadratischen Gitters mit `3` festgelegt. Die initial belegten Zellen sind (1,1), (2,1) und (3,3). Ihre Belegungen werden durch die Atome `number(1,1,1)`, `number(2,1,1)` und `number(3,3,0)` repräsentiert. Anhand der Problemstellung erkennen wir, dass drei Pfade gesucht sind, wobei die von (1,1) und (2,1) ausgehenden Pfade jeweils einmal und der von (3,3) ausgehende Pfad keinmal abbiegen soll, bevor alle drei Pfade an einem Treffpunkt zusammenlaufen.

Die (eindeutige) Lösung für diese Problemstellung ist in dem unteren der abgebildeten Gitter dargestellt. Sie besteht aus den Pfaden $\langle$(1,1),(1,2),(2,2),(3,2)$\rangle$, $\langle$(2,1),(3,1),(3,2)$\rangle$ und $\langle$(3,3),(3,2)$\rangle$. Wir erkennen, dass die freie Zelle (3,2) der gemeinsame Treffpunkt der Pfade ist, die von ihren Startpunkten aus nur über frei Zellen führen, sich nicht verzweigen oder überschneiden, nicht umkehren und so oft abbiegen, wie es die Zahl auf dem jeweiligen Startpunkt verlangt. Die drei Pfade der Lösung werden durch die Atome `link(3,1,3,2)`, `link(2,1,3,1)`, `link(2,2,3,2)`, `link(1,2,2,2)`, `link(1,1,1,2)` und `link(3,3,3,2)` repräsentiert, die jeweils die (direkt) miteinander verbundenen Paare von Zellen angeben. Z.B. steht das Atom `link(1,1,1,2)` dafür, dass der vom Startpunkt (1,1) ausgehende Pfad zuerst die Zelle (1,2) erreicht (bevor er von dort aus über die Zelle (2,2) zum Treffpunkt (3,2) führt). Beobachten Sie, dass die Reihenfolge von Werten in Atomen der Form `link(X1,Y1,X2,Y2)` mit der Reihenfolge korrespondiert, in der einer der gesuchten Pfade die horizontal oder vertikal aneinandergrenzenden Zellen (`X1`,`Y1`) und (`X2`,`Y2`) miteinander verbindet. (Das bedeutet z.B., dass `link(1,2,1,1)` nicht zur Repräsentation des von (1,1) ausgehenden Pfades gehört.) Neben den die Pfade einer Lösung repräsentierenden Atomen der Form `link(X1,Y1,X2,Y2)` darf eine Antwortmenge keine weiteren Atome des Prädikats `link/4` enthalten. (Das bedeutet z.B., dass `link(1,3,2,3)` nicht zur Repräsentation der obigen Pfade gehört.)

Aufgabenstellung. Schreiben Sie ein ASP-Encoding in der Eingabesprache des Grounders `gringo` (bzw. `clingo`), sodass die Antwortmengen für eine gegebene Instanz eins-zu-eins zu den Lösungen des durch die Instanz beschriebenen Puzzles korrespondieren. Die Instanz besteht dabei aus Fakten der Form `cell(1..n,1..n).` und `number(X,Y,N).`, die die Zellen eines quadratischen Gitters und die mit nicht-negativen ganzen Zahlen belegten Zellen des Gitters definieren. Eine Lösung soll wie oben beschrieben durch Atome der Form `link(X1,Y1,X2,Y2)` repräsentiert werden.

Framework. In dem Archiv `meeting.zip` finden Sie die Datei `meeting.lp` und zehn Beispielinstanzen. Die Datei `meeting.lp` ist von Ihnen mit Ihrem ASP-Encoding zu ergänzen. Wenn Sie Ihr Encoding einreichen, dann müssen die folgenden Zeilen in `meeting.lp` enthalten sein: [1]

```
#hide.
#show link/4.
```

Sie müssen Ihr Encoding in einer Datei mit dem Namen `meeting.lp` einreichen. Neben `link/4` dürfen in der eingereichten Version keine weiteren Prädikate in der Ausgabe eingeblendet sein!

[1] Diese Anweisungen sorgen dafür, dass alle Atome, die nicht zur Repräsentation von Lösungen gehören, bei der Ausgabe von Antwortmengen ausgeblendet werden.

Weiterhin finden Sie in `meeting.zip` das Python-Skript `graphic1.py`, welches die erste Antwortmenge mit ASCII-Zeichen visualisiert. Dieses Skript kann optional zum Debuggen des Encodings genutzt werden. Dazu wird die Ausgabe in das Skript "gepiped", z.B.:

```
[user@local ~] clingo meeting.lp grid_01.lp | python -O graphic1.py -s 3
```

Die Option `-s` steht für die Größe des Puzzles, z.B. `3` für `grid_01.lp`. Anders als bei der Einreichungsversion sollte für die Visualisierung auch das Prädikat `number/3` in der Ausgabe eingeblendet sein:

```
#hide.
#show link/4.
#show number/3.
```

Formalitäten. Sie können die Praktikumsaufgabe in Gruppen von **bis zu zwei** Studenten gemeinsam bearbeiten. Verschiedene Gruppen müssen verschiedene Lösungen einreichen. Bei Plagiaten wird die Praktikumsaufgabe für alle beteiligten Gruppen als "nicht bestanden" gewertet. Reichen Sie Ihr Encoding bitte bis zum **06.12.2010** über die von der Moodle-Seite zur Vorlesung aus verlinkte YETI-Plattform ein. (Alle Gruppenmitglieder müssen bei YETI einen Account haben und als Gruppenmitglieder registriert sein!) Achten Sie darauf, dass Sie Ihr Encoding in einer Datei mit dem Namen `meeting.lp` einreichen, wobei der Dateiname ausschließlich Kleinbuchstaben enthält.

Neben den zehn vorgegebenen Instanzen (mit jeweils einer eindeutigen Lösung) testen wir Ihr Encoding auf weiteren Ihnen unbekannten Instanzen. Auch auf diesen weiteren Instanzen muss Ihr Encoding korrekt funktionieren, damit die Praktikumsaufgabe als bestanden gilt. Wenn Sie Ihr Encoding bei YETI hochladen, wird es dort automatisch getestet (mit geringer zeitlicher Verzögerung). Falls dabei Fehler auftreten, können Sie diese den Statusmeldungen von YETI entnehmen. Zur Analyse der Effizienz von Encodings werden wir ein Vergleichsencoding bei YETI hochladen, das Ihnen einen Anhaltspunkt bzgl. der Performanz Ihres eigenen Encodings gibt.

Encodings, die eine ähnliche Performanz wie unser Vergleichsencoding erzielen, werden mit **bis zu 2 Bonuspunkten auf die Klausur** honoriert.

Empfehlungen und Hinweise:

- Überlegen Sie sich, welche Bedingungen Sie zur Charakterisierung von Lösungen testen wollen und definieren Sie geeignete Prädikate, die diese Tests ermöglichen.
- Verwenden Sie die Kommandozeilenoption `--solution-recording` zur Effizienzsteigerung bei der Berechnung aller Lösungen, z.B. wie in dem folgenden Aufruf:

  ```
  [user@local ~] clingo meeting.lp grid_01.lp --solution-recording 0
  ```

- Wenn Sie an einer Stelle nicht weiterkommen, können Sie sich gern an uns wenden. Wir versuchen alle Fragen bestmöglich zu beantworten. Fragen können Sie entweder persönlich an uns richten oder per Mail an: `asp1@cs.uni-potsdam.de`.
- Fangen Sie bald mit der Bearbeitung der Aufgabe an, damit Ihnen die Zeit nicht davonläuft. (Sollten Sie trotzdem Schwierigkeiten mit der Einhaltung des Termins haben, dann wenden Sie sich bitte an uns, anstatt eine beliebige Lösung zu kopieren!)