

Answer Set Solving in Practice

Prof. Torsten Schaub, Javier Romero
University of Potsdam — Winter Semester 2014/2015
Project 2 (Sitzplan)

Problembeschreibung. Die Studenten eines Kurses überlegen sich, wie der Sitzplan bei der Klausur aussehen müsste, damit möglichst viele von ihnen abschreiben können. Zur Ermittlung aussichtsreicher Abschreibgelegenheiten schätzt jeder Student vorab seine Note. Wenn ein Teilnehmer nicht zu nahe an der Klausuraufsicht sitzt, kann er von einem besseren Studenten direkt vor oder neben sich abschreiben. Um die Abschreibgelegenheiten objektiv zu optimieren, soll der Sitzplan mittels Antwortmengenprogrammierung (ASP) erstellt werden. Die ASP-Lösung ist allerdings auch für die Kursleitung praktisch, um ihrerseits einen Sitzplan mit möglichst wenig Abschreibgelegenheiten zu gestalten.

<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>										1	Student	Note	<table><tr><td>S</td><td></td><td>3</td></tr><tr><td></td><td>2</td><td>1</td></tr><tr><td>5</td><td>4</td><td>6</td></tr></table>	S		3		2	1	5	4	6
S		3																				
	2	1																				
5	4	6																				
1	2	3	1	27	1	2	3															
			2	14																		
			3	26																		
			4	39																		
			5	27																		
			6	34																		

Abbildung 1: Ein Beispielszenario (links) und ein optimaler Sitzplan (rechts).

Ein beispielhaftes Szenario ist in Abbildung 1 (links) dargestellt. Neben dem Raum, in dem die Klausur geschrieben wird, sind die geschätzten Noten, die Anzahl der für die Aufsicht zuständigen Personen und der Modus angegeben. Die Aufgabe besteht darin, Studenten und Beobachter so zu platzieren, dass möglichst viel (bzw. wenig) abgeschrieben werden kann, wobei die folgenden formalen Kriterien angelegt werden:

1. Verschiedene Beobachter müssen (aus Kostengründen) mindestens drei Plätze voneinander entfernt sitzen, wobei die Manhattan-Distanz (Summe der Koordinatendifferenzen) als Maß für den Abstand zwischen Plätzen verwendet wird.
2. Ein Student, der horizontal, vertikal oder diagonal unmittelbar neben einem Beobachter sitzt, kann nicht abschreiben. Andernfalls kann der Student von direkt vor oder neben sich platzierten besseren Teilnehmern (mit kleinerer Note) abschreiben.
3. Im Studentenmodus soll die Anzahl der Abschreibgelegenheiten, ermittelt anhand von Paaren voneinander abschreibender Studenten, möglichst groß und im Kursleitungsmodus möglichst klein sein.

Eine optimale Lösung (für Studenten) mit 6 Abschreibgelegenheiten ist in Abbildung 1 (rechts) dargestellt, wobei die Nummern für Studenten und *S* für den Beobachter stehen. Konkret kann bei der gezeigten Platzierung wie folgt abgeschrieben werden:

- Student 1 von Student 2 und 3
- Student 4 von Student 2, 5 und 6
- Student 6 von Student 1

Repräsentation in ASP. In ASP repräsentieren wir Probleminstanzen durch Fakten über den folgenden Prädikaten:

```
cols(C)           % Anzahl Sitze je Reihe
rows(R)           % Anzahl der Sitzreihen
mode(K)           % Modus (students - maximiere Abschreiben,
                  %          uni          - minimiere Abschreiben)
numstudents(N)    % Anzahl der Studenten
numsupervisors(M) % Anzahl der Beobachter
exp_grade(S,G)    % geschätzte Note G von Student S
```

Für jeden der von 1 bis N (aus `numstudents(N)`) fortlaufend nummerierten Studenten S enthält eine Instanz genau einen Fakt `exp_grade(S,G)` mit der geschätzten Note G von S.

Z.B. wird das Szenario in Abbildung 1 (links) durch folgende Fakten beschrieben:

```
cols(3). rows(3). mode(students).
numstudents(6). numsupervisors(1).
exp_grade(1,27). exp_grade(2,14). exp_grade(3,26).
exp_grade(4,39). exp_grade(5,27). exp_grade(6,34).
```

Eine Lösung, d.h. eine Platzierung von Studenten und Beobachtern, wird durch Atome der folgenden Form repräsentiert:

```
at(student(1),X1,Y1) ... at(student(N),XN,YN) % Studentenpositionen
at(supervisor,X1,Y1) ... at(supervisor,XM,YM) % Beobachterpositionen
```

Während die Nummern von Studenten in den Atomen enthalten sind, werden Beobachter nicht einzeln benannt. Außerdem müssen die Positionen in den Atomen paarweise verschieden voneinander sein.

Die in Abbildung 1 (rechts) dargestellte Lösung wird z.B. durch folgendes Atome innerhalb einer Antwortmenge beschrieben:

```
at(student(1),3,2) at(student(2),2,2) at(student(3),3,1)
at(student(4),2,3) at(student(5),1,3) at(student(6),3,3)
at(supervisor,1,1)
```

Framework. In dem Archiv `plan.tar.gz` finden Sie die Datei `plan.lp` und sieben Beispielinstanzen. Die Datei `plan.lp` ist von Ihnen mit Ihrem ASP-Encoding zu ergänzen, sodass *optimale* Antwortmengen zu Lösungen mit je nach Modus maximal oder minimal vielen Abschreibgelegenheiten korrespondieren. Wenn Sie Ihr Encoding einreichen, dann müssen die folgenden Zeilen in `plan.lp` enthalten sein:¹

```
#hide.
#show at(student(_),_,_).
#show at(supervisor,_,_).
```

Sie müssen Ihr Encoding in einer Datei mit dem Namen `plan.lp` einreichen. Neben `at/3` dürfen in der eingereichten Version keine weiteren Prädikate in der Ausgabe enthalten sein!

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

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 **31.12.2012** über YETI 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 `plan.lp` einreichen, wobei der Dateiname ausschließlich Kleinbuchstaben enthält.

Neben den sieben vorgegebenen Instanzen testen wir Ihr Encoding auf weiteren Ihnen unbekannten Instanzen. Auch auf diesen weiteren Instanzen muss Ihr Encoding *korrekt* funktionieren, d.h. (im Falle der Terminierung innerhalb weniger Minuten) **eine** optimale Lösung liefern, 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 dies den Statusmeldungen von YETI entnehmen. Bitte korrigieren Sie eventuelle Fehler umgehend selbständig oder (bei Schwierigkeiten) in Rücksprache mit uns.

Bonus: Zur Analyse der resultierenden 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 Ermittlung (optimaler) Lösungen testen wollen und definieren Sie geeignete Prädikate, die diese Tests ermöglichen.
- Die resultierende Effizienz von Encodings kann nicht unabhängig von dem zugrunde liegende Lösungsverfahren ermittelt werden. Die folgenden Faustregeln können dennoch (unabhängig von Lösungsverfahren) als universell betrachtet werden:
 1. Platzbedarf und Laufzeit korrespondieren zueinander. Wenn die gleichen Sachverhalte mit weniger (instanziierten) Regeln ausgedrückt werden können, dann sollte die Suche nach Antwortmengen davon profitieren. (Verwenden Sie möglichst Default-Negation anstelle von Auflistungen komplementärer Werte.)
 2. Regeln und Integritätsbedingungen sind umso stärker, je weniger (nicht statisch bestimmte) Vorbedingungen in ihren Körpern auftreten. (Vermeiden Sie für die Korrektheit unerhebliche Fallanalysen in Regelkörpern.)
 3. Die Suche profitiert i.d.R. davon, dass nicht-triviale inhärente Eigenschaften von (optimalen) Lösungen in “redundanten” Bedingungen explizit gemacht werden, weil sie dann nicht erst auf Fallbasis über Suche ermittelt werden müssen.
- Wenn Sie an einer Stelle nicht weiterkommen, können Sie sich gern an uns wenden. Wir versuchen alle Fragen bestmöglich zu beantworten. Fragen und Bemerkungen können Sie persönlich an uns richten, in Moodle (Forum/Wiki) stellen oder per Mail an `asp1@cs.uni-potsdam.de` senden.
- 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!)