

Handlungsplanung

Prof. Dr. B. Nebel, Dr. M. Helmert
R. Mattmüller
Wintersemester 2006/2007

Universität Freiburg
Institut für Informatik

Projekt P1

Abgabe: Mittwoch, 17. Januar 2007

Ziel dieses Projekts ist es, mehrere Planer kennenzulernen und mit ihnen zu arbeiten. Dabei geht es im Gegensatz zu den Übungen nicht so sehr um theoretische Fragen, sondern um den praktischen Umgang mit den Planern. Konkret wollen wir hier einen experimentellen Vergleich der Laufzeiten und Planlängen verschiedener Planer anstellen.

Um das Projekt durchführen zu können, benötigen Sie die Planer (a) SatPlan, (b) SGPlan, (c) FF und (d) LPG und zusätzlich (e) einige Wrapper-Skripte, die es Ihnen erlauben, die Planer mit einer einheitlichen Schnittstelle zu verwenden, (f) das Planvalidierungstool VAL, (g) Gnuplot zum Plotten der Ergebnisse, (h) einen Compiler/Interpreter für eine Programmiersprache Ihrer Wahl (C, C++, Java, Python, Ruby, Bash, SML, Prolog, ...¹), sowie (i) die Benchmark-Probleme des Internationalen Planungswettbewerbs 2006.

Die Planer und sonstigen Downloads sind auf der Übungswebsite verlinkt.

Aufgabe P1.1 (Planen – 2 Punkte)

Die Wrapper `ff.sh`, `lpg.sh`, `satplan.sh` und `sgplan.sh` werden einheitlich wie folgt aufgerufen:

```
./PLANNER.SH dom prob soln time mem
```

Dabei sind `dom`, `prob` und `soln` die Namen der Domänen-, Problem- und Lösungsdateien, `time` ein Timeout in Sekunden, `mem` eine Speicherbegrenzung in KB.

Schreiben Sie ein Programm, das systematisch alle Probleme von allen Planern lösen lässt. Das Programm soll als `./generateplans config` aufgerufen werden können. Mit dem Argument `config` wird der Name einer Konfigurationsdatei übergeben, deren Inhalt nach der Grammatik in Tabelle 1 aufgebaut ist.

Die Nichtterminale `<int>`, `<name>` und `<path>` stehen für Zeichenketten, die als Ganzzahl, als alphanumerische Zeichenkette bzw. als Pfadangabe geparkt werden können. Zeilen können durch ein führendes `#` auskommentiert werden. Jeder Eintrag `PLANNER <name> <path>` steht für einen zu testenden Planer mit Namen `<name>` und ausführbarer Datei `<path>`. `RUNS` gibt an, wieviele Male jedes Planungsproblem von jedem Planer gelöst werden soll. `TIMEOUT` setzt einen Timeout für jeden einzelnen Planerlauf in Sekunden fest, `MEMORY` eine Obergrenze für den verwendeten Hauptspeicher in KB. Mit `PREFIX` wird ein Verzeichnispräfix angegeben, relativ zu dem die später folgenden Angaben der Domänen- und Problemdateinamen zu verstehen sind. Domänen werden

¹andere Programmiersprachen nach Absprache (mattmuel@informatik.uni-freiburg)

```

<config> ::= <planner>* <runs> <timeout>
           <memory> <prefix> <domain>*
<planner> ::= PLANNER <name> <path>
<runs>    ::= RUNS <int>
<timeout> ::= TIMEOUT <int>
<memory>  ::= MEMORY <int>
<prefix>  ::= PREFIX <path>
<domain>  ::= <domainfile> <problemfile>*
<domainfile> ::= DOMAIN <name> <path>
<problemfile> ::= PROBLEM <int> <path>

```

Tabelle 1: Aufbau Konfigurationsdatei

mit `DOMAIN <name> <path>` deklariert, wobei `<name>` der Name der Domäne und `<path>` der relative Pfad der Domänenendatei ist. Jeder Eintrag `PROBLEM` gibt die Problemnummer und den relativen Pfad der Problemdatei an. Jedes Problem gehört zur zuletzt erwähnten Domäne. Jede Domäne wird nur einmal deklariert.

Die Datei `benchmarks.dat` in dem Archiv `projekt01.tar.gz` auf der Vorlesungs-Website ist entsprechend aufgebaut.

Ihr Programm `generateplans` soll im aktuellen Arbeitsverzeichnis ein Unterverzeichnis `results` anlegen, das je ein mit dem Namen des Planers benanntes Unterverzeichnis pro Planer enthält. Diese Verzeichnisse enthalten entsprechend benannte Unterverzeichnisse für jede Domäne, in denen die von den Planern erzeugten Lösungsdateien abgelegt werden. Die Lösungsdateien enthalten Pläne zusammen mit Metainformationen wie Anzahl der Aktionen oder für das Planen verbrauchte Zeit. Die Lösungsdatei des k -ten Planerlaufs auf Problemdatei `prob.pddl` sollte unter dem Namen `prob-k.soln` gespeichert werden. Für wegen Überschreitung des Timeouts abgebrochene Planungsvorgänge werden keine Lösungsdateien angelegt.

Starten Sie `generateplans` mit der Konfigurationsdatei `benchmarks.dat`. Passen Sie vorher ggf. die Pfadangaben in `benchmarks.dat` an.

Aufgabe P1.2 (Auswertung – 1,5 Punkte)

Schreiben Sie ein Programm `evaluateplans`, das die Pläne auswertet. Die Konfigurationsdatei, mit der `generateplans` aufgerufen wurde, wird auch an `evaluateplans` übergeben.

Rufen Sie in `evaluateplans` für jeden Plan `VAL` auf, um die Pläne auf Korrektheit zu überprüfen. Geben Sie eine Zusammenfassung der Korrektheitsprüfungen auf der Standardausgabe aus. Entnehmen Sie für jeden Plan der Lösungsdatei die Planlänge und die Laufzeit, die der jeweilige Planer benötigt hat, um den Plan zu berechnen.

Berechnen Sie für jeden Planer und jedes Problem die Mittelwerte der Laufzeiten und Planlängen über alle Läufe des Planers auf dem Problem.

Erstellen Sie pro Planer und Domäne im entsprechenden Ergebnisverzeichnis eine Datei `plans.dat`, die als Eingabe für Gnuplot dienen soll und für jedes Problem aus der Domäne eine Zeile der Form `num time actions` enthält. Dabei

ist `num` die Problemnummer, `time` der Mittelwert der mit dem aktuellen Planer zur Lösung von Problem `num` benötigten Zeiten und `actions` der entsprechende Mittelwert der Planlängen.

Aufgabe P1.3 (Ausgabe – 0,5 Punkte)

Erstellen Sie mit Hilfe von Gnuplot für jede Domäne je einen Plot, in dem die Laufzeiten, und einen, in dem die Planlängen verglichen werden. Auf der x -Achse sind die Probleme nach aufsteigender Nummer aufgetragen, auf der y -Achse die mittleren Laufzeiten bzw. Planlängen. Die Plots enthalten je eine Kurve pro untersuchtem Planer.

Beispiel: Sind `ff-plans.dat`, `lpg-plans.dat`, `satplan-plans.dat` und `sgplan-plans.dat` die Vergleichsdateien der Planer, so können Sie einen Vergleichsplot der Planlängen erhalten durch den Gnuplot-Befehl

```
plot "ff-plans.dat"      u 1:3 w lines title "FF", \
    "lpg-plans.dat"      u 1:3 w lines title "LPG", \
    "satplan-plans.dat"  u 1:3 w lines title "SAT", \
    "sgplan-plans.dat"   u 1:3 w lines title "SGP"
```

Aufgabe P1.4 (Bonusaufgabe – 0 Punkte)

- Spielen Sie mit der Konfigurationsdatei `benchmarks.dat` und kommentieren Sie weitere, bisher auskommentierte Planungsprobleme ein. Wiederholen Sie nun die Tests.
- Ermitteln Sie bei der Berechnung der Mittelwerte von Laufzeiten und Planlängen die 95%-Konfidenzintervalle² für die Mittelwerte und schreiben Sie die Grenzen der Intervalle zusätzlich zu den Mittelwerten in die Datei `plans.dat`. Visualisieren Sie die Intervalle bei der Gnuplot-Ausgabe.

Das Projekt darf und sollte in Gruppen von zwei Studenten bearbeitet werden. Geben Sie in Ihrer Lösung beide Namen an.

Geben Sie eine komprimierte Archiv-Datei ab, die alle von Ihnen geschriebenen Dateien (nur Quellen, keine Binaries, ggf. zusammen mit einem Makefile o.ä.) enthält. Das Archiv sollte ferner alle erzeugten Pläne aus Aufgabe P1.1 und Auswertungsdateien aus P1.2 sowie die Plots aus Aufgabe P1.3 (als Postscript-Dateien o.ä.³) enthalten.

²etwa mit Hilfe von Bootstrap-Resampling (Perzentil-Methode), vgl. <http://webrum.uni-mannheim.de/sowi/shikanos/Publication/BootstrapMethodenbuch-20-12-05.pdf>

³Postscript-Ausgabe ist in Gnuplot mit dem Befehl `set terminal postscript eps` vor dem Plotbefehl möglich. Die Ausgabedatei kann mit `set output <file>` festgelegt werden.