

Handlungsplanung

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

Universität Freiburg
Institut für Informatik

Projekt P3

Abgabe: Freitag, 16. März 2007

Die LOGISTICS-Domäne ist eine Standard-Benchmark-Domäne. Eine Aufgabe aus dieser Domäne besteht aus einer Menge von Städten, die jeweils eine Menge von Orten enthalten. Das Ziel besteht darin, Pakete von Anfangs- zu Zielpositionen zu transportieren. Die Pakete können innerhalb von Städten mit Hilfe von LKW unbeschränkter Kapazität transportiert werden. Zwischen ausgezeichneten Orten in den Städten (Flughäfen) kommen Flugzeuge unbeschränkter Kapazität zum Einsatz. Alle Orte einer Stadt sind paarweise über Straßen miteinander verbunden. Ebenso sind je zwei Flughäfen mit Flugzeugen direkt voneinander zu erreichen. LKW können sich nur innerhalb einer Stadt bewegen, Flugzeuge nur zwischen Flughäfen.

In diesem Projekt schreiben Sie einen domänenspezifischen Planer für die LOGISTICS-Domäne. Da optimales Planen in der LOGISTICS-Domäne NP-schwer ist, darf Ihr Planer auch suboptimale Pläne erzeugen.

Aufgabe P3.1 (Domänenspezifischer Planer – 3 Punkte)

Schreiben Sie einen Planer für LOGISTICS-Aufgaben. Sie können dabei domänenspezifische Algorithmen und Heuristiken verwenden, um möglichst kurze Pläne zu erzeugen.

Geben Sie Pläne in dem in Projektaufgabe P2.3 beschriebenen Format aus. Ihr Planer sollte Aufgabenbeschreibungen lesen können, die der STRIPS-Kodierung der LOGISTICS-Domäne aus dem Internationalen Planungswettbewerb 1998 entsprechen (s. Anhang).

Die Punkte für diese Aufgabe orientieren sich an der Güte (Kürze) der berechneten Pläne.

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

Logistics-Domäne

```
(define (domain logistics-strips)
  (:requirements :strips)
  (:predicates (OBJ ?obj) (TRUCK ?truck) (LOCATION ?loc)
               (AIRPLANE ?airplane) (CITY ?city) (AIRPORT ?airport)
               (at ?obj ?loc) (in ?obj1 ?obj2) (in-city ?obj ?city))

  (:action LOAD-TRUCK
    :parameters (?obj ?truck ?loc)
    :precondition (and (OBJ ?obj) (TRUCK ?truck) (LOCATION ?loc)
                      (at ?truck ?loc) (at ?obj ?loc))
    :effect (and (not (at ?obj ?loc)) (in ?obj ?truck)))

  (:action LOAD-AIRPLANE
    :parameters (?obj ?airplane ?loc)
    :precondition (and (OBJ ?obj) (AIRPLANE ?airplane) (LOCATION ?loc)
                      (at ?obj ?loc) (at ?airplane ?loc))
    :effect (and (not (at ?obj ?loc)) (in ?obj ?airplane)))

  (:action UNLOAD-TRUCK
    :parameters (?obj ?truck ?loc)
    :precondition (and (OBJ ?obj) (TRUCK ?truck) (LOCATION ?loc)
                      (at ?truck ?loc) (in ?obj ?truck))
    :effect (and (not (in ?obj ?truck)) (at ?obj ?loc)))

  (:action UNLOAD-AIRPLANE
    :parameters (?obj ?airplane ?loc)
    :precondition (and (OBJ ?obj) (AIRPLANE ?airplane) (LOCATION ?loc)
                      (in ?obj ?airplane) (at ?airplane ?loc))
    :effect (and (not (in ?obj ?airplane)) (at ?obj ?loc)))

  (:action DRIVE-TRUCK
    :parameters (?truck ?loc-from ?loc-to ?city)
    :precondition (and (TRUCK ?truck) (LOCATION ?loc-from)
                      (LOCATION ?loc-to) (CITY ?city) (at ?truck ?loc-from)
                      (in-city ?loc-from ?city) (in-city ?loc-to ?city))
    :effect (and (not (at ?truck ?loc-from)) (at ?truck ?loc-to)))

  (:action FLY-AIRPLANE
    :parameters (?airplane ?loc-from ?loc-to)
    :precondition (and (AIRPLANE ?airplane) (AIRPORT ?loc-from)
                      (AIRPORT ?loc-to) (at ?airplane ?loc-from))
    :effect (and (not (at ?airplane ?loc-from)) (at ?airplane ?loc-to)))
)
```