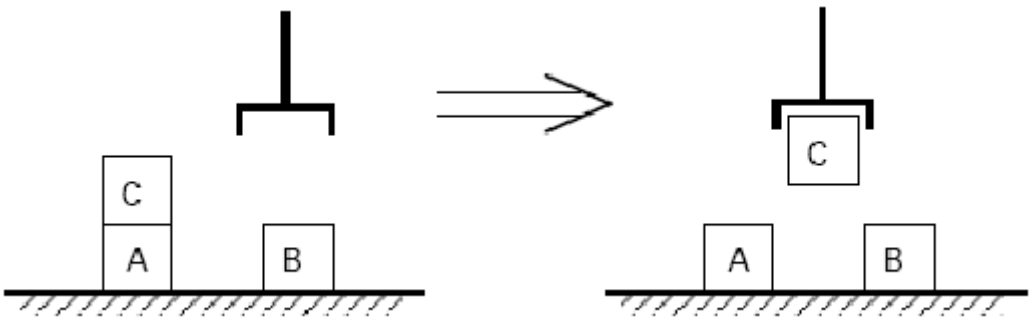
Einführung
in die logische Programmierung

Aufgabenblock IV

Die nachfolgenden Aufgaben befassen sich mit einer kleinen Welt, die Blocksworld. Es gibt dort Blöcke die auf einem Tisch sind und die zu Türmen aufeinandergestapelt werden können. Für das Stapeln, bzw. für dessen Aktionsplanung ist ein Roboter zuständig. (Siehe Bild)

Blöcke können sich auf dem Tisch befinden: on(a,table), bedeutet "block a is on the table". Oder blöcke können auf einander gestapelt sein: on(a,b), bedeutet "block a is on block b. Um ein Block bewegen zu können, muss er frei sein: es darf kein anderer Block auf dem Block gestapelt sein. Ein block a kann auf Block b gestapelt werden (bewogen werden) wenn beide frei sind. (jeder Block der frei ist kann auf den Tisch bewogen werden, der Tisch darf selbstverständlich nicht auf Blöcke bewogen werden). Wenn ein Block nicht frei ist, dann muss man ihn freimachen (clear), bevor man ihn bewegen kann.



Ein Prolog Programm, das die Aktionen für den Roboter relativ zu einem Anfangszustand und einen Auftrag, der die Zielzustand der Welt beschreibt, finden Sie in **`Aufgaben_2021-04.pl`**. Laden Sie diese Datei in Ihrem Prolog und probieren Sie die vorhandenen Prädikate aus. Lesen Sie die Dokumentation und die inline Kommentare in der Datei sorgfältig. Experimentieren Sie auch mit noch mehr Blöcken und mehrere Türme.

Aufgabe 1

Auf dem ersten Blick funktioniert das Programm einwandfrei. Dies ist allerdings nur bei gültigen Zielzuständen der Fall. Falls man beim ursprünglichen Programm einen nicht-valide Weltzustand herstellen will, dann kann das Programm in einer Endlosschleife geraten.

Als Beispiel diene: do([on(b,table), on(c,b), on(a,b)]). Dieser Zustand ist ungültig, weil nicht zwei Blöcke (hier: c und a) direkt auf einem Block (hier: b) stehen können.

Um Endlosschleifen zu vermeiden ist das Prädikat valid/1 vorgesehen. Dies existiert nur als Platzhalter im Programm. Schreiben Sie den Code für valid/1.

Aufgabe 2

Ändern Sie das Blocksworld Programm so ab, dass es ohne assert und retract auskomm. Dies erreichen Sie wenn Sie zusätzliche Argumente in den Prozeduren für Weltzustand und Aktionsplan mitführen.

Aufgabe 3

Wenn Sie ein wenig mit dem Programm herumspielen, werden Sie schnell merken, dass es keine sonderlich intelligenten Pläne erstellt. Versuchen Sie das Programm dahingehend zu optimieren, dass es keine überflüssigen Aktionen plant. (Hint: untersuchen sie die Reihenfolgen der Fakten in der Zielzustandsbeschreibung).