

Grundlagen der Künstlichen Intelligenz

Prof. Dr. B. Nebel, Prof. Dr. W. Burgard
B. Frank, A. Karwath, G. Röger
Sommersemester 2009

Universität Freiburg
Institut für Informatik

Übungsblatt 7

Abgabe: Dienstag, 23. Juni 2009

Aufgabe 7.1 (Situationskalkül und Golog)

Aktionssprachen werden verwendet, um das Verhalten eines Systems (z.B. eines Roboters) auf einer hohen Abstraktionsebene festzulegen. Golog ist eine solche Sprache, die (in ihren theoretischen Aspekten) auf dem Situationskalkül basiert. Die Implementierung von Golog basiert auf Prolog (daher der Name).

IndiGolog ist eine Gologvariante, die es erlaubt, Sensing und exogene Ereignisse zu modellieren. IndiGolog-Programme werden online ausgeführt, das heißt, es werden nicht erst alle Aktionen vorberechnet, bevor die Programmausführung beginnt.

Sie finden einen Tarball mit IndiGolog auf der Übungswebseite. Da es nur eine Linuxversion gibt, können Sie entweder einen Poolrechner verwenden oder ihre eigene Linuxinstallation. Im Pool sind bereits alle notwendigen zusätzlichen Programme installiert, auf ihrem eigenen Computer müssen Sie eventuell Software nachinstallieren (in erster Linie SWI-Prolog, siehe Übungswebseite).

Auf unsere Webseite finden Sie auch genauere Informationen, wie Sie das im Tarball enthaltene Wumpusworldbeispiel laufen lassen können.

- (a) Bringen Sie das Beispiel zum Laufen. Sehen Sie sich danach die Datei `wumpus.pl` an (Sie müssen nicht alles verstehen) und finden Sie heraus, was die folgenden Ausschnitte bedeuten:
- `poss(moveFwd, neg(inTheEdge(locRobot,dirRobot)))`.
 - `causes(turn, dirRobot, Y, rotateRight(dirRobot,Y))`.
 - `interrupt(isGold(locRobot)=true, [pickGold])`

Wahrscheinlich kommen Sie bereits mit scharfem Hinsehen und ein bisschen Nachdenken auf die richtige Lösung. Falls nicht, hilft Ihnen eine Suche in der Datei

<http://www.dis.uniroma1.it/~degiacom/papers/2009/IndiGologChapter09.pdf>.

- (b) Wenn Sie sich die Action history und Sensing results im Applet ansehen, sehen Sie, dass der Roboter auch dann noch `smell`-Aktionen ausführt, wenn er den Wumpus bereits getötet hat. Modifizieren Sie die Prozedur `mainControl(4)` so, dass diese unnötigen Aktionen nicht mehr ausgeführt werden. (Fügen Sie eine if-Abfrage ein, die testet, ob der Wumpus noch lebendig (`alive`) ist. Für den else-Teil benötigen Sie eine leere Aktion, die Sie zum Beispiel durch `?(true)` simulieren können.)
- (c) Bisher kann der Roboter sich nur mit `turn`-Aktionen (im Uhrzeigersinn) drehen. Es sind aber bereits Aktionen `turnRight` und `turnLeft` vorgesehen, die momentan aber noch nicht ausführbar (und noch nicht fertig implementiert) sind. Machen Sie diese Aktionen ausführbar und die ursprüngliche `turn`-Aktion nicht-ausführbar, indem Sie die Vorbedingungen modifizieren. Fügen Sie Code ein, der den Effekt der neuen Aktionen beschreibt. Für `turnLeft` benötigen Sie dabei ein neues Prädikat `rotateLeft` analog zu `rotateRight`. Die `turn`-Aktion wurde bisher von den Prozeduren `move(D)` und `shoot(D)` verwendet. Prozedur `move(D)` ist

dabei zum Beispiel folgendermaßen zu lesen: Suche nach einer Aktionssequenz, die `turn` maximal 4-mal ausführt, um `dirRobot=D` wahr zu machen. Führe danach die Aktion `moveFwd` aus.

Sie sollen diese Prozeduren nun so verändern, dass die neuen Aktionen verwendet werden, und zwar so, dass der Roboter möglichst wenig gedreht wird. Verwenden Sie hierzu das Konstrukt `ndet(aktionen1,aktionen2)`, mit dem die Aktionssequenzsuche wählen kann, ob sie `aktionen1` oder `aktionen2` ausführt. Ziehen Sie dabei nicht die Testaktion (die mit dem Fragezeichen) in das `ndet`.

Testen Sie ihre Implementierung, indem Sie das Programm laufen lassen.

Ihre Abgabe für Teil (b) und (c) (zusammen) sollte alle Änderungen in der Datei `wumpus.pl` und die *Action history und Sensing results* eines Laufs in der finalen Version enthalten.

Aufgabe 7.2 (Allens Intervalkalkül)

- (a) Die Komposition zweier binären Relationen R und S (über X) ist im Allgemeinen definiert als

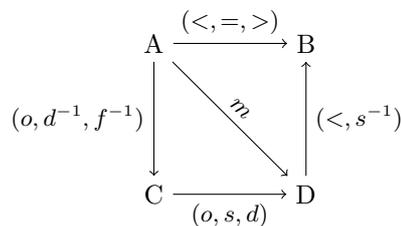
$$R \circ S = \{(x, z) \mid \exists y \in X \text{ so dass } (x, y) \in R \text{ und } (y, z) \in S\}.$$

Allens Intervalkalkül ist *unter Komposition abgeschlossen*, das heißt, jede Komposition von Allenrelationen (also auch von Vereinigungen der 13 Basisrelationen) kann wieder als Vereinigung von Basisrelationen dargestellt werden. Zum Beispiel ist $f \circ s = d$, da für beliebige Intervalle A, B und C mit AfB und BsC auch AdC gelten muss. Es ist aber nicht so, dass die Komposition von Basisrelationen wieder eine einzelne Basisrelation ergeben muss, wie man am Beispiel $f^{-1} \circ d = (o, d, s)$ sehen kann. Bestimmen sie die folgenden Kompositionen:

- (1) $o \circ m$
- (2) $m \circ f$
- (3) $(o, m) \circ f$

- (b) Die Komposition wird auch bei der Constraint-Propagation verwendet. Nutzen Sie diese Technik, um das folgende Constraint-Netzwerk 3-konsistent zu machen.

Hinweis: Ist von einem Intervall zu einem anderen keine gerichtete Kante vorhanden, bedeutet dies implizit, dass für diese Intervalle die All-Relation gilt, d.h. die Vereinigung aller 13 Basisrelationen. Sie dürfen zudem verwenden, dass $(IRJ, ISJ)^{-1} = (IR^{-1}J, IS^{-1}J)$ (für I, J Intervalle und R, S Basisrelationen des Intervalkalküls).



Aufgabe 7.3 (Entscheidungsbäume)

- (a) Geben sie Entscheidungsbäume an, die die folgenden Booleschen Funktionen repräsentieren:
- (1) $A \text{ XOR } B$
 - (2) $(A \wedge B) \vee (C \wedge D)$
- (b) Diese Übungsaufgabe behandelt die grundlegenden informationstheoretischen Konzepte, die bei der Erstellung von Entscheidungsbäumen Verwendung finden. Betrachten Sie die folgende Menge von Trainingsbeispielen:

a_1	a_2	Classification
T	T	+
T	T	+
T	F	-
F	F	+
F	T	-
F	T	+

Wie groß ist der Informationsgehalt (information content) dieser Sammlung von Trainingsbeispielen bezüglich der Zielfunktion *Classification*? Wie hoch ist der Informationsgewinn (information gain) von a_2 relativ zu diesen Trainingsbeispielen?

Die Übungsblätter dürfen und sollten in Gruppen von drei (3) Studenten bearbeitet werden. Bitte füllen Sie das Deckblatt¹ aus und heften Sie es an Ihre Lösung.

¹<http://www.informatik.uni-freiburg.de/~ki/teaching/ss09/gki/coverSheet-german.pdf>