

## Ausarbeitung der Gruppe „Wissensmanagement mit ATMS“

### Bestandsaufnahme

#### 1. Unter welchen Umständen ergänzt der GA (welche) Wissensbasis? Wie lassen sich die Einträge begründen?

- Der Instruktions- CRIL-Graph wird in „mem“ gespeichert.  
Bei der Perzeption inferiert der GA aufgrund von wahrgenommenen Wegen zu möglichen Bewegungspfaden (in „cPG“ durch "recognize") und speichert diese im „cPG“. Danach werden Teile aus dem „cPG“ in „mem“ übernommen (Wahrnehmung + Matching). (Abschnitt 1.1)
- Die Actionplan WB wird ergänzt durch die Instruktionen (Abschnitt 1.2, 6.2) aus dem Instruktion File.
- Wissen über den Aufenthaltsort, was auf der Selbstlokalisierung gründet, wird in curloc und MyCurrentPositionNode vermerkt.  
(Abschnitt 1.3)
- Die Liste aller Knoten, die schon einmal gegangen wurden (P-Graph) ist in der alreadygonePaths Methode enthalten.
- Bei Fortsetzung der Weg wird die Wahrnehmung ergänzt, welches wohl evtl. die angedachte Revision widerspiegeln kann/sollte.
- Der GA führt geometrische Inferenzen bezüglich der Objekte und Regionen aus (in RobotMind durch "ObjectLocalization").

#### 2. Welches „Wissen“ gelangt nicht in (welche) Wissensbasis (sollte es aber)?

- Der GA hat kein Handlungsgedächtnis, also kein Wissen über zuvor getätigte Aktionen, somit können Handlungen mehrfach am Stück durchgeführt werden, was zu unsinnigem Verhalten führen kann. Ebenso ist dem GA Start und Ziel seiner Instruktionsbeschreibung unbekannt. (Abschnitt 5)
- Der GA besitzt kein Wissen über seine Herkunft. Sobald er sich bewegt, werden alle anderen Knoten des „cPG“ vergessen (da von einer statischen Umgebung ausgegangen wird, mit Ausnahme der Bewegung des GA) und es kann kein Weg zurückgegangen werden. Der GA hat nur das Wissen zuvor besuchter Knoten aus alreadygonePath. (Abschnitt 7).

- Weiterhin ist es momentan so, dass in currloc/MyCurrentPositionNode jeweils ein Knoten gespeichert wird, der die aktuelle Position des geometrischen Agenten beschreibt. Möglicherweise könnte hier besser eine Liste von möglichen angenommenen Positionen des GA behandelt werden.
- Der GA inferiert, wenn er Teilgraphen des P-Graphen und des I-Graphen matcht, wobei P- und I-Graph zu einer jeweiligen Konjunktion von Propositionen korrespondieren. Matching liefert nur die höchstrangigen Bewertungen (jeder Match enthält eine Bewertung) zurück. Wenn sich diese allerdings als nicht „gut“ erweisen, gibt es keine Möglichkeiten auf alte Werte zurückgreifen zu können. Hier ist es vielleicht hilfreich, auch andere niederrangige Matches zu beurteilen, um Entscheidungsalternativen zu besitzen und diese der ATMS zugänglich zu machen. (Abschnitt 8.3)
- Um eine Aktion ausführen zu können, muss sich der GA (unter Zuhilfenahme weiterer Kriterien) für genau ein Match entscheiden (zwischen mehreren die denselben höchsten Rang besitzen). Nur die diesem Match zugehörigen Knoten werden in mem übernommen. Wenn es sich hierbei um den falschen Weg handelt, kann sich der GA total „verrennen“. Vielleicht wäre es auch sinnvoll, Wissen über andere Matches zu behalten und der ATMS zukommen zu lassen.

### **3. Unter welchen Umständen liefert der GA „Inkonsistenz“-Information? Wie sind sie begründet?**

- Wie kann der GA Wege richtig koreferenzieren, wenn er nur Teile des Weges sehen kann, weil z.B. Abschnitte durch Bäume oder Gebäude verdeckt werden. Werden diese richtig verfolgt, oder evtl. schon früher abgebrochen? (Abschnitt 7)
- Beim localPlanning müssen Objekte koreferenziert werden, damit die Instruktionen abgearbeitet und Aktionen durchgeführt werden können. Finden sich keine Objekte, kann der GA nichts matchen, also keine Knoten aus dem I-Graphen und dem P-Graphen verbinden. Deswegen muss der GA solange Überbrückungsfunktionen durchführen, bis er etwas koreferenzieren kann. Geschieht dies nicht entsteht Inkonsistenz. (Abschnitt 8.1.1)