

Carola Eschenbach, Christopher Habel

## **Kennenlernen des Geometrischen Agenten**

Voraussetzung:

Eclipse wurde gestartet, der GeoAgent wurde gestartet.

### **1) Was der GA so sieht**

Aus dem Menü links zu wählen:

Modus -> Fest vorgegebene Abläufe -> Immer der Nase nach

Aus dem Menü zu wählen:

GA Position -> bei Pavillon 6

Und jetzt sollte etwas passieren:

Der GA macht die Augen auf und baut eine interne Repräsentation von dem, was er so sieht auf.

Im Fenster 'Geometric Environment' wird durch Linien angedeutet, worauf sein Blick fällt. (Bitte erstmal nicht in dieses Feld hineinklicken !)

Im Fenster 'Field of View' wird dargestellt, was er sieht, also sowas wie seine 'Netzhautbilder', bei Bäumen und Häusern angereichert um die Vorstellung, wie das gesehene Objekt insgesamt aussieht.

(Das kann der geometrischen Umgebung recht nahe kommen, aber auch sehr stark abweichen. )

Im Fenster 'CRIL-Graph' erscheint nun sein 'mentales Modell' (eventuell muss man scrollen!).

Die Rechtecke stehen für die Objekte, die er sieht, die Kanten für die Relationen zwischen den Objekten.

Die Rechtecke kann man mit der Maus packen und bewegen.

Zu den 'Netzhautbildern' werden CRIL-Knoten mit den Sorten landmark, bzw. track erzeugt.

Aus den 'Netzhautbildern' können weitere Prädikate übernommen werden (z.B. bei Gebäuden und Bäumen Höheninformation, bei Bäumen das Prädikat 'baum', bei Gebäuden das Prädikat 'building' und die zugeordneten Label). Dieses geschieht bislang eher unsystematisch.

(Viel hängt davon ab, was in der Umgebungsdatei enthalten ist. Da wir keine komplexe Wahrnehmung implementieren wollten, lassen wir zu, dass auch Prädikate darin enthalten sind. z.B. das Prädikat 'uebergang' oder 'pfoertnerhaus' bei den entsprechenden Gebäuden. Auch diese Prädikate kann der GA 'wahrnehmen', wenn er die Objekte sieht.)

Es gibt auch immer einen Knoten (Sorte 'position') für die aktuelle Position des GA. (Dieser Knoten trägt selbst aber nicht die Information, dass er die aktuelle Position beschreibt.)

Sieht der GA einen Weg (track), dann fügt er automatisch Knoten für die Endpunkte des Wegs ein und setzt sie entsprechend in Relation zu dem Weg (ept). Liegt für zwei Knoten vom Typ position dieselbe geometrische Spezifikation vor, dann werden sie miteinander identifiziert (genauer: dann gibt es nur einen Knoten.)

Steht der GA auf einem Weg (den er dann auch automatisch wahrnimmt), dann werden weitere Knoten für mögliche Bewegungspfade eingefügt. Beginnend bei dem Pfad, auf dem er steht, bildet er für jeden Endpunkt den Pfad von der aktuellen Position zu dem Endpunkt (steht er auf einem Endpunkt, resultiert nur ein Pfad.)

Diese Pfade führen den Weg entlang ('along') und sind gerichtet (sie haben auch eine entsprechende geometrische Beschreibung.) Entsprechend werden die beiden Endpunkte als Startpunkt (spt) vs. finaler Punkt (fpt) in Relation zu dem Pfad gesetzt. Für alle Tracks, die vom finalen Punkt eine mögliche Fortsetzung bilden werden weitere Pfade eingeführt. Alle gebildeten Pfade starten bei der aktuellen Position und haben keine Schleifen.

Versucht, nachzuvollziehen, wie dieses Netz zu den gesehenen Objekten korrespondiert. Im ersten Schritt ist das noch vergleichsweise einfach, da der GA wenig sieht. Wenn alles klar ist, lasst ihn loslaufen, indem Ihr in 'Geometric Environment' hineinklickt.

Durch die Wahl von erst

Modus -> Fest vorgegebene Abläufe -> Immer der Nase nach  
dann

GA Position -> frei bestimmt

kann man sich dann eine eigene Position aussuchen (Klick!), der nächste Klick gibt an, wohin der GA schauen soll.

Im Sommer kann man erfahrungsgemäß nicht durch die Bäume durchsehen, im Winter schon.

So geht es auch dem GA. Unter dem Menu-Eintrag

Environment -> Sommer

Environment -> Winter

kann man die Jahreszeit wählen und damit auch steuern, wie viel Information der GA durch das 'Sehen' erhält. Entsprechend kann man hier wählen, ob der GA eigentlich die Bezeichnungen an den Häusern sieht, oder nicht.

Environment -> Label sichtbar

Environment -> Label nicht sichtbar

Benutzt 'immer der Nase nach' um einen Eindruck davon zu kriegen, was der GA in den verschiedenen Bedingungen 'sieht' und welche Inhalte er in seine interne Repräsentation packt.

## **2) Instruierte Navigation (fest verdrahtet).**

Aus dem Menü links zu wählen:

Modus -> Fest vorgegebene Abläufe -> Demo 1: von der Mensa zu Haus E

Danach läuft ein fester Ablauf ab, der durch das Klicken in 'Geometric Environment' weitergeschaltet wird.

Diesen kann man durch

Modus -> Fest vorgegebene Abläufe -> Demo 1: von der Mensa zu Haus E  
immer wieder von vorne starten.

Im Fenster 'CRIL-Graph' erscheint eine netzartige Repräsentation, die aus einer Instruktion gewonnen sein könnte. Die Farbe 'rosa' bei den Knoten zeigt an, dass die Objekte in der Instruktion genannt wurden.

Erster Klick!

Im ActionControl-Fenster teilt der GA mit, dass er annimmt, dass er sich an der Position befindet, die im Instruktionsgraphen durch 'n5' repräsentiert ist. (Das ist der Endpunkt eines Pfades, der in der Mensa beginnt und außerhalb endet.)

Klick !

Jetzt macht der GA wirklich die Augen auf, und wie bei 'Immer der Nase nach' erscheint der entsprechende Perzeptionsgraph. Allerdings wird ein Knoten des Perzeptionsgraphen schon mit dem Instruktionsgraphen über eine 'koref-Kante' in Beziehung gesetzt. Dies ist der Knoten, der für die eigene Position des GA in diesem Augenblick steht.

Der Wert 1.0 sagt dabei, dass er eine sehr hohe Übereinstimmung gefunden hat. Alle Kanten bekommen dieselbe Bewertung.

Einerseits hat der GA gerade angenommen, dass er am Ende eines Weges aus der Mensa heraus steht (und sich seitdem nicht mehr bewegt). Deshalb gibt es eine koref-Kante zu n5.

Da der GA als nächstes einen Weg nach links sucht, nimmt er auch noch an, dass er sich schon am Anfang des gesuchten Weges befindet. (s.a. die Meldung im ActionControl-Fenster)

Dies wird auch durch eine entsprechende koref-Kante ausgedrückt.

Klick !

Im ActionControl-Fenster teilt der GA nun mit, nach welchem Weg er sucht.

Durch ein dunkleres Gelb wird ganz kurz ein Teil des Instruktionsnetzes hervorgehoben, der den Erwartungen des GA, was er jetzt sehen / erkennen soll, entspricht. (In diesem Teilgraphen ist auch der Knoten enthalten, von dem er annimmt, dass er sich dort befindet.) Da in diesem Teilnetz auch ein Regionenknoten vorkommt, der für die Region links vom Ausgangspunkt des Weges liegt, generiert der GA nun im Perzeptionsgraphen einen entsprechenden Knoten. Im 'Field of View' wird entsprechend ein Sektorensystem eingezeichnet, das der geometrischen Repräsentation dieser Region entspricht. Zudem bestimmt der GA, welche der Punkte und Wege, die er sieht, in der Region liegen, und trägt entsprechende Lokalisierungs-Kanten in den Perzeptionsgraphen ein ('loc' ist enthalten in, 'ploc' ist teilweise enthalten in, 'nloc' ist gar nicht enthalten in).

Klick !

Der GA führt einen Vergleich der Netze durch und entscheidet sich für eine Zuordnung von Perzeptionsknoten zu Instruktionsknoten, trägt die entsprechenden koref-Kanten ein und löscht / vergisst den Rest (von dem, was er gerade gesehen hat).

Im ActionControl-Fenster teilt er auch noch mit, wie viele Alternativen er gehabt hätte, die ebenso gut bewertet waren.

Klick !

Der GA läuft los und macht dann gleich 'die Augen auf' und es werden entsprechende Modelle aufgebaut.

Dabei gibt es dann wieder einen Knoten im Perzeptionsnetz, der der eigenen Position entspricht, und mit dem Endpunkt eines Weges (n9) und dem Anfangspunkt eines anderen Weges (n10) koreferenziert ist.

Zu beachten ist vielleicht auch noch, dass n9 jetzt mit zwei Perzeptions-Knoten koreferenziert ist. Der eine steht für den Punkt, den er beim ersten 'Schauen' als Endpunkt eines Weges gesehen hat, dem er dann gefolgt ist. Der zweite steht für den Punkt, den er beim zweiten 'Schauen' als Repräsentation seiner Position eingefügt hat. !! Derzeit werden Knoten, die aus verschiedenen Wahrnehmungsereignissen stammen, nicht miteinander verglichen oder in der Repräsentation identifiziert. !!

Klick !

Der GA sucht den nächsten Weg und bestimmt dafür die Region zwischen zwei Häusern.

Klick !

Er entscheidet sich für eine Zuordnung und trägt sie ein.

Im weiteren hat der GA 'keinen Plan mehr' und läuft einfach nur weiter.  
Die Grundstruktur des Schließens des GA sollte nun aber halbwegs klar sein.

### **3) Instruierte Navigation**

Aus dem Menü links zu wählen:  
Modus -> ActionPlanner -> AP\_n

Die vier Varianten folgen demselben Grundschema. Sie unterscheiden sich darin, welche Überbrückungshandlungen der GA durchführt, wenn er nicht das sieht, was er zu sehen erwartet.

Nun bittet der GA um eine Instruktion  
Dafür gibt es im Grunde 3 Möglichkeiten:

1) InstructionFile -> nb\_test\_n  
wählen.

2) InstructionFile -> open test file  
und dann eine entsprechende Datei öffnen

3) Eintippen der Instruktion in 'Enter the instruction here' und dann mit 2\* return oder Klick auf NLI abschicken und warten.

(Die Sprachverarbeitung ist natürlich sehr eingeschränkt.)

Wenn er die Instruktion bekommen und verstanden hat, erscheint wieder ein entsprechendes CRIL-Netz und auch der 'Aktionsplan', der aus der Instruktion extrahiert wurde im ActionControl-Fenster.

Links stehen die Anweisungen, rechts die Information über deren temporale Ordnung.

Nun möchte der GA auch noch in der Umgebung plaziert werden.  
GA Position -> X

Anschließend beginnt er gleich mit der Abarbeitung des Plans. (Dabei unterbricht er sich nicht so oft, wie bei der fest verdrahteten Demo).

Im ActionControl-Fenster kommentiert er seinen internen Zustand und seine Handlungen.

Zwischendrin könnt Ihr auch die Wahrnehmungsbedingungen variieren und damit prüfen, wie sie sich auf den 'Informationszustand' und die Möglichkeiten des GA, sich zu entscheiden, auswirken.

Unter idealen Bedingungen, läuft der GA schon ganz gut. Wenn er an die falsche Stelle gesetzt wird und/oder in der Wahrnehmung stark eingeschränkt wird, kann man das nicht so sagen.

### **4) CRIL-Netz**

Enthalten Knoten mit Identifikatoren (IDs) der Sorten

- building ('b')
- track (Strasse, Weg, 't')
- position (Punkt, 'n')
- region ('r')
- path (Trajektorie, gerichtet, 'w')

denen zugeordnet sein können

- Namen, Label: 'E'

- Eigenschaften, Prädikate: 'haus' (Der Unterschied zwischen Sorte und Eigenschaft besteht nur darin, dass wir die 'Sorte' benutzen, um den internen IDs zu erzeugen.)

Enthalten Kanten (mit Bezeichnungen) für

- Relationen zwischen den durch die Knoten repräsentierten Objekte

Wenn zwischen zwei Knoten zwei Relationen bestehen, dann werden leider die Bezeichnungen übereinander dargestellt, so dass das ganze nicht gut lesbar ist.

Die Eigenschaften und Relationen, die repräsentiert werden können, sind viel differenzierter als die in der Navigation tatsächlich berücksichtigten Eigenschaften und Relationen.

Mit folgenden Relationen kann der GA in der Navigation mehr oder weniger anfangen:

## **Räumliche Relationen**

Binäre Relation zwischen einem Referenzobjekt und einer Raumregion

left\_space, right\_space, prec, succ, at

Ternäre Relation zwischen zwei Referenzobjekt und einer Raumregion

between

Binäre Relationen zwischen einem Objekt und einer Raumregion

loc ((vollständig) lokalisiert in)

nloc ((vollständig) nicht lokalisiert in)

ploc (teilweise lokalisiert in)

Binäre Relationen zwischen einem Pfad und einer Position

spt (Startpunkt)

fpt (Endpunkt)

ipt (innerer Punkt)

Binäre Relationen zwischen einem Pfad und einer Landmarke (Track oder Building)

along

## **Andere Relationen**

Koreferenzrelationen zwischen Knoten der Instruktion und Knoten der Perzeption

koref

## **Aktionsplan**

Liste von Aktionsbeschreibungen der folgenden Typen

go a1 w1 (gehe (gerichteten) Pfad w1)

be\_at a2 r2 (stelle sicher, dass Du Dich in Region r2 befindest)

view a3 b3 (stelle sicher, dass Du b3 siehst)

ch\_orient a4 w4 (richte Dich entsprechend w4 aus)

## **Linear CRIL**

ist die Form, in der die Informationen von der Sprachverarbeitungskomponente (in PROLOG) an den GA (in Java) übergeben werden.

## **5) Übersetzung Deutsch -> CRIL-Instruktionen**

Aus dem Menü links zu wählen:

Modus -> ActionPlanner -> AP\_n

Jetzt in dem Instruktions-Feld des Fensters eine einfache Instruktion eingeben (z.B. 'geh nach links') und durch den NLI-Knopf oder 2\*Return abschließen.

Danach in das Eclipse-Fenster wechseln, und unten die 'Console' anzeigen lassen. Hier erscheint die Information des Netzes in der linearen Form.  
(Die anschließende Warnung bitte ignorieren.)

Wie diese Repräsentation aufgebaut wird, soll uns in diesem Projekt nicht interessieren. Versucht aber aus der Beziehung zwischen sprachlichem Ausdruck und CRIL-Repräsentation einen Eindruck von der Bedeutung der CRIL-Ausdrücke zu gewinnen.

Längere Beispiele finden sich im Ordner 'Instruktionen' der von Eclipse aus zugänglich ist. Zeilen, die mit 'echo' beginnen, werden dem Benutzer angezeigt.

'node' deklariert einen neuen Knoten und ordnet die Sorte zu.

'descr' erlaubt die Ergänzung von Relationen.

'type' erlaubt die Ergänzung von Prädikaten/Eigenschaften

'name' erlaubt die Ergänzung von Namen/Labeln

'node' deklariert und beschreibt eine neue Aktion.

'temp' gibt zeitliche Information über die Aktionen (wird zur Zeit ignoriert.)

**! Beim Einlesen entsprechender Sequenzen werden für die Bezeichner ('r155', ...) neue KnotenIDs eingeführt !**

Hier der Inhalt von ce\_test\_02.txt:

```
echo Vom Eingang zu Haus E
echo wenn du beim pfoertnerhaus stehst dann siehst du ein hohes haus dot gehe an dem haus
entlang dot geh nach links dann siehst du haus e
node building r150
node building r20
node building r8
node path r130
node path r61
node region r135
node region r2
node region r93
node position r156
node position r157
descr fpt r130 r157
descr spt r130 r156
descr ploc r61 r93
descr at r20 r93
descr at r8 r2
descr nloc r156 r135
descr loc r157 r135
descr left_space r156 r135
descr alongR r93 r61
type pfoertnerhaus r8
type haus r20
type hoch r20
type haus r150
name E r150
action be_at r12 r2
action view r16 r20
action go r60 r61
action go r125 r130
action view r148 r150
temp seq r12 r16
temp seq r16 r60
temp seq r60 r125
temp seq r125 r148
```

## **6) Entwicklung von CRIL-Instruktionen**

Im Ordner 'Instruktionen' gibt es die Datei `gi_text_01.txt`.

Diese darf verändert werden, um die Abbildung des linearen CRIL auf die Netz-Darstellung von CRIL auszuprobieren.

Entwickeln Sie eine Routeninstruktion in linearem CRIL. Die prinzipiell verwendbaren Prädikate und Relationen sind in der Klasse `GATaxonomies` im Paket `geoAgent.cril` zu finden. Allerdings kann er bei der Navigation nur die oben angegebenen Relationen nutzen.

Ziel der nächsten Übungen wird es sein, per Hand graphische Instruktionen in lineares CRIL abzubilden.

## 7) Übersetzung einer Wegeskizze in lineares CRIL

Zielsetzung: Vertieftes Kennenlernen des GA (und von CRIL) und Zieldefinition für das Projekt.  
Aufgabe:

- 1) 'Übersetzen' Sie eine der in der ersten Sitzung betrachteten Wegeskizze (per Hand) in lineares CRIL, so dass der GA mit dem Resultat möglichst etwas anfangen kann.
- 2) Dokumentieren Sie dabei auftretende Fragen (und ggf. Antworten). Welche Aspekte konnten nicht übertragen werden?
- 3) Bereiten Sie eine Kurzvorstellung Ihres Ergebnisses und Ihrer Fragen im Plenum am 6.11.08 vor.

Dazu ist zu beachten:

1) Es müssen Aktionsbeschreibungen (insbesondere go-Aktionen) enthalten sein, da der GA nur etwas tut, wenn er einen Plan hat.

2) go-Aktionen werden mit einem pfad (w1) spezifiziert

```
action go a1 w1
```

Solche Pfade können nur in der Umgebung sinnvoll identifiziert werden, wenn sie räumlich beschrieben sind. Dazu können die Pfade mit Punkten (position), Regionen oder Straßen (track) in Beziehung gesetzt werden.

Start-/Endpunktzuordnung:

```
descr spt/fpt w1 p1
```

Relation zu Region:

```
descr loc/ploc/nloc w1 r1
```

Relation zu Track:

```
descr along t1 w1
```

Auch die Punkte können in Relation zu Regionen gesetzt werden und so indirekt der räumlichen Spezifikation des Pfades dienen:

Relation zu Region:

```
descr loc/ploc/nloc p1 r1
```

3) Regionsobjekte sind basierend auf einem Referenzobjekt und einer räumlichen Funktion definiert. Die Links-Region von lm1:

```
descr left_space lm1 r1
```

Als Referenzobjekte können insbesondere auch Anfangspunkte von Pfaden dienen, dann ist eine Region, links von diesem Punkt gemeint.

```
descr left_space p1 r1
```

Reihenfolgerestriktionen:

node-Deklarationen müssen erfolgen, bevor das entsprechende Objekt in einer anderen Anweisung auftaucht.

action-Angaben müssen vor seq-Angaben auftreten.

## **8) Skizzen-Baukasten für den Geometrischen Agenten**

Zielsetzung: Sammlung von Ideen zum Inhalt der graphischen Instruktionen

### **Landmarken**

- Welche Arten von Objekten sollen für die Instruktion verwendbar sein (Häuser, Bäume, Lampen, Schilder, Rasenflächen, Blumenbeete, Schranken, ...)?
- Welche geometrischen Objekte sollen für die Landmarken verwendet werden (Rechtecke, Polygone, Ellipsen, ...)?
- Über welche Art von Information sollen die Objekte spezifizierbar sein (Höhe, Farbe, Label, ...)?

### **Tracks und Track-Konstellationen**

- Welche Arten von Objekten, auf denen sich der GA sich bewegen kann, sollen für die Instruktion verwendbar sein (Straßen, Wege, Kreuzungen, Einmündungen, Plätze)?
- Welche geometrischen Objekte sollen für die Landmarken verwendet werden (Linie, Doppellinie, ...)?
- Wie können Wege zu Kreuzungen zusammengesetzt werden, bzw. Kreuzungen in Wege (Wegabschnitte) zerlegt werden?
- Sollen Tracks und Track-Konstellationen über weitere Arten von Information spezifizierbar sein (Straßenbelag, Zebrastreifen, ...)?

### **Aktionen und Pfade**

- Wie sollen Start, Ziel und Wegverlauf eingegeben und dargestellt werden?
- Sollen Alternativ-Pfade möglich sein?
- Soll die Gliederung in Einzelaktionen durch den Instrukteur oder durch das GI-Paket erfolgen?

### **Räumliche Relationen (wird auch/insbesondere in der Sitzung am Freitag behandelt)**

- Welche räumlichen/geometrischen Relationen aus der Graphik müssen in die Instruktion kodiert werden?
  - zwischen Landmarken
  - zwischen Track und Landmarken
  - zwischen Pfad/Aktion und Tracks/Trackkonstellationen
  - zwischen Pfad/Aktion und Landmarken
- Wann sind die Objekte als Ganzes in Relation zu setzen, wann sind charakteristische Punkte (Start-/End-/Mittelpunkt) eines Objektes in Relation zu setzen?