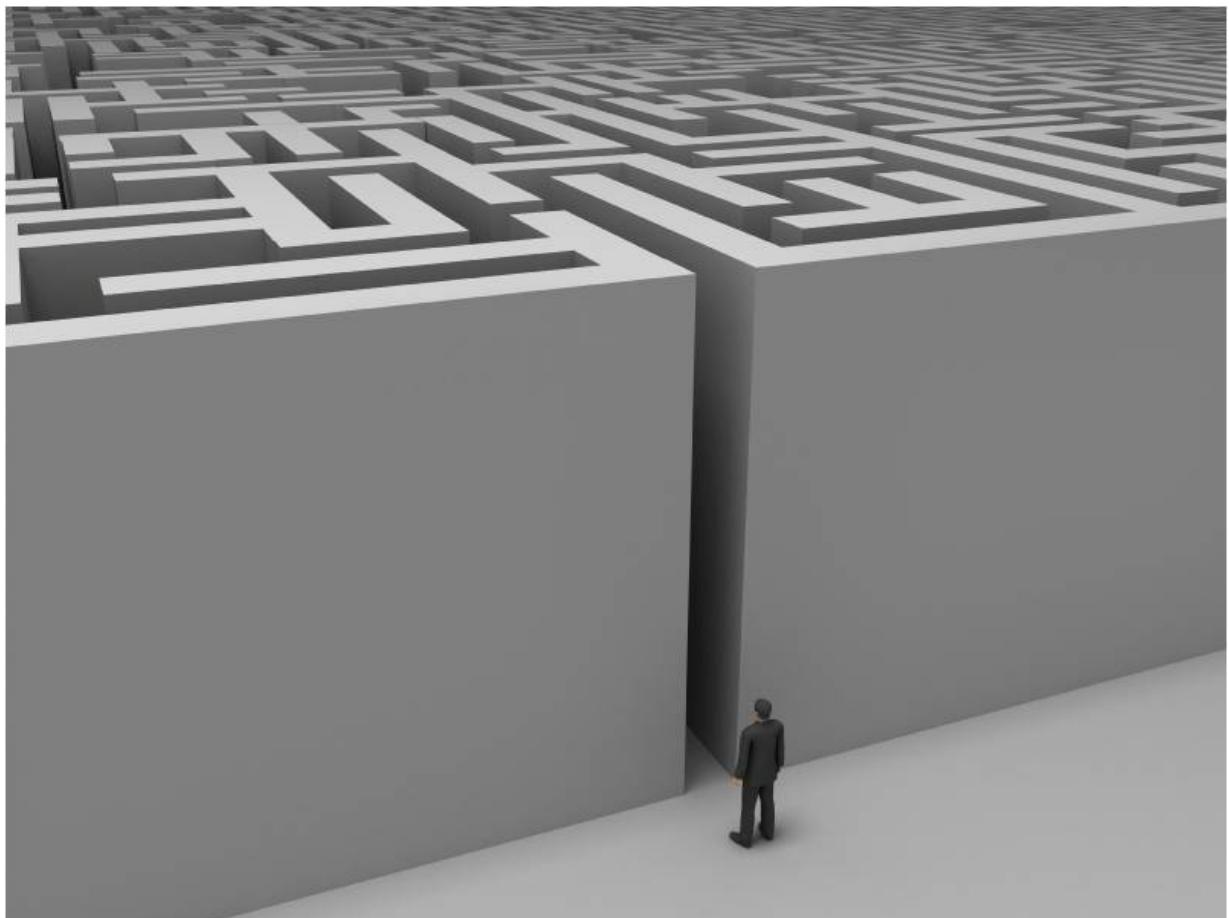


Artificial Intelligence - FS2018 - Berner Fachhochschule

Projektaufgabe Prolog

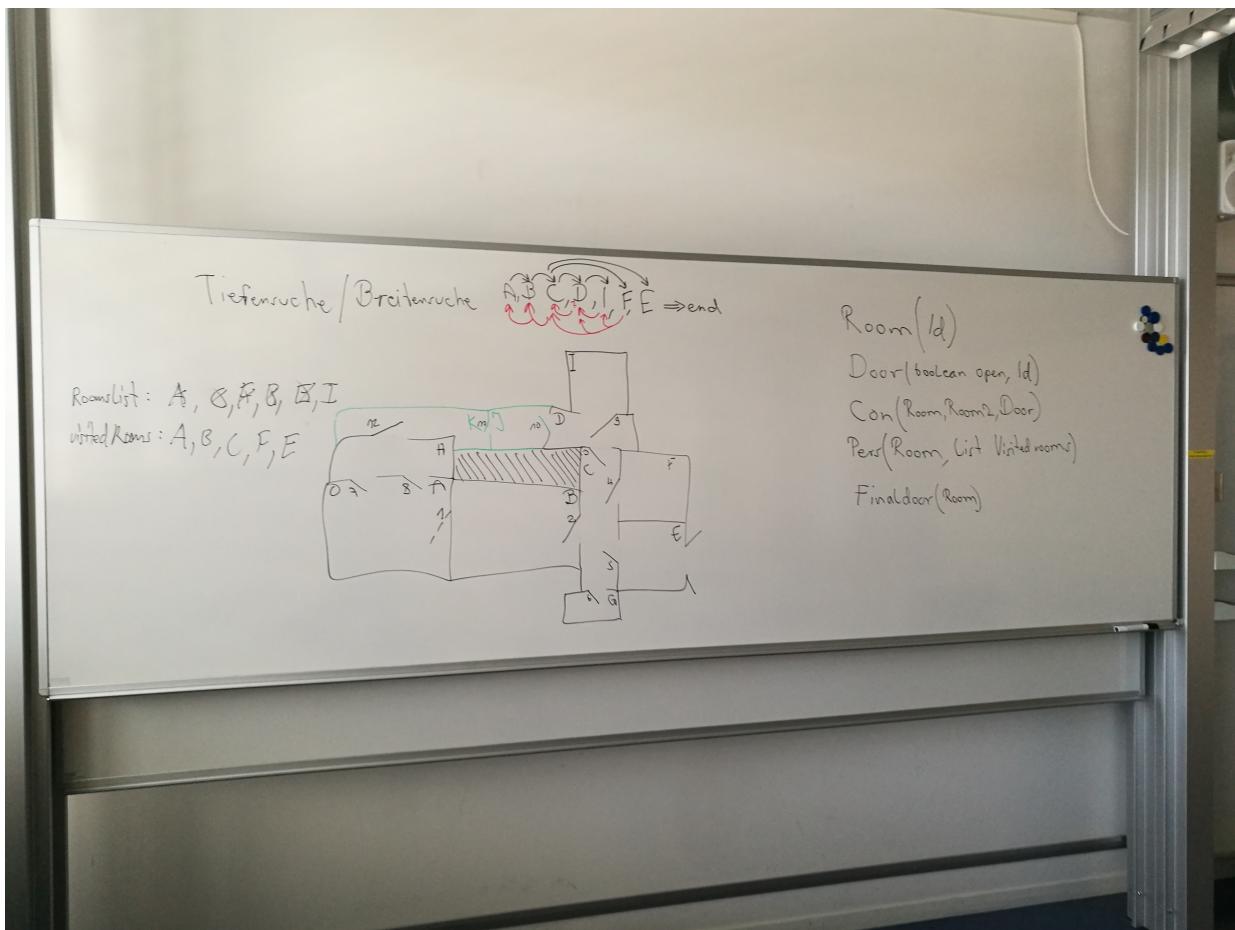
Filip Hofer, Kevin Suter



Die Aufgabenstellung

Ein Agent muss den Weg aus einer labyrinthartigen Welt in die Freiheit finden. Die Welt besteht aus Räumen, die über Türen miteinander verbunden sind. Türen können offen oder verschlossen sein. Es gibt Wächter, die durch das Labyrinth laufen und Räume besetzen. Wenn Wächter durch Türen laufen, vergessen sie diese jeweils zu schließen und somit können vorher verschlossene Türen plötzlich offen sein. Der Agent kann sich nicht in einen Raum bewegen, in dem sich ein Wächter befindet. Es ist jedoch erlaubt, dass ein Wächter einen Raum betritt, in dem sich der Agent befindet.

Die Spielwelt



Folgende Prädikate wurden für die Modellierung der Welt verwendet:

| Prädikat | Beschreibung | Parameter |
|------------------------|---|---|
| room(NAME) | Bezeichnet einen Raum im Labyrinth | NAME: Name des Raumes |
| door(NUMMER) | Bezeichnet eine Türe, welche zwei Räume verbindet | NUMMER: Nummer der Türe |
| con(ROOM1,ROOM2, DOOR) | Verbindet zwei Räume über eine Türe. | ROOM1: Der erste Raum ROOM2: Der zweite Raum DOOR: Türe, welche die beiden Räume verbindet. |

Die Suche des Agenten

Für die Suche des Agenten aus dem Labyrinth wurden zwei Regeln in Prolog definiert:

startGame

Beschreibung

Initialisiert das Spiel und startet die Suche mit dem Aufruf der Regel *gameLoop*. Es werden ebenfalls die offenen Türen und Wächterpositionen für jeden Spielschritt übergeben (Liste von Listen).

Parameter

- **StartRoom:** Raum, in dem sich der Agent am Anfang befindet

Bedingungen (UND)

- **gameLoop:** Sobald der Game Loop *true* zurückgibt, ist das Spiel erfolgreich durchgelaufen.

gameLoop

Beschreibung

Die eigentliche Wegsuche bzw. der Ablauf des Spiels. Jeder rekursive Aufruf entspricht einem Spielschritt.

Parameter

- **CurrentRoom:** Der aktuelle Raum, in dem sich der Agent befindet.
- **VisitedRooms:** Liste aller Räume, die vom Agenten schon besucht wurden.
- **NextRoom:** Der nächste Raum, welcher vom Agenten aufgesucht wird
- **OpenDoors:** Liste derjenigen Türen, die pro Spielschritt (Index in der Liste) offen sind.
- **GuardsPositions:** Liste derjenigen Räume, die pro Spielschritt (Index in der Liste) mit Wächtern besetzt und somit vom Agenten nicht betreten werden können.

Bedingungen (UND)

- **con(CurrentRoom,NextRoom,Door):** Der aktuelle Raum ist über die Tür *Door* mit einem weiteren Raum *NextRoom* verbunden.
- **member(Door,[_|OpenDoors]):** Die Tür, über die der CurrentRoom mit dem NextRoom verbunden ist, ist offen.
- **not(member(NextRoom,GuardsPositions)):** Im nächsten Raum befindet sich kein Wächter.
- **not(member(CurrentRoom,VisitedRooms)):** Der nächste Raum wurde nicht bereits besucht.
- **exit = NextRoom:** Der nächste Raum ist die Freiheit.

ODER

gameLoop(NextRoom,[CurrentRoom|VisitedRooms], NewOpenDoors, NewGuardsPositions): Gehe in den nächsten Raum und hoffe, dass du von dort aus den Ausgang findest (nächster Spielschritt mit neuen offenen Türen und neuen Wächterpositionen).