

Programmierung von Agenten in GUPU

Allgemeine Anforderungen

- Seiteneffektfrei — Alle beabsichtigten Effekte müssen explizit sichtbar sein
- Zustand des Agenten explizit!
- Monotonie?

Agentenarchitektur — Agentenschnittstelle

Schnittstelle des Agenten A : Zu definierende Relationen

$\text{init}_A(S_0)$:

S_0 ... Anfangszustand des Agenten

$\text{percept_action}_A(\text{Percept}, \text{Action}, S_0, S)$:

Percept ... die momentanen Wahrnehmungen

Action ... die Reaktion auf Percept

S_0 ... Zustand *vor* Wahrnehmung und Aktion

S_1 ... Zustand *danach*

$\text{maybe_obj}_A(\text{Position}, \text{Object}, S)$:

Im Zustand S glaubt Agent, dass Object sich an Position befinden kann.

Agentenarchitektur

Praktische Umsetzung durch Prädikat mit einer Klausel pro Relation

a007(init(S0)) ←

... .

a007(percept_action_(Percept, Action, S0, S)) ←

... .

a007(maybehere_obj_(Position, Object, S)) ←

... .

Verwendung:

← a007(init(S0)), a007(percept_action_(perc(no,no,no,no,no)), Action, S0, S).

Was tut Agent als erstes?

Agentenarchitektur — Randbedingungen

Verhalten des Agenten sollte eindeutig bestimmt sein.

Randbedingungen für Relationen

- Eindeutige Aktionen

$\nexists \text{ dif}(A_I, A_{II}), \text{a007}(\text{percept_action_}(P, A_I, S_0, -)), \text{a007}(\text{percept_action_}(P, A_{II}, S_0, -))$.

- Allgemein deterministisch

$\nexists \text{ dif}(S_I, S_{II}), \text{a007}(\text{percept_action_}(P, -, S_0, S_I)), \text{a007}(\text{percept_action_}(P, -, S_0, S_{II}))$.

Nachteil: Meist nicht direkt beweisbar (\nexists statt \nexists), da Lösungsmenge von S_0 bereits unendlich groß.

- Referentiell transparent

Kein impliziter Einfluß zwischen Zuständen

Gilt, falls Zustände immer variablenfrei sind.

Agentenarchitektur — Darstellung der zeitlichen Abfolge

← WAH = s(w(0):ag0/1,H), snapshot_(WAH,Expl).

← snapshot_(s(w(0):ag0/1,H),Expl).

@@ % Expl = "Welcome, agent!":0, H = [].

@@ % Expl = "Breeze!": -1, H = "f".

@@ % Expl = "Breeze!": -2, H = "ff".

@@ % Expl = []: -3, H = "fff".

@@ % Expl = "Stench! Breeze! Glitter!": -4, H = "ffff".

@@ % Expl = "Stench! Breeze! Got some gold!": -5, H = "ffffg".

@@ % Expl = "Stench! Breeze!": -6, H = "ffffgr".

@@ % Expl = "Stench! Breeze!": -7, H = "ffffgrr".

@@ % Expl = []: -8, H = "ffffgrrf".

@@ % Expl = "Breeze!": -9, H = "ffffgrrff".

@@ % Expl = "Breeze!": -10, H = "ffffgrrffr".

@@ % Expl = []: -11, H = "ffffgrrffrf".

@@ % Expl = "I am outta here.":988, H = "ffffgrrffrfc".

@@ %% 13 Lösungen

- Lösungssequenz entspricht zeitlichem Ablauf
- Durch Geschichte (H) Ablauf eindeutig

Programmierung des Agenten

Agent, der nichts vergisst:

a007(percept_action_(Percept, Action, History, [Percept-Action|History])) ←
history_percept_action(History, Percept, Action).