

## Лекция 9: Планиране на действията

**Същност на планирането.** Декомпозиране на задачата на подзадачи и построяване на поредица от елементарни (в определен смисъл) стъпки, чието последователно изпълнение може да доведе до решаване на първоначално поставената задача.

*Примерна предметна област:* свят на кубовете.

### СИТУАЦИОННО СМЯТАНЕ

Ситуационното смятане (Situation Calculus, McCarthy & Hayes, 1969) е подмножество на предикатното смятане от първи ред, създадено с цел формализиране на описанието на ситуациите и действията при решаване на задачи от областта на планирането.

Основни елементи и характеристики на ситуационното смятане:

- **Действията и фактите** (т.е. елементите на плана и елементите на описанието на състоянието на сцената) се представят чрез термове, например:  $puton(A,B)$ ,  $on(A,B)$ ;
- **Верните факти в дадена ситуация** се описват с помощта на предиката *holds*. Например  $holds(on(A,B),S)$  означава, че  $A$  се намира върху  $B$  в ситуацията  $S$ ;
- **Ситуациите** са термове, описващи състоянията на света (предметната област, сцената). При зададена начална ситуация, например  $s0$ , всяка следваща ситуация се получава чрез функцията *result*. Например  $result(puton(A,B),s0)$  представлява ситуацията, получена в резултат на прилагане на действие  $puton(A,B)$  в ситуация  $s0$ . По този начин могат да се записват и по-сложни ситуации:  $result(puton(C,A),result(puton(A,B),s0))$ ;
- Удобно представяне на връзката между ситуацията и плана за достигането ѝ на Пролог: началната ситуация се представя чрез празния списък  $[]$ , действието  $puton(A,B)$  в ситуация  $S$  води до ситуация  $[puton(A,B)|S]$ . Така всяка ситуация директно описва **плана** (списъка от действията, натрупани в обратен ред) за своето достигане;
- **Аксиомите** описват **действията** (с техните предусловия и ефекти). Те се представят чрез импликации, свързващи предусловията и ефектите (т.е. промените, които предизвикват), изразени с помощта на предиката *holds*. Например:  $holds(clear(A),S) \wedge holds(clear(B),S) \rightarrow holds(on(A,B),result(puton(A,B),S))$
- В резултат от изпълнението на дадено действие ситуацията се променя, следователно е необходимо да се постулира също кои факти остават верни и в новата ситуация (т.е. какво от състоянието на сцената не се променя при извършването на действието). Това става чрез т. нар. **фреймови аксиоми**. Например:  
 $holds(clear(X),S) \wedge \neg eq(X,B) \rightarrow holds(clear(X),result(puton(A,B),S))$
- За всеки факт, който се използва при представянето (т.е. в описанието на състоянието на сцената), трябва да се зададе съответна фреймова аксиома. Следователно, възниква необходимост от голям брой фреймови аксиоми (това е

същността на т. нар. рамков или **фреймов проблем, frame problem**). По такъв начин представянето се лишава от елегантност, а търсенето става неефективно.

Системата STRIPS може да бъде разглеждана в частност като сполучлив опит за решаване на фреймовия проблем. Тук се предполага, че всяка формула от описанието на сцената, която е била истина преди изпълнението на действието и не принадлежи на списъка на изтриванията на това действие, е истина и след неговото изпълнение.

## **ПОДХОД, ИЗПОЛЗВАН В СИСТЕМАТА STRIPS**

### **Описание на състоянията и целта**

Състоянията (начални, междинни, целеви) се представят чрез конюнкции от термове, в които се използват предикатите *on*, *ontable*, *clear*, *holding*, *handempty*.

Примери:

$on(A,B) \wedge ontable(B) \wedge clear(A) \wedge holding(C)$

$ontable(A) \wedge on(B,A) \wedge handempty$

$on(x,A) \wedge on(y,x)$

### **Моделиране на действията на робота**

Елементарните действия на ръката на робота се представят с помощта на продукционни правила, всяко от които има три компонента:

- формула на предварителното условие (Precondition);
- списък на изтриванията (Delete List);
- формула на добавянията (Add Formula).

Формулата на предварителното условие е аналог на лявата страна на правилото при традиционните продукционни правила. Списъкът на изтриванията и формулата на добавянията са аналог на дясната страна на правилото при традиционните продукционни правила.

### **Работа на интерпретатора на правилата**

Извършва извод, управляван от целите (обратен извод). Поддържа се *стек на целите* и усилията по решаването на задачата се съсредоточават върху целта във върха на стека. Работата на интерпретатора на правилата се прекратява при получаване на празен стек на целите.

- *Първоначално състояние на стека на целите*: съдържа само главната цел.
- *Действие при удовлетворена цел на върха на стека на целите*: такава цел се изтрива от върха на стека и евентуално използваната субституция на съответствието се прилага към елементите на стека, разположени под разглежданата цел.
- *Действие при неудовлетворена цел на върха на стека на целите*: зависи от типа на неудовлетворената цел (еднолитерална или съставна). Преодоляване на проблемите, свързани с т. нар. *взаимодействие на целите*.

### **Йерархично планиране**

Планът се изгражда на няколко равнища – като последователност от по-сложни действия, всяко от които може също да бъде обект на планиране.

Въвежда се *отлагане на изпълнението* (и проверката) на някои от предварителните условия. За целта се определя *йерархия на условията*, свързана обикновено с трудността на постигането на съответната цел.

## ***РЕАКТИВНО ПЛАНИРАНЕ***

Осъществява се при системи, в които състоянието на сцената се променя не само в резултат на действията на агента, чийто действия се планират. Например, в света на кубовете, на сцената освен робота може да действа и малко дете, което разбърква по произволен начин предметите върху масата.

Пример за реактивна планираща система с реална сложност: симулатор на военни действия.

Методите за планиране в такива системи са съвсем други (класическите методи за планиране не вършат работа в тези системи).