

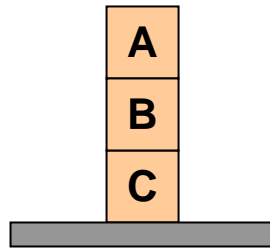
Изкуствен интелект - летен семестър, 2023/2024 учебна година

***Тема 4, част 2:
Планиране на действия***

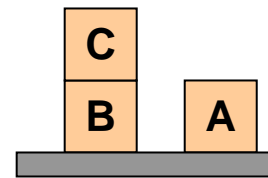
Дефиниция на задачата за планиране

- 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.
- Частен случай на търсене в пространството на състоянията, при което се използва логическо представяне на състоянията (в частност целите) и операторите.

Примерна предметна област: планиране на действията на робот в „света на кубчетата“.



начално състояние



целево състояние

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:

$$\mathit{holds}(\mathit{clear}(A), S) \wedge \mathit{holds}(\mathit{clear}(B), S) \rightarrow \\ \mathit{holds}(\mathit{on}(A, B), \mathit{result}(\mathit{puton}(A, B), S))$$

- 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:

$$\text{holds}(\text{clear}(X), S) \wedge \neg \text{eq}(X, B) \rightarrow \\ \text{holds}(\text{clear}(X), \text{result}(\text{puton}(A, B), S))$$

- 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.

Алгоритъмът за планиране в системата STRIPS представлява конкретизация на алгоритъма MEA.

Подход, използван в системата STRIPS

Описание на състоянията и целта

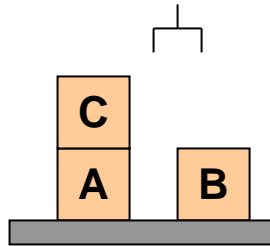
Състоянията (начални, междинни, целеви) се представят чрез конюнкции от термове, в които се използват предикатите *on*, *ontable*, *clear*, *holding*, *handempty*.

Примери:

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

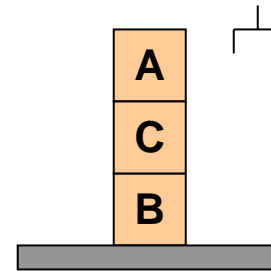
$ontable(A) \wedge on(B,A) \wedge handempty$

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



начално състояние

***on(C,A) ∧ ontable(A) ∧
ontable(B) ∧ clear(C) ∧
clear(B) ∧ handempty***



целево състояние

on(C,B) ∧ on(A,C)

Моделиране на действията на работа

Елементарните действия на ръката на работа се представят с помощта на оператори - правила, всяко от които има три компонента:

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

- ***Precondition***: ППФ от предикатното смятане от първи ред, която трябва да бъде логическо следствие от фактите в описанието на текущото състояние, за да бъде съответното правило приложимо към това състояние на сцената.
- ***Delete List***: списък от литерали, който задава елементите от описанието на състоянието на сцената, които ще престанат да бъдат верни след изпълнението на правилото (оператора).
- ***Add Formula***: конюнкция от литерали, които трябва да се добавят към описанието на текущото състояние, за да се получи новото състояние, което е резултат от изпълнението на даденото правило.

**Списък на правилата (операторите), описващи
елементарните действия на ръката на робота**

pickup(x)

P&D: *ontable(x), clear(x), handempty*

A: *holding(x)*

putdown(x)

P&D: *holding(x)*

A: *ontable(x), clear(x), handempty*

stack(x,y)

P&D: *holding(x), clear(y)*

A: *handempty, on(x,y), clear(x)*

unstack(x,y)

P&D: *handempty, on(x,y), clear(x)*

A: *holding(x), clear(y)*

Алгоритъм за планиране

Поддържа се *стек на целите* и усилията по решаването на задачата се съсредоточават върху целта във върха на стека. Работата на алгоритъма за планиране се прекратява при получаване на празен стек на целите.

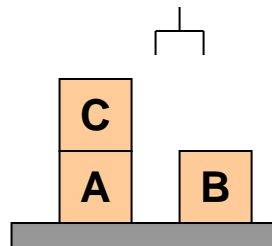
- *Първоначално състояние на стека на целите:* съдържа само главната цел.
- *Действие при намиране на удовлетворена цел на върха на стека на целите:* такава цел се изтрива от върха на стека и евентуално използваната субституция на съответствието се прилага към елементите на стека, разположени под разглежданата цел.

- *Действие при намиране на неудовлетворена цел на върха на стека на целите:* зависи от типа на неудовлетворената цел (еднолитерална или съставна).
 - Ако целта е съставна, в стека се добавят всички нейни компоненти.
 - Ако целта е еднолитерална, се търси правило, което съдържа във формулата на добавянията си литерал, съпоставим с тази цел. Разглежданата еднолитерална цел се заменя със съответния частен случай на избраното правило и над това правило се поставя съответният частен случай на предусловието на същото правило.

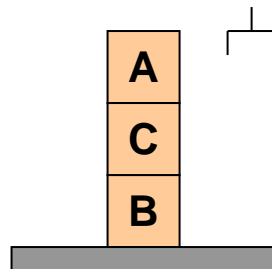
- *Действие при намиране на правило на върха на стека на целите:* правилото се изпълнява върху текущото състояние на сцената и в резултат състоянието се променя. Правилото се изтрива от върха на стека на целите и се добавя към плана.
- Преодоляване на проблемите, свързани с т. нар. *взаимодействие на целите*.

Пример

Начално състояние: $on(C,A) \wedge ontable(A) \wedge ontable(B) \wedge clear(C) \wedge$
 $clear(B) \wedge handempty$



Целево състояние: $on(C,B) \wedge on(A,C)$



План:

{unstack(C,A), stack(C,B), pickup(A), stack(A,C)}

В системата STRIPS се използват различни *управляващи стратегии*:

- За избор при наличие на алтернативни възможности (например: при наличие на конюнкция от литерали и възможност за избор на реда на поставянето им във върха на стека на целите; при наличие на повече от един приложим оператор и др.);
- За откриване и премахване на поредици от излишни действия.

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).