

Планирование в пространстве состояний

Первым планировщиком, осуществляющим планирование в пространстве состояний, является STRIPS (STanford Research Institute Problem Solver). STRIPS изначально разрабатывался для решения задачи формирования плана поведения робота, перемещающего предметы через множество помещений.

Собственно, идея алгоритма STRIPS заимствована из системы GPS [1]. Метод, использованный в GPS, назывался анализ средств и целей (means-ends analysis). Он подразумевает рассмотрение в текущем состоянии тех действий, которые имеют отношение к цели. Однако, при таком подходе возникает следующая проблема: применять ли действия, связанные с целью, немедленно, как только они найдены или же приостановить применение действия до тех пор, пока не будут найдены все действия имеющие отношение к цели? STRIPS применяет действия не откладывая, достигая каждой цели по отдельности.

МакДермот показал, что эффективность планирования с использованием метода анализа средств и целей может быть намного повышена задержкой применения действия до тех пор, пока не будут найдены все релевантные (относящиеся к цели действия) и повторением поиска релевантных действий заново, после каждого применения действия. Для решения проблемы фрейма STRIPS предлагает в состоянии, к которому применяется правило (действие), изменять выполнимость лишь тех формул, которые описаны в эффекте действия, а выполнимость всех остальных оставлять неизменной.

Рассмотрим постановку задачи планирования при классических допущениях в терминах STRIPS. Как и выше, фактом будем называть замкнутую атомарную формулу языка исчисления предикатов 1-го порядка (ИПП), а состоянием — некоторое множество фактов. Неформально, состояние представляет модель среды, в которой действует интеллектуальный агент. Приведём пример описания среды в терминах STRIPS:

$$s = \{ATR(a), AT(B, b), AT(C, c), \forall u \forall x \forall y ((AT(u, x) \wedge (x \neq y)) \rightarrow \neg AT(u, y))\}.$$

Здесь, $ATR(a)$ означает, что «робот находится в комнате a », $AT(B, b)$ — «ящик B находится в комнате b », $AT(C, c)$ — «ящик C находится в комнате c ». Имена

конкретных объектов из этого множества: a, b, c — соответственно «комната a », «комната b », «комната c »; A, B, C — соответственно, «ящик A », «ящик B », «ящик C ».

Действия агента будем описывать с помощью правил, при этом, для упрощения таких описаний, примем некоторые соглашения. При описании STRIPS-задачи планирования как в множествах добавляемых, так и в множествах удаляемых фактов будем использовать лишь атомарные формулы без функциональных символов. Пример правила:

Имя правила:	$Push(x, y, z)$
Условие:	$C(R) = \{ATR(y), AT(x, y)\}$
Список добавлений:	$A(R) = \{ATR(y), AT(x, y)\}$
Список удалений:	$D(R) = \{ATR(z), AT(x, z)\}$

В приведённом примере STRIPS-правило $Push(x, y, z)$ описывает действие робота по перемещению ящика x из комнаты y в комнату z . Здесь, x, y, z — переменные. Выполнение агентом действия сводится к применению правила. Применение правила R преобразует состояние s в s' следующим образом:

$$s' = (s \setminus (D(R)\theta)) \cup (A(R)\theta).$$

Это преобразование обозначается так: $s \xrightarrow{R, \theta} s'$, где через θ обозначена подстановка элементов предметной области вместо переменных.

Определение 0.0.1 (STRIPS-допущение) При применении некоторого правила R к состоянию s выполнимость факта $f \in s$ изменяется, только если факт f описан либо в списке удалений $D(R)$, либо в списке добавлений $A(R)$.

Технически, при проверке применимости некоторого правила R , STRIPS выполняет полную подстановку на места всех переменных индивидов предметной области. Возможны различные варианты подстановок. Некоторые варианты подстановки могут давать примеры правил, применимых (или же неприменимых) в состоянии s . Однако, в алгоритм STRIPS можно внести незначительные модификации для применения не полностью означенных правил. В этом случае, в состоянии s появились бы факты с переменными в описании. Как будет видно далее, неполная подстановка активно используется планировщиками в пространстве планов. Соответствующее свойство этих планировщиков получило название *малого связывания* (least commitment).

Приведем постановку задачи STRIPS-планирования.

Определение 0.0.2 Будем называть доменом планирования $P = \langle s_0, \Sigma R \rangle$, где s_0 — начальное состояние, ΣR — конечное множество правил.

Определение 0.0.3 Будем называть задачей планирования $T = \langle P, G \rangle$, где G — описание целевого факта агента, или просто цель.

Решение задачи планирования T заключается в нахождении плана, который достигает цели G .

Определение 0.0.4 План $Plan$ — это последовательность состояний s_0, \dots, s_n , последовательность правил R_1, \dots, R_n и последовательность подстановок $\theta_1, \dots, \theta_n$, такая что, G выполнима в s_n . Длина плана $Plan$ равна n :

$$Plan : s_0 \xRightarrow{R_1, \theta_1} s_1 \xRightarrow{R_2, \theta_2} s_2 \dots \xRightarrow{R_n, \theta_n} s_n.$$

На вход алгоритма STRIPS подаётся множество правил ΣR , начальное состояние s_0 , цель G .

Алгоритм 1 Алгоритм STRIPS

Вход: $\Sigma R, s, G$

Выход: $Plan$

- 1: $s = s_0$
- 2: **пока** G не выполнимо в s
- 3: выбрать компоненту g из G
- 4: выбрать правило $R \in \Sigma R$ такое, что $g \in A(R)$
- 5: STRIPS($\Sigma R, s, C(R)$)
- 6: применить R к s
- 7: добавить R в $Plan$

вернуть $Plan$

Будем полагать, что в множестве ΣR все правила полностью конкретизированы. Вначале в стек целей помещается главная цель G . Если цель не является простой, т. е. содержит конъюнкцию литералов, то система STRIPS добавляет в стек в некотором порядке каждый из литералов составной цели (шаг 3). Когда верхняя цель стека является однолитеральной, система ищет действие (шаг 4), которое содержит в списке добавлений литерал, сопоставимый с этой целью. Если такое действие не применимо к текущему состоянию, тогда его предусловие помещается в стек целей, иначе действие применяется к текущему состоянию (шаг 6) и помещается в план $Plan$. Если верхняя цель в стеке соответствует текущему состоянию, то она удаляется из стека. Алгоритм STRIPS завершается, когда стек пуст.

Существуют задачи, для которых STRIPS либо не может построить план, либо находит не минимальный план. Причина этого кроется в том, что STRIPS удовлетворяет каждую компоненту составной цели по отдельности, без учёта их взаимосвязи. Особенность предметной области, где цели взаимосвязаны (взаимодействуют) получила название взаимосвязи целей.

Список литературы

1. *Fikes R. E., Nilsson N. J.* STRIPS: A new approach to the application of theorem proving to problem solving // Artificial Intelligence. — 1971. — Vol. 2, no. 3-4. — Pp. 189-208.