

# НОВЫЕ МЕТОДЫ ОПРЕДЕЛЕНИЯ И ВЫЯВЛЕНИЯ ЗНАЧИМОГО КОНТЕКСТА РАССУЖДЕНИЙ \*

Трофимов И.В.  
ИЦИИ ИПС РАН

152020, Россия, г. Переславль-Залесский, igor@warlock-98.botik.ru

Одним из важнейших направлений исследований в области автоматического планирования является разработка методов, позволяющих сократить пространство поиска решения. Неоднократно предпринимались попытки положить в основу таких методов различные механизмы абстрагирования, однако не всем механизмам было уделено должное внимание. Ранее автор предлагал [1, 2] использовать фокусировку в качестве механизма абстрагирования. Результат абстрагирования был назван значимым контекстом рассуждений, а процесс абстрагирования подразумевал автоматический отбор тех объектов, которые имеют отношение к решению текущей задачи. Однако предложенные методы имеют ряд ограничений. В данной работе предпринята попытка расширить содержание понятия «значимый контекст» и формализовать его определение, а также рассматриваются новые, более гибкие подходы к определению и выявлению значимого контекста.

## Введение

В работах [1, 2] описан подход, осуществляющий фокусировку путем разделения множества объектов на релевантные (значимый контекст) и нерелевантные. Выделение значимого контекста предлагалось реализовывать при помощи эвристических процедур или на базе методологии рассуждений по прецедентам. Такого решения может быть достаточно в некоторых предметных областях (например, для домена «мир кубиков» существует довольно простая эвристика). Тем не менее, анализ новых доменов показал, что требуются более гибкие механизмы выявления и более «избирательные» способы определения контекста.

В данной работе рассматривается способ определения значимого контекста как множества конкретных действий (а не множества объектов или схем действий), а также метод выявления контекста на базе множества правил. Ниже будет рассмотрена формальная модель фокусировки в задачах планирования. Далее в рамках этой модели рассматриваются способы определения значимого или незначимого контекста. В заключение описывается механизм выявления контекстов на базе декларативных правил.

## Модель фокусировки для задачи планирования

Для того чтобы рассуждать о том, какие действия имеют или не имеют отношение к решению текущей задачи планирования, необходимо определиться с формализацией самой задачи планирования и механизма абстрагирования. Здесь будет использоваться упрощенная формулировка задачи планирования, так как ряд аспектов, обычно затрагиваемых в классических способах формулировки (например, PDDL), в рамках данной работы интереса не представляют и приводят лишь к ненужным усложнениям.

При решении задачи планирования предполагается наличие *агента*, способного на некоторую деятельность, и *внешней среды*, в которой агент действует. Агент способен выполнять конечное число видов деятельности, описываемых *схемами действий*. Схема действий — это параметризованное описание некоторого вида деятельности; подстановка имен конкретных объектов среды вместо параметров порождает определение *конкретного действия*.

Обозначим **Schemas** — конечное множество схем действий, описывающих все возможные виды деятельности для данного агента. Каждому элементу множества **Schemas** соответствует целое неотрицательное число **k**, обозначающее количество объектов, вовлеченных в описание действий, соответствующих данной схеме. Будем называть это число *арностью схемы действий* или *количеством параметров схемы действий*. Арность схемы действий удобно моделировать при помощи функции:

---

\* Работа выполнена при поддержке РФФИ (проекты № 07-01-00763-а, 06-07-89083-а) и Президиума РАН (проект № 223).

**h: Schemas** → **Z<sup>+</sup>**.

При письме для обозначения арности схемы действий будет использоваться верхний индекс. Например, запись **schema<sup>k</sup>** означает, что речь идет о такой **schema**, для которой **h(schema) = k**.

Каждому описанию схемы действий соответствует непустое множество описаний конкретных действий. Описание действия представляет собой кортеж вида: **⟨schema<sup>k</sup>, obj<sub>1</sub>, ..., obj<sub>k</sub>⟩**, где **obj<sub>i</sub>** — имена объектов, вовлеченных в описание действия. Способы формирования описаний конкретных действий из схем могут быть различны и здесь не рассматриваются. Множество всех действий, соответствующих некоторой схеме действий **schema**, будем обозначать **actions[schema]**, а множество всех возможных действий агента обозначим **Actions**.

$$Actions = \bigcup_{schema \in Schemas} actions[schema]$$

Во всякий момент времени внешняя среда находится в каком-либо состоянии. Множество всех возможных состояний, в которых может пребывать внешняя среда, обозначим **States**. Для каждого состояния в **States** определено множество действий из **Actions**, которые выполнимы в данном состоянии. Другими словами существует функция

**g: Actions** × **States** → {*выполнимо*, *невыполнимо*},

которая для каждого действия из **Actions** и каждого состояния среды из **States** определяет, выполнимо ли действие в данном состоянии. Функцию **g** будем называть *функцией выполнимости*.

Выполнение всякого действия приводит к изменению состояния среды. Обозначим **AS** множество всех пар **⟨action, state⟩** (**action** ∈ **Actions**, **state** ∈ **States**), для которых выполняется условие (**x** ∈ **AS** → (**g(x)** = *выполнимо*)). Определим функцию

**f: AS** → **States**,

причем если **f(action, state<sub>1</sub>) = state<sub>2</sub>**, то **state<sub>1</sub> ≠ state<sub>2</sub>**. Функцию **f** будем называть *функцией переходов*.

Определим задачу планирования как кортеж вида:

**⟨States, Actions, g, f, S<sub>0</sub>, SG⟩**,

где **S<sub>0</sub> ∈ States** — состояние, называемое *начальным*; **SG ⊂ States** — множество состояний, являющихся *целевыми* для данной задачи планирования (в этих состояниях выполняется целевое условие). **S<sub>0</sub> ∉ SG**.

*Решением задачи планирования* (или *планом*) будем называть последовательность<sup>1</sup> вида

**⟨S<sub>0</sub>, A<sub>0</sub>, S<sub>1</sub>, A<sub>1</sub>, ..., S<sub>N</sub>, A<sub>N</sub>, S<sub>N+1</sub>⟩**,

для которой выполняются следующие условия:

1. **S<sub>i</sub> ∈ States, A<sub>j</sub> ∈ Actions**, где  $0 \leq i \leq N+1, 0 \leq j \leq N$ ;
2. **S<sub>N+1</sub> ∈ SG**;
3. **S<sub>i</sub> ∉ SG**, где  $0 \leq i \leq N$ ;
4. **g(A<sub>i</sub>, S<sub>i</sub>) = выполнимо**, где  $0 \leq i \leq N$ ;
5. **f(A<sub>i</sub>, S<sub>i</sub>) = S<sub>i+1</sub>**, где  $0 \leq i \leq N$ .

Пусть **P** — некоторая задача планирования, тогда **Solutions(P)** будем обозначать множество всех возможных решений задачи планирования **P**.

Определив таким образом постановку задачи планирования и результат ее решения, перейдем к определению понятия фокусировки (как разновидности абстрагирования) для задачи планирования.

Для данной задачи **P = ⟨States, Actions, g, f, S<sub>0</sub>, SG⟩** результатом фокусировки будет абстрактная задача **P<sub>1</sub> = ⟨States<sub>1</sub>, Actions<sub>1</sub>, g, f, S<sub>0</sub>, SG<sub>1</sub>⟩**, отвечающая следующим требованиям:

1. **States<sub>1</sub> ⊂ States**;
2. **Actions<sub>1</sub> ⊂ Actions**;
3. **a ∈ Actions<sub>1</sub>, s ∈ States<sub>1</sub>, g(a, s) = выполнимо** → **f(a, s) ∈ States<sub>1</sub>**;
4. **SG<sub>1</sub> ⊂ States<sub>1</sub>**;
5. **SG<sub>1</sub> ⊂ SG**;

---

<sup>1</sup> В данной работе для простоты рассматриваются только планы без параллельных ветвей деятельности (линеаризованные планы).

6.  $\text{Solutions}(\mathbf{P}_1) \subseteq \text{Solutions}(\mathbf{P})$ ;

7.  $\text{Solutions}(\mathbf{P}) \neq 0 \rightarrow \text{Solutions}(\mathbf{P}_1) \neq 0$ .

Предложенная модель фокусировки предполагает, что абстрактная задача отличается от исходной содержанием трех множеств: множества состояний, множества действий и множества целевых состояний. В данной работе рассматривается механизм фокусировки, сокращающий *только* множество **Actions**, то есть предполагается выполнение следующих условий  $\mathbf{States}_1 = \mathbf{States}$ ,  $\mathbf{SG}_1 = \mathbf{SG}$  и  $\mathbf{Actions}_1 \subset \mathbf{Actions}$ .

Согласно требованию 7, если существует решение исходной задачи, то множество **Actions**<sub>1</sub> должно содержать набор действий, из которых можно получить хотя бы одно решение абстрактной задачи. Это абстрактное множество действий будем называть *возможным значимым контекстом* и обозначать **SCx**. Если для получения решения нельзя исключить ни одного действия, то абстрагирование невозможно. Исключенные (абстрагированные) действия составляют множество, которое будем называть *допустимым незначимым контекстом* и обозначать **NCx**. Эти два вида контекстов связаны следующим соотношением.

$$\mathbf{SCx} = \mathbf{Actions} \setminus \mathbf{NCx}$$

Таким образом, для того чтобы осуществить абстрагирование, нам достаточно найти либо один из возможных значимых контекстов, либо какой-нибудь допустимый незначимый контекст.

Чтобы воспользоваться предложенным механизмом абстрагирования необходимо решить две подзадачи:

- найти адекватное *средство определения* для контекстов;
- найти *механизм выявления* **NCx** или **SCx** для текущей задачи планирования.

### Способы определения контекстов

По определению возможные значимый и допустимый незначимый контексты являются множествами. Два основных способа определения множеств — перечисление элементов и определение посредством характеристического свойства.

В первом случае контекст представляет собой простое перечисление действий. Это удобно тем, что для осуществления планирования сразу имеется готовое множество действий, которыми можно оперировать. С другой стороны, это множество может быть огромной мощности и потребует больших затрат на хранение. В открытых средах, где количество объектов (и оперирующих ими конкретных действий) не определено и считается бесконечным, такой подход к определению контекста становится неприемлемым. То же самое касается сред, в которых параметрами действий могут выступать числовые величины (например, вещественные числа или целые из неограниченных интервалов).

Второй способ позволяет иметь дело с бесконечными множествами (действий). Однако в процессе планирования при выборе очередного действия необходимо выполнять проверку на предмет удовлетворения этого действия ограничениям характеристического свойства, чтобы выяснить, принадлежит ли действие контексту. А это требует некоторых вычислительных ресурсов.

В общем виде, определение контекста **Cx** посредством характеристического свойства можно записать следующим образом:  $\mathbf{Cx} = \{\mathbf{a} \mid \mathbf{a} \in \mathbf{Actions}, \mathbf{filter}\}$ , где **filter** — требование к элементам **Cx** (характеристическое свойство). То есть контекст представляет собой такое множество элементов из **Actions**, которые отвечают требованию **filter**. Рассмотрим подробнее, какими могут быть требования.

В силу того, что мы моделируем действие как составную сущность, а именно, как кортеж вида  $\langle \mathbf{schema}^k, \mathbf{obj}_1, \dots, \mathbf{obj}_k \rangle$ , мы можем накладывать ограничения на элементы, из которых состоит действие. Например, можно потребовать, чтобы элементами контекста были действия, порожденные от одной определенной схемы действий. Это можно сформулировать следующим образом.

$$\mathbf{Cx} = \{\mathbf{a} \mid \mathbf{a} \in \mathbf{Actions}, \mathbf{a} \in \mathbf{actions}[\mathbf{schema}]\}, \text{ где } \mathbf{schema} \in \mathbf{Schemas}.$$

В качестве еще одного примера рассмотрим требование, постулирующее, что **n**-ым параметром действия является имя конкретного объекта **Obj**. Это также удобно определить посредством множества. Обозначим  $\mathbf{actions}_{i, \mathbf{obj}}[\mathbf{schema}^k]$  множество действий, порожденных от

схемы  $schema^k$ , у которых в качестве  $i$ -ого параметра (и, следовательно,  $i+1$ -ого элемента кортежа, описывающего действие) выступает имя объекта **Obj** (здесь  $1 \leq i \leq k$ ). Тогда множеством всех действий, у которых  $i$ -ый параметр равен **Obj**, является объединение множеств  $actions_{i,Obj}[schema^k]$  по всем схемам действий с арностью большей либо равной  $i$ . Будем обозначать это множество  $actions_{i,Obj}$ .

$$actions_{i,Obj} = \bigcup_{\substack{schema^k \in Schemas, \\ k \geq i}} actions_{i,Obj}[schema^k]$$

Соответственно, контекст будет определяться как  $Cx = \{a \mid a \in \mathbf{Actions}, a \in actions_{i,Obj}\}$ .

Аналогичным образом можно определить целый ряд ограничивающих множеств; в силу ограниченности объема они здесь не приводятся (подробнее см. [3]). Так как мы задаем характеристическое свойство посредством операции принадлежности или непринадлежности какому-либо множеству, более сложные требования к контексту можно строить из предложенных множеств (а также из индивидуальных действий), используя операции над множествами.

Способ задания контекста посредством характеристического свойства довольно гибок. Переходя к более сложным моделям задачи планирования, можно составлять характеристические требования другого рода — оперирующие понятиями новой модели.

### Выявление значимого контекста на базе правил

Предлагавшиеся ранее методы выявления опирались либо на закодированные эвристические процедуры, либо на методологию рассуждений по прецедентам. Единственным преимуществом методов на базе закодированных эвристик является то, что они могут опираться на знания о предметной области, не представленные в постановке задачи планирования явным образом (по крайней мере, не представленные формально). Однако такой путь требует разработки новых планировщиков для новых предметных областей и задач. Методы на базе методологии рассуждений по прецедентам также имеют ограничения, отмеченные в [2].

Шагом в сторону более гибкого подхода является создание эвристик, опирающихся на набор декларативных правил, определяющих связь между формальной постановкой задачи и контекстом. Декларативное представление знаний позволяет легко сменить предметную область, написав новые правила, но не изменяя алгоритмов. Такой подход предполагает наличие единого механизма интерпретации правил, а значит, и наличие некоторого формального языка описания правил с определенными выразительными способностями.

Отметим некоторые ключевые аспекты языка описания правил. Во-первых, такой язык может опираться на описания начального состояния и цели. Для простоты будем полагать, что описание всякого состояния представляет собой множество «атомарных» фактов, выраженных кортежами вида:  $\langle predicate^k \ arg_1, \ arg_2, \ \dots, \ arg_k \rangle$ , где  $arg_i$  — имена объектов моделируемой предметной области ( $1 \leq i \leq k$ ), а  $predicate$  — имя представляющего данный факт отношения между этими объектами. Как и в случае с моделью схемы действий, верхний индекс у предиката означает арность описываемого им отношения. Аналогичным образом задается и целевое условие, с той лишь разницей, что кортежи помечаются модальностями «должен» или «не должен» факт присутствовать в целевом состоянии. Синтаксически, будем помечать модальность «не должен» значком отрицания « $\neg$ »; если значок отсутствует, то это значит, что имеет место модальность «должен». Цель планировщика — достичь такого состояния, описание которого содержало бы все кортежи целевого условия с модальностью «должен» и не содержало ни одного кортежа целевого условия с модальностью «не должен».

Для того чтобы работать с этими описаниями, введем две новые функции. Пусть  $P$  — некоторая задача планирования. Тогда функция  $InInit(P)$  возвращает описание начального состояния в виде множества кортежей, описывающих атомарные факты. В свою очередь функция  $InGoal(P)$  возвращает описание цели в виде множества целевых условий, представленных аналогичными кортежами, которые (дополнительно) могут быть помечены модальностью «не должен».

Во-вторых, было бы полезно иметь возможность писать правила как ориентированные на один определенный класс задач, так и правила, осуществляющие классификацию задач (для

одного класса задач контекст формируется по одним правилам, для другого — по другим). Чтобы реализовать это, можно выбрать следующий общий вид для правил:

*Если* «классификатор задачи», *то* «определитель контекста».

Здесь «классификатор задачи» — это некоторое условие, позволяющее определить, принадлежит ли данная задача определенному классу. Если это условие определено как «Истина», то мы получим универсальное правило домена, которое срабатывает при решении задач всех классов.

В качестве примера работы классификатора можно привести следующую ситуацию. Пусть в некотором домене мы хотим различать задачи перемещения и ремонта. Обозначим «at» — имя отношения, отражающего местоположения объектов в домене, а «broken» — имя отношения, отражающее состояние исправности объекта. Тогда левые части правил определения контекста могли бы иметь следующий вид.

*Если*  $\exists X, Y, Z. (\langle at X Y \rangle \in \mathbf{InInit}(P) \wedge \langle at X Z \rangle \in \mathbf{InGoal}(P) \wedge (Y \neq Z))$ , *то* ...

*Если*  $\exists X. (\langle broken X \rangle \in \mathbf{InInit}(P) \wedge \neg \langle broken X \rangle \in \mathbf{InGoal}(P))$ , *то* ...

Здесь  $P$  — решаемая задача планирования. Первое правило, фактически, декларирует следующее «если задача является перемещением, то...», второе — выполняет проверку принадлежности задачи к классу «ремонт». Для первого класса задач возможный значимый контекст можно определить как, все действия, касающиеся перемещений, а для второго — ремонтов. Правые части предложенных правил должны содержать соответствующие конструкции для формирования такого рода контекстов.

Теперь рассмотрим вторую часть правила — «определитель контекста». Она должна определять, из чего состоит контекст в зависимости от конкретной решаемой задачи. Здесь речь идет не о зависимости классифицирующего характера (как в левой части правила), а о зависимости от конкретного содержания предметной области. Так как в классическом подходе к планированию (см. синтаксис PDDL) объекты предметной области, с которыми сталкивается планировщик, объявляются как часть задачи планирования (а не домена), то, решая различные задачи даже одного класса, мы можем иметь дело с различными объектами. Таким образом, ограничения на контекст вида  $\mathbf{actions}_{i,obj}$  (и другие, в которых фигурируют имена объектов) должны быть связаны с конкретной текущей задачей (и объявленными в ней объектами).

В связи с этим, элемент «определитель контекста» удобно разделить на две компоненты: «декларатор объектов» и «модификатор контекста». Синтаксически в правилах будем разделять их стрелкой:

«определитель контекста» ::= [«декларатор объектов»  $\rightarrow$ ] «модификатор контекста».

Элемент «декларатор объектов» является необязательным. Он определяет множества объектов, которыми будет пользоваться «модификатор контекста». Если «модификатор контекста» не содержит ограничений, в которых фигурируют объекты, то «декларатор объектов» опускается.

Рассмотрим на примере, что может представлять из себя «определитель контекста». Пусть мы хотим определить допустимый незначимый контекст как «множество действий, вторым аргументом которых являются объекты, которые в начальном состоянии были красного цвета». Этот определитель можно записать при помощи конструкции вида:

$$\text{Set1} = \{X \mid \langle red X \rangle \in \mathbf{InInit}(P)\} \rightarrow \mathbf{NCx} = \bigcup_{obj \in \text{Set1}} \mathbf{actions}_{2,obj}$$

Здесь в левой части определителя (до стрелки) декларируется множество красных объектов (Set1), а в правой части объявляется, что допустимый незначимый контекст ( $\mathbf{NCx}$ ) состоит из действий, манипулирующих (по второму аргументу) объектами из Set1.

Выразительные способности этого языка позволяют устанавливать зависимость между контекстом и свойствами/отношениями объектов, что делает метод довольно избирательным.

### Список литературы

1. Трофимов И.В. *Значимый контекст рассуждений в задаче планирования*. //Труды конференции САИТ-2005 — М.:КомКнига, 2005, т.1, стр. 227-229.
2. Трофимов И.В. *Значимый контекст рассуждений в задаче планирования: эксперименты и перспективы* // Труды КИИ-2006 — М.: Физматлит, 2006, т.2, стр. 700—708.
3. [http://ai-center.botik.ru/planning/workshops/2007\\_Trofimov\\_SignificantContextAsAbstraction.pdf](http://ai-center.botik.ru/planning/workshops/2007_Trofimov_SignificantContextAsAbstraction.pdf)