

УДК 004.832

## ИСПОЛЬЗОВАНИЕ ФОКУСИРОВКИ ПРИ РЕШЕНИИ ЗАДАЧ ПЛАНИРОВАНИЯ\*

И.В.Трофимов<sup>1</sup>

В работе рассматриваются результаты экспериментов с автоматическим планировщиком, осуществляющим сокращение поискового пространства задачи посредством абстрагирования. В качестве механизма абстрагирования использована фокусировка. Принципы абстрагирования декларируются на специальном языке.

### Введение

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 принципы абстрагирования [Sacerdoti, 1974][Yang et al., 1990][Knoblock, 1994]; в качестве механизма абстрагирования использовалась «избирательность» (selectivity) [Zucker, 2003]. Автором было предложено [Трофимов, 2007] использовать методы абстрагирования, опирающиеся на механизм «фокусировки» (focusing или focalization) [Zucker, 2003]. Результат абстрагирования был назван значимым контекстом рассуждений (ЗКР) и представляет собой множество конкретизированных (ground) действий, релевантных конкретной решаемой задаче. Методы выявления ЗКР могут быть различны, но наиболее перспективным кажется метод, опирающийся на декларативное описание принципов абстрагирования, определяющих связь между формальной постановкой конкретной задачи и значимым контекстом.

Ранее [Трофимов, 2007] была формально определена модель фокусировки, а также были отмечены ключевые аспекты языка для декларации принципа абстрагирования (ЯДПА). В данной статье кратко описан ЯДПА, а также рассматривается попытка применить метод выявления ЗКР (опирающийся на интерпретацию принципа

---

\* Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00763-а) и Президиума РАН (проект № 223)

<sup>1</sup> 152020, г. Переславль-Залесский, ИПС РАН, igor@warlock-98.botik.ru

абстрагирования) к известным задачам планирования, использовавшимся в качестве тестовых треков на соревнованиях планировщиков IPC.

## 1. Язык декларации принципа абстрагирования

Задача языка определить принцип выявления множества действий, представляющих собой значимый контекст рассуждений.

ЯДПА является скриптовым языком. Выявление ЗКР осуществляется при помощи выполнения инструкций и операций. Инструкции отвечают за управление последовательностью выполнения операций и именование результатов операций. Операции формируют ЗКР, а также промежуточные вспомогательные множества.

ЯДПА содержит три инструкции: условие, переход и присваивание.

Инструкция «условие» имеет вид:

*Если (идентификатор) инструкция*

и интерпретируется следующим образом. Если соответствующее идентификатору множество не пусто, то выполняется инструкция после скобок.

Инструкция «переход» имеет вид:

*Переход метка\_перехода.*

Интерпретация данной инструкции приводит к тому, что следующей будет выполнена инструкция, помеченная «меткой перехода».

Инструкция «присваивание» имеет вид:

*идентификатор = операция.*

Ее интерпретация приводит к тому, что полученное в результате выполнения операции множество будет иметь имя, заданное идентификатором. Выполнение операций всегда связано с инструкцией присваивания.

Инструкции должны формировать скрипт таким образом, чтобы всякая последовательность выполнения операций приводила к формированию специального множества с именем *SCx*, соответствующего значимому контексту рассуждений.

Инструкции «условие» и «переход» позволяют реализовать комплексную «операцию транзитивного замыкания» при построении множеств, а также осуществлять классификацию задач планирования.

Операции ЯДПА содержательно идентичны операциям реляционной алгебры [Дейт, 2001]. В язык включены операции выборки, проекции, декартова произведения, объединения, пересечения и разности. Дополнительно введены еще две операции: выборка одного произвольного элемента и (реляционная) группировка с выборкой одного произвольного элемента в каждой группе.

Операции можно выполнять над так называемыми базовыми множествами или множествами, полученными при помощи ранее выполненных операций.

Базовыми считаются следующие множества.

- Множество всех объектов (*AllObjects*) из постановки задачи планирования. Элементами этого множества являются кортежи из двух элементов <имя объекта, непосредственный тип объекта>.

- Множество всех конкретизированных действий (*AllActions*) задачи планирования. Это множество состоит из кортежей, длина которых равна максимальной арности схем действий + 1. Первый элемент кортежа содержит имя действия, остальные — аргументы действия (если количество аргументов у действия меньше длины кортежа, то оставшиеся элементы кортежа принимают специальное значение «пусто»). Результирующее множество  $SCx$  должно являться подмножеством *AllActions*.

- Для каждого предиката  $P$  арности  $k$  в определении домена существует пара множеств *InInit<sub>P</sub>* и *InGoal<sub>P</sub>*, элементами которых могут быть кортежи длины  $k$ . Элементами данных множеств являются кортежи объектов. Если начальное состояние задачи содержит атомарную формулу, производную от  $P$ , то аргументы этой формулы составляют кортеж, который является элементом *InInit<sub>P</sub>*. Аналогично для цели задачи. Такая формулировка налагает ограничение на спектр задач, к которым может быть применен метод — начальное состояние и цель задачи планирования должны определяться списком атомарных формул.

Все базовые множества могут быть построены автоматическими средствами на основании формулировки задачи планирования.

Следует отметить, что элементы кортежей именованы. В *AllObjects* первый элемент именуется *obj*, второй — *type*. В *AllActions* первый элемент именуется *aName*, остальные — *arg1*, *arg2*, ..., *argN*, где  $N$  — максимальная арность схем действий. Во множествах вида *InInit<sub>P</sub>* и *InGoal<sub>P</sub>* элементам кортежей присваиваются имена аргументов предиката  $P$ , согласно его описанию в домене.

Рассмотрим подробнее спектр операций.

Операция «выборка» возвращает подмножество исходного множества, элементы которого удовлетворяют заданному ограничению. Так как элементами множеств являются кортежи, ограничения накладываются именно на значения элементов кортежей. Операция выборки имеет следующий вид:

*Выборка* (<список имен элементов кортежа> | оперативное множество | условие выборки).

Здесь и далее оперативное множество — это множество, к которому применяется операция. Список имен элементов кортежа в выборке такой

же, как и в оперативном множестве, однако в результат попадают только кортежи, удовлетворяющие условию. В качестве условия может выступать проверка на равенство (неравенство) заданного элемента кортежа некоторой константе или другому элементу кортежа. Условие может содержать несколько проверок перечисляемых через запятую.

Операция «проекция» возвращает множество кортежей исходного множества, из которых исключены заданные позиции. Операция имеет следующий вид:

*Проекция* (<список имен элементов кортежа> | оперативное множество).

Список имен элементов кортежа должен содержать меньшее число элементов, чем в оперативном множестве.

Операции «декартово произведение», «объединение», «пересечение» и «разность» интерпретируются как классические теоретико-множественные операции и имеют следующие форматы, соответственно:

*ДекПроизв* (<список имен элементов кортежа> | оперативное множество | оперативное множество),

*Объединение* (<список имен элементов кортежа> | оперативное множество | оперативное множество),

*Пересечение* (<список имен элементов кортежа> | оперативное множество | оперативное множество),

*Разность* (<список имен элементов кортежа> | оперативное множество | оперативное множество).

Для операций, в которых важен порядок операндов (декартово произведение и разность), первое оперативное множество соответствует левому операнду, а второе — правому.

Операция «выборки одного произвольного элемента» возвращает множество, содержащее один кортеж из исходного множества. Операцией не детерминировано, из какого именно кортежа будет сформирован результат. Операция имеет вид:

*Один* (<список имен элементов кортежа> | оперативное множество).

Операция «группировки с выборкой одного произвольного элемента в каждой группе» осуществляет группировку (по аналогии с реляционной алгеброй) по заданному набору элементов кортежа и из каждой группы помещает в результат один произвольный кортеж. Операция имеет вид:

*ГруппОдин* (<список имен элементов кортежа> | оперативное множество | список имен элементов кортежа).

Здесь первый список имен элементов кортежа такой же, как и в оперативном множестве, а второй — определяет перечень элементов кортежей, по которым производится группировка.

Во всех операциях за исключением проекции разрешается переименование элементов кортежа, формируемого в результате выполнения операции.

Интерпретация такого языка легко реализуема на базе любой современной СУБД, и именно такая реализация была использована в экспериментах.

В качестве простого примера использования ЯДПА рассмотрим процесс отбора «полезных» действий  $stack$  из домена «мир кубиков». Если в цели задачи указан атом ( $on\ x\ y$ ), то действие  $stack(x,y)$  является «полезным». Для получения множества всех таких действий сначала из множества всех действий отбираются действия  $stack$ .

$StackActions = \text{Выборка}(\langle aName, arg1, arg2 \rangle \mid AllActions \mid aName = 'stack')$

Затем строится декартово произведение этого множества с множеством кортежей объектов для предиката ON в цели.

$Stack\_x\_OnInGoal = \text{ДекПроизв}(\langle aName, arg1, arg2, x, y \rangle \mid StackActions, InGoal\_ON)$

Далее из этого множества отбираются те записи, у которых  $arg1 = x$  и  $arg2 = y$ .

$StackEx = \text{Выборка}(\langle aName, arg1, arg2, x, y \rangle \mid Stack\_x\_OnInGoal \mid arg1 = x, arg2 = y)$

Теперь формируем окончательное множество действий  $stack$  (отбросив лишние столбцы).

$FinalStack = \text{Проекция}(\langle aName, arg1, arg2 \rangle \mid StackEx)$

## 2. Результаты экспериментов

В ходе экспериментов оценивались следующие показатели:

- количество решенных задач,
- время поиска плана,
- длина плана,

полученные при помощи оригинального планировщика FF версии 2.3 [Hoffmann et al., 2001] и этого же планировщика, дополненного модулем абстрагирования. В качестве тестовых задач взяты *strips*-задачи с соревнований планировщиков IPC.

Экспериментальная площадка имела следующие характеристики: ПК с одним процессором Intel Pentium M 1,7 ГГц и 1,5 Гб ОЗУ; ОС Windows XP Pro; компилятор *vc8*.

### 2.1 Blocks World 4 — мир кубиков с 4 операторами

Задачи для мира кубиков использовались в качестве тестов на втором соревновании IPC. Трек состоял из 35 задач (конфигурации от 4 до 17

кубиков). Эксперимент также проводился на 67 внеконкурсных дополнительных задачах (от 17 до 50 кубиков). В сумме 102 задачи.

После абстрагирования ЗКР составляли действия  $stack(x,y)$ , если цель задачи содержала атом (on x,y); действия  $unstack(x,y)$ , если начальное состояние задачи содержало атом (on x,y); все действия  $pickup$  и  $putdown$ .

В эксперименте задача считалась нерешенной, если планировщик работал более 10 минут или тратил более 1 Гб ОЗУ.

Оригинальный FF не решил 25 задач (причем 4 нерешенных задачи из основного трека). Абстрагирующий планировщик решил все задачи. Для совместно решенных задач планировщики получали планы почти одинаковой длины и (за редким исключением) схожее время работы.

## 2.2 Depots — склады

Домен «склады» является объединением хорошо известных логистического домена (Logistics) и мира кубиков. В домене грузовики могут перевозить ящики, а ящики должны быть уложены в стопки на платформах (pallets) в пунктах их назначения. Укладка в стопки осуществляется подъемниками.

Задачи домена «склады» использовались в качестве тестов на третьем соревновании IPC. Трек состоял из 22 задач.

В экспериментах были опробованы два принципа абстрагирования. Первый позволял выгружать и складировать ящики только в их пункте назначения (АП-1). Второй — дополнял первый принцип тем, что грузить ящики позволялось только в месте их изначального нахождения и допускалось использование только одного грузовика (АП-2). Результаты представлены на рисунках 1 и 2.

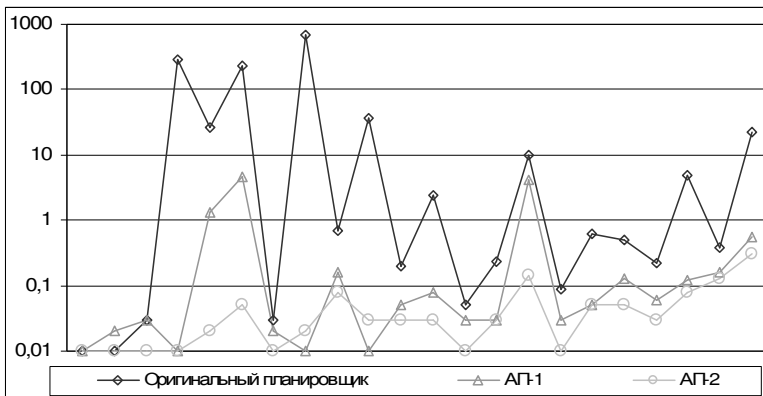


Рисунок 1. Домен Depots. Время поиска решения планировщиками на тестовом треке. Ось абсцисс — задачи, ось ординат (логарифмическая) — время поиска решения в секундах.

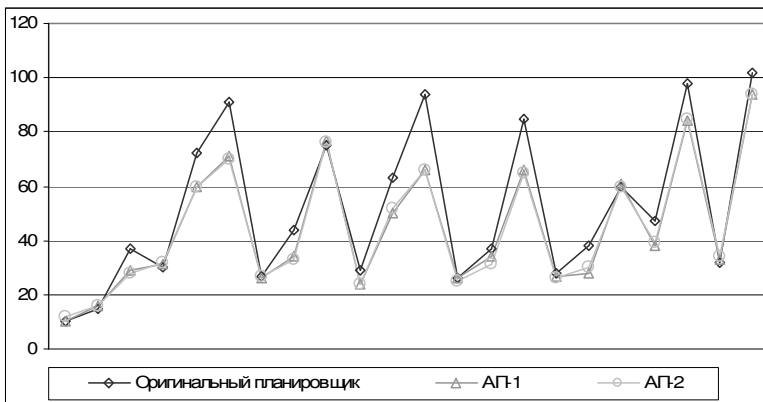


Рисунок 2. Домен Depots. Длины планов, найденных планировщиками на тестовом треке. Ось абсцисс — задачи, ось ординат — длина плана в шагах.

Оригинальный FF и оба абстрагирующих планировщика решили все задачи из тестового трека, однако абстрагирующие планировщики значительно превосходят оригинал по скорости поиска решения и на многих задачах находят более короткие планы.

Эксперимент для домена «склады» был продолжен на наборе внеконкурсных задач (22 задачи из каталога HandCoded) значительно большего размера. Для этого блока задач был выставлен лимит времени в 10 минут и ограничение по памяти в 1 Гб.

Из внеконкурсных задач оригинальному планировщику удалось решить только одну (первую). AP-1 решил 8 задач; AP-2 — 12 задач. Самый длинный план был получен AP-2 на 17-ой задаче — 388 шагов (за 531 сек).

### 2.3 Satellite — спутник

Задачи состоят в планировании наблюдений для группы спутников, имеющих различное оборудование. Фотографировать можно любые цели, выполнять калибровку инструментов можно только по определенным целям.

Задачи домена «спутник» использовались в качестве тестов на третьем соревновании IPC. Трек состоял из 20 задач. Дополнительно в экспериментах с абстрагированием использовались 16 внеконкурсных задач (каталог HandCoded).

Основу принципа абстрагирования составило ограничение множества поворотов спутников на цели и ограничение количества используемых инструментов. Инструменты выбирались по одному для каждого режима съемки, а повороты спутник осуществлял только на цели съемки,

калибровки или глобальные цели задачи, указывающие окончательную ориентацию спутников.

И оригинальный FF, и абстрагирующий планировщик решили все задачи. Абстрагирующий планировщик продемонстрировал значительное преимущество в скорости (на некоторых задачах в 100 раз) при почти одинаковых длинах решений. Результаты экспериментов представлены на рисунках 3 и 4.

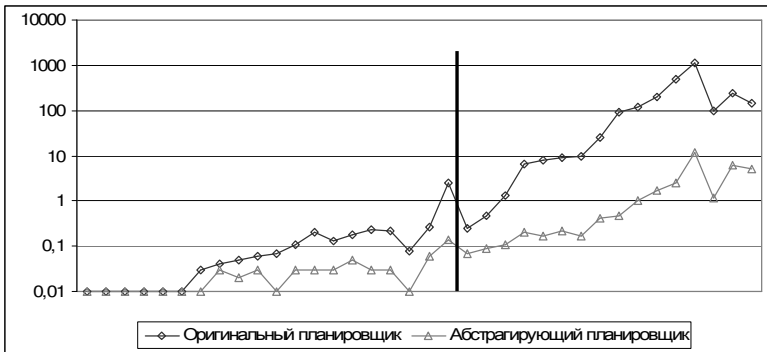


Рисунок 3. Домен Satellite. Время поиска решения планировщиками на тестовом (слева от вертикальной черты) и внеконкурсном (справа от черты) треке. Ось абсцисс — задачи, ось ординат (логарифмическая) — время поиска решения в секундах.

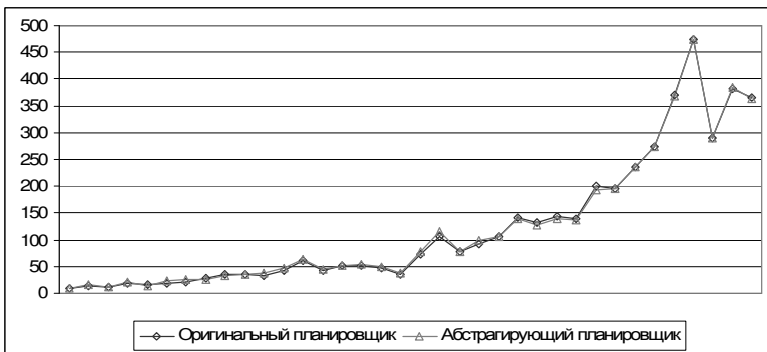


Рисунок 4. Домен Satellite. Длины планов, найденных планировщиками. Ось абсцисс — задачи, ось ординат — длина плана в шагах.

### 3. Заключение

Эксперименты с использованием фокусировки для сокращения пространства поиска задачи планирования показывают, что применение такого подхода может значительно ускорять процесс поиска решения. Можно предположить, что значительный выигрыш в скорости будет



получен на задачах, с большим количеством объектов, которые могут подвергаться воздействию, но лишь небольшая их часть необходима для получения решения, а также в сложных доменах с большим количеством различных действий и целей. Последнее может быть достигнуто за счет классифицирующих элементов ЯДПА (инструкций «условие» и «переход»).

Перспективным направлением дальнейших исследований представляется расширение ЯДПА в направлении поддержки широкого спектра элементов современного PDDL (в частности количественных величин и временного фактора). Другим интересным направлением является интеграция предлагаемого механизма абстрагирования в планировщики других типов (например, HTN).

### Список литературы

- [Дейт, 2001] Дейт К. Дж. Введение в системы баз данных, 7-е издание.: Пер. с англ. – М.:Издательский дом «Вильямс», 2001.
- [Трофимов, 2007] Трофимов И.В. Новые методы определения и выявления значимого контекста рассуждений. // Тр. Второй международной конференции "Системный анализ и информационные технологии" САИТ-2007: В 2-х томах. – М.: ЛКИ, 2007.
- [Hoffmann et al., 2001] Hoffmann J., Nebel B. The FF Planning System: Fast Plan Generation Through Heuristic Search. // Journal of Artificial Intelligence Research. 2001. № 14.
- [Knoblock, 1994] Knoblock C. Automatic generation of abstraction for planning. // Artificial Intelligence. 1994. № 68.
- [Sacerdoti, 1974] Sacerdoti E. Planning in a hierarchy of abstraction spaces. // Artificial Intelligence. 1974. № 5.
- [Yang et al., 1990] Yang Q., Tenenber J.D. Abtweak: Abstracting a nonlinear, least commitment planner. // Proc. of the Eighth National Conference on Artificial Intelligence. Boston. 1990.
- [Zucker, 2003] Zucker J.-D. A grounded theory of abstraction in artificial intelligence. // Philosophical Transactions of the Royal Society B: Biological Sciences. 2003. № 358(1435).