ТЕХНИЧЕСКАЯ
КИБЕРНЕТИКА
№ 3 • 1985

УДК 681.3 : 519 : 007.52

ФОРМАЛЬНО-ЛИНГВИСТИЧЕСКАЯ МОДЕЛЬ ДЛЯ РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ. I. ИНСТРУМЕНТАРИЙ ФОРМАЛИЗАЦИИ. ЯЗЫК ТРАЕКТОРИЙ

ШТИЛЬМАН Б. М.

Введение. Для решения задач дискретной оптимизации, к которым сводятся многие важные практические задачи, разработаны различные методы математического программирования. В большинстве случаев эти методы порождают алгоритмы, сложность которых экспоненциально растет с ростом размерности задачи (с повышением точности решения — для приближенных методов). Принципиальный характер этих трудностей подтверждается результатами теории вычислительной сложности [1]. Актуальна разработка новых приближенных методов дискретной оптимизации, например, моделируя человеческие методы в тех областях, где эти методы достигли высокого совершенства.

Исследования, проводимые под руководством М. М. Ботвинника [2—4], привели к созданию метода ПИОНЕР для решения переборных задач большой размерности. В основе разработок моделирование и обобщение метода поиска хода шахматным мастером. С помощью метода ПИОНЕР в поставленной задаче вводится динамическая многоступенчатая структура; при этом сложная система, описанная в условиях задачи, сводится к иерархии ступеней-подсистем, которые имеют свои локальные цели, отличные от цели всей системы (т.н. точной цели игры), но общие по типу с новой специально введенной «неточной» целью игры. Введенная структура позволяет сформулировать достаточно сильные критерии отсечения ветвей и приоритета включения в перебор. Эксперименты с реализациями метода (в шахматах [4], в задачах планирования ремонтов энергооборудования [5—7]) показывают, что его использование сводит перебор вариантов к минимуму (порядка 100 вариантов) при высоком качестве получаемого решения. Метод ПИОНЕР был представлен набором понятий и процедур, описанных в виде рациональных рассуждений. В настоящей работе предложена формализация некоторых элементов метода ПИОНЕР с использованием формально-лингвистического подхода, проведено исследование разработанной математической модели.

1. Подход к формализации. В работе формально описан класс задач, охваченный предлагаемой моделью. Представляет интерес (особенно, учитывая выводы [8]), что в этот класс входят как игровые задачи, так и задачи исследования операций, имеющие неигровой характер. Иерархия подсистем, изменяющаяся в процессе перебора, моделируется иерархией формальных языков (траекторий и зон). При переходах изучаемой системы из одного состояния в другое языки изменяются с помощью переводов. Центральное место в работе занимает формальное решение для данной модели проблемы границ (Frame problem) [9—13]. Создатели различных систем, выполняющих перебор альтернативных вариантов, сталкиваются с задачей дифференциальной обработки информации: производится разделение неизменной и устаревающей информации об объекте при его переходе в новое состояние, а затем путем перестройки устаревшей информации вырабатывается информация, адекватная новому состоянию. Для систем искусственного интеллекта проблема границ состоит в таком представлении знаний, чтобы система могла эффективно (без полного перебора) определить, какие факты должны измениться в результате некоторого действия, а какие нет. Универсального решения этой проблемы не существует, т. к. такое решение должно быть основано на взаимосвязях между фактами, существующих в конкретной задаче [12]. Практически при разработке конкретных программ проблема границ решается эвристически. В настоящей работе для метода ПИОНЕР она решена формально путем введения отношений достижимости точек и связи траекторий, порождения языков траекторий и зон, выделения изменяемой и неизменной частей языка зон при переводах. Решение проблемы границ для данной модели позволило осуществить дифференциальную обработку информации в процессе перебора и, тем самым, обеспечить высокую эффективность алгоритмов, реализующих модель для конкретных задач.

Несмотря на то, что полученные результаты применимы непосредственно лишь для метода ПИОНЕР, выделим основные принципы, на которых основано решение проблемы границ и которые, по-видимому, применимы в других задачах. Прежде всего это представление информации об объекте (иерархической системе) в виде иерархии языков в том смысле, что цепочка языка данного уровня определяет терминальный символ языка более высокого уровня. Переход объекта в новое состояние вызывает перестройку всей иерархии языков: отдельные цепочки не изменяются, другие изменяются частично (сохраняя в своем составе терминальные символы, соответствующие неизменным цепочкам языков низшего уровня), некоторые цепочки исключаются полностью. Перестройка завершается пополнением пересчитанного подмножества новыми цепочками, отсутствовавшими в старом состоянии. Формально перестройка состоит в выполнении перевода языков (отображения), выделении неподвижного подмножества, ядра (прообраза пустой цепочки), вычислении коядра перевода (дополнения образа перевода до полного языка). Эффективность этой процедуры для данной модели объясняется тем, что при переходе в новое состояние иерархия изменяется «кустообразно» — должен быть пересчитан «куст» взаимосвязанных цепочек и подцепочек (зон и траекторий). Значительно большая часть иерархии остается неизменной. Возможность эффективного описания изменяющегося «куста» обеспечивается введенными в систему взаимосвязями (отношениями достижимости точек и связи траекторий).

В первой части работы описана область применимости метода, введен класс используемых в работе формальных грамматик, определен и исследован нижний уровень иерархии языков — язык траекторий. Во второй части рассмотрены семейство траекторных сетей, язык зон, переводы языков и на основе полученных результатов дано формальное решение проблемы границ. Изложение иллюстрировано примерами из задач планирования ремонтов электростанций и программирования шахматной игры.

2. Постановка задачи.

Определение 1. Назовем *сложной системой* следующую восьмерку:

$$\langle X,\ P,\ R_p,\ ON,\ \psi,\ S_0,\ Sg,\ M\rangle \qquad (2.1)$$

Здесь $X=\{x_i\}$ — конечное множество «точек»; $P=\{p_i\}$ — конечное множество «элементов», $P=P_1\cup P_2$, $P_1\cap P_2=\varnothing$; $R_p(x,\ y)$ семейство бинарных отношений достижимости в $X(x\in X,\ y\in X,\ p\in P)$, $R_p(x,\ y)$ означает, что точка y достижима из точки x для элемента p; $ON(p)=x$ частичная функция из P в X; $\psi: P\rightarrow \mathbf{Z}_+$; S_0 и Sg соответственно описания множеств начальных и целевых состояний системы; M — описание операторов перехода системы из одного состояния в другое. Под описанием состояния системы мы понимаем описание на языке исчисления предикатов первого порядка, ставящем в соответствие каждому отношению некоторую правильно построенную формулу — ППФ. Так, каждое исходное состояние описывается некоторым набором ППФ вида $\{ON(p_i)=x_j\}$. Аналогично описывается каждое целевое состояние. Σ — множество всех состояний системы (2.1). Оператор перехода из M состоит в замене одного множества ППФ другим, а также ППФ применимости этого оператора:

$$\text{ПЕРЕХОД } (p, x, y)$$

$$\text{ППФ применимости:} \quad (ON(p)=x)\wedge(((\forall q((ON(q)\neq y)\wedge(\chi(p, q)=0)))\cdot$$

$$\vee(\exists q((ON(q)=y)\wedge(\chi(p,q)=0))))\wedge R_p(x, y). \tag{2.2}$$

ППФ перехода: изъять $ON(p)=x$, $ON(q)=y$; добавить $ON(p)=y$. Операторы обратного перехода обозначим ПЕРЕХОД$^{-1}$ (p, x, y). (Функция $\chi(p, q)$ определена на $P\times P$ и равна 1, если p и q оба принадлежат P_1 или P_2; $\chi(p, q)=0$ в остальных случаях.) Переходы выполняют по очереди с участием элементов p из P_1 и P_2, причем разрешен пропуск хода.

Пусть $S\in\Sigma$, определим функцию m: $\Sigma\to\mathbf{Z}_+$,

$$m(S)=\sum_i\psi(p_i^1)-\sum_j\psi(p_j^2), \quad \text{где} \quad p_i^1\in P_1, \quad p_j^2\in P_2 \text{ и } p_k^l$$

из набора $\{ON(p_k^l)=x_m\}$ ППФ данного состояния S. Потребуем, чтобы система удовлетворяла следующему условию:

$$|m(S_1)|<|m(S_2)|\,\forall S_1\in\Sigma\backslash Sg \text{ и } \forall S_2\in Sg.$$

Пусть

$$Sg=Sg^1\cup Sg^2,\ Sg^1=\{S\in Sg\,|\,m(S)>0\},\ Sg^2=\{S\in Sg\,|\,m(S)<0\}.$$

При переходах с участием $p\in P_1$ целевыми являются состояния из

$$Sg^1, \text{ при } p\in P_2 - \text{из } Sg^2.$$

Далее в работе систему из определения 1 мы называем системой (2.1) или просто системой.

Пример 1. *Задача планирования ремонтов энергооборудования.* В этой задаче задан период планирования $T_{\max}$, например, 30 дней при месячном планировании. Задана функция $f(i)$ резерва мощности (ремонтной площадки), где i — номер дня периода планирования. Требуется составить план ремонтов, если имеется массив заявок, каждая из которых содержит 3 числа (для j-го агрегата): w_j — заявленная мощность агрегата, h_j — снижение рабочей мощности энергосистемы из-за ремонта, $x_j^{\max}$ — заявленная продолжительность ремонта. Здесь для простоты мы пренебрегаем прочими параметрами заявки. При реализации метода в виде программы ПИОНЕР-2 все параметры были учтены [6]. Значения всех исходных данных принадлежат $\mathbf{Z}_+$. Критерий оптимизации плана — максимум суммарной мощности отремонтированных агрегатов.

В терминах системы (2.1): $X=(Y\cup\{g\})\times Y\times(P_{\text{рем}}\cup Q_{\text{рем}}\cup\{r\})$, $Y=\{0, 1, 2, \ldots, T_{\max}\}$, $P=P_1\cup P_2$, $P_1\cap P_2=\varnothing$, $P_1=P_{\text{рем}}\cup P_{\text{резерв}}$, $P_{\text{резерв}}=\bigcup\limits_{i=1}^{T_{\max}}P_i^{\text{рез}}$, $P_2=$ $=Q_{\text{рем}}\cup Q_{\text{сниж}}$, $Q_{\text{сниж}}=\bigcup\limits_{i=1}^{T_{\max}}\bigcap\limits_{j=1}^{|Q_{\text{рем}}|}Q_{ij}^{\text{сниж}}$, где $|Q_{\text{рем}}|=|P_{\text{рем}}|$ — число агрегатов, содержащихся в заявках; между элементами $Q_{\text{рем}}$ и $P_{\text{рем}}$ установлено взаимно-однозначное соответствие $(q_j\leftrightarrow p_j)$, т. е. каждому агрегату $p_j\in P_{\text{рем}}$ поставлен в соответствие его дубликат $q_j\in Q_{\text{рем}}$. (Значения $|P_{\text{резерв}}|$ и

$|Q_{\text{резерв}}|$ определены ниже.)

$$R_p(x, y)=\begin{cases} ((x=(0, y_2, p))\wedge(y-(0, y_2+1, p)))\vee((x=(y_1, y_2, p))\wedge \\ \wedge((y=(y_1+1, y_2, p)))\vee((x=(x_i^{\max}, y_2, p))\wedge(p=p_i)\wedge \\ \wedge(y=(g, 0, p))), \quad \text{если } p\in P_{\text{рем}}; \\ ((x=(y_1, 0, q_j))\wedge(y_1>0))\wedge((y=(y_1-y_2, y_2, p_j))\wedge \\ \wedge((y_1-y_2)>0))\vee(y=(y_1, 1, r))), \\ \qquad \text{если } p\in Q_{y_1 j}^{\text{сниж}}\subset Q_{\text{сниж}}; \\ (((x=(y_1, 0, r))\wedge(y=(y_1, 1, r))\vee((x=(y_1, 1, r))\wedge \\ \wedge(y=(y_1, 0, q_j)))\wedge(y_1>0), \quad \text{если } p\in P_{\text{резерв}}; \\ F\,(\text{ложно}), \quad \text{если } p\in Q_{\text{рем}}. \end{cases}$$

Отметим, что отношение достижимости R_p здесь несимметрично, т. е. $\exists p, x, y$ такие, что $R_p(x, y)\neq R_p(y, x)$. Для задания частичной функции $ON(p)$ достаточно выписать ее значения в начальном состоянии S_0.

$$ON(p)=\begin{cases} (0, 0, p), & \text{если } p\in P_{\text{рем}}; \\ (g, 0, p_j), & \text{если } p=q_j\in Q_{\text{рем}}; \\ (y_1, 0, r), & \text{если } p\in P_{y_1}^{\text{рез}}\subset P_{\text{резерв}}; \\ (y_1, 0, q_j), & \text{если } p\in Q_{y_1 j}^{\text{сниж}}\subset Q_{\text{сниж}}. \end{cases}$$

$$\psi(p)=\begin{cases} w_i, & \text{если } p=q_i\in Q_{\text{рем}} \\ \sum_{i=1}^{|Q_{\text{рем}}|} w_i, & \text{если } p\in P_{\text{рем}}; \\ \psi_0, & \text{если } p\in P_{\text{резерв}}\cup P_{\text{сниж}}. \end{cases}$$

Здесь ψ_0 — квант резерва снижения мощности — общий делитель всех значений $f(i)$ резерва мощности и всех значений h_j снижения мощности (для всех агрегатов), например, $\varphi_0=1$ МВт. Теперь можно определить $|P_{\text{резерв}}|$ и $|Q_{\text{сниж}}|$, задав $|P_i^{\text{рез}}|$ и $|Q_{ij}^{\text{сниж}}|$. Итак, $|P_i^{\text{рез}}|=\dfrac{f(i)}{\psi_0}$, $|Q_{ij}^{\text{сниж}}|=\dfrac{h_j}{\psi_0}$.

Функционирование системы легко описать, используя формулы для оператора перехода (2.2). Здесь начальное состояние S_0 соответствует нулевому дню периода планирования, а целевые (из Sg^1) соответствуют максимуму отремонтированной мощности. Таким образом, состояния из Sg^1 можно описать как состояния к концу периода планирования, в которых истинны ППФ $ON(p_i)=(g, 0, p_i)$ для таких номеров i, что $\sum_i \psi(q_i)$ максимальна ($q_i\in Q_{\text{рем}}$). Переход системы в новое состояние с изъятием ППФ вида $ON(q_i)=(g, 0, p_i)$ означает завершение ремонта агрегата p_i. Переход с добавлением ППФ $ON(p_i)=(1, y_2, p_i)$ означает вывод в ремонт агрегата p_i в день y_2, а с добавлением ППФ $ON(p_i)=(0, y_2+1, p_i)$ и изъятием $ON(p_i)=(0, y_2, p_i)$ означает, что в день y_2+1 агрегат p_i еще не выведен в ремонт в данном варианте плана. Для пояснения этого примера обратимся к рис. 1. На нем изображена постановка задачи планирования ремонтов для двух агрегатов на период в 3 дня: $w_1=5$, $w_2=2$; $h_1=3$, $h_2=2$; $x_1^{\max}=x_2^{\max}=2$; $T_{\max}=3$; $f(1)=4$, $f(2)=5$, $f(3)=3$. Из рисунка видно, что для составления плана ремонтов элементам p_i надо из точек $(0, 0, p_i)$ пройти к точкам $(g, 0, p_i)$. В частности, чтобы элементу p_2 пройти к точке $(g, 0, p_2)$ по любому из путей $(0, 0, p_2)\to(1, 0, p_2)\to(2, 0, p_2)\to(g, 0, p_2)$

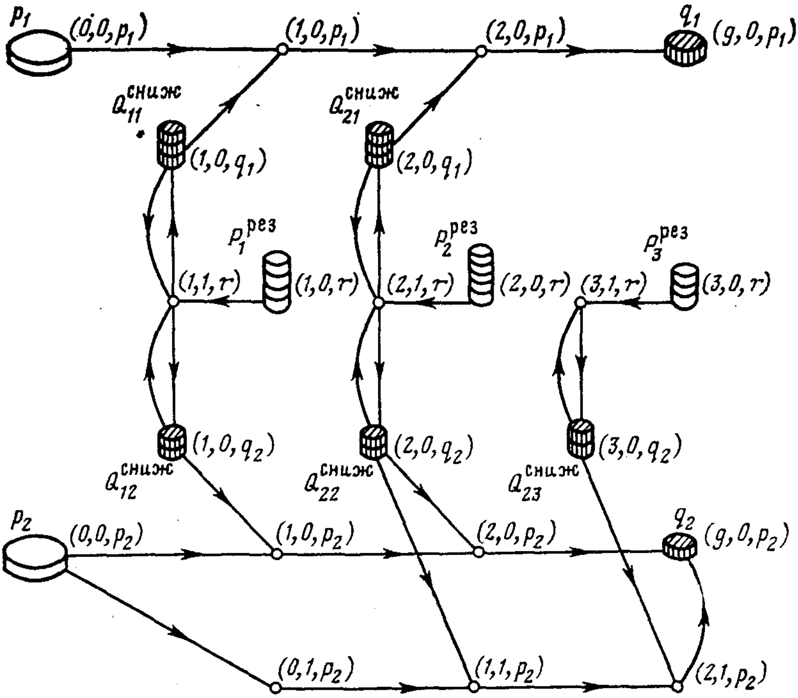

Рисунок. Интерпретация сложной системы в задаче месячного планирования ремонтов энергооборудования

или $(0,0,p_2)\to(0,1,p_2)\to(1,1,p_2)\to(g,0,p_2)$, нужно предварительно уничтожить элементы из множества (пирамиды) $Q_{12}^{сниж}$, стоящие в точке $(1,0,q_2)$, а также пирамид $Q_{22}^{сниж}$, $Q_{32}^{сниж}$ в точках $(2,0,q_2)$, $(3,0,q_2)$. Элементы этих пирамид контролируют точки путей элемента p_2 к цели. Очевидно, пирамиды элементов из $Q_{сниж}$ соответствуют снижению мощности в энергосистеме на время ремонта. Для ликвидации элементов из $Q_{сниж}$ служат множества (пирамиды) $P_1^{рез}$, $P_2^{рез}$ и $P_3^{рез}$ в точках $(1,0,r)$, $(2,0,r)$ и $(3,0,r)$, соответствующие резерву мощности по дням. Надо произвести переход, т. е. передвижение элемента из $P_1^{рез}$ в точку $(1,1,r)$, затем передвинуть элемент из $Q_{12}^{сниж}$ в эту же точку, т. е. произвести взятие, затем следующий элемент из $P_1^{рез}$ и т. д. Если удастся разменять в точке $(1,1,r)$ все элементы из $Q_{12}^{сниж}$, то точка $(1,0,p_2)$ станет свободно проходимой для элемента p_2. Если же это не удается (как и показано на рис. 1) из-за того, что 3 элемента пирамиры $P_1^{рез}$ были истрачены на снятие контроля с точки $(1,0,p_1)$, т. е. ликвидацию $Q_{11}^{сниж}$, а оставшегося одного не хватает для уничтожения двух элементов $Q_{12}^{сниж}$, то элемент p_2 вынужден совершить перемещение в точку $(0,1,p_2)$. Таким образом, в первый день периода планирования в ремонт может быть выведен лишь один из агрегатов (например, p_1) из-за недостаточного резерва мощности. Второй агрегат p_2 будет выведен в ремонт во второй день (передвижение $(0,1,p_2)\to(1,1,p_2)$). Различным вариантам плана ремонтов соответствуют различные варианты передвижений элементов из P по точкам из X.

Пример 2. *Задача программирования шахматной игры.* Здесь X — 64 поля доски. P_1 и P_2 — белые и черные фигуры; $R_p(x,y)$ задаются правилами игры, разрешающими или запрещающими фигуре p сделать ход с поля x на поле y; $ON(p)=x$, если фигура p стоит на поле x, $\psi(p)$ — стоимость фигуры p(п.— 1, К — 3, С — 3, Л — 5, Ф — 9, Кр — 200). S_0 — исходная позиция для анализа или начальная позиция; Sg — множество позиций, которые могут быть получены из всевозможных матовых позиций через два полухода взятием короля (если разрешить это взятие). ПЕРЕХОД (p, x, y) — ход фигурой p с поля x на поле y, причем, если на поле y стоит фигура другого цвета, выполняется взятие. Шахматная задача не вполне соответствует определению 1: мы пренебрегли таким важным шахматным понятием, как блокада (в системе (2.1) несколько фигур одного цвета могут стоять на одном поле), а также некоторой спецификой шахмат — рокировкой, взятием на проходе, превращением пешки и т. п. При опробовании метода на шахматной модели (программа ПИОНЕР), естественно, все это было учтено [3, 4].

При такой постановке задачи, как в определении 1, для поиска оптимальной последовательности переходов в целевое состояние можно было бы использовать методы доказательства теорем в исчислении предикатов. Теоретически это дало бы нам точное решение задачи. Но поиск пришлось бы вести в пространстве огромной размерности (для нетривиальных примеров), т. е. практически решение не было бы получено. Займемся поиском приближенного решения. Для этого перейдем к новому более глубокому представлению задачи на специальном языке, для которого представление (2.1) послужит описанием предметной области.

3. Основные определения. В задачах распознавания образов для представления иерархической структурной информации, содержащейся в каждом образе, т. е. для описания образов с помощью более простых подобразов, был предложен лингвистический подход [14]. Этот подход выявляет аналогию между иерархической структурой образов и синтаксисом языков. Правила, управляющие объединением подобразов в образы, обычно задаются т.н. грамматикой описания образов, причем мощность такого описания объясняется рекурсивной природой грамматик. Применяя аналогичный подход для описания иерархических систем, воспользуемся теорией формальных языков в аспекте, развитом в [15—19]. Начнем с определения класса используемых грамматик, который аналогичен грамматикам из [18, 19].

Определение 2. *Недетерминированной формально-параметрической программной грамматикой* (НФППГ) называется следующая совокупность Г:

$$\Gamma = (V_T,\ V_N,\ S,\ V_{PR},\ E,\ H,\ Parm,\ \mathscr{L},\ R),$$

V_T — алфавит терминальных символов; V_N — алфавит нетерминальных символов; S — начальный символ $(S \in V_N)$.

V_{PR} — алфавит исчисления предикатов первого порядка PR, $V_{PR} = Pred \cup Con \cup Var \cup F \cup \{«(», «)», «,», «=», «\vee», «\wedge», «\rceil», «\forall», «\exists»\}$, где $Pred$ — предикатные буквы, Con — константные буквы, Var — переменные, F — функциональные буквы $(F = Fcon \cup Fvar)$. Функциональной формулой исчисления PR называется: а) буква из Con или Var, б) $f(\varphi_1, \varphi_2, \ldots, \varphi_k)$, где $f \in F$, φ_i — функциональные формулы, $k = k(f)$ — количество аргументов данной функциональной буквы. Предикатами из PR (или ППФ) называются буквы из $Pred$, выражения вида $\varphi_1 = \varphi_2$, где φ_1, φ_2 — функциональные формулы, а также выражения, составляемые из них с помощью обычных правил.

E — перечислимое множество, называемое предметной областью.

H — интерпретация исчисления PR на множестве E, т. е. некоторое соответствие следующего вида. H относит каждой а) предикатной букве $Q \in Pred$ k-местное отношение на множестве E, $k = k(Q)$; б) константной букве $w \in Con$ элемент из E; в) функциональной букве $f \in F$ множество $D(f) \subset E^k$, $k = k(f)$ — число аргументов функциональной буквы f. Если $f \in Fcon$, то задано вычислимое отображение $H(f)$: $: D(f) \to E$. Если $f \in Fvar$, то для нее определяется $\gamma(f) : D(f) \to E$, изменяющееся в процессе вывода в грамматике Г. Для упрощения обозначений будем писать f вместо $H(f)$ или $\gamma(f)$, т. к. из контекста всегда ясно, что имеется в виду: формула или ее интерпретация. Таким образом, интерпретация H позволяет вычислить значение любой функциональной формулы (оно лежит в E) и любого предиката (F — «истина» или T — «ложь»), если заданы значения всех входящих в них переменных.

Parm — отображение $V_T \cup V_N$ в 2^{Var}, ставящее в соответствие каждому символу алфавита $V_T \cup V_N$ набор формальных параметров, причем $Parm(S) = Var$.

$\mathscr{L}$ — конечное множество, называемое множеством меток.

R — множество продукций, т. е. конечное множество наборов вида:

$$(l, Q, A, B, \pi_a, \pi_f, F_S, F_F)$$

Здесь $l \in \mathscr{L}$ — метка продукции (метки разных продукций различны, и далее множества меток будут отождествляться с множествами помеченных ими продукций); Q — ППФ исчисления предикатов PR — условие применимости продукции, причем в Q входят лишь те переменные из Var, которые принадлежат $Parm(A)$; $A \in V_N$; $B \in (V_T \cup V_N)^*$ — цепочка в алфавите грамматики Г, выражение вида $A \to B$ называется ядром продукции); π_f и π_a — последовательности функциональных формул, соответствующих всем формальным параметрам каждой функциональной буквы из $Fvar$ и каждого вхождения символов из $V_T \cup V_N$ в цепочку B соответственно (фактические параметры); F_S, $F_F \subset 2^{\mathscr{L}}$ — множества переходов в случае успеха и неудачи соответственно.

Результатом вывода в НФППГ служит конечное множество цепочек из V_T^*, в которых каждому формальному параметру (для каждого вхождения терминального символа в цепочку) приписано значение из E и каждой букве $f \in Fvar$ поставлено в соответствие отображение $\gamma(f)$.

Вывод в НФППГ происходит следующим образом. Началом вывода служит цепочка S, причем ее формальным параметрам приданы начальные значения из E и заданы исходные отображения $\gamma(f)$ для всех $f \in Fvar$. В качестве начального множества допустимых продукций принимается все множество R. К текущей цепочке применяется каждая из продукций из множества допустимых, символ A для которой входит в цепочку. В результате применения продукции образуется новая цепочка и новое множество допустимых продукций. Вывод для каждой из цепочек, полученных из данной, происходит в дальнейшем независимо. Если ни одну из допустимых продукций нельзя применить, то вывод данной цепочки прекращается. Если эта цепочка состоит только из терминальных символов, то она поступает в множество результатов вывода, в противном случае — отбрасывается.

Применение продукции происходит следующим образом. Выбирается самое левое вхождение символа A в цепочку. Вычисляется значение предиката Q. Если $Q=F$, то множеством допустимых продукций становится F_F, и применение продукции заканчивается. Если $Q=T$, то символ A заменяется на цепочку B; происходит вычисление значений всех формул из π_a, соответствующих параметрам символов, и параметры принимают вычисленные значения; новые отображения $\gamma(f)$, $f \in Fvar$ задаются с помощью формул из π_f; множеством допустимых продукций становится F_S, и применение продукции заканчивается. В записи продукции формулы из π_f, оставляющие $\gamma(f)$ неизменными, опускаются.

В конструкциях, которыми оснащена НФППГ легко просматриваются аналогии с языком программирования СНОБОЛ-4.

Определение 3. *Языком* $L[\Gamma]$, *порожденным* НФППГ Г, называется объединение всех множеств, являющихся результатами вывода в НФППГ.

Для построения языка траекторий нам понадобится функция, заданная на компонентах сложной системы (2.1).

Определение 4. *Картой множества* X с параметрами x и p для системы (2.1) называется отображение: $\mathrm{MAP}_{x,p}: X \to \mathbf{Z}_+$, где $x \in X$, $p \in P$, которое строится следующим образом. Пусть $x \in X$, $p \in P$; рассмотрим семейство непустых подмножеств $M^i_{x,p}$ множества $X: M^1_{x,p} = \{m | R_p(m, x) = T\}$, $M^i_{x,p} = \{m | m \in X \setminus \bigcup_{j=1}^{i-1} M^j_{x,p},\ R_p(m, m_{i-1}) = T\}$, $i > 1$. Начиная с некоторого k, $M^k_{x,p} = \varnothing$, т. к. X — конечно $(n = |X|)$. Очевидно, $M^i_{x,p} \cap M^j_{x,p} = \phi\, (i \neq j)$, $\bigcup_i M^i_{x,p} \subseteq X$.

$$\mathrm{MAP}_{x,p}(v) = \begin{cases} i, & \text{если } v \in M^i_{x,p}; \\ 2n, & \text{если } v \notin M^j_{x,p} \text{ и } v \neq x; \\ 0, & \text{если } v = x. \end{cases}$$

Легко проверить, что карта множества X для фиксированного элемента p задает на X несимметричную функцию расстояния:

1. $\mathrm{MAP}_{x,p}(y) \geqslant 0$ при $x \neq y$; $\mathrm{MAP}_{x,p}(x) = 0$;
2. $\mathrm{MAP}_{x,p}(y) + \mathrm{MAP}_{y,p}(z) \geqslant \mathrm{MAP}_{x,p}(z)$.

Если отношение R_p системы (2.1) симметрично, то $\mathrm{MAP}_{x,p}(y) = \mathrm{MAP}_{y,p}(x)$, т. е. $\mathrm{MAP}_{x,p}(y)$ — метрика, а X — метрическое пространство. В этом случае каждый из элементов $p \in P$ задает на X свою метрику. Метрика позволит нам строить из точек множества X «траектории для передвижения» по ним элементов из P.

4. Язык траекторий.

Определение 5. *Траекторией* в системе (2.1) для элемента $p \in P$ с началом в $x \in X$ и концом в $y \in X (x \neq y)$ длиной l называется цепочка символов с параметрами вида: $t_0 = \alpha(x)\alpha(x_1)\ldots\alpha(x_l)$, где $x_1, x_2, \ldots, x_l$ — элементы X, $x_i \neq x_j$ (при $i \neq j$); $x_l = y$; кроме того, ППФ $ON(p) = x$, $R_p(x, x_1)$, $R_p(x_i, x_{i+1})$ — истинны (для $i = 1, 2, \ldots, l-1$). Обозначим $t_p(x, y, l)$ — множество траекторий, у которых совпадают p, x, y, l; $\mathscr{P}(t_0) = \{x, x_1, \ldots, x_l\}$ — упорядоченное множество параметрических букв траектории t_0.

Определение 6. Траектория $t \in t_p(x_0, y_0, l_0)$ называется *кратчайшей* для заданных x_0, y_0, p, если для любой другой траектории из $t_p(x_0, y_0, l)$ имеет место: $l \geqslant l_0$.

Для конкретных систем (2.1), как правило, легко строятся грамматики класса НФППГ, порождающие кратчайшие траектории. В случае, если R_p — симметрично, такую грамматику можно построить в общем виде.

Теорема 1. Пусть в системе (2.1) $x_0 \in X$, $y_0 \in X$ $(x_0 \neq y_0)$, $p \in P$ и $ON(p) = x_0$. Условие

$$\mathrm{MAP}_{x_0, p}(y_0) = l_0 (l_0 < 2n,\ n = |X|) \tag{4.1}$$

является *необходимым и достаточным* для существования кратчайших траекторий $t_p(x_0, y_0, l_0)$. Если отношение R_p симметрично, т. е. для всех $x \in X$, $y \in X$, $p \in P$ $R_p(x, y) = R_p(y, x)$, то все кратчайшие траектории $t_p(x_0, y_0, l_0)$ порождаются *грамматикой* $\Gamma^{(1)}_t$ (табл. 1).

Доказательство. Предположим, что $t_0 \in t_p(x_0, y_0, l_0)$ существует и является кратчайшей. Докажем (4.1). Доказательство проведем индукцией по l_0. При $l_0 = 1$ утверждение легко проверяется. Предположим, что при $l_0 < m$ утверждение верно. Пусть $l_0 = m$ и $t_m \in t_p(x_0, y_0, m)$ — кратчайшая. Докажем, что $\mathrm{MAP}_{x_0,p}(y_0) = m$. Рассмотрим укороченную траекторию $t_{m-1} \in t_p(x_0, x_{m-1}, m-1)$, $t_{m-1} = \alpha(x_0)\alpha(x_1)\ldots\alpha(x_{m-1})$, которая получается из t_m после отбрасывания последнего символа. Если $t_m \in t_p(x_0, x_m, m)$ — кратчайшая $(x_m = y_0)$, то и t_{m-1} — кратчайшая. Но из предположения сле-

Грамматика кратчайших траекторий $\Gamma_t^{(1)}$

$\mathscr{L}$	Q	Ядро	π_f	F_S	F_F
1	Q_1	$S(x, y, l) \rightarrow A(x, y, l)$		*two* !	$\varnothing$
2_i	Q_2	$A(x, y, l) \rightarrow \alpha(x)\, A(\mathrm{next}_{i,\, u_0,\, p}(x, l), y, f(l))$		*two* $\{2_i\}$	3
3	Q_3	$A(x, y, l) \rightarrow \alpha(y)$		$\varnothing$	$\varnothing$

$V_T = \{\alpha\}$, $V_N = \{S, A\}$, $Pred = \{Q_1, Q_2, Q_3\}$, $Var = \{x, y, l\}$,
$Fcon = \{f, \mathrm{next}_1, \ldots, \mathrm{next}_n\}$ $(n = |X|)$, $Fvar = \{x_0, y_0, l_0, p\}$.
Для краткости принято $u_0 = (x_0, y_0, l_0)$.
$E = \mathbf{Z}_+ \cup X \cup P$.
$f(l) = l - 1$, $D(f) = \mathbf{Z}_+ \setminus \{0\}$.
next_i определяется следующим образом.

$D(\mathrm{next}_i) = X \times \mathbf{Z}_+ \times X^2 \times \mathbf{Z}_+ \times P$,

$\mathrm{next}_i(x_0, l_0) = x_0$,

$\mathrm{SUMMA} = \{v \mid v \in X,\ \mathrm{MAP}_{x_0,\, p}(v) + \mathrm{MAP}_{y_0,\, p}(v) = l_0\}$,

$\mathrm{ST}_k(x) = \{v \mid v \in X,\ \mathrm{MAP}_{x,\, p}(v) = k\}$,

$\mathrm{MOVE}_l(x) = \mathrm{ST}_1(x) \cap \mathrm{ST}_{l_0 - l}(x_0) \cap \mathrm{SUMMA}$.

Если $\mathrm{MOVE}_l(x) = \{m_1, m_2, \ldots, m_r\} \neq \varnothing$, то

$$\begin{cases} \mathrm{next}_i(x, l) = m_i & \text{при } i \leqslant r, \\ \mathrm{next}_i(x, l) = m_r & \text{при } r < i \leqslant n, \end{cases}$$

иначе $\mathrm{next}_i(x, l) = x$.

$Q_1(x, y, l) = (\mathrm{MAP}_{x,\, p}(y) = l) \wedge (0 < l \leqslant n)$,
$Q_2(l) = (l \geqslant 1)$
$Q_3 = T$
$\mathscr{L} = \{1, 3\} \cup two$, $two = \{2_1, 2_2, \ldots, 2_n\}$.
$Parm: S \rightarrow Var$, $A \rightarrow Var$, $\alpha \rightarrow \{x\}$.
В начале вывода: $x = x_0$, $y = y_0$, $l = l_0$, $x_0 \in X$, $y_0 \in X$, $l_0 \in \mathbf{Z}_+$, $p \in P$.

дует, что $\mathrm{MAP}_{x_0,p}(x_{m-1}) = m-1$. Тогда из определения 4 следует, что $x_{m-1} \in M^{m-1}_{x_0,p}$. Т. к. $R_p(x_{m-1}, y_0)$ истинна, то $y_0 \in (\bigcup_{j=1}^{m-1} M^j_{x_0,p}) \cup M^m_{x_0,p}$. Если $y_0 \in \bigcup_{j=1}^{m-1} M^j_{x_0,p}$, то траектория t_m некратчайшая, т. к. найдется траектория $t' \in t_p(x_0, y_0, m-1)$ длиной $m-1$. Получаем противоречие. Итак, $y_0 \in M^m_{x_0,p}$, т. е. $\mathrm{MAP}_{x_0,p}(y_0) = m$.

Обратно, пусть выполнено (4.1). Покажем, что существует траектория, принадлежащая $t_p(x_0, y_0, l_0)$, и она является кратчайшей. Доказательство проведем по индукции. При $l_0 = 1$ утверждение очевидно. Пусть оно верно при $l_0 < m$. Пусть теперь $l_0 = m$ и $\mathrm{MAP}_{x_0,p}(y_0) = m$. Тогда по определению 4 существует $x_{m-1} \in M^{m-1}_{x_0,p}$ такая, что $R_p(y_0, x_{m-1}) = T$. Но из того, что $x_{m-1} \in M^{m-1}_{x_0,p}$, вытекает: $\mathrm{MAP}_{x_0,\, p}(x_{m-1}) = m-1$. Следовательно, по предположению индукции существует кратчайшая траектория $\alpha(x_0)\alpha(x_1)\ldots\alpha(x_{m-1})$ длины $m-1$. В таком случае и траектория $\alpha(x_0)\alpha(x_1)\ldots\alpha(x_{m-1})\alpha(y_0)$ длины m будет кратчайшей.

Для завершения доказательства теоремы остается показать, что все траектории $t_p(x_0, y_0, l_0)$ порождаются грамматикой из табл. 1, если R_p симметрично. Эта грамматика в соответствии с определением 2 принадлежит классу НФППГ. Отметим, что множество функциональных букв $Fvar$ в ней является четверкой нульместных функций p, x_0, y_0, l_0, т. е. $\Gamma_t^{(1)}=\Gamma(p, x_0, y_0, l_0)$. Очевидно, что каждая из цепочек, порождаемых $\Gamma_t^{(1)}$ является траекторией из $t_p(x_0, y_0, l_0)$. В самом деле, для каждой порожденной цепочки $\alpha(x_0)\alpha(x_1)\ldots\alpha(y_0)$ элементы $x_i\in ST_i(x_0)=M^i_{x_0,p}$ следовательно, эта цепочка является кратчайшей траекторией.

Как уже отмечалось выше, все подцепочки кратчайшей траектории являются кратчайшими траекториями с началом в x_0 и концом в $x_i(i=1, 2, \ldots, l_0)$ соответственно. Учитывая симметричность отношения R_p, все «обращенные» подцепочки с началом в y_0 и концом в $x_i(i=l_0-1, \ldots, 1, 0)$ также будут кратчайшими траекториями. Следовательно, $x_i\in M^{l_0-i}_{y_0,p}$. Это означает, что для любой кратчайшей траектории $\alpha(x_0)\alpha(x_1)\ldots$ $\ldots\alpha(y_0)\in t_p(x_0, y_0, l_0)$ $x_i\in M^i_{x_0,p}\cap M^{l_0-i}_{y_0,p}$, т. е. $MAP_{x_0,p}(x_i)=i$ и $MAP_{y_0,p}(x_i)=l_0-i$, а значит

$$MAP_{x_0,p}(x_i)+MAP_{y_0,p}(x_i)=l_0. \tag{4.2}$$

Обратно, если для некоторого $x\in X$ выполнено (4.2), то он обязательно входит в множество $\mathscr{P}(t_i)$ хотя бы одной из кратчайших траекторий вида $t_i\in t_p(x_0, y_0, l_0)$. Это следует из того, что $MAP_{x_0,p}(x)\geqslant 0$ и $MAP_{y_0,p}(x)\geqslant 0$, а их сумма равна l_0. Т. е. существует $j(0\leqslant j\leqslant l_0)$, что $MAP_{x_0,p}(x)=j$, $MAP_{y_0,p}(x)=l_0-j$. Тогда существуют две кратчайшие траектории $t^1\in t_p(x_0, x, j)$ и $t^2\in t_p(y_0, x, l_0-j)$; траектория, построенная из тех же символов, что t^2, но в обратном порядке, $t^3\in t_p(x, y_0, l_0-j)$ также будет кратчайшей. Конкатенация t^1 и t^3 дает искомую кратчайшую траекторию, содержащую x. Итак, любой элемент множества X входит в совокупность параметрических букв $\bigcup_{t^i}\mathscr{P}(t^i)$ для кратчайших траекторий вида $t^i\in t_p(x_0, y_0, l_0)$ в том и только в том случае, если выполнено (4.2). Эти рассуждения обосновывают алгоритм вычисления функции $\operatorname{nex} t_i(x, l)$ (табл. 1).

Далее воспользуемся индукцией. Очевидно, грамматика траекторий порождает первый символ $\alpha(x_0)$ всех кратчайших траекторий из $t_p(x_0, y_0, l_0)$. Предположим, что она порождает m первых символов любой кратчайшей траектории из $t_p(x_0, y_0, l_0)$. Покажем, что она порождает и $(m+1)$-й символ $\alpha(x_m)$. Имеем $MOVE(x_{m-1})=ST_1(x_{m-1})\cap ST_m(x_0)\cap SUMMA$. Поскольку $t_p(x_0, y_0, l_0)$ кратчайшие, для x_m выполнено: $x_m\in ST_m(x_0)=M_{x_0,p}$, но $x_m\in SUMMA$ (из (4.2)) и $x_m\in ST_1(x_{m-1})=M^1_{x_{m-1}}, p'$ поскольку $R_p(x_{m-1}, x_m)=T$ по определению 5. Итак, $x_m\in MOVE(x_{m-1})$, т. е. $(m+1)$-й символ порождается грамматикой траекторий $\Gamma_t^{(1)}$. Теорема доказана.

Определение 7. Траектория $t_0=\alpha(x_0)\alpha(x_1)\ldots\alpha(x_{l_0})$ из $t_p(x_0, y_0, l_0)$ в системе (2.1) называется *допустимой степени* k, если существует подмножество $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{k-1}}\}\subset\mathscr{P}(t_0)$, $i_1<i_2<\ldots<i_{k-1}$, $k\leqslant l_0$ такое, что соответствующие подцепочки $\alpha(x_0)\ldots\alpha(x_{i_1})$, $\alpha(x_{i_1})\ldots\alpha(x_{i_2}), \ldots, \alpha(x_{i_{k-1}})\ldots$ $\ldots\alpha(x_{l_0})$ являются кратчайшими траекториями.

Рассуждая неформально, можно провести аналогию: кратчайшая траектория аналогична отрезку прямой, соединяющей две точки на плоскости, допустимая траектория степени k аналогична k-звенной ломаной, соединяющей эти точки. Очевидно, если $k=l_0$, множество допустимых траекторий степени k совпадает с множеством всех траекторий длины l_0 с началом в x и концом в y для элемента p. В конкретных задачах особую роль играют допустимые траектории степени 2. Как и в случае кратчайших для конкретных систем (2.1), как правило, легко строятся грамматики, порождающие такие траектории (пример 3). Если же R_p симметрично, построим порождающую грамматику в общем виде.

Теорема 2. Если отношение R_p в системе (2.1) *симметрично*, все допустимые траектории степени 2 вида $t_p(x_0, y_0, l_0)$ порождаются *грамматикой* $\Gamma_t^{(2)}$ (табл. 2).

Грамматика допустимых траекторий степени 2 $\Gamma_t^{(2)}$

$\mathscr{L}$	Q	Ядро	π_f	F_S	F_F
1	Q_1	$S(x, y, l) \to A(x, y, l)$		*two*	$\varnothing$
2_i	Q_2	$A(x, y, l) \to A(x, \mathrm{med}_{i,p}(x, y, l), l\,\mathrm{med}_{i,p}(x, y, l))$ $A(\mathrm{med}_{i,p}(x, y, l), y, l - l\,\mathrm{med}_{i,p}(x, y, l))$		*tr*	*tr*
3_j	Q_3	$A(x, y, l) \to \alpha(x)\, A(\mathrm{next}_j(x, l), y, f(l))$		*tr*	4
4	Q_4	$A(x, y, l) \to \alpha(y)$		*tr*	$\varnothing$

$V_T = \{\alpha\}$, $V_N = \{S, A\}$, $Pred = \{Q_1, Q_2, Q_3, Q_4\}$, $Var = \{x, y, l\}$,
$Fcon = \{f, \mathrm{next}_1, \ldots, \mathrm{next}_n, \mathrm{med}_1, \ldots, \mathrm{med}_n, \mathrm{lmed}_1, \ldots, \mathrm{lmed}_n\}$,
$Fvar = \{x_0, y_0, l_0, p\}$, $E = \mathbf{Z}_+ \cup X \cup P$.
$f(l) = l - 1$, $D(f) = \mathbf{Z}_+ \setminus \{0\}$.

next_i определяется также как в $\Gamma_t^{(1)}$ (табл. 1).

med_i определяется следующим образом.

$D(\mathrm{med}_i) = X \times X \times \mathbf{Z}_+ \times P$,
$\mathrm{DOCK} = \{v \mid \mathrm{MAP}_{x,p}(v) + \mathrm{MAP}_{y,p}(v) = l,\ v \in X\}$.
Если $\mathrm{DOCK} = \{v_1, v_2, \ldots, v_m\} \neq \varnothing$, то

$$\begin{cases} \mathrm{med}_{i,p}(x, y, l) = v_i & \text{при } 1 \leqslant i \leqslant m; \\ \mathrm{med}_{i,p}(x, y, l) = v_m & \text{при } m < i \leqslant n, \end{cases}$$

иначе $\mathrm{med}_{i,p}(x, y, l) = x$ при $1 \leqslant i \leqslant n$.

$\mathrm{lmed}_{i,p}(x, y, l) = \mathrm{MAP}_{x,p}(\mathrm{med}_i(x, y, l))$, $D(\mathrm{lmed}_i) = X \times X \times \mathbf{Z}_+ \times P$.
$Q_1(x, y, l, p) = (\mathrm{MAP}_{x,p}(y) \leqslant l < 2\mathrm{MAP}_{x,p}(y)) \wedge (l < 2n)$,
$Q_2(x, y, l, p) = (\mathrm{MAP}_{x,p}(y) \neq l)$,
$Q_3(x, y, l, p) = (\mathrm{MAP}_{x,p}(y) = l) \wedge (l \geqslant 1)$,
$Q_4 = T$.
$\mathscr{L} = \{1, 4\} \cup two \cup tr$, $two = \{2_1, 2_2, \ldots, 2_n\}$, $tr = \{3_1, 3_2, \ldots, 3_n\}$.
$Parm:\ S \to \{x, y, l\}$, $A \to \{x, y, l\}$, $\alpha \to \{x\}$.

В начале вывода

$x = x_0$, $y = y_0$, $l = l_0$, $x_0 \in X$, $y_0 \in X$, $l_0 \in \mathbf{Z}_+$, $p \in P$.

Доказательство теоремы 2 приведено в приложении.

Определение 8. *Пучком траекторий* назовем совокупность траекторий вида $\{t_p(x_0, y_0, l)\}$, где $1 \leqslant l \leqslant l_0$.

Определение 9. Пусть $S \in \Sigma$. *Язык траекторий* $L_t^H(S)$ — это множество всех траекторий, порождаемых грамматикой $\Gamma_t^{(2)}$ для всех p, x, y, l, $1 \leqslant l \leqslant H$ в состоянии S. При этом число $H \in \mathbf{Z}_+$ называется *горизонтом*.

Пример 3. *Планирование ремонтов*. Несмотря на несимметричность отношения R_p (см. пример 1), в этой задаче можно определить язык траекторий, воспользовавшись грамматиками $\Gamma_t^{(1)}$ и $\Gamma_t^{(2)}$. Функция $\mathrm{MAP}_{x,p}(y)$ задается в соответствии с определением 4 и является несимметричной функцией расстояния. В частности, для примера из рис. 1 $\mathrm{MAP}_{(0,0,p_2),p_2}(1, 0, p_2) = 1$, $\mathrm{MAP}_{(0,0,p_1),p_1}(g, 0, p_1) = 3$ и т. д. Здесь определена грамматика кратчайших траекторий, совпадающая с $\Gamma_t^{(1)}$ (табл. 1), но с другой функцией next. Мы не приводим здесь (ввиду громоздкости) определения этой функции для общего случая. (Отметим лишь, что здесь она не зависит от i.) Для примера на рис. 1 функция $\mathrm{next}_{u_0,p}(x, l)$ для всех $(x, l) \in X \times \mathbf{Z}_+$ задана стрелками диаграммы. Примером кратчайшей траектории (рис. 1) здесь может служить $t_{p_2}((0, 0, p_2), (g, 0, p_2), 3) = \{\alpha(0, 0, p_2) \times \alpha(1, 0, p_2)\alpha(2, 0, p_2)\alpha(g, 0, p_2)\}$, т. е. вывод в ремонт агрегата p_2 в первый день и его завершение во второй день. Другой пример: $t_{p_2}((1, 0, r), (1, 0, q_2), 2) = \{\alpha(1, 0, r)\alpha(1, 1, r)\alpha(1, 0, q_2)\}$. Для элементов из $P_{\text{рем}}$ здесь определены допустимые траектории степени 2, порождаемые с помощью грамматики, совпадающей с $\Gamma_t^{(2)}$ (табл. 2), но с другой функцией med : $\mathrm{med}_p((0, y_2, p), (g, 0, p), x_p^{\max}) = (0, y_2+1, p)$, в остальных случаях положим $\mathrm{med}_p(x, y, l) = x$. Примером допустимой траектории степени 2 на рис. 1 может служить траектория $t_{p_2}((0, 0, p_2), (g, 0, p_2), 4) = \{\alpha(0, 0, p_2)\alpha(0, 1, p_2)\alpha(1, 1, p_2)\alpha(2, 1, p_2)\alpha(g, 0, p_2)\}$; движение по такой траектории соответствует варианту плана с выводом в ремонт агрегата p_2 во второй день.

Пример 4. *Шахматная модель* [2]. Траекторией здесь является цепочка символов с параметрами — координатами полей шахматной доски, на которые должна встать фигура, двигаясь от начального поля к конечному. На множестве X введена система координат: $x \in X$, $x = (x_1, x_2)$, $x_1 = 1, 2, \ldots, 8$, $x_2 = 1, 2, \ldots, 8$, где x_1, x_2 — номера вертикалей и горизонталей шахматной доски соответственно. При реализации $\Gamma_t^{(1)}$ и $\Gamma_t^{(2)}$ в алгоритме ПИОНЕР для повышения эффективности алгоритма потребовалось обеспечить табличное задание функции MAP. Для этого в соответствии с отношениями R_p (правилами передвижения фигур) были заданы семь квадратных таблиц — массивов 15×15. Каждый из массивов заполнен числами для одного из типов шахматных фигур по принципу: фигура ставится на центральное поле массива (там записан 0), на остальных полях записываются числа, равные числу передвижений, которое необходимо фигуре, чтобы добраться с центрального поля до данного кратчайшим путем. Эти массивы в совокупности можно задать трехмерной таблицей $\mathrm{T15}(v_1, v_2, f)$ размерностью $15 \times 15 \times 7$, тогда $\mathrm{MAP}_{x,p}(y) = \mathrm{T15}(v_1, v_2, f)$, где $x = (x_1, x_2)$, $y = (y_1, y_2)$, $v_1 = 8 - x_1 + y_1$, $v_2 = 8 - x_2 + y_2$, $f = f(p)$ — тип фигуры. Из-за симметричности отношения R_p в данной модели $\mathrm{MAP}_{x,p}(y)$ задает метрику, и имеют место теоремы 1 и 2. (Для пешки — симметрия более сложная: $R_p(x, y) = R_q(y, x)$, где p и q — черная и белая пешки.) Семь массивов 15×15 задают на X семь метрик. Грамматика $\Gamma_t^{(1)}$ (табл. 1) порождает кратчайшие траектории для передвижения фигур по пустой доске (состоящие из полей остановки), а $\Gamma_t^{(2)}$ (табл. 2) — траектории, составленные из двух кратчайших. Обе грамматики здесь являются фактически недетерминированными (порождают более одной траектории), поэтому для их реализации удобно использовать ЭВМ с параллельными вычислениями.

ПРИЛОЖЕНИЕ

Доказательство теоремы 2. Очевидно, если l_0 — таково, что ППФ Q_3 — истинна, то все кратчайшие траектории, соединяющие x_0 и y_0, порождаются продукциями с метками 1, 3_j, 4 (табл. 2). Остается рассмотреть случай, когда истинна ППФ Q_2, т. е. случай некратчайших допустимых траекторий степени 2 длины l_0. Пусть $t_0 = \alpha(x_0)\alpha(x_1)\ldots\alpha(y_0)$ — некратчайшая допустимая траектория степени 2 длины l_0. Тогда найдется $x_m \in \mathscr{P}(t_0)$, что $\alpha(x_0)\ldots\alpha(x_m)$ и $\alpha(x_m)\ldots\alpha(y_0)$ — кратчайшие. По теореме 1 $\mathrm{MAP}_{x_0, p}(x_m) = m$ и $\mathrm{MAP}_{x_m, p}(y_0) = l_0 - m$. Тогда и $\mathrm{MAP}_{y_0, p}(x_m) = l_0 - m$, отсюда $\mathrm{MAP}_{x_0, p}(x_m) + \mathrm{MAP}_{y_0, p}(x_m) = l_0$. Отсюда $x_m \in \mathrm{DOCK}$ (табл. 2). Следовательно, найдется $v_i \in \mathrm{DOCK}$ и $v_i = x_m$, а значит, одна из продукций с метками из множества *two* будет применена в процессе вывода. Таким образом, «точка стыковки» $\alpha(x_m)$ двух кратчайших траекторий будет порождена. Сами же эти кратчайшие траектории будут порождены применением продукции 3_j и 4 в соответствии с теоремой 1.

Обратно, пусть грамматика из табл. 2 порождает некоторую цепочку символов с параметрами. Из определения функции $\mathrm{next}_i(x, l)$ вытекает, что эта цепочка — траектория. Если в процессе вывода продукции с метками из *two* не применялись ни разу, то из теоремы 1 следует, что данная траектория является кратчайшей. Предположим теперь, что эти продукции применялись. Из табл. 2 вытекает, что для данной траектории такое применение было однократным. (Продукция с меткой 2_i может быть применена лишь после продукции с меткой 1, которая применяется первой и однократно; если *two* содержит более одного элемента, происходит вывод сразу нескольких цепочек. Итак, однократное применение продукции с меткой 2_i порождает в цепочке нетерминал $A(x, \mathrm{med}_{i,p}(x, y, l), \mathrm{lmed}_{i,p}(x, y, l))$, причем для $v_i = \mathrm{med}_{i,p}(x_0, y_0, l_0)$ имеет место: $\mathrm{MAP}_{x_0,p}(v_i) + \mathrm{MAP}_{y_0,p}(v_i) = l_0$. Следовательно, $\alpha(v_i)$ является «точкой стыковки» двух кратчайших траекторий, сумма длин которых равна l_0. Теорема 2 доказана.

ЛИТЕРАТУРА

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
2. *Ботвинник М. М.* Алгоритм игры в шахматы. М.: Наука, 1968.
3. *Ботвинник М. М.* О кибернетической цели игры. М.: Сов. радио, 1975.
4. *Ботвинник М. М.* О решении неточных переборных задач. М.: Сов. радио, 1979.
5. *Резницкий А. И., Бордюгов В. М., Штильман Б. М.* Метод планирования ремонтов оборудования электростанций.– Электричество, 1983, № 2.
6. *Резницкий А. И., Штильман Б. М.* Применение метода ПИОНЕР в автоматизации планирования ремонтов энергооборудования.– Автоматика и телемеханика, 1983, № 11.
7. *Ботвинник М. М., Петряев Е. И., Резницкий А. И. и др.* Применение нового метода решения переборных задач к планированию ремонтов оборудования электростанций.– Экономика и матем. методы, 1983, XIX, № 6.
8. *Поспелов Г. С.* Некоторые вопросы реализации диалоговых систем планирования, управления и проектирования. М.: ВЦ АН СССР, 1980.
9. *Хайес П.* Логика действий.– В сб.: Интегральные работы, вып. 2. М.: Мир, 1975.
10. *Попов Э. В.*, Фирдман Г. Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. М.: Наука, 1976.
11. *Рафаэл Б.* Думающий компьютер. М.: Мир, 1976.
12. *Попов Э. В.* Общение с ЭВМ на естественном языке. М.: Наука, 1982.
13. *Дойл Дж.* Система поддержания истинности.– Кибернет. сб. М.: Мир, 1983, вып. 20.
14. *Фу К.* Структурные методы в распознавании образов. М.: Мир, 1977.
15. *Хомский Н.* Формальные свойства грамматик.– Кибернет. сб., М.: Мир, 1966, вып. 2.
16. *Кнут Д. Э.* Семантика контекстно-свободных языков.– В сб.: Семантика языков программирования. М.: Мир, 1978.
17. *Розенкранц Д.* Программные грамматики и классы формальных языков.– Сб. переводов по вопросам информационной теории и практики. М.: ВИНИТИ, 1970, № 16.
18. *Кузин Л. Т.* Основы кибернетики. Т. 2. М.: Энергия, 1979.
19. *Волченков Н. Г.* Интерпретатор бесконтекстных управляемых параметрических программных грамматик.– В сб.: Вопросы кибернетики. Интеллектуальные банки данных. М.: Науч. сов. по комп. пробл. «Кибернетика» АН СССР, 1979.

Москва

Поступила в редакцию
24.IV.1984