Э. У. КАЗАРЯН, С. С. ПОГОСЯН

ПЛАНИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СХЕМ С УЧЕТОМ ПРЕДИКАТНЫХ УСЛОВИЙ

При разработке специального математического обеспечения систем программирования большой интерес представляют вопросы организации планирования расчетных схем на основе формализованной предметной области [1—4].

Во многих областях науки, в частности химической кинетике, при определении наиболее вероятного механизма сложных цепных реакций возникает необходимость учета в процессе планирования условий, налагаемых на параметры модулей. При этом модуль предметной области характеризуется следующими свойствами:

—в качестве модулей выступают элементарные химические реакции;

—входами модуля являются исходные вещества соответствующей химической реакции, а выходами—продукты этой реакции;

—на выходные параметры модуля налагается предикатное условие (например, свойство радикальности вещества — в спланированной группе каждый радикал должен обязательно быть исходным веществом какой-либо реакции в выбранной группе).

В работе изложены алгоритмы процесса планирования, реализованные в базовой диалоговой системе АНИ-81. Интерпретацией вводимых далее понятий могут служить вышеуказанные свойства.

Основные понятия

Пусть заданы:

1) конечное множество переменных $X=\{x_1, x_2, \ldots\}$ для представления параметров;
2) конечное множество $F=\{f_1, f_2, \ldots f_i \ldots\}$ операций арности $m_i \times n_i$, $m_i, n_i \geqslant 1$ для представления модулей;
3) с каждой операцией $f \in F$ арности $m \times n$ связаны наборы $\mathrm{In}(f) \subseteq X$ входных и выходных $\mathrm{Out}(t) \subseteq X$ переменных соответственно мощностей m и n.

Пару (X, F) назовем вычислительной моделью. Пусть $X \subseteq X$. Определим множество термов $T(X, F)=\{t, r, \ldots\}$, порожденных множеством X. С каждым $t \in T(X, F)$ определим множества $\mathrm{In}(t)$, $\mathrm{Out}(t)$ список $S(t)$ и глубину терма $d(t)$:

1. $f \in F \rightarrow t \in T(X, F) \rightleftarrows \mathrm{In}(f) \subseteq X$; полагаем

$\mathrm{In}(t)=\mathrm{In}(f)$, $\mathrm{Out}(t)=\mathrm{Out}(f)$, $S(t)=\langle f\rangle$, $d(t)=1$.

Пусть $t\in T(X,F)$ и с t связаны $\mathrm{In}(t)$, $\mathrm{Out}(t)$, $S(t)$, $d(t)$.

Пусть $f\in F$ и f не входит в список $S(t)$. Тогда

$$t\circ f\in T(X,F)\rightleftarrows \mathrm{In}(f)\subseteq \mathrm{Out}(t)\cup X;$$

Определяем: $\mathrm{In}(t\circ f)=\mathrm{In}(t)\cup(\mathrm{In}(f)\setminus \mathrm{Out}(t))$;

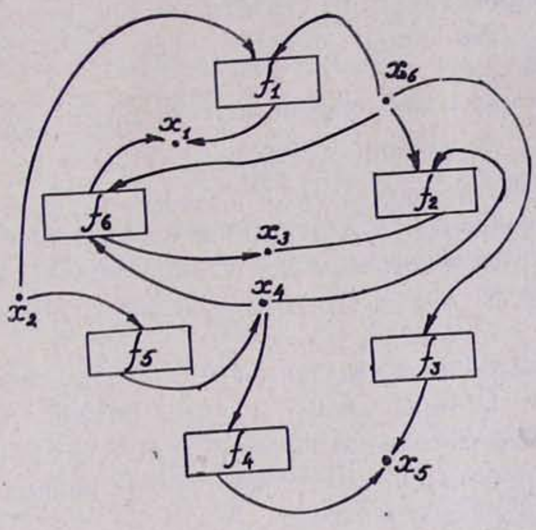
$\mathrm{Out}(t\circ f)=\mathrm{Out}(t)\cup\mathrm{Out}(f)$; $S(t\circ f)=S(t)\dot{\oplus}\langle f\rangle$; $d(t\circ f)=d(t)+1$;

Символ $\dot{\oplus}$ обозначает добавление с конца к списку $S(t)$ операции f).

Для примера (рис. 1) множество $T(\{x_2\},F)$ состоит из термов:

$t_1:\mathrm{In}(t_1)=\{x_2\}$, $\mathrm{Out}(t_1)=\{x_4\}$, $S(t_1)=\langle f_5\rangle$, $d(t_1)=1$,

$t_2:\mathrm{In}(t_2)=\{x_2\}$, $\mathrm{Out}(t_2)=\{x_4,x_5\}$, $S(t_2)=\langle f_5,f_4\rangle$, $d(t_2)=2$.



$F=\{f_1,f_2,f_3,f_4,f_5,f_6\}$, $X=\{x_1,x_2,x_3,x_4,x_5,x_6\}$

$\mathrm{In}(f_1)=\{x_2,x_6\}$, $\mathrm{Out}(f_1)=\{x_1\}$,

$\mathrm{In}(f_2)=\{x_4,x_6\}$, $\mathrm{Out}(f_2)=\{x_3\}$,

$\mathrm{In}(f_3)=\{x_6\}$, $\mathrm{Out}(f_3)=\{x_5\}$,

$\mathrm{In}(f_4)=\{x_4\}$, $\mathrm{Out}(f_4)=\{x_5\}$,

$\mathrm{In}(f_5)=\{x_2\}$, $\mathrm{Out}(f_5)=\{x_4\}$,

$\mathrm{In}(f_6)=\{x_4,x_6\}$, $\mathrm{Out}(f_6)=\{x_1,x_3\}$.

Рис. 1. Простейшая вычислительная модель.

Будем говорить, что терм t порожден списком $\langle f_{i_1}, \ldots f_{i_k} \rangle$ если $S(t) = \langle (f_{i_1}, \ldots f_{i_k}) \rangle$.

Обозначим через $S^*(t)$ множество операций списка $S(t)$. Каждая операция $f \in F$ интерпретируется как модуль вычисления $\mathrm{Out}(f)$ из параметров $\mathrm{In}(f)$. Множество $T(X, F)$ представляет различные варианты вычислений из начальных данных X, осуществляемые модулями множества F.

Произвольную пару (X, Y) назовем задачей, если $Y \setminus X \neq \varnothing$, $X, Y \subseteq \boldsymbol{X}$.

Пусть термы $t, r \in T(X, F)$.

Терм t называется подтермом r, если $S^*(t) \subseteq S^*(r)$ (обозначается $t \prec r$). Два терма $t, r \in T(X, F)$, называются эквивалентными, если $t \prec r \wedge r \prec t$, т. е. $S^*(t) = S^*(r)$ (обозначается $t \sim r$).

С каждым термом $t \in T(X, F)$ связывается предикат $P_t(x)$, определенный на множестве $\mathrm{Out}(t)$, причем: если $t \prec r$, то $\forall x \in \mathrm{Out}(t)$, $P_t(x) = 1 \to P_r(x) = 1$.

Замечание. Из этого условия вытекает, что

$$\text{если } t \prec r, \text{ то } \forall x \in \mathrm{Out}(r),\ P_r(x) = 0 \to P_t(x) = 0.$$

Будем говорить, что задача (X, Y) разрешима, если существует терм $t \in T(X, F)$, такой, что $Y \subseteq X \cup \mathrm{Out}(t)$ и $\forall x \in \mathrm{Out}(t)$, $P_t(x) = 1$. При этом терм t называется вычислительной схемой задачи (X, Y). Обозначим через $\tilde{T}(X, Y, F)$ множество всех вычислительных схем задачи (X, Y).

Вычислительная схема t задачи (X, Y) называется тупиковой, если не существует вычислительной схемы r задачи (X, Y) такой, что $S^*(r) \subset S^*(t)$. Вычислительную схему $t \in \tilde{T}(X, Y, F)$ назовем замыкающей, если $d(t) = \max\limits_{r \in \tilde{T}(X, Y, F)} d(r)$ и соответственно минимальной, если $d(t) = \min\limits_{r \in \tilde{T}(X, Y, F)} d(r)$.

Задача нахождения минимальной вычислительной схемы является алгоритмически не разрешимой. Вместе с тем минимальная вычислительная схема является тупиковой. Естественным поэтому является нахождение тупиковых вычислительных схем. При условии $P_t(x) \equiv 1$, $\forall x \in \mathrm{Out}(t)$, $\forall t \in T(X, F)$, $X \subseteq \boldsymbol{X}$ вопросы нахождения тупиковых вычислительных схем рассматриваются в работе [5].

Пусть задача (X, Y) не является разрешимой в вычислительной модели $(\boldsymbol{X}, F)$. Переопределенной для нее назовем задачу (X', Y) такую, что

1) $X \subset X' \subseteq \boldsymbol{X}$, $X' \cap Y = X \cap Y$;

2) (X', Y) — разрешима в $(\boldsymbol{X}, F)$,

3) $\forall x \in X' \setminus X$ задача $(X' \setminus \{X\}, Y)$ — не разрешима.

Алгоритм построения замыкающей вычислительной схемы.

Теорема. Для того, чтобы вычислительная схема $t^* \in \tilde{T}(X, Y, F)$ являлась замыкающей необходимо и достаточно, чтобы

$$t \prec t^*, \ \forall t \in \tilde{T}(X, Y, F).$$

Доказательство. Достаточность очевидна.

Необходимость. Пусть t не является подтермом t^* и $S(t^*) = \langle f_{i_1}, \ldots f_{i_\nu} \rangle$, $\nu \geqslant 1$, $S(t) = \langle f_{l_1}, \ldots f_{l_e} \rangle$, $e \geqslant 1$. Рассмотрим список $S' = \langle f_{k_1}, \ldots f_{k_p} \rangle$, составленный из тех операций списка $S(t)$, которые не встречаются в списке $S(t^*)$, с сохранением их порядка в списке $S(t)$. В силу предположения $p \geqslant 1$. Рассмотрим список $S'' = S(t^*) \dot{\oplus} S'$. Легко доказать, что список S'' порождает терм t' с $d(t') = d(t^*) + p$ и что $t' \in \tilde{T}(X, Y, F)$. Тогда $d(t') = \nu + p > d(t^*) = \nu$, что противоречит определению замыкающей вычислительной схемы t^*. Следовательно $t \prec t^*$.

Следствие. Замыкающая вычислительная схема единственна с точностью до эквивалентности.

Обозначим через

$$R(t, 0) = \{f \in S^*(t) / \exists x \in \mathrm{Out}(f), \ P_t(x) = 0\}.$$

Алгоритм построения замыкающей вычислительной схемы задачи (X, Y) представляется в следующей последовательности шагов:

ш.1—полагаем список $\tilde{M}$ пустым;

ш.2—ставится задача (X', Y), где $X' = X$;

ш.3—выбирается множество всех термов $M \subseteq T(X, F)$, для которых глубина равна 1;

ш.4—если $M = \varnothing$ и $\tilde{M} = \varnothing$, то переход на ш.12;

ш.5—если $M = \varnothing$, то переход на ш.8;

ш.6—полагаем $M = \tilde{M} \dot{\oplus} M$ (добавляются справа элементы множества M в произвольном порядке);

ш.7—полагаем $X' = X' \bigcup\limits_{t \in M} \mathrm{Out}(t)$, $F' = F' \setminus M$, переход на ш.3;

ш.8—списком $\tilde{M}$ порождается терм t_0;

ш.9—$Y \subseteq \mathrm{Out}(t_0)$, то переход на ш.12;

ш.10—если $R(t_0, 0) \neq \varnothing$, то задача (X, Y) разрешима и терм является ее вычислительной схемой; переход на ш. 13;

ш.11—$F = S^*(t_0) \setminus R(t_0, 0)$, переход на ш.1;

ш.12—задача (X, Y) не разрешима в (X, F);

ш.13—конец алгоритма.

Пусть в процессе функционирования алгоритма на ш.8 построена последовательность термов $t_0^1, \ldots t_0^p$, а на ш.11—последовательность множеств операций $F_1, \ldots, F_p$, где t_0^i—соответствует терму t_0 и F_i—множеству F $(F_1 = F)$, построенным при i-ом проходе алгоритма. Тогда имеет место следующая лемма: если $t^* \in \tilde{T}(X, Y, F)$—замыкающая вычислительная схема, то

$$t^* \prec t_0^i, \ i = \overline{1, p}.$$

Доказательство. На основании функционировании ш.2—ш.8 алгоритма вытекает, что t_0^l является термом с максимальной глубиной в множестве $T(X, F_l)$. В силу этого, вышеуказанное соотношение верно для $i=1$. Предположим, что оно верно для $i=l<p$. Т. к. $F_{l+1}=S^*(t_0)\setminus R(t_0^l, 0)$, то на основании замечания 1 и индуктивного предположения следует, что $t^*\in\widetilde{T}(X, Y, F_{l+1})$. Следовательно $t^*<t_0^{l+1}$.

Теорема. Вышеизложенный алгоритм построения замыкающей вычислительной схемы является корректным.

Доказательство. Если при функционировании алгоритма есть переход на ш.12 (это возможно, если выполняется условие ш.4), то $T(X, F_p)=\varnothing$ или $Y\not\subseteq \mathrm{Out}(t_0^p)$, что и означает не разрешимость задачи (X, Y). Если нет перехода на ш.12, тогда на основании утверждения предыдущей леммы t_0^p является замыкающей вычислительной схемой.

Алгоритм построения замыкающей вычислительной схемы задачи (X, Y) позволяет решать ряд других задач процесса планирования реализованного в базовой диалоговой системе автоматизации научных исследований АНИ-81. В данной статье рассматриваются следующие задачи: построение тупиковой вычислительной схемы задачи (X, Y); выявление переопределенной для не разрешимой задачи (X, Y);

Алгоритм построения тупиковой вычислительной схемы.

Алгоритм построения тупиковой вычислительной схемы представляется в следующей последовательности шагов:

ш.1—полагаем список $S_0=\varnothing$, $j=1$, $X'=X$, $F'=F$;

ш.2—функционирует алгоритм построения замыкающей вычислительной схемы t^* задачи (X', Y) в вычислительной модели (X, F);

ш.3—если t^* не существует, то переход на ш.7;

ш.4—полагаем $j=2$, $F'=S^*(t^*)\setminus \mathrm{begin}\, S(t^*)$, где $\mathrm{begin}\, S(t^*)$ означает 1-ый по порядку элемент списка $S(t)$;

ш.5—если множество $F'\neq\varnothing$, то переход на ш.2;

ш.6—список $S_0\dot{\oplus}\mathrm{begin}\, S(t^*)$ порождает тупиковую вычислительную схему задачи (X, Y) в вычислительной модели (X, F), переход на ш.9;

ш.7—если $j>1$, то $S_0=S_0\dot{\oplus}\mathrm{begin}\, S(t^*)$; $X'=X'\cup \mathrm{Out}(\mathrm{begin} S(t^*))$ переход на ш.2;

ш.8—задача (X, Y) не разрешима;

ш.9—конец алгоритма.

Алгоритм выявления переопределенной задачи.

Каждому $x_i\in X\setminus(X\cup Y)$ поставим в соответствие новую операцию φ_i определяемую следующим образом: $In(\varphi_i)=\bar{x}$, $\mathrm{Out}(\varphi_i)=x_i$, где $\bar{x}$ это некоторая фиктивная переменная, не принадлежащая X.

Алгоритм выявления переопределенной для не разрешимой задачи (X, Y) представляется в следующей последовательности шагов:

ш.1—полагаем множества $\Phi'=\Phi$, $\Phi''=\Phi$, $Y_1=Y$, $X''=X$;

ш.2—ставится задача (X_1, Y_1) в вычислительной модели $(X\cup\{x\}, \Phi')$ при ограничении $P_t(x)=1$, $\forall x\in \mathrm{Out}(t)$, $\forall t\in T(X, \Phi')$, где $X_1=X$;

ш.3—выбирается произвольный терм t из множества $(T(X_1, \Phi')$, для которого глубина равна 1;

ш.4—если таковой отсутствует, то переопределенная для задачи (X, Y) в вычислительной модели не существует, переход на ш.11;

ш.5—полагаем $X_1 = X_1 \cup \mathrm{Out}(t)$;

ш.6—если задача (X_1, Y_1) не разрешима в вычислительной модели (X, F), то переход на ш.3;

ш.7—полагаем $S_1 = S_1 \oplus \mathrm{Out}(t)$, $X'' = X'' \cup \mathrm{Out}(t)$;

ш.8—если (X'', Y_1) разрешима в вычислительной модели (X, F), то переход на ш.10;

ш.9—$\Phi' = \Phi'' \setminus \Phi'$, $\Phi'' = \Phi$, переход на ш.2;

ш.10—задача (X_1, Y) является переопределенной для задачи (X, Y) в вычислительной модели (X, F);

ш.11—конец алгоритма.

Для задачи (X, Y) переопределенная задача является неоднозначной. Задача нахождения минимальной по мощности множества X' переопределенной задачи (X', Y) является алгоритмически неразрешимой.

Используя вышеизложенные алгоритмы и результаты работы возможно описать алгоритмы для нахождения некоторого множества не эквивалентных друг другу вычислительных схем и некоторого множества переопределенных задач данной задачи (X, Y).

Է. Հ. ՂԱԶԱՐՅԱՆ, Ս. Ս. ՊՈՂՈՍՅԱՆ

ՀԱՇՎՈՂԱԿԱՆ ՍԽԵՄԱՆԵՐԻ ՊԼԱՆԱՎՈՐՈՒՄԸ՝ ՊՐԵԴԻԿԱՏԱՅԻՆ ՊԱՅՄԱՆՆԵՐԸ ՆԿԱՏԻ ԱՌՆԵԼՈՒ ԴԵՊՔՈՒՄ

Ամփոփում

Աշխատանքում դիտարկված են ԱՆԻ—81 երկխոսական գիտական հետազոտությունների ավտոմատացման համակարգում իրագործված հաշվարկային սխեմաների բացահայտման հիմնական ալգորիթմները։ Բերված են էլքափակող և փակուղային հաշվարկային սխեմաների կառուցման ալգորիթմները։ Ապացուցված է հետևյալ թեորեմը՝

Որպեսզի $t^* \in \tilde{T}(X, Y, F)$ *հաշվարկային սխեման լինի էլքափակող անհրաժեշտ է և բավարար, որ տեղի ունենա հետևյալ պայմանը՝*

$$\forall t \in \tilde{T}(X, Y, F) \rightarrow t \leqslant t^*.$$

Աշխատանքում նկարագրված ալգորիթմները հիմնված են այդ թեորեմի հետևանքի վրա։

ЛИТЕРАТУРА

1. *К. А. Абгарян.* Базовая система автоматизации научно-технических и опытно-конструкторских разработок.— В. кн.: Математические вопросы кибернетики и вычислительной техники, вып. XI, Ереван. 1982.
2. *Э. Х. Тыугу.* Решатель вычислительных задач.—Ж. вычисл. математики и мат. физики; 1971, т. II, № 14.
3. *Ю. А. Бухштаб, А. И. Горлин* и др. Об одном методе планирования расчетных схем.—Программирование, 1981, № 3.
4. *В. А. Вальковский.* О синтезе оптимальных программ на базе вычислительных моделей. Программирование, 1980, № 6.
5. *Э. У. Казарян, С. С. Погосян.* Планирование расчетных цепочек в системе АНИ-81. Изд-во АН АрмССР, Ереван, 1983.