ЛОГИЧЕСКИЕ МЕТОДЫ В ПРОГРАММИРОВАНИИ
(Вычислительные системы)
1987 год Выпуск 120

УДК 658.012.011

ЛОГИСТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ И ЕГО ПРИМЕНЕНИЕ ДЛЯ СИНТЕЗА ПРОГРАММ

А.А. Рось

В настоящее время становится очевидным, что решение многих задач систем реального времени невозможно без привлечения информации, для которой нецелесообразна количественная форма представления.

Характерные особенности большинства задач систем реального времени заключаются в следующем: цели функционирования сложных систем не всегда могут быть выражены количественными критериями; между объектами и явлениями не удается установить количественные зависимости; ряд отношений между состояниями объектов и явлениями, характерных для сложных динамических систем, например временные и причинно-следственные отношения, не поддаются описанию с помощью аналитических зависимостей; последовательность действий системы по достижении целей функционирования является логическим следствием содержания решаемых задач.

Указанные особенности обусловливают целесообразность использования формально-логических методов для формализации задач управления и обработки информации. При этом основу технологии разработки программного обеспечения (ПО) будет составлять синтез плана действий. В этом случае, аналогично [1], задача синтеза содержательно может быть сформулирована следующим образом. Представим процесс управления (обработки информации), реализуемый на ЭВМ как последовательность действий, характеризующихся условиями, результатами выполнения, объектом, субъектом и другими атрибутами. При этом под действием будем понимать отдельную операцию или группу операций, образующих программный модуль с одним входом и одним выходом.

В особую группу выделим программные модули с одним входом и двумя выходами, предназначенные для проверки условий и результатов выполнения действий. Имеющийся набор автономно отлаженных программных модулей интерпретирует содержание понятий теории среды. Известны поведение управляющей системы и цели ее функционирования, описанные некоторой аксиоматической моделью в терминах имен программных модулей. Необходимо на основе описания поведения управляющей системы из набора программных модулей синтезировать управляющую программу (программу обработки информации), реализующую достижение целевой установки.

Сформулированную задачу предлагается решать в два этапа: планирование действий системы управления по достижении целевой установки и построение проблемной программы. Необходимо отметить, что применение формально-логических методов для практического синтеза программ реального времени предъявляет к аппарату формализации следующие основные требования: в связи с тем, что задача планирования вычислительного процесса по управлению и обработке информации относится к классу оптимизационных, математический аппарат должен позволять оперировать как с количественными ограничениями на значения управляемых переменных, так и с ограничениями, выраженными понятийными категориями; формализм должен позволять описывать динамичность поведения систем реального времени; логический вывод должен осуществляться за приемлемые временные ресурсы.

В задаче требуется не просто найти доказательство, а в некотором смысле оптимальное доказательство. Математическая формулировка задачи в общем виде может быть записана следующим образом: минимизировать скалярную компоненту $\varphi(x)$ целевой функции $F(x)/\varphi(x)$ на множестве $x = \{x: q_i(x) \geq 0,\ i = 1,2,\ldots,n;\ P_k(x),\ k = 1,2,\ldots,l\}$, где $F(x)$ – формула, описывающая понятийными категориями; целевое состояние $\varphi(x)$, $q_i(x)$ – скалярные функции; $P_k(x)$ – формулы, описывающие ограничения содержательного характера.

С помощью понятийных категорий описываются закономерности функционирования моделируемых объектов, свойства и возможности как отдельных элементов объекта, так и объекта в целом, рекомендации по планированию действий, направленных на достижение целей функционирования, и др.

Из формулировки задачи видно, что она не является в чистом виде задачей математического программирования или задачей логического характера. Задачи по нахождению экстремумов функций на мно-

жествах, определяемых ограничениями, выраженными как количественными характеристиками, так и понятийными категориями (содержательно), будем называть задачами логистического программирования. Этот термин оправдан тем, что в основе формализации таких задач лежит разработка логистической системы.

Предлагаемый метод решения задачи логистического программирования основан на анализе перспективности ветвей дерева вывода с использованием количественных характеристик в процессе логического вывода, осуществляемого на основе информации семантического характера. При этом в качестве процедуры доказательства используется алгоритм вывода, основой которого является процедура MESON (см. [2]), расширенная правилами, учитывающими изменение состояний и условий функционирования динамических систем [3]. Данный алгоритм при построении И-ИЛИ дерева реализует метод перебора в глубину с отсечением неперспективных вариантов по методу ветвей и границ.

Нижняя граница значения функции выигрыша для совокупности ветвей $\Gamma_{\delta_1^k}$ может быть определена следующим образом:

$$\delta(\Gamma_{\delta_1^k}) = L(\delta_1^k) + \min_{\{\delta_{1,i}^{n-k}\}} L(\delta_{1,i}^{n-k}) .$$

Здесь $L(\delta_1^k)$ и $L(\delta_{1,i}^{n-k})$ – длины маршрутов вдоль участков ветвей, включающих первых k и $(n-k)$ оставшихся вершин соответственно.

Основная сложность использования метода ветвей и границ в процессе логического вывода заключается в определении оптимистической оценки. Это обусловлено тем, что ее приходится определять не при наличии полного набора вариантов достижения целевой установки, а в процессе их формирования.

Как правило, сложные системы управления решают ряд взаимосвязанных задач, представляющих сложную иерархическую структуру. Следовательно, и целевая функция для всей системы в общем случае должна быть комплексной и отражать соответствующую иерархию подцелей. Это предъявляет определенные требования к структуре аксиоматической теории среды.

Структура аксиоматической теории среды. Особенностью рассматриваемой аксиоматической теории среды является ее многоуровневость, обусловленная необходимостью выделения в особые классы

утверждений, описывающих факты действительности (сложившуюся ситуацию) и различные суждения об этих фактах. Аксиоматическая теория среды, отражающая динамичность условий функционирования систем реального времени, наряду с удовлетворением общим требованиям полноты, непротиворечивости и разрешимости, должна обеспечивать возможность разбиения рассматриваемой предметной области, ввиду ее многосортности, на ряд непересекающихся собственных классов понятий, задания первичных отношений между индивидами рассматриваемой предметной области, определения способа (программы) для установления истинности каждого первичного отношения.

Задание первичных отношений между индивидами предметной области требует прежде всего понятия первичного отношения. При этом первичность отношения должна определяться возможностью его "понимания" ЭВМ. Назовем первичным (вычислимым) предикат $P(x_1, x_2, \ldots, x_n)$ от n предметных переменных, если в памяти управляющей ЭВМ содержится соответствующая программа, выполнение которой обеспечивает нахождение его значения истинности. Проверка истинности (означивание) первичного предиката возможна всякий раз при подстановке предметных констант вместо предметных переменных. Означивание первичных предикатов может производиться программным путем по данным, поступающим от источников информации о среде, или другой информации, содержащейся в базе данных.

Формулы аксиоматической теории среды, составленные с помощью логических связок $\&, \vee, \rightarrow$ и кванторов $\forall, \exists$ из первичных предикатов, назовем первичными. Их особенностью является то, что при относительно небольшом конечном числе индивидов предметной области они могут быть означены программным путем. Совокупность первичных формул образует теорию первого уровня T_1. Последующее расширение T_1 осуществляется введением дополнительных понятий и отношений посредством определений, задания аксиом, выраженных в терминах вновь введенных понятий. Получаемая таким образом функционально полная аксиоматическая теория среды может быть представлена, например, следующими уровнями: T_1 – теория, включающая множество первичных предложений Γ_1, которые описывают факты действительности; T_2 – теория, построенная из множества Γ_2^1, которое описывает возможные состояния объектов внешней среды, их возможные действия и результаты; T_3 – теория, построенная из множества Γ_2^2, которое описывает возможное развитие процессов управления и т.д. При этом $\Gamma_1 \subset \Gamma_2^1 \subset \Gamma_2^2 \subset \ldots \subset \Gamma_2$.

Все множество допустимых вопросов Σ, требующих своего разрешения в процессе управления подчиненными объектами, можно разбить на два класса: Σ_1 - вопросы, разрешимые на множестве Γ_1; Σ_2 - вопросы, выраженные формулами, значения истинности которых не зависят от истинности формул первичной теории T_1. Ответ на вопросы первого класса определяется сложившейся в данный момент ситуацией.

Истинность формул, составляющих ответы на вопросы второго класса, определяется непосредственно теорией среды и от сложившейся ситуации не зависит. С учетом этого требование функциональной полноты аксиоматической теории среды выполнено, если для любой формулы $S_1 \in \Sigma_1$ имеет место $\Gamma_1 \vdash S_1$ или $\Gamma_1 \vdash \sim S_1$, а для любой формулы $S_2 \in \Sigma_2$ имеет место $\Gamma_2 \vdash S_2$ или $\Gamma_2 \vdash \sim S_2$.

Такой подход позволяет ограничить процесс вывода ответа на вопрос выделением формулы, составленной из первичных предикатов, истинность которой определяется путем означивания.

Рассмотренная структура аксиоматической теории среды позволяет осуществлять декомпозицию общей задачи, решаемой системой реального времени, на ряд подзадач путем построения в процессе их формализации многоуровневой структуры соответствующих иерархически взаимосвязанных аксиоматических моделей. При этом содержание имени подзадачи, используемое на i-м уровне иерархии, описывается аксиоматической моделью, расположенной на (i+1)-м уровне. Этим обеспечиваются взаимосвязь частных целевых установок с общей целью функционирования системы реального времени и реализация принципа нисходящего проектирования ПО. При этом каждая аксиоматическая модель должна строиться в соответствии с изложенными принципами построения аксиоматической теории среды.

Рассмотрим процедуру доказательства, являющуюся основой рассматриваемого метода решения задачи логистического программирования.

Процедура доказательства. Анализ процесса управления показывает, что для формализации содержания задач, решаемых системами реального времени, целесообразно использовать множество предикатов, которые можно классифицировать следующим образом: первичные предикаты, описывающие результаты и условия выполнения действий, контролируемые отношения (значения истинности которых путем определенных воздействий можно изменить) и исходные условия функционирования системы управления; предикаты, содержание которых опи-

сано программным модулем с одним входом и одним выходом алгоритмически в терминах имен программных модулей, аксиоматическими моделями (имена аксиоматических моделей); предикаты, содержание которых описано аксиоматически, а также предикаты, введенные по определению в виде правильно построенных формул (вторичные предикаты).

В особую группу следует выделить предикаты, введенные для организации доказательства (системные предикаты). Содержание первичных предикатов представлено в ЭВМ программными модулями с одним входом и двумя выходами.

Приведенная классификация обеспечивает возможность декомпозиции задач управления и позволяет разработать многоуровневую иерархическую структуру взаимосвязанных моделей. Все множество предикатов разобьем на подмножества: S_P - подмножество, включающее первичные предикаты; S_H - подмножество, включающее все предикаты, кроме первичных, которые описывают исходные условия функционирования системы управления.

В основу рассматриваемой процедуры доказательства положены правила вывода процедуры MESON [2], которые модифицированы и расширены в связи с необходимостью оперирования в процессе вывода условной формой записи утверждений теории проблемной среды. Последняя обусловлена характером задачи и обработки информации. Поэтому здесь приводится описание тех механизмов, которые являются дополнением процедуры MESON.

Схематически процесс доказательства можно представить древовидным графом, у которого есть три типа вершин: начальная, называемая корнем дерева (этой вершине соответствует формула, подлежащая доказательству); конечные, называемые листьями; промежуточные, составляющие крону дерева. Графически дерево изображается корнем вверх, причем в каждую вершину, кроме корневой, сверху вниз входит одна линия, соединяющая ее с ближайшим предком, из каждой вершины, кроме листьев, выходит одна или несколько линий, соединяющих ее с ближайшими потомками.

Будем говорить, что вершины находятся на одном уровне дерева доказательства, если у них одинаковое число предков. Корневая вершина находится на первом уровне. Линии, соединяющие вершины разных уровней, можно интерпретировать логической связкой-имплика - цией, имеющей направление от потомка к предку. ..... ли, соединяю -

щие вершины одного уровня, можно трактовать как конъюнкции сопоставляемых им литералов, а сами вершины будем называть партнерами. Несколько линий, выходящих из одной вершины, интерпретируются как дизъюнкция соответствующих конъюнкций. Линии - импликации - позволяют найти путь от листьев к корню и построить процесс доказательства в виде цепочки импликаций, напоминающей выводимую формулу теоремы дедукции.

Пусть мы на i-м уровне и нам нужно установить доказуемость литерала l_k. По списку предложений с учетом унификации аргументов отыскивается вхождение l_k в какое-либо предложение, которое в случае удачи преобразуется к виду:

$$L_1 \& \ldots \& L_{k-1} \& L_{k+1} \& \ldots \& L_N \rightarrow l_k, \qquad (1)$$

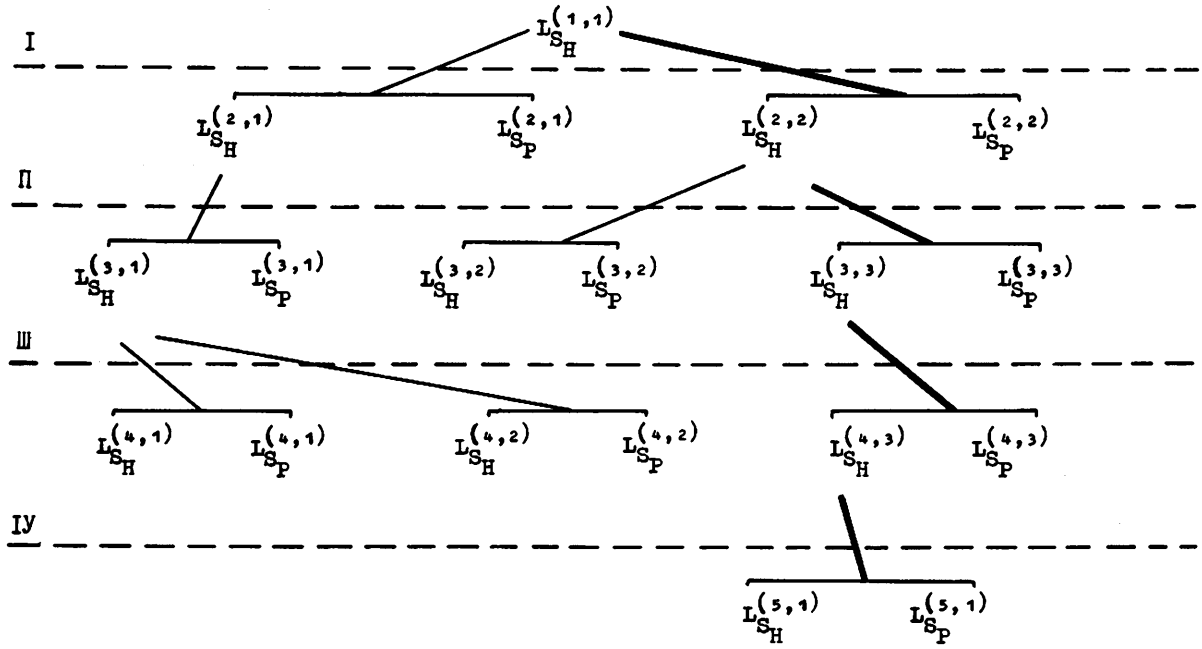
где $L_1, \ldots L_N$ - дополнения литералов $l_1, \ldots, l_N$ соответственно. В силу исходной классификации предикатов выражение (I) можно переписать в виде:

$$L_{S_H} \& L_{S_P} \rightarrow l_k, \qquad (2)$$

где L_{S_H}, L_{S_P} - конъюнкции литералов, образованных из предикатов, которые принадлежат соответственно подмножествам S_H и S_P. Формулы L_{S_H} и L_{S_P} являются ближайшими потомками литерала l_k и образуют (i+1) -й уровень дерева доказательства. Значение истинности формулы L_{S_P} установить несложно путем означивания составляющих ее первичных предикатов. Выводимость формулы L_{S_H} требуется определить в процессе дальнейшего доказательства. В целом процесс доказательства можно изобразить схематически, как показано на рисунке. Перепишем формулу (2) следующим образом:

$$\underbrace{L_{S_H}}_{\text{ядро}} / \underbrace{L_{S_P}}_{\text{условие}} \rightarrow l_k. \qquad (3)$$

В соответствии с вложенным в выражение (3) смыслом будем его читать так: "литерал l_k доказуем в случае доказуемости формулы-ядра L_{S_H} при наличии условий, описываемых формулой L_{S_P}". В дальнейшем запись формулы в виде (3) будем называть условно-истинностной формой [4]. С учетом динамичности функционирования систем реального времени условно-истинностную форму в общем виде можно записать следующим образом:



Схематическое изображение процесса доказательства.

$$L_{S_H}^{(N,j_N)} / L_{S_P}^{(N,j_N)} \to (\ldots (L_{S_H}^{(i,j_i)} / L_{S_P}^{(i,j_i)} \to$$

$$\to \left[\ldots (L_{S_H}^{(2,j_2)} / L_{S_P}^{(2,j_2)} \to L_{S_H}^{(1,j_1)}) \ldots \right]) \ldots), \quad (4)$$

где $L_{S_H}^{(i,j_i)}$, $L_{S_P}^{(i,j_i)}$ - формулы, расположение которых на дереве доказательства определяется i-м уровнем и j_i-м местонахождением на этом уровне. Заметим, что правило перестановки посылок к выражению (4) применять нельзя, так как дерево доказательства определяет последовательность действий во времени с учетом динамичности условий функционирования системы управления. С учетом этого замечания в выражение (4) вкладывается следующее содержание: "формула, заключенная в квадратные скобки, доказуема, если существуют условия ее доказуемости, определяемые формулой $L_{S_P}^{(i,j_i)}$, и к моменту их выполнения истинна формула $L_{S_H}^{(i,j_i)}$". Понятие условно-истинностной формы, обусловленное динамичностью характеристик проблемной среды, является фундаментальным в описываемой процедуре доказательства и определяет систему правил установления выводимости формул, которые можно разбить на две группы: правила окончания вывода и правила разметки вершин дерева доказательства.

Сформулируем критерии окончания вывода: истинность анализируемого литерала задана аксиоматически; литерал образован из первичного предиката, описывающего исходные условия функционирования системы управления; литерал является дополнением литерала, являющегося предком данного литерала; литерал уже встречался в качестве предка; нет предложений, которые можно использовать для продолжения вывода данного литерала.

С целью определения выводимости литерала каждой вершине дерева ставится в соответствие одна из следующих меток: "выводима", "условно выводима", "допустима", "условно допустима", "недопустима", "логически не определена", "противоречива". Правила присвоения меток вершинам дерева можно разбить на три группы: правила разметки листьев дерева, правила сопоставления меток конъюнкциям литералов и правила сопоставления меток вершинам на основе анализа меток вершин-преемников (дизъюнкции формул).

Сопоставление меток листьям дерева осуществляется по следующим правилам: вершина, образованная литералом, истинность которого задана аксиоматически, сопровождается меткой "выводима"; вершине, образованной литералом, являющимся дополнением одного из литералов-предков, ставится в соответствие метка "допустима"; вершина, образованная из литерала, который встречался в качестве литерала-предка, сопровождается меткой "не допустима"; вершине, образованной из первичного предиката, ставится в соответствие метка "условно выводима"; вершина, образованная из литерала, дальнейший вывод которого невозможен из-за отсутствия предложений, включающих этот литерал, метится "логически не определена"; вершина, образованная литералом, помеченным как "выводим", причем ранее установлено, что его дополнение также "выводимо", сопровождается меткой "противоречива".

Следующие две группы правил представлены в табл. I и 2. Символ "+" указывает на обязательное присутствие хотя бы одной вершины с данной меткой, символ "-" - на обязательное отсутствие вершин с данной меткой, а символ "х" - на то, что безразлично, будут ли среди группы анализируемых вершин вершины с соответствующей меткой. Кроме перечисленных правил, введем еще одно, которое трудно включить в таблицу: если вершина сопровождается меткой "допустима" или "условно допустима", то, если на дереве вывода она не расположена так, что один из ее потомков или она сама является дополнением одного из ее предков, она переименовывается на "выводима" или "условно выводима" соответственно. Это правило применяется всякий раз перед этапом сопоставления метки конъюнкции литералов.

Дерево, построенное в процессе доказательства и представляющее собой план действий, содержит всю информацию об их последовательности, направленной на достижение целевой установки. Используемая классификация предикатов обеспечивает однозначность при извлечении из дерева доказательства проблемного алгоритма. Оценка целесообразности продолжения вывода на основе количественного анализа осуществляется всякий раз с привлечением предиката, который сопровождается значением затрачиваемых ресурсов на реализацию соответствующего действия.

Отладка синтезируемых алгоритмов. При описываемом подходе синтезируемый алгоритм однозначно соответствует описанию задач систем реального времени, а следовательно, любые изменения модели проблемной среды отразятся на его структуре. Очевидно, что алгоритм управления представляет некоторую интерпретацию аксиоматиче-

Таблица I

Правила сопоставления меток конъюнкции литералов

Выводима	Условно выводима	Допустима	Условно допустима	Логически не определена	Не допустима	Заключение
x	x	x	x	x	+	Не допустима
x	x	x	x	+	–	Логически не определена
x	x	x	+	–	–	Условно допустима
x	+	+	–	–	–	Условно допустима
x	–	+	–	–	–	Допустима
x	+	–	–	–	–	Условно выводима
+	–	–	–	–	–	Выводима

Таблица 2

Правила сопоставления меток родительскими вершинами (дизъюнкции литералов)

Выводима	Условно выводима	Допустима	Условно допустима	Логически не определена	Не допустима	Заключение
+	x	x	x	x	x	Выводима
–	+	x	x	x	x	Условно выводима
–	–	+	x	x	x	Допустима
–	–	–	+	x	x	Условно допустима
–	–	–	–	+	x	Логически не определена
–	–	–	–	–	+	Не допустима

ской теории. Возможность его построения определяется наличием таких свойств теории, как непротиворечивость и функциональная полнота, а независимость аксиом гарантирует отсутствие дублирующих ветвей в синтезируемом алгоритме. Таким образом, отладка алгоритма сводится к исследованию аксиоматической модели на непротиворечивость, функциональную полноту и независимость.

Рассмотрим множество предложений $S^C = \{C_1, C_2, \ldots, C_q\}$. Каждому предложению $C_i \in S^C$ сопоставлено соответствующее предложение $B_i \in S$ в условно-истинностной форме, причем формулы-условия L_{P_i} предполагаются тождественно истинными. Иначе говоря, каждое C_i есть формула-ядро соответствующего предложения B_i. Рассмотрим также некоторую формулу G, выводимую из S^C. На основе теоремы дедукции, правил отделения и силлогизма, доказывается следующая

ТЕОРЕМА. Если формула G выводима из S^C, то формула

$$(L_{P_{\xi_1}} \& L_{P_{\xi_2}} \& \ldots \& L_{P_{\xi_m}}) \to G \qquad (5)$$

выводима из множества предложений S.

В выражении (5) $L_{P_{\xi_1}} \& \ldots \& L_{P_{\xi_m}}$ есть конъюнкция формул-условий, соответствующих предложениям $C_{\xi_1}, C_{\xi_2}, \ldots, C_{\xi_m}$, которые входят в вывод G из S^C. Посылку импликации (5) назовем общим условием выводимости формулы G из S и обозначим его через L_{P^0}, а саму формулу G будем называть условно выводимой. Выражение (5) запишем следующим образом: G/L_{P^0}, а в запись $S \vdash G/L_{P^0}$ будем вкладывать следующий смысл: формула G выводима из множества предложений S, если общее условие выводимости L_{P^0} имеет значение "истина". В общем условии вводимости G из S для систем, изменяющих свое состояние во времени, существенное значение имеет порядок записи условия частей формул, входящих в вывод. Предположим, что в формуле L_{P^0} составляющие ее формулы-условия упорядочены слева направо по времени. Введем следующие операции для объединения формул-условий, отражающих изменение состояния среды во времени.

Пусть P_1 и P_2 - формулы-условия. Тогда

I. $P_1 \uparrow P_2$ - операция одновременного выполнения условий. Формула $P_1 \uparrow P_2$ принимает значение "истина", если формула P_1 истинна в момент времени t и формула P_2 истинна в тот же момент времени.

2. $P_1 \rightarrow P_2$ - операция строгого следования условий. Формула $P_1 \rightarrow P_2$ принимает значение "истина", если формула P_1 истинна в момент времени t_1, формула P_2 истинна в момент времени $t_2 > t_1$.

3. $P_1 \rightharpoonup P_2$ - операция нестрогого следования условий. Формула $P_1 \rightharpoonup P_2$ принимает значение "истина", если истинна одна из формул:

$$P_1 \rightharpoonup P_2 \text{ при } t_2 > t_1 ;$$

$$P_2 \rightharpoonup P_1 \text{ при } t_2 < t_1 ;$$

$$P_1 \uparrow P_2 \text{ при } t = t_1 = t_2 .$$

Основные свойства систем аксиом динамических теорий, такие как непротиворечивость, функциональная полнота и независимость, определяются с использованием понятия условной выводимости.

Пусть формулы A и $\sim$A условно выводимы из систем аксиом S. Формуле A соответствует общее условие выводимости $L_{P_1^0}$, а формуле $\sim$A соответствует $L_{P_2^0}$. Система аксиом S в этом случае считается противоречивой, если формулы $L_{P_1^0}$ и $L_{P_2^0}$ совместимы. Эта совместимость означает, что существуют такие наборы значений переменных, при которых $L_{P_1^0}$ и $L_{P_2^0}$ одновременно принимают значение "истина". Следовательно, доказательство противоречивости системы аксиом заключается в следующем: устанавливается выводимость некоторой формулы A и ее отрицания $\sim$A в предположении, что условные части формул, входящие в вывод A и $\sim$A из S, истинны; проверяется совместимость соответствующих формул-условий $L_{P_1^0}$ и $L_{P_2^0}$ на всех наборах значений предметных переменных с учетом последовательности их выполнения во времени. В случае, когда все возможные для данной теории формулы рассмотрены, а противоречия не обнаружено, система аксиом считается непротиворечивой.

Аналогично проводится проверка независимости системы аксиом. Пусть $B_j \in S$ условно выводима в системе S^j, полученной после удаления B_j из списка аксиом. Аксиому B_j можно считать зависимой,

если условная часть формулы B_j истинна всякий раз, когда истинна формула-условие $L_{P_j^0}$, полученная в результате вывода B_j из S^j.

Изложенные принципы положены в основу разработанных методов и алгоритмов отладки аксиоматических моделей. Использование этих методов позволяет выявить и устранить большинство ошибок алгоритмического характера, допущенных на этапе формализации задач, решаемых системами реального времени.

Заключение. Описанный метод решения задачи логистического программирования и принципы синтеза программ в полной мере реализованы в диалоговой автоматизированной информационно-логической системе ДАИЛОС [4]. Входной язык ПЛАТАН-ДІ располагает средствами описания иерархической структуры многосортной предметной области. Логической основой языка является многосортное исчисление предикатов первого порядка, расширенное операторами модальной и временной логики [5]. Информация для количественной оценки перспективности ветвей дерева вывода содержится в спецификациях программных модулей, отражающих среднее время выполнения и их объем.

Производство ПО при использовании ДАИЛОС можно представить следующими основными этапами: формализацией и согласованием формализованного технического задания на разработку ПО между заказчиком и разработчиком; разработкой логической структуры базы данных; разработкой системы понятий теории управления; разработкой и отладкой аксиоматических моделей управляемых объектов и проблемной среды; разработкой спецификаций на программные модули, интерпретирующие базовые понятия теории управления; проверкой соответствия управляющих алгоритмов формализованному техническому заданию; разработкой и отладкой программных модулей; синтезом управляющих программ из программных модулей.

Такая технология разработки ПО позволяет на основных этапах технологического цикла использовать одну и ту же модель процесса управления, а для ее анализа и обработки привлекать один и тот же инструмент - ДАИЛОС, при этом выполняется требование единства системы автоматизации программирования и отладки всего технологического цикла разработки программного обеспечения систем реального времени. Кроме этого, процесс формализации и структура аксиоматической теории среды согласованы с современными принципами нисходящего проектирования и модульного программирования.

Используемое при выводе понятие условно-истинностной формы позволяет автоматически сгенерировать те условия функционирования, на которые спроектирован синтезируемый алгоритм. Это позволяет автоматизировать контроль входной информации. Особое значение приобретает исследование аксиоматических моделей на независимость,что эквивалентно оптимизации логической структуры программ.

Литература

I. НЕПЕЙВОДА Н.Н. Об одном методе построения правильной программы из правильных подпрограмм //Программирование. - 1979. - №1. - С. 15-25.

2. LOVELAND D. Automated Theorem Proving: A Logical Basis. - New York,1978.- 405 p.

3. РОСЬ А.А. Принципы синтеза программ для систем реального времени на основе формально-логического подхода к описанию проблемной среды //Кибернетика. - 1985. - №5. - С. 31-36.

4. РОСЬ А.А., БОГДАНОВ Ю.Г., КОСИНОВ А.Н., СНОПКОВ М.В., ВОЛОШИНА Т.П. Диалоговая автоматизированная информационно-логическая система //Управляющие системы и машины. - 1985.-№4. - С. 97-103.

5. ЯРУШЕК В.Е. О формализованной модели для планирования действий управляемых объектов в динамической среде //Проблемы бионики. - 1982. - Вып. 29. - С. 88-95.

Поступила в ред.-изд.отд.
31 декабря 1985 года