

УДК 007:519.816

РЕАЛИЗАЦИЯ ЛОГИКИ ВЕТВЯЩЕГОСЯ ВРЕМЕНИ

И.Е. Куриленко (*ivan@appmat.ru*)
Московский Энергетический Институт (Технический
Университет), Москва

Рассматривается современный подход к моделированию темпоральных (временных) рассуждений в интеллектуальных системах. Предлагаются алгоритмы вывода для ветвящейся временной логики, построенной на основе качественной точечной модели времени. Разработанные алгоритмы построены с учетом специфики применения в составе интеллектуальных систем поддержки принятия решений реального времени (ИСППР РВ).

Введение

Современные исследования и разработки по созданию перспективных интеллектуальных (экспертных) систем поддержки принятия решений (ИСППР), в частности, ИСППР реального времени (РВ) [Вагин и др., 2001] опираются на разработку и применение интегрированных моделей и неклассических логик. Так как многие базовые понятия, такие как «изменение», «причина», «следствие» и отношения между ними описываются в терминах времени, ведутся активные исследования по построению интегрированных моделей рассуждений учитывающих временной фактор при описании проблемной ситуации и в процессе поиска решения [Поспелов и др., 1986]. ИС типа ИСППР РВ в процессе своего функционирования вынуждены оперировать с большим количеством информации, изменяющейся со временем (это могут быть показания датчиков, значения управляющих параметров, выполняемые операторами действия и т.д.) [Башлыков и др., 1994]. При этом поиск решения таких задач ИСППР как диагностика, мониторинг, планирование, прогнозирование и др. должен обеспечиваться в условиях достаточно жестких временных ограничений (определяемых реальным управляемым процессом) и различного вида неопределенностей в исходных данных и знаниях [Еремеев и др., 2003]. Таким образом, временные логики и временной вывод могут применяться во многих блоках ИСППР РВ (анализаторе, блоках поиска решения, объяснения, моделирования и прогнозирования) и позволяют повысить скорость отклика системы, что

увеличивает эффективность принятия решений ЛПР в различных проблемных (аномальных) ситуациях.

Для ИСППР РВ требуются модели (логики) ветвящегося времени, которые полезны при прогнозировании последствий принимаемых решений или развития некоторой ситуации в условиях дефицита времени. К сожалению, алгоритмы вывода для известных темпоральных логик [Torsun, 1998] ветвящегося времени обладают большой сложностью. В данной работе основное внимание уделяется вопросу создания темпоральной логики ветвящегося времени, применимой в составе современных ИС.

1. Основные определения и алгоритмы

На данный момент известно большое количество разнообразных временных логик [Pnuelli, 1977, Смирнов, 1979, Allen, 1983, Gereviny et al., 1993]. Однако на практике препятствием к их широкому внедрению является высокая сложность алгоритмов вывода. В этом плане выделяются логики, построенные на основе представления информации о времени как ограничений (зависимостей) между временными примитивами. В качестве таких примитивов могут использоваться временные моменты, интервалы или их комбинация. Зависимости между временными примитивами трактуются как ограничения на их расположение во времени. Благодаря существованию полиномиальных алгоритмов вывода, а также тому, что в них представимы сложные типы временных зависимостей, они являются предметом активных исследований. Среди них наиболее простой и приемлемой по вычислительной сложности для применения в составе современных ИС является точечная временная логика [Еремеев и др., 2009]. Реализация алгоритмов вывода в точечной временной логике основывается на переходе к задаче согласования временных ограничений (ЗСВО). ЗСВО является конкретизацией более общей задачи согласования ограничений (ЗСО), что позволяет использовать для решения ЗСВО методы, применяемые для ЗСО [Еремеев и др., 2003]. ЗСВО задается как $Z = (V, D, BTR, C)$ [Еремеев и др., 2005]:

- 1) $V = \{V_1, V_2, \dots, V_m\}$ – конечное множество временных переменных (интерпретируемых как моменты времени);
- 2) D – область значения переменных (множество целых чисел);
- 3) $BTR = \{r_1, r_2, \dots, r_n\}$ – конечное множество взаимоисключающих бинарных базовых временных ограничений, полное объединение которых является универсальным ограничением U (не накладывающим каких-либо ограничений);
- 4) $C = \{C_{ij} / C_{ij} = \{r_1, \dots, r_k\}; k > 0; r_1, \dots, r_k \in BTR; i, j \leq m\}$, конечное множество ограничений, где C_{ij} – ограничение над временными переменными V_i и V_j , интерпретируемое как $(V_i, r_1, V_j) \vee \dots \vee (V_i, r_k, V_j)$. Если C_{ij} состоит

только из одного дизъюнкта, то оно называется *точным*.

Для решения задачи выполнимости **SAT** необходимо найти множество не противоречащих друг другу ограничений $C^* = \{C_{ij}^* | C_{ij}^* = \{r_l\}, r_l \in C_{ij}\}$. Если такого множества построить нельзя, то ЗСВО является *несогласованной*. Если ЗСВО имеет по крайней мере одно решение, то она называется *согласованной*.

Основными операциями над временными ограничениями являются:

- 1) отрицание (\neg): $\neg L_{ij} = U \setminus L_{ij}$;
- 2) инвертирование (\sim): $\sim(r_1, \dots, r_k) = (\sim r_1, \dots, \sim r_k)$;
- 3) пересечение: $S \cap T = \{r/r \in S, r \in T\}$;
- 4) композиция: $T \bullet S = \{t_1, \dots, t_k\} \bullet \{s_1, \dots, s_q\} = \{t_1 \bullet s_1, t_1 \bullet s_2, \dots, t_k \bullet s_q\}$.

Известно [Gereviny et al., 1993], что множество всех возможных типов временных ограничений для двух временных примитивов, состоит из 2^{BTR} типов ограничений, замкнуто относительно операций \neg , \sim , \cap , \bullet и образует *алгебру* временных ограничений.

ЗСВО называют *единичной* ЗСВО (ЕЗСВО), тогда и только тогда, когда в множество C входят только точные ограничения. При этом сама ЗСВО сводится к проверке непротиворечивости (согласованности) ограничений из множества C . Задачу определения ограничения r , справедливого для переменных V_i и V_j , для которых задано ограничение $C_{ij} = \{r_1, \dots, r_k\}$, при $k > 1$, называют *задачей вычисления неточного ограничения* C_{ij} .

Ограничение C_{ij} *выполнимо* для переменных V_i и V_j тогда и только тогда, когда существует хотя бы одно решение ЗСВО, в котором C_{ij} является ограничением для этих переменных. *Минимальным ограничением* C_{ij}^{min} называется множество, состоящее только из выполнимых ограничений для V_i и V_j . ЗСВО называют *минимальной*, если все ее ограничения минимальны. Задачу вычисления минимальной ЗСВО называют *задачей поиска минимального представления* **MIN**. Известно, что для любой ЗСВО всегда можно найти эквивалентную минимальную или показать несогласованность ограничений.

Построим на базе точечной временной логики логику ветвящегося времени, применимую в составе современных интеллектуальных систем [Куриленко, 2004, Куриленко, 2009]. Сначала определим на базе ЕЗСВО сценарий как $S_i = (V_i, C_i, S_j)$, где $S_0 = (V_0, D, BTR, C_0)$ – ЕЗСВО, интерпретируемая как начальный сценарий; S_i – наследуемый сценарий, который расширяет множество переменных V_j и множество единичных ограничений C_j сценария S_j , $j < i$, множествами V_i и C_i соответственно; D – область определения переменных (множество целых чисел), BTR – множество базовых временных ограничений. Таким образом, сценарий является композитной ЕЗСВО, которую можно построить по иерархии наследования, задаваемой в определении сценария. Для каждого сценария, исходя из определения ЕЗСВО, можно поставить задачи поиска

минимального представления и проверки согласованности, т.е. можно говорить о согласованном или несогласованном сценарии и сценарии в минимальном представлении. Будем называть текущими сценариями такие сценарии, для которых не существует наследуемых сценариев.

Определим ветвящуюся ЗСВО как множество альтернативных сценариев, унаследованных от одного начального сценария $S_0 \in VZ = \{S : S \text{ – сценарий}\}$. Определим для ветвящейся ЗСВО следующие подзадачи:

- Проверка согласованности – проверка существования как минимум одного текущего согласованного сценария $S_i \in VZ$.
- Проверка истинности каких-либо утверждений для конкретного текущего сценария.
- Преобразование всех ЕЗСВО, соответствующих согласованным сценариям, к минимальному виду.
- Проверка истинности каких-либо утверждений для всех текущих сценариев или хотя бы для одного текущего сценария.

Рассмотрим пример (рис. 1). Данная задача содержит 16 сценариев, из которых 5 являются несогласованными. Следует отметить, что, исходя из практических соображений, следует допускать наследование только согласованных сценариев. Поэтому сценарии S_{26} и S_{27} не участвовали в порождении сценариев-наследников.

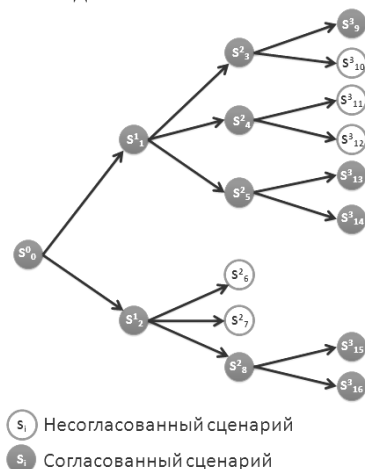


Рис. 1. Ветвящаяся ЗСВО

Таким образом, каждый сценарий соответствует возможному варианту развития событий (задаваемому соответствующим множеством ограничений над моментами времени).

Существуют разнообразные стратегии формирования сценариев в ветвящейся ЗСВО. В качестве источника ветвления на каждом шаге являются дизъюнктивные утверждения. Не дизъюнктивные утверждения

добавляются к каждому текущему сценарию и не приводят к ветвлению. В случае же если вносятся дизъюнктивные утверждения, то к каждому сценарию добавляется ряд наследников. Например, при добавлении в ветвящуюся ЗСВО ограничения типа $C_{ij} \vee C_{xy}$ для каждого текущего сценария должны быть добавлены три сценария – сценарий, в котором обязательно наличие ограничения C_{ij} , сценарий, в котором обязательно наличие C_{xy} и сценарий в котором обязательно наличие как C_{ij} так и C_{xy} . Естественно, что при такой постановке ветвление является существенной проблемой, т.к. ведет к сильному росту числа возможных сценариев (особенно при большом числе дизъюнктов). Однако существует ряд приемов, которые могут быть использованы для ограничения ветвления. Мы рассмотрим их немного позже. На практике может быть предусмотрен вариант внесения ограничений, связанных исключаящим или, т.е. в рассмотренном примере нет необходимости порождения сценария, в котором обязательно наличие комбинации C_{ij} и C_{xy} . Другой возможной стратегией является независимая достройка сценариев (когда каждый сценарий достраивается и разветвляется так, как это необходимо исходя из практической задачи). В этом случае внесение дизъюнктивного ограничения приводит к разветвлению конкретного сценария и может не затронуть все оставшиеся сценарии.

Алгоритмы решения задач ветвящейся ЗСВО могут быть построены на базе алгоритмов решения ЕЗСВО [Gereviny et al., 1993, Еремеев и др., 2005, Куриленко, 2007, Куриленко, 2008]. В частности каждую ЕЗСВО в процессе решения ветвящейся ЗСВО можно решать с помощью перехода к задаче на графе времени TL-графе (Подробно подход к решению ЕЗСВО на основе перехода к задаче на графе рассмотрен в работах [Gereviny et al., 1993, Еремеев и др., 2005]). Однако подобный подход не является оптимальным, т.к. на практике придется хранить в памяти TL-граф и Time-граф для каждого сценария. С учетом возможного роста числа сценариев хотелось бы построить такое внутреннее представление задачи, которое позволяло бы экономить ресурсы за счет учета наследования сценариев, заложенного в само определение ветвящейся ЗСВО. Также при построении алгоритмов следует учесть, что создание новых сценариев на каждом шаге должно выполняться как можно быстрее, т.к. в некоторых приложениях, типа ИСППР РВ, глубина анализа (т.е. рассмотрение как можно большего числа сценариев) способствует получению более качественного решения. В работах [Куриленко 2008, 2009] предлагаются алгоритмы позволяющие экономить вычислительные ресурсы при последовательном дополнении ЕЗСВО ограничениями, за счет анализа производимых изменений. В случае ветвящейся ЗСВО такие алгоритмы отлично подходят для решения ЕЗСВО для каждого сценария-наследника $S_i=(V_i, C_i, S_j)$, так как они определяются как расширение множества

ограничений в сценарии-предке. Коротко рассмотрим этот подход. Для представления ЕЗСВО используется ТЛ граф, определяемый как $G=(W,E,L)$, где W – непустое множество вершин (соответствующее моментам времени), E – множество направленных связей вида (w_i, l, w_j) , где $w_i, w_j \in W, l \in L, L = \{<, >, \leq, \geq\}$ [Куриленко, 2008]. Путем $\pi_{xy} = \langle x_0, x_1, \dots, x_k \rangle$ из вершины x в вершину y в графе G называется последовательность вершин, в которой $x_0 = x, x_k = y$ и $(x_i, x_{i+1}) \in E$ для всех $0 \leq i < k$. В качестве веса пути примем $w(\pi_{xy}) = \prod_{i=0}^{k-1} w_{x_i x_{i+1}}$, где $w_{x_i x_{i+1}}$ – пометка ребра (x_i, x_{i+1}) . В случае если при вычислении веса w_{xy} в графе G существует направленная связь (y, l, x) , то в качестве w_{xy} принимается $\sim l$. Будем считать путь π_{xy} *существенным*, если $w(\pi_{xy}) \neq \cup$ и $w(\pi_{xy}) \neq \emptyset$.

Определим множество P всех направленных путей между всеми вершинами в графе G . Определим: P' – множество существенных путей в графе $G, P' \subseteq P$; множество входящих в вершину x путей $L_p(x) = \{\pi_{lx} | l \neq x, \pi_{lx} \in P'\}$ и множество выходящих из вершины x путей $R_p(x) = \{\pi_{xm} | m \neq x, \pi_{xm} \in P'\}$. Алгоритмы создания и удаления связей строятся таким образом, что их выполнение производится над множеством существенных путей, и что связи, которые могут привести к несогласованности, отсекаются на этапе внесения и не участвуют в порождении путей. Ниже приводится алгоритм, применяемый для создания связей (алг. 1), сложность которого составляет $O(|R_p(y)| \cdot |L_p(x)|)$. Требуемый для работы объем памяти – $O(e^2)$. Существенным отличием алг. 1 является то, что согласованность проверяется на этапе создания связей, и несогласованность определяется сразу при внесении приводящих к ней связей.

Алгоритм 1. Алгоритм создания связей

Входные данные: (x, r, y) – связь для создания, $r \in \{<, >, \leq, \geq\}$; P' – множество всех существенных путей.

01: $r' \leftarrow$ **Алгоритм 1** (x, y, P')

02: **if** $(r' \cap r = \emptyset)$ **return** *inconsistent*

03: **if** $(r' \Rightarrow r)$ **return** // Существующее ограничение сильнее вносимого

04: **foreach** $(\pi_{lx} \in L_p(x))$ {

05: **if** $(\pi_{ly}$ существенный) $P = P \cup \pi_{ly}$

06: **foreach** $(\pi_{ym} \in R_p(y))$

07: **if** $(\pi_{lm}$ - существенный) $P = P \cup \pi_{lm}$

08: } // **foreach** $\pi_{lx} \in L_p(x)$

09: **foreach** $(\pi_{ym} \in R_p(y))$

10: **if** $(\pi_{xm}$ – существенный) $P = P \cup \pi_{xm}$

Алгоритм создания связей в результате своей работы вносит в множество существенных путей пути, которые образуются в ТЛ-графе в результате создания связи (w, l, v) . Алгоритмы вычисления выполнимого ограничения и удаления ограничения приведены в работе [Куриленко,

2008]. Эти алгоритмы построены так, что они позволяют поддерживать в актуальном состоянии множество всех существенных путей на TL-графе G , соответствующему ЕЗСВО, после каждого изменения. Основным преимуществом этого подхода является то, что после каждого изменения не требуется вызов дополнительных алгоритмов проверки согласованности и вычисления неявных ограничений.

Таким образом, для каждого текущего сценария можно держать множество существенных путей и для достройки сценария использовать приведенные выше алгоритмы. Следует отметить, что применяемая внутренняя структура данных (множество существенных путей) может быть с легкостью адаптирована для ветвящейся ЗСВО за счет хранения для каждого сценария-наследника $S_i=(V_i, C_i, S_j)$ только тех существенных путей, которые появились при дополнении ЕЗСВО сценария-предка S_j входящими в C_i ограничениями. Такой подход очень просто реализуется, позволяет значительно уменьшить объем памяти, требуемый для решения задачи.

Рассмотрим некоторые методы, позволяющие уменьшить ветвление. В примере, приведенном на (рис. 2) показана ситуация, когда в ветвлении нет необходимости. Следует отметить, что все три сценария S_{j+1} , S_{j+2} и S_{j+3} будут идентичны S_j , т.к. ограничения $(V_2, <, V_5)$ и $(V_1, <, V_4)$ следуют из ЕЗСО, соответствующей сценарию S_j . Более сложной версией проиллюстрированного правила выводимости, является случай, когда в дизъюнктивном ограничении не все элементарные ограничения следуют из S_j . Но и в этом случае можно уменьшить ветвление. Рассмотрим пример на рис.3. Сценарии S_{j+2} и S_{j+3} будут аналогичными, т.к. из ЕЗСВО, соответствующей S_j , следует ограничение $(V_2, <, V_5)$. Сценарий S_{j+1} идентичен S_j , но сохранен в качестве текущего сценария, чтобы не потерять вариант, когда выполнение ограничения $(V_5, <, V_4)$ не требуется. В этом заключается сложность логики ветвящегося времени.

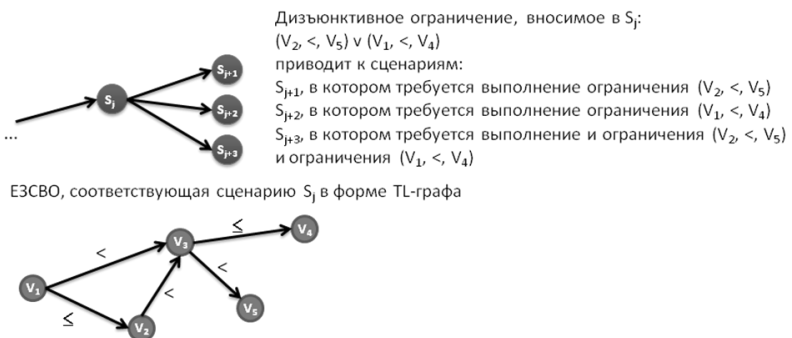


Рис. 2. Пример случая ограничения ветвления

Приведенные алгоритмы позволяют организовать решение ветвящейся ЗСВО и ее подзадач. В частности, т.к. при любых изменениях поддерживается множество текущих сценариев и для каждого сценария за счет применения приведенных ранее алгоритмов для получения множества существенных путей поддерживается актуальное решение соответствующей ЕЗСВО, автоматически решается подзадача проверки согласованности. В любой момент для любого из текущих сценариев можно проверить истинность какого-либо утверждения.

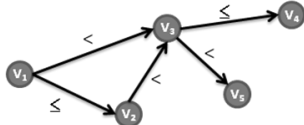
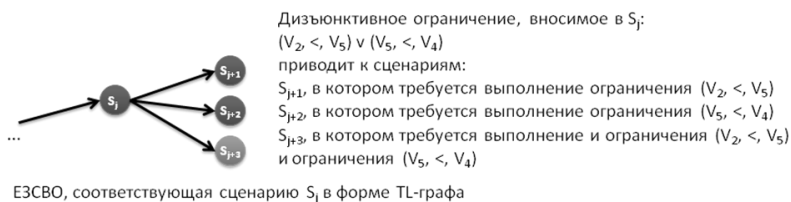


Рис. 3. Пример случая ограничения ветвления

Заключение

Построенная ветвящаяся логика на базе точечной временной логики является менее выразительной, чем известные аналоги, но практически реализуемой. Рассмотренные алгоритмы реализованы в СВР PointTime, интегрирующей различные модели представления временных зависимостей (для метрического, интервального, смешанного представления времени, для линейной и ветвящейся структур времени, количественных и качественных временных зависимостей и т.д.). Данная СВР разрабатывается в МЭИ (ГУ) для применения в составе прототипа ИСППР РВ для мониторинга и управления сложными техническими объектами [Еремеев и др., 2005].

Благодарности. Работа выполнена при финансовой поддержке РФФИ (проект № 08-01-00437-а).

Список литературы

- [Вагин и др., 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени. // Изв. РАН. Теория и системы управления, 2001, N6, с.114-123.
- [Поспелов и др., 1986] Поспелов Д.А., Литвинцева Л.В. Представление знаний о времени и пространстве в интеллектуальных системах / Под ред. Д.А. Поспелова. – М.: Наука, 1986.

- [**Башлыков и др., 1994**] Башлыков А.А., Еремеев А.П. Экспертные системы поддержки принятия решений в энергетике. М.: Изд-во МЭИ, 1994.
- [**Еремеев и др., 2003**] Еремеев А.П., Троицкий В.В. Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // Изв. РАН. Теория и системы управления, 2003, №5, с.75-88
- [**Torsun, 1998**] Torsun I.S. Foundations of Intelligent Knowledge-Based Systems // ACADEMIC PRESS, London, 1998.
- [**Pnuelli, 1977**] Pnuelli A. The temporal logic of programs. Proc. 18th Ann. Symp. Found. Computational Science, 1977. Pp. 46-57.
- [**Смирнов, 1979**] Смирнов В.А. Логические системы с модальными временными операторами // Материалы II Советско-финского коллоквиума по логике «Модальные и временные логики». – М.: Институт философии АН СССР, 1979.
- [**Allen, 1983**] Allen J.F. Maintaining Knowledge about Temporal Intervals. // Communications of the ACM. 1983. Vol. 26, no. 11. – Pp. 832-843.
- [**Gereviny et al., 1993**] Gereviny A. and Schubert L. Efficient Algorithms for Qualitative Reasoning about Time. Technical report 496, Department of Computer Science, University of Rochester, Rochester, NY, 1993.
- [**Еремеев и др., 2009**] Еремеев А.П., Куриленко И.Е. Применение темпоральных моделей в интеллектуальных системах / Интеллектуальные системы. Колл. монография. Выпуск третий. / Под. Ред. В.М. Курейчика. – М.: Физматлит, 2009, 195 с., 124-139.
- [**Куриленко И.Е., 2004**] Куриленко И.Е. Некоторые принципы построения систем временных рассуждений для СППР РВ // Сб. тр. научной сессии МИФИ–2004 в 15 т. – Т.3. – М.:МИФИ, 2004. – С. 58–59.
- [**Куриленко, 2007**] Куриленко И.Е. О методах улучшения алгоритмов вывода для системы временных рассуждений // Сб. тр. IV–й междунар. научно–практической конф. Интегрированные модели и мягкие вычисления в искусственном интеллекте в 2 т. – Т.1. – М.:ФизМатЛит, 2007 – С.149–156.
- [**Куриленко, 2009**] Куриленко И.Е. Система временного вывода для интеллектуальных систем поддержки принятия решений реального времени // Сб. док. междунар. научно-практ. конф. Интегрированные модели и мягкие вычисления в искусственном интеллекте в 2 т. – Т.1. – М.:ФизМатЛит, 2009. С 241 – 252.
- [**Куриленко, 2008**] Куриленко И.Е. «Пошаговые алгоритмы временных рассуждений для точечной модели времени» // Сб. тр. научной сессии МИФИ–2008 в 15 т. – Т.10. – М.:МИФИ, 2008. – С. 132–133.
- [**Еремеев и др., 2005**] Еремеев А.П., Куриленко И.Е. Реализация временных рассуждений для интеллектуальных систем поддержки принятия решений реального времени // Программные продукты и системы 2005, №2, с.4-16