

УДК 004.65

## ОЦЕНКА ОБЛАСТИ ЗНАЧЕНИЙ АКТИВНЫХ ПРАВИЛ ПО РЕГУЛЯРНЫМ ВЫРАЖЕНИЯМ

**Зудов А.Б.**

*ФГБОУ ВПО «Пензенский государственный университет», Пенза, e-mail: antony.zudov@mail.ru*

В статье описывается способ оценивания области значений правил со строковыми параметрами, обрабатывающих события в активных базах данных. Показано, как оценивание области значений правил может быть использовано на стадии статической проверки, когда определяются потенциальные сценарии взаимодействия активных правил и выявляются параметры событий, при обработке которых взаимодействие правил приведёт к конфликтным ситуациям. Обоснована возможность использования регулярных выражений для задания активных правил и описан формат активных правил, основанный на регулярных выражениях. Разработана схема соединения конечных автоматов, синтезированных по входящим в правило выражениям. Выделены возможные операции с данным конечным автоматом, необходимые для проведения статической проверки набора активных правил. Приведены примеры областей применения активных правил, заданных в соответствии с описанным в статье форматом.

**Ключевые слова:** активные правила, активные базы данных, регулярные выражения, статическая проверка

## ACTIVE RULES RANGE ESTIMATION ON REGULAR EXPRESSIONS

**Zudov A.B.**

*Penza State University, Penza, e-mail: antony.zudov@mail.ru*

The purpose of the research is to describe an range estimation method of active database rules with string parameters. Demonstrated, how rules range estimation can be used in static analysis, when potential rules interactions scenario is determined and parameters of events, which processing cause to conflict situations, is identified. Possibility of use of regular expressions to set the active rule is justified and active rules format, which based on regular expressions, is described. Connection scheme of finite state machines, which synthesized on regular expressions, is developed. A possible operation on these finite state machines, needed in static analysis, is highlighted. Examples of applications of active rules in describes format are given.

**Keywords:** active rules, active database, regular expressions, static analysis

Вычислительные системы, ориентированные на обработку событий, позволяют описывать и обрабатывать события на разных уровнях абстракции. Источником событий является некоторый объект мониторинга, факт смены состояний которого не может быть заранее определён по параметрам управляющих воздействий и текущему состоянию объекта.

Обработка событий, возникающих в базах данных (БД) и соответствующих по уровню абстракции терминам предметной области, требует применения событийно-ориентированной системы, по отношению к которой база данных является объектом мониторинга.

В некоторых случаях для решения данной задачи достаточно функциональности систем управления базами данных (СУБД) и механизмов триггеров. Логика обработки событий при этом должна быть относительно простой, ограниченной жёсткими временными рамками и не предполагающей возникновения большого числа промежуточных событий.

Концепция активных баз данных (АБД) подразумевает такой способ обработки событий, при котором порождаемое событие может иметь как более высокий, так и более низкий уровень абстракции по отношению к исходному событию [4]. В качестве обра-

ботчиков событий в активных базах данных используются активные правила, хранимые наравне с традиционным наполнением БД и обеспечивающие реагирование на события без необходимости в командах от внешнего приложения. Концепция не накладывает ограничений на количество промежуточных событий, если оно остаётся конечным.

Отличительной особенностью систем управления активными базами данных (СУАБД) является наличие специальной подсистемы статической проверки, которая при создании очередного активного правила выявляет возможные сценарии взаимодействия с другими правилами, выбирает среди них конфликтные [1] и определяет параметры событий, приводящих к ним.

Для активных правил, все параметры которых являются вещественными, существуют методы статической проверки, основанные на оценивании области значений правил, заданной в виде замкнутых вещественных интервалов [3].

Если активное правило в процессе обработки события генерирует новое событие, которое тоже обрабатывается тем же или другим правилом, то данные правила называют иницирующим и иницируемым

соответственно [5]. Поведение того или иного правила зависит от значений параметров обрабатываемого события. Параметры промежуточного события являются выходными параметрами иницирующего правила и входными параметрами иницируемого правила. Поэтому поведение правила зависит от того, является ли оно иницируемым. В случае нескольких промежуточных событий образуется последовательность иницирующих правил. Область значений выходных параметров каждого иницируемого правила зависит от области значений иницирующего. Таким образом, путём сравнения областей значений разных правил можно выявить входящие в них дублирующие события или конфликтующие операции.

В случае правил со строковыми параметрами реализация методов статической проверки требует решения проблемы оценки областей значений. Существует два аспекта данной проблемы. Первый состоит в способе описания параметров активных правил [2]. Второй аспект относится к способу описания оценок области значений и их построению на основе описания правил.

#### Формат активных правил

Классической нотацией описания активных правил является ЕСА (Event-Condition-Action, Событие – Условие – Действие). Активные правила вне зависимости от типов параметров задаются в следующем формате:

```
on <Событие>
when <Условие>
do <Действие>;
```

Для описания условия по строковому параметру, заданного с помощью регулярного выражения, разработана функция `regexр`, которая применяет регулярное выражение `rg1` к параметру `p1`. Для связывания условий нескольких параметров функция `regexр` имеет входной параметр для подстановки значений и выходной параметр для сохранения найденных подстрок. Сигнатура функции имеет следующий вид:

```
bool regexр(
    TRegExp rg --регулярное выражение
    , string text --текст, по которому идёт поиск
    , string[] subvalues --подставляемые значения
    , out string[] res --найденные подстроки
);
```

Выходные параметры задаются аналогично блоку замены в регулярном выражении, то есть в виде строки, состоящей из найденных выражений подстрок. Различие состоит лишь в том, что в активном правиле доступны результаты для нескольких выра-

жений. Таким образом, выходные параметры `out_p1, ..., out_pM` задаются как строковые функции от `res1, ..., resN`. Исходя из этого, специальный формат активных правил описан следующим образом:

```
on <Событие>
when regexр(rg1, p1,
subvalues1, res1)
and regexр(rg2, p1,
subvalues2, res2)
...
and regexр(rgN, pN,
subvaluesN, resN)
do out_p1(res1, res2, ..., resN)
, out_p2(res1, res2, ..., resN)
...
, out_pM(res1, res2, ..., resN);
```

Данный формат правил уступает по выразительным возможностям универсальному формату, в котором условие является единой булевой функцией, а действие – единой процедурой. Однако разработанный специальный формат позволяет получать оценку области значений путём синтеза конечного автомата.

#### Построение конечного автомата

Оценивание областей значений правил, заданных в таком формате, основано на синтезе конечных автоматов по регулярным выражениям. У автомата, соответствующего области значений правила, не менее одного входа, так как регулярное выражение `rg` обрабатывает один входной параметр правила. Кроме этого, если в выражении используются подстроки, найденные другими выражениями, то автомат может иметь дополнительные входы по количеству подставляемых значений `subvalues`. Выходов может быть несколько по числу сохраняющих конструкций в выражении, сопоставленных с найденными подстроками `res`. Если выражение не содержит таких конструкций, то выходов у автомата быть не должно.

Если подстрока, найденная выражением `rg1`, используется другим выражением `rg2`, из этого должно следовать, что один из выходов автомата  $M1$  совпадает с дополнительным входом автомата  $M2$ . Сопоставляя входы и выходы автоматов, автоматы объединяют их в группы. Для каждой такой группы предлагается строить объединённый автомат  $M'$ , реализующий ту же логику, что и группа связанных автоматов.

Если все автоматы связаны между собой, то в результате получается один объединённый автомат, и он целиком описывает логику, заложенную в блоке условия правила. Если же таких автоматов несколько, то их предлагается объединить путём параллельного соединения. В обоих случаях получается единый автомат блока условия  $C$ .

Существует несколько схем параллельного соединения. Различия схем заключаются в том, совпадает ли вход одного из соединяемых автоматов с входом другого, и преобразуются ли их выходы в один [1].

Автомат  $C$  должен иметь столько же входов, сколько и у активного правила. Также он должен иметь несколько выходов в соответствии с количеством сохраняющих конструкций в регулярных выражениях. Эти выходы необходимы, так как они найденные выражения подстроки могут быть использованы в действии правила. Поэтому предлагается схема параллельного соединения, при которой сохраняется несколько входов и выходов, по следующей системе рекуррентных соотношений:

$$\begin{cases} q[\tau+1] = (q_1[\tau+1]; q_2[\tau+1]) = (\psi_1(q_1[\tau]; x_1[\tau]); \psi_2(q_2[\tau]; x_2[\tau])); \\ y[\tau] = (y_1[\tau]; y_2[\tau]) = (\varphi_1(q_1[\tau]; x_1[\tau]); \varphi_2(q_2[\tau]; x_2[\tau])). \end{cases}$$

где  $\tau$  – такт;  $\varphi$  – функция состояний;  $\psi$  – функция переходов;  $A_i$  – автомат, построенный для  $i$ -го выходного параметра;  $C_j$  –

автомат, построенный для  $j$ -го входного параметра.

Функции выходных параметров заданы так же, как задаётся блок замены в регулярном выражении. Поэтому для каждой такой функции синтезируется конечный автомат  $A$ , имеющий несколько входов по количеству используемых в функции подстрок и один выход, соответствующий выходному параметру правила. Каждый автомат  $A$  соединяется с автоматом  $C$  по той же схеме, по которым строятся автоматы  $M'$ : выходы автомата  $C$  сопоставляются с входами автомата  $A$ .

При параллельном соединении таких автоматов по схеме с общим входом синтезируется автомат  $R$ , целиком реализующий

$$\begin{cases} q[\tau+1] = (q_1[\tau+1]; q_2[\tau+1]) = (\psi_1(q_1[\tau]; x[\tau]); \psi_2(q_2[\tau]; x[\tau])); \\ y[\tau] = (y_1[\tau]; y_2[\tau]) = (\varphi_1(q_1[\tau]; x[\tau]); \varphi_2(q_2[\tau]; x[\tau])). \end{cases}$$

Результирующая схема рекуррентных соотношений представлена на рисунке.

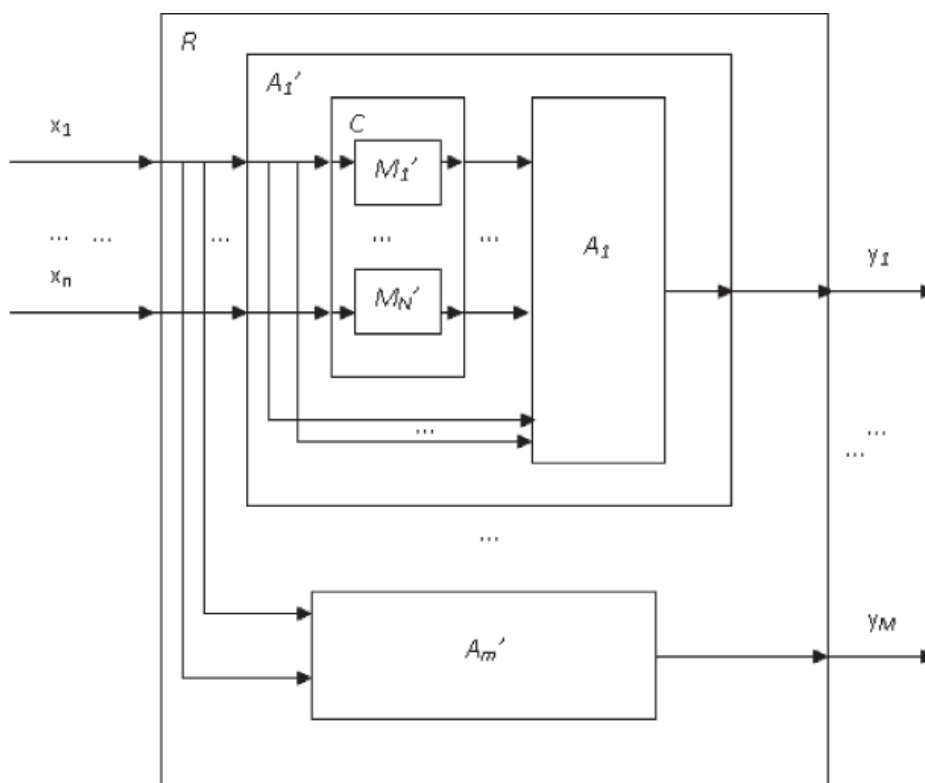


Схема соединения конечных автоматов

Результирующая схема рекуррентных соотношений является следующей:

$$\begin{cases} q[\tau+1] = \prod_i \left( \left( \varphi_{A_i}(q_{A_i}[\tau], \prod_j \varphi_{C_j}(q_{C_j}[\tau], x[\tau])) \right), \prod_j \psi_{C_j}(q_{C_j}[\tau], x[\tau]) \right); \\ y[\tau] = \prod_i \varphi_{A_i} \left( q_{A_i}[\tau], \left\langle \prod_j \varphi_{C_j}(q_{C_j}[\tau], x[\tau]), x[\tau] \right\rangle \right). \end{cases}$$

Область значений является пустой, если в задающем автомате графе не существует пути из начальной вершины в конечную. Такая проверка может быть проведена как для автомата  $R$ , так и для автоматов, синтезированных по регулярным выражениям. Если любой  $M_i$  задаёт пустую область значений, то же верно и для результирующего автомата  $R$ . Это обстоятельство может быть использовано, чтобы получить оценку области значений на более ранней стадии алгоритма оценивания с целью повышения эффективности процесса.

Область значений суперпозиции правил может быть получена путём последовательно соединения соответствующих правил автоматов  $R_1$  и  $R_2$  по следующей схеме:

$$\begin{cases} q[\tau+1] = (q_1[\tau+1]; q_2[\tau+1]) = ((\psi_1(q_1[\tau]; x_1[\tau]); \psi_2(q_2[\tau]; \varphi_1(q_1[\tau]; x_1[\tau])))); \\ y[\tau] = \varphi(q[\tau]; x[\tau]) = \varphi_2(q_2[\tau]; \varphi_1(q_1[\tau]; x_1[\tau])). \end{cases}$$

Для статической проверки указанные операции являются основными.

### Заключение

Предложен формат активных правил со строковыми параметрами, основанный на регулярных выражениях, и модель описания областей значений активных правил, отличающаяся представлением входных и выходных параметров правил в виде входов и выходов конечного автомата. Модель позволяет проводить оценивание областей значений путём синтеза конечного автомата по регулярным выражениям. С помощью оценки области значений возможно установить факт инициирования правилами друг друга путём проверки на пустоту их последовательного соединения.

Активные правила предложенного формата могут использоваться в областях, для которых характерно большое число промежуточных событий. Такими областями являются геоинформационные системы, базы данных движущихся объектов, базы данных операторов сотовой связи, социальные сети, гетерогенные базы данных. В частности, описанный формат был применён при разработке правил, обрабатывающих события, возникающие при работе с адресным реестром в электронной карте, которая используется в Администрации города Пензы и поддерживается МУП «Объединённая городская служба архитектуры, градостроительства и технической инвентаризации».

### Список литературы

1. Макарычев П.П., Волгина Н.А. Моделирование сетей массового обслуживания на основе маркированных графов // Известия вузов. Поволжский регион. Технические науки. – 2008. – № 3. – С. 33–39.
2. Мелихов А.Н. Ориентированные графы и конечные автоматы. – М.: Наука, Физматлит, 1971. – 416 с.
3. Bailey J. Termination analysis of active rules modular sets / J. Bailey, Ra Poulouvassilis, P. Newson // Proc. 1st Int.

Conf. on Computational Logic (DOOD stream), LNCS 1861. – London, UK, 2000. – P. 1106–1120.

4. Debray S. Constraint-Based Validation Analysis for Cyclic Active Database Rules / Saumya K. Debray, Timothy J. Hickey // Proceedings of the First International Conference on Computational Logic. – London, UK, 2000. – P. 1121–1136.

5. Paton N.W. Active database systems / N.W. Paton, O. Diaz // ACM Computing Surveys (CSUR). – 1999. – № 31. 1 – P. 63–103.

6. Ray I. Detecting Termination of Active Database Rules Using Symbolic Model Checking / I. Ray, I. Ray // 5th East European Conference, Proceedings. – 2001. – № 13. – P. 266.

### References

1. Makarychev P.P., Volgina N.A. Modelirovaniye setej massovogo obsluzhivaniya na osnove markirovannykh grafov // Izvestiya vuzov. Povolzhskiy region. Tehnicheskie nauki. no. 3, 2008, pp. 33–39.
2. Melihov A.N. «Orientirovannye grafy i konechnye avtomaty» [Directed graphs and finite state machines], Nauka, Fizmatlit, Moscow, 1971. 416 p.
3. Bailey J. Termination analysis of active rules modular sets / J. Bailey, Ra Poulouvassilis, P. Newson // Proc. 1st Int. Conf. on Computational Logic (DOOD stream), LNCS 1861 – London, UK, 2000. pp. 1106–1120.
4. Debray S. Constraint-Based Validation Analysis for Cyclic Active Database Rules / Saumya K. Debray, Timothy J. Hickey // Proceedings of the First International Conference on Computational Logic London, UK, 2000. pp. 1121–1136.
5. Paton N.W. Active database systems / N.W. Paton, O. Diaz // ACM Computing Surveys (CSUR). 1999 no. 31.1 pp. 63–103.
6. Ray I. Detecting Termination of Active Database Rules Using Symbolic Model Checking / I. Ray, I. Ray // 5th East European Conference, Proceedings. 2001. no. 13. pp. 266.

### Рецензенты:

Зинкин С.А., д.т.н., профессор кафедры «Вычислительная техника», ФГБОУ ВПО «Пензенский государственный университет», г. Пенза;

Макарычев П.П., д.т.н., профессор, зав. кафедрой МОиПЭВМ, ФГБОУ ВПО «Пензенский государственный университет», г. Пенза.

Работа поступила в редакцию 02.03.2015.