

РЕАЛИЗАЦИЯ ОПЕРАТОРОВ СЛИЯНИЯ И РАЗЛИЯНИЯ В ГРАФИЧЕСКОМ РЕДАКТОРЕ АЛГОРИТМИЧЕСКИХ СЕТЕЙ

О. Ф. КОРОЛЕВ

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

<korolf@rambler.ru>

УДК 004.4'236

Королев О. Ф. Реализация операторов слияния и разлияния в графическом редакторе алгоритмических сетей // Труды СПИИРАН. Вып. 4. — СПб.: Наука, 2007.

Аннотация. В статье рассматривается реализация в графическом редакторе операторов слияния и разлияния, широко используемых в практике построения моделей на основе алгоритмических сетей. В предыдущей версии графического редактора эти операторы были реализованы в упрощенной форме. Рассматриваются сложности при программной реализации операторов в полном варианте и возможные их решения. — Библ. 4 назв.

UDC 004.4'236

Korolev O. F. The Realization of Effluence and Confluence Operators in the Graphical Editor of the Algorithmic Networks // SPIIRAS Proceedings. Issue 4. — SPb.: Nauka, 2007.

Abstract. In the article the graphical editor realization of effluence and confluence operators widely used in the practice of building algorithmic networks models is considered. In the previous version of the graphical editor these operators were realized only in the restricted form. The program realization full form operator difficulties and their possible ways to solve are considered. — Bibl.4 items.

1. Определение операторов разлияния и слияния

Одними из применяемых операторов формализма алгоритмических сетей (АС) являются операторы слияния и разлияния [1].

Графическое изображение оператора разлияния представлено на рис. 1.

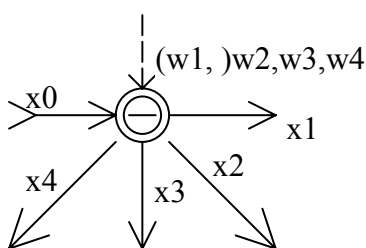


Рис. 1. Оператор разлияния.

Вычисление выходов оператора разлияния происходит по формулам:

$$\begin{cases} x_j = x_0 \cdot w_j, j = 1..n, \\ \sum_{i=1}^n w_i = 1. \end{cases}$$

Существенным недостатком такого определения является наложение ограничения на значения весов. Поскольку при реализации систем на основе АС процессы построения модели и ввода данных разделяются, то синтаксическая

проверка правильности ввода оператора возможна только после ввода информации о значениях весов. Поэтому при программной реализации предпочтительней следующее вычисление выходов оператора:

$$\begin{cases} x_1 = x_0 \cdot (1 - \sum_{i=2}^n w_i), \\ x_j = x_0 \cdot w_j, j = 2..n. \end{cases}$$

В данном определении условие нормировки весов уже заложено в определении оператора и проверка правильности ввода графической конструкции возможна на этапе построения модели.

Графическое изображение оператора слияния представлено на рис. 2.

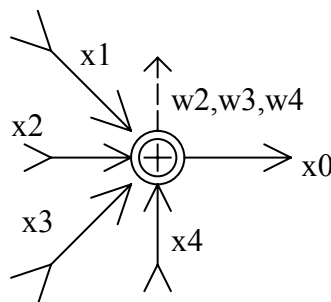


Рис. 2. Оператор слияния.

Поскольку оператор слияния является обратным для оператора разливания, то формулы для вычисления выходов оператора получаются следующим образом:

$$\begin{cases} w_j = x_j / x_0, j = 2..n, \\ x_1 = x_0 \cdot (1 - \sum_{i=2}^n w_i) = x_0 \cdot (1 - \sum_{i=2}^n \frac{x_i}{x_0}) = x_0 - \sum_{i=2}^n x_i, \Rightarrow \\ x_0 = \sum_{i=1}^n x_i, \\ w_j = x_j / x_0, j = 2..n. \end{cases}$$

В вырожденном случае одного входа оба оператора на выходе выдают входное значение.

2. Сложности программной реализации операторов разливания и слияния

С точки зрения реализации ввода подобных графических конструкций в графическом редакторе эти операторы являются нестандартными по следующему ряду причин:

1. Графическое представление сети состоит из базисных графических примитивов (знаки операторов, дуги, метки переменной и значки разметки [2–4]). Графическое изображение операторов разливания и слияния содержит пунктирную дугу, не входящую в число базовых примитивов.

2. Каждой дуге АС соответствует одна переменная АС. В случае пунктирной дуги одной дуге соответствует несколько переменных-весов.
3. В [3, 4] одной из составляющих общей синтаксической проверки АС было выполнение требования ввода минимальной формы для каждого оператора сети. Данное требование подразумевает контроль ввода необходимого числа входных и выходных дуг АС. В случае же, например, оператора разливания необходимое число входных дуг-весов зависит от числа введенных выходных дуг.
4. Непонятен способ осуществления соответствия между весами и выходами (входами) оператора.

В целях ухода от решения перечисленных проблем в первой версии графического редактора АС [2] были реализованы упрощенные варианты операторов слияния и разливания. Операторы имели фиксированное число входов и выходов, равное двум. Графическое изображение упрощенного оператора разливания изображено на рис. 3, а оператора слияния — на рис. 4.

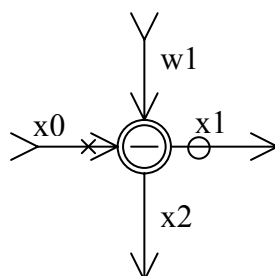


Рис. 3. Упрощенный оператор разливания.

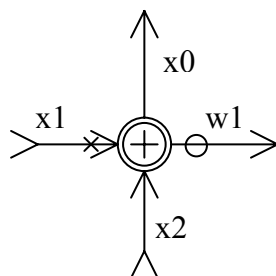


Рис. 4. Упрощенный оператор слияния.

Вычисление выходов оператора разливания происходило по формулам

$$\begin{cases} x_1 = x_0 \cdot w_1, \\ x_2 = x_0 \cdot (1 - w_1), \end{cases}$$

а оператора слияния

$$\begin{cases} w_1 = x_1 / (x_1 + x_2), \\ x_0 = x_1 + x_2. \end{cases}$$

Соответствие весов и выходов (входов) задавалось с помощью значков разметки.

Однако предложенное решение неудобно при вводе в графическом редакторе накопленного ряда созданных ранее моделей, поскольку операторы разливания и слияния, не удовлетворяющие ограничениям, приходится заменять другими операторами.

3. Вариант программной реализации операторов

Рассмотрим решение отмеченных в пункте 2 сложностей при реализации данных операторов в графическом редакторе.

Из введенного ограничения, что знак оператора имеет восемь точек возможных связей с дугами [3, 4], следует, что весов может быть не более пяти. Например, для разливания: одна входная дуга соответствует разливаемой переменной; пунктирная дуга предназначена для весов; шесть выходных дуг соответствуют долям разливаемой переменной, а число весов должно быть на единицу меньше, т.е. не более пяти. Таким образом, с учетом весов один оператор может иметь до двенадцати входов и выходов, т.е. строка матрицы операторов будет состоять из двенадцати элементов.

На практике данное ограничение не является жестким и позволяет вводить большинство имеющихся моделей на основе АС. В случае необходимости разливания (слияния) на (из) более чем шесть (шести) составляющих, возможным выходом является работа с векторными переменными. Например, оператор разливания будет иметь графическое изображение, представленное на рис. 5.

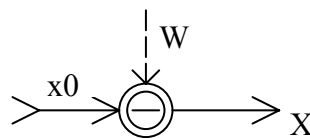


Рис. 5. Векторный оператор разливания.

Если переменная X является вектором размерности n , то W является вектором размерности $n-1$. Вычисление элементов вектора X происходит по формулам:

$$\begin{cases} x_1 = x_0 \cdot (1 - \sum_{i=1}^{n-1} w_i), \\ x_j = x_0 \cdot w_{j-1}, j = 2..n. \end{cases}$$

Оператор слияния будет иметь графическое изображение, представленное на рис. 6.

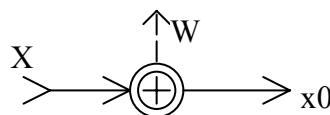


Рис. 6. Векторный оператор слияния.

Переменная X является вектором размерности n , W является вектором размерности $n-1$, а вычисление x_0 и элементов W происходит по формулам:

$$\begin{cases} x_0 = \sum_{i=1}^n x_i, \\ w_j = x_{j+1} / x_0, j = 1..n-1. \end{cases}$$

Однако векторный вариант операторов неудобен тем, что в АС требуется использовать дополнительные операторы по формированию вектора из скаляров и для чтения составляющих его элементов.

3.1. Пунктирная дуга

Если вводить новый пунктирный тип дуги, то либо добавляется новый базисный элемент (дополнительно к знаку оператора, дуге, значку разметки и метке переменной), либо вводится дополнительная характеристика дуги (пунктир). В любом случае требуется пересмотр всех синтаксических проверок и всей структуры данных, поскольку данная дуга не подчиняется общим правилам и ей соответствует несколько имен переменных.

Возможным выходом является прорисовка пунктирной дуги как фиктивной составляющей образа знака оператора. При этом необходимо изменение реализации синтаксических правил А1–Е1 и А1–А9 [3, 4], обеспечивающих непересечение знаков операторов с другими графическими примитивами и пересечение с окончаниями и началами дуг только в точках возможных связей. Вместо простой проверки расстояния от центра знака оператора до других элементов АС предложено следующее решение:

- фиктивная пунктирная составляющая добавляется включением в графический образ значка оператора в файле настройки системы новых элементов (достаточно несколько линий);
- при синтаксической проверке на непересечение знака оператора с другими элементами АС проверка выполняется для каждого элемента, составляющего графический образ значка оператора.

3.2. Проблема переменных-весов

Возможным решением является изменение диалоговой формы ввода метки переменной, а именно возможность ввода имени веса, соответствующего дуге. При этом необходим анализ, требует ли дуга, для которой задается метка переменной, задание веса или нет. Задание веса разрешается при выполнении следующих условий:

- дуга является выходной (входной) для оператора разлития (слияния);
- дуга не является первой выходной (входной) для оператора разлития (слияния) (для обеспечения равенства суммы весов единице для первой дуги вес не задается);
- для сохранения равенства суммы весов единице в случае удаления первой дуги следующая дуга должна становиться первой и у нее уничтожается введенный вес.

Для правильного сбора матрицы операторов и синтаксической проверки при вводе веса в АС создается соответствующая дуга с признаком «вес», состоящая из отрезка нулевой длины с координатами, совпадающими с координатами знака оператора. Совпадение координат необходимо для определения вхождения и выхода весов из знака оператора, а нулевая длина — для не отображения дуг, которым соответствуют веса, на рабочем поле (пришлось изменить процедуры отображения и удаления графического элемента так, чтобы они игнорировали нахождение данных фиктивных дуг). В случае удаления или замены знака оператора необходимо удалять соответствующие дуги-веса.

Признак «вес» необходим, чтобы различать обычные дуги, имеющие графическое отображение, и веса, не имеющие такового.

Также при вводе (редактировании) имен весов было организовано автоматическое отображение их на рабочем поле рядом со знаком оператора.

3.3. Проверка минимальной формы

Минимальное и максимальное число входных (выходных) дуг оператора задается в файле настройки системы и влияет на синтаксическую проверку (выполнение ограничений минимальной и максимальной форм), а также используется при шаблонном вводе.

Необходимо различать минимальное и максимальное число дуг, которые можно ввести графически и допустить аналитически. Например, для оператора разливания графически можно ввести только одну входную дугу, а аналитически он имеет один разливаемый вход и до пяти задаваемых весов. При вводе имени выходной дуги и имени соответствующего веса создается входная дуга-вес, при этом система благодаря признаку «вес» отличает ее и требует ввода еще входной дуги для разливаемой величины.

Также добавляется условие равенства числа входных и выходных дуг. Это необходимо для исключения случая, когда, например, для оператора разливания не указаны имена соответствующих выходным дугам весов и соответственно не созданы соответствующие входные дуги-веса.

3.4. Соответствие весов и входов (выходов) операторов

В структуре данных в оперативной памяти (а также в структуре хранения на диске) было реализовано сохранение связи между переменной и соответствующим ей весом. В случае оперативной памяти в качестве характеристики дуги хранится указатель на соответствующую ей дугу-вес. При записи на диск вместо указателя сохраняется текущий адрес дуги-веса в оперативной памяти и этот же адрес записывается как характеристика дуги-веса. При последующем открытии на основании этой информации восстанавливаются динамические ссылки. Пример хранения на диске для модели на рис. 7 приведен в листинге 1.

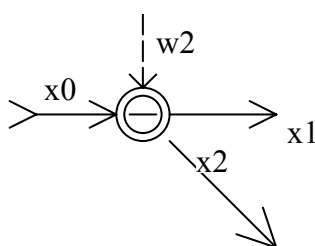


Рис. 7. Модель с оператором разливания.

```

[oper]
0 0 0
#arc (5 3) (5 3) (0 0)
#name (w2 ) (0 0 0)
#weight_index 17387944
#arc (4 3) (4.80 3) (0 0)
#name (x0 ) (4.260 2.880 1)
[oper]
60 5 3
#arc (5.20 3) (6 3) (0 0)
#name (x1 ) (6.080 3.140 1)
#arc (5.20 3.20) (6 4) (0 0)
#name (x2 ) (5.400 3.400 1)
#weight_index_ref 17387944
#extra :names<w2>
...

```

Листинг 1. Формат хранения модели на диске.

Литература

1. *Иванищев В. В.* Автоматизация моделирования потоковых систем. Л.: Наука, 1986. 142 с.
2. *Быков Я. А.* Система автоматизации моделирования на основе визуализированного представления моделей в формализме алгоритмических сетей. Дис. ... канд. техн. наук. СПб., 2001. 140 с. (СПИИРАН, СПб.).
3. *Королев О. Ф.* Синтаксис графического построения алгоритмических сетей // Информационные технологии и интеллектуальные методы. СПб.: СПИИРАН, 1999. Вып. №3. С.112–131.
4. *Королев О. Ф.* Методы графического представления моделей на основе алгоритмических сетей и их программная реализация: Дис. ... канд. техн. наук. СПб., 2003. 162 с. (СПИИРАН, СПб.).