УДК 519.685.3

ПРЕОБРАЗОВАНИЕ НЕДЕТЕРМИНИРОВАННЫХ СИНТАКСИЧЕСКИХ ДИАГРАММ В ДЕТЕРМИНИРОВАННЫЕ

Ю. Д. Рязанов

Белгородский государственный технологический университет им. В.Г. Шухова

Поступила в редакцию 14.10.2014 г.

Аннотация. Рассматривается задача эквивалентного преобразования синтаксических диаграмм в детерминированные синтаксические диаграммы. Определены причины (конфликты), по которым диаграмма может быть недетерминированной и предложен способ и алгоритм устранения конфликтов типа «переход - переход».
Ключевые слова: формальный язык, синтаксическая диаграмма, эквивалентные преобразования, детерминированная синтаксическая диаграмма.
Annotation. This paper presents the problem of equivalent transformation of syntax diagrams into deterministic syntax diagrams. The author identifies the possible reasons (conflicts) for diagrams to be nondeterministic and suggests a method and algorithm of solving the «transition - transition» type of conflicts.
Keywords: formal language, syntax diagram, equivalent transformations, deterministic syntax diagramg.

ВВЕДЕНИЕ

Одним из удобных способов спецификации формальных языков является синтаксическая диаграмма (СД), которую впервые применил Н. Вирт для описания языка Паскаль [1]. СД используются как для документирования языков программирования, так и при проектировании обработчиков языков (распознавателей, трансляторов, интерпретаторов) [2–4].

Для класса детерминированных СД (ДСД) [5, 6] существуют алгоритмы синтеза МП-распознавателей с одним и множеством состояний [7] рекурсивных и нерекурсивных программ-распознавателей линейной сложности [5, 8]. Исходные СД могут не принадлежать классу ДСД, поэтому, для синтеза программ-распознавателей линейной сложности, их нужно преобразовать в ДСД. Но не любую СД можно преобразовать в ДСД, так как даже проблема распознавания, существует ли для данной СД, не являющейся ДСД, эквивалентная ей ДСД, неразрешима (в противном случае была бы разрешима проблема распознавания, существует ли для данной КС-грамматики, не являющейся $LL(1)$-грамматикой, эквивалентная ей $LL(1)$-грамматика, неразрешимость которой показана в [9]). Однако для некоторых СД такое преобразование возможно. В данной работе определяются причины (конфликты), по которым СД не принадлежат классу ДСД и предлагается способ и алгоритм, который с помощью эквивалентных преобразований либо устраняет конфликты определенного типа в СД, либо останавливается с сообщением о невозможности устранить конфликты предложенным способом. Это позволит расширить класс СД, по которым можно проектировать распознаватели линейной сложности.

ОСНОВНЫЕ ПОНЯТИЯ

Синтаксическую диаграмму будем задавать пятеркой $D = (T, N, G, S, F)$, где

T – конечное множество терминалов;

N – конечное множество нетерминалов;

$S \in N$ – начальный нетерминал;

$G = (V, E)$ – ориентированный граф, где

© Рязанов Ю. Д., 2015

$V = V_T \cup V_N \cup V_u \cup V_{вход} \cup V_{выход}$, где

$V_{вход}$ – множество точек входа, $|V_{вход}| = |N|$;

V_T – множество терминальных вершин;

V_N – множество нетерминальных вершин;

V_u – конечное множество узлов;

$V_{выход}$ – множество точек выхода, $|V_{выход}| = |N|$;

$E = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$, где

$E_1 \subseteq \{(a,b) \mid a \in V_{вход}, b \in V_u\}$ – множество входных дуг;

$E_2 \subseteq \{(a,b) \mid a \in V_u, b \in V_{выход}\}$ – множество выходных дуг;

$E_3 \subseteq \{(a,b) \mid a \in V_u, b \in V_T \cup V_N\}$ – множество дуг, выходящих из узлов;

$E_4 \subseteq \{(a,b) \mid a \in V_T \cup V_N, b \in V_u\}$ – множество дуг, входящих в узлы;

$E_5 \subseteq \{(a,b) \mid a \in V_u, b \in V_u\}$ – множество ε-дуг, соединяющих узлы;

$F : V_T \cup V_N \to T \cup N$ – отображение множества вершин на множество терминалов и нетерминалов.

Пример синтаксической диаграммы в графической форме представлен на рис. 1.

Каждому нетерминалу соответствует связная компонента графа. Компонента именуется соответствующим нетерминалом, имеет только одну точку входа и одну точку выхода и конечное множество вершин других типов. Точки входа и выхода на диаграмме компоненты не изображаются. Терминальная вершина изображается кружочком, в который вписан терминальный символ, нетерминальная – прямоугольником, в который вписан нетерминальный символ. Узел изображается на диаграмме жирной точкой. В точку входа не входит ни одна дуга и выходит конечное множество дуг (входные дуги компоненты). Узлы, в которые входят входные дуги, называются начальными. Из точки выхода не выходит ни одна дуга и входит конечное множество дуг (выходные дуги компоненты). Узлы, из которых выходят выходные дуги, называются заключительными.

СД называется псевдодетерминированной (ПСД) [10], если все ее компоненты псевдодетерминированные. Компонента СД псевдодетерминированная, если:

1) содержит только один начальный узел;

2) не имеет ε-дуг;

3) все ее узлы – псевдодетерминированные.

Узел компоненты СД псевдодетерминированный, если любые две дуги, выходящие из него, входят в вершины, содержащие различные символы.

Любую СД можно преобразовать в эквивалентную ей ПСД [10].

Цепочку языка можно получить, «двигаясь» по дугам СД от точки входа начальной компоненты к ее точке выхода. При этом если дуга идет в терминальную вершину, то вписанный в нее символ добавляем в цепочку,

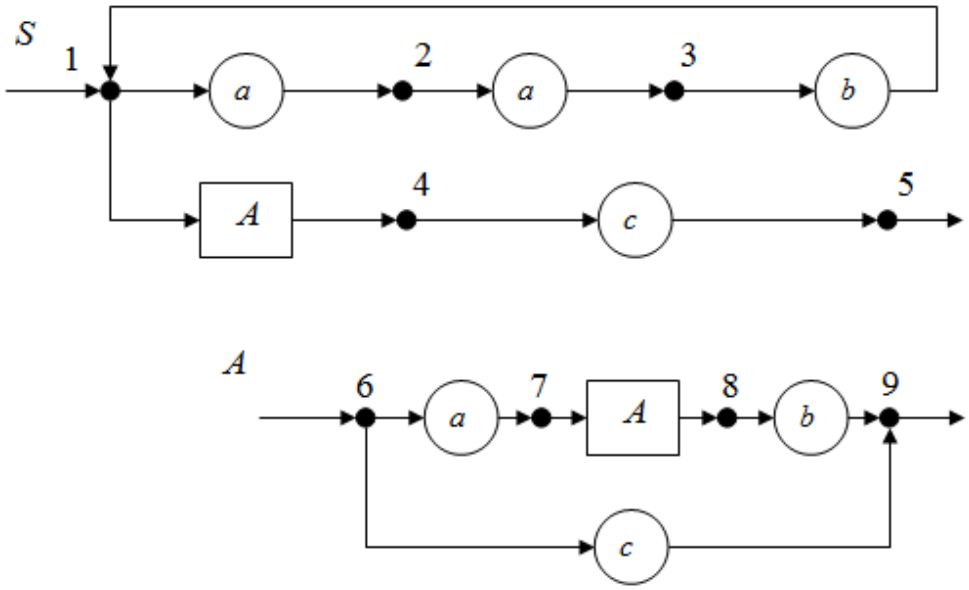

Рис. 1. Синтаксическая диаграмма

если дуга идет в нетерминальную вершину, то переходим в соответствующую компоненту и движемся по ней аналогичным образом до точки выхода, после чего возвращаемся в предыдущую компоненту и продолжаем движение. После прохождения выходной дуги начальной компоненты в цепочку добавляем концевой маркер (-|).

Символ x, который может быть добавлен в цепочку после прохождения выходящей из узла дуги e, принадлежит множеству выбора дуги e (ВЫБОР(e)). Узел u называется детерминированным, если множества выбора любых двух дуг, выходящих из узла u, не пересекаются. СД является детерминированной (ДСД), если в ней все узлы детерминированные. В работах [5, 6] описаны алгоритмы вычисления множеств выбора дуг, выходящих из узлов, и определения принадлежности СД классу ДСД.

КОНФЛИКТЫ «ПЕРЕХОД-ПЕРЕХОД» И «ПЕРЕХОД-ВЫХОД»

ПСД не является детерминированной, если она содержит в себе хотя бы одну недетерминированную компоненту. Компонента СД является недетерминированной, если в ней есть хотя бы один недетерминированный узел. Узел компоненты ПСД недетерминированный, если из него выходят хотя бы две дуги, множества выбора которых пересекаются. Если множества выбора дуг (u,x) и (u,y), выходящих из узла u, пересекаются (ВЫБОР(u,x) $\cap$ ВЫБОР$(u,y) \neq \varnothing$), то будем говорить, что в узле u имеет место конфликт типа «переход – переход». Если узел u заключительный и множества выбора выходной дуги (u, *выход*) и дуги (u,x) пересекаются, то будем говорить, что в узле u имеет место конфликт типа «переход – выход». В ДСД конфликты типа «переход – переход» и «переход – выход» отсутствуют.

Для того чтобы преобразовать СД в ДСД, нужно устранить конфликты типа «переход – переход» и «переход – выход». В данной работе предлагается способ устранения конфликтов типа «переход – переход». Отметим еще раз, что не каждую СД можно преобразовать в ДСД, поэтому в работе описан алгоритм, который либо устраняет конфликты типа «переход – переход», либо останавливается с сообщением о невозможности устранить конфликты типа «переход – переход» предложенным способом.

УСТРАНЕНИЕ КОНФЛИКТОВ ТИПА «ПЕРЕХОД – ПЕРЕХОД»

Предлагаемый способ устранения конфликтов заключается в следующем. Если в компоненте K есть узел u, такой, что ВЫБОР (u,A) $\cap$ ВЫБОР$(u,X) \neq \varnothing$, где A – нетерминал, X – терминал или нетерминал, то вершина с нетерминалом A заменяется на соответствующую компоненту и полученная компонента преобразуется в псевдодетерминированную, которая может быть детерминированной. Если компонента A имеет конфликт «переход – переход», то нет смысла выполнять такую замену, так как в результате в компоненту K добавятся узлы с конфликтами. Если же в компоненте A нет конфликтов, то после замены и преобразования компоненты K в псевдодетерминированную все равно могут появиться узлы с конфликтами. Процесс разрешения конфликтов может зациклиться. Обнаружить истинное зацикливание сложно, это требует глубокого анализа всех результатов преобразований. Одним из способов избежания зацикливаний может быть следующий. Каждому узлу u, имеющему конфликт, поставить в соответствие характеристику χ, которая представляет собой множество терминалов и нетерминалов, записанных в вершинах, в которые идут дуги из узла u. В результате получим конечное множество характеристик $\{\chi_1, \chi_2, \ldots, \chi_n\}$. Каждой характеристике χ_i поставить в соответствие счетчик c_i. Если в процессе устранения конфликтов выполняется замена нетерминальной вершины, в которую идет дуга из узла u с характеристикой χ_i, то увеличить счетчик c_i. Попытку устранения конфликтов можно считать неудачной, если один из счетчиков превысил некоторое критическое значение. В качестве такого значения можно

взять, например, количество узлов в компоненте.

Алгоритм устранения конфликтов типа «переход – переход»

Вход: псевдодетерминированная синтаксическая диаграмма.

Выход: 1) F – признак возможности устранения конфликтов (0 – конфликты нельзя устранить, 1 – конфликты устранены);

2) если конфликты устранены — таблицы СД без конфликтов «переход – переход».

1. $F := 1$ (предполагаем, что конфликты устранить можно).

Сформировать множество M компонент, имеющих конфликты «переход – переход». Если $M = \varnothing$, то конфликтов «переход – переход» нет, конец алгоритма, иначе выполнить п. 2.

2. Выбрать компоненту $K \in M$, для которой истинно следующее условие: если узел u имеет конфликт «переход – переход» и множество выбора дуги, идущей из узла u в нетерминальную вершину с нетерминалом A пересекается с множеством выбора какой либо другой дуги, выходящей из узла u, то компонента, соответствующая нетерминалу A, не принадлежит множеству M.

3. Выполнить попытку устранения конфликтов «переход – переход» в компоненте K следующим образом.

3.1. $M\chi := \varnothing$, где $M\chi$ – множество характеристик.

3.2. Сформировать множество MU узлов с конфликтами.

3.3. Для каждого узла из множества MU определить характеристику χ и, если она не принадлежит множеству $M\chi$, то включить ее в множество $M\chi$ и обнулить соответствующий ей счетчик.

3.4. Обработать каждый узел u из множества MU. Если счетчик c, соответствующий характеристике узла u превысил критическое значение, то попытку устранения конфликтов считать неудачной, $F := 0$, конец алгоритма, иначе обработать каждую дугу, идущую из узла u в нетерминальную вершину. Если ВЫБОР(u, A) $\cap$ ВЫБОР$(u, X) \neq \varnothing$, где A – нетерминал, X – терминал или нетерминал, то вершину с нетерминалом A заменить на соответствующую компоненту и увеличить счетчик c.

3.5. Компоненту K преобразовать в псевдодетерминированную.

3.6. Сформировать множество MU узлов с конфликтами. Если $MU = \varnothing$, то попытку устранения конфликтов в компоненте K считать удачной, иначе выполнить п. 3.3.

4. Если конфликты в компоненте K устранены, то исключить ее из множества M $(M := M \setminus \{K\})$. Если $M \neq \varnothing$, то выполнить п. 2, иначе конфликты «переход – переход» устранены, конец алгоритма.

Рассмотрим устранение конфликтов «переход – переход» в ПСД, представленной на рис. 1. В этой ПСД только компонента S имеет конфликт «переход – переход», так как ВЫБОР$(1, a) = \{a\}$, ВЫБОР$(1, A) = \{a, c\}$ и ВЫБОР$(1, a) \cap$ ВЫБОР$(1, A) = \{a\} \cap \{a, c\} = \{a\}$, поэтому $M = \{S\}$. Компоненту S можно выбрать для устранения конфликта, так как в ней только узел 1 имеет конфликт и компонента A не принадлежит множеству M. Определим характеристику узла 1: $\chi = \{a, A\}$. Включим ее в множество $M\chi$ и обнулим соответствующий ей счетчик $(c = 0)$. Значение счетчика не превышает количества улов в компоненте $(c < 5)$, поэтому вершину с нетерминалом A заменяем соответствующей компонентой и получаем компоненту S, представленную на рис. 2.

После преобразования ее в псевдодетерминированную (алгоритм преобразования описан в [10]), получим компоненту S, представленную на рис. 3.

Увеличиваем счетчик на единицу $(c = 1)$. В компоненте S (рис. 3) есть узел с конфликтом и характеристикой $\chi = \{a, A\}$, значение счетчика этой характеристики меньше пяти, поэтому вершину с нетерминалом A заменяем соответствующей компонентой и получаем компоненту S, представленную на рис. 4.

После преобразования ее в псевдодетерминированную, получим компоненту S, представленную на рис. 5.

Увеличиваем счетчик на единицу $(c = 2)$. В полученной компоненте S конфликтов нет. Исключаем ее из множества M. Множество M стало пустым, детерминированная СД получена.

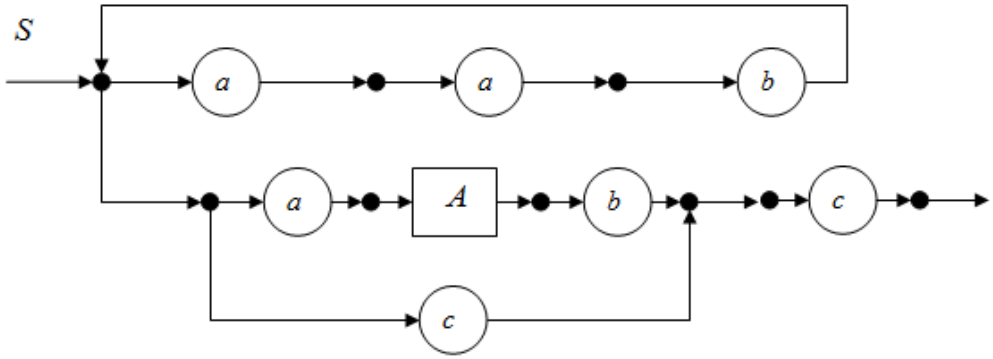

Рис. 2. Компонента S

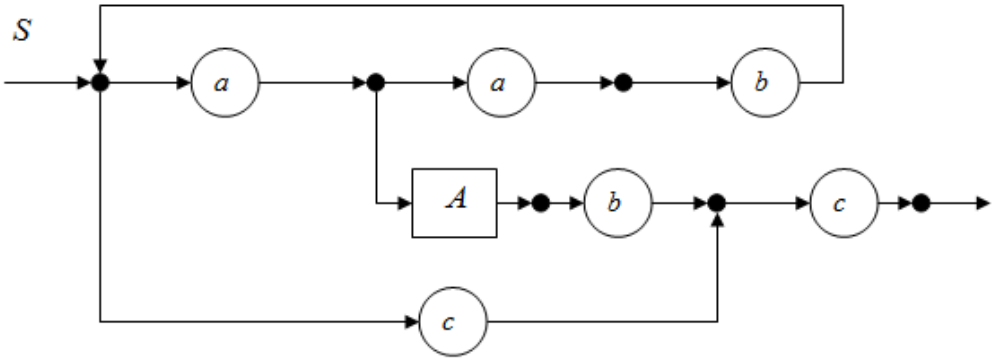

Рис 3. Псевдодетерминированная компонента S

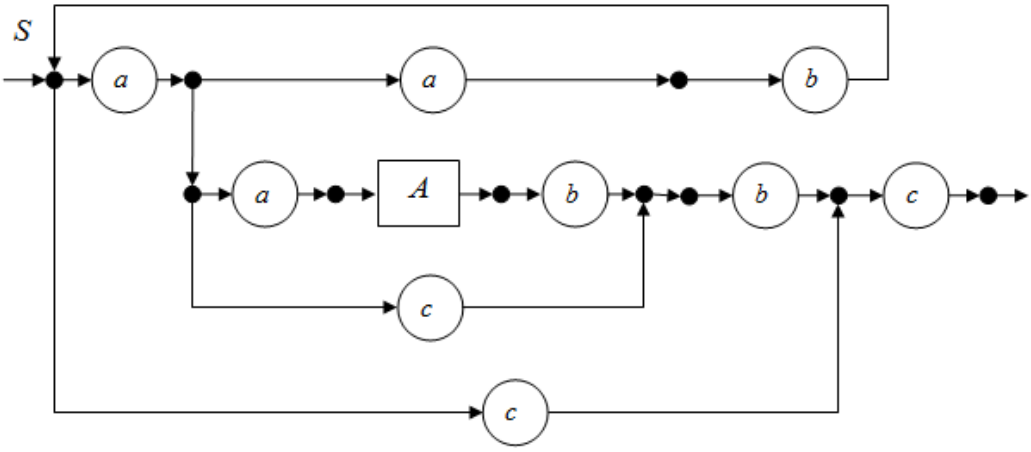

Рис. 4. Компонента S

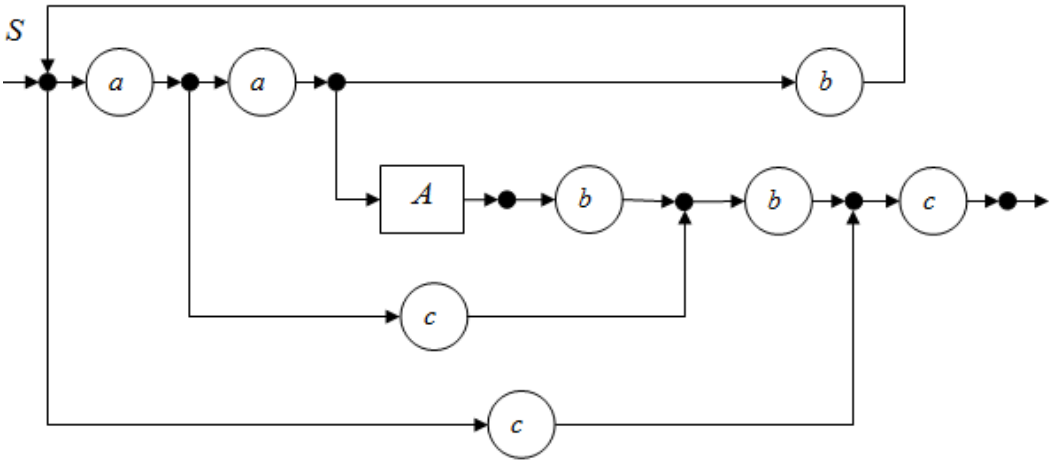

Рис. 5. Псевдодетерминированная компонента S

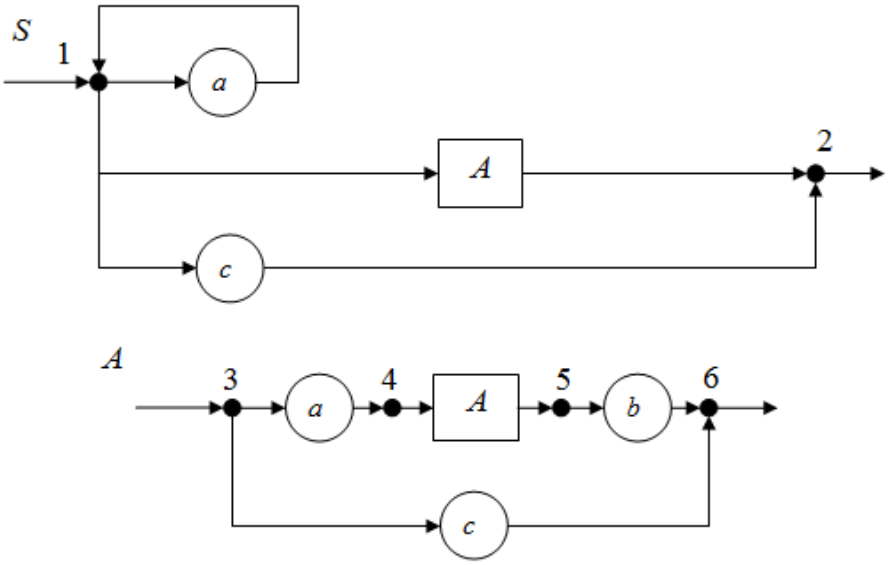

Рис. 6. Синтаксическая диаграмма

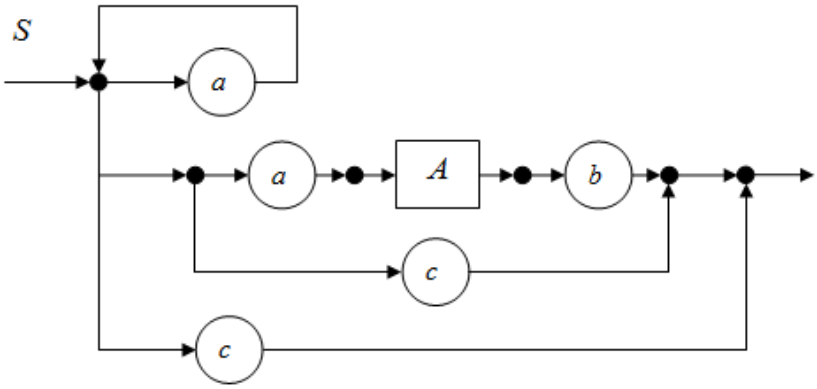

Рис. 7. Компонента S

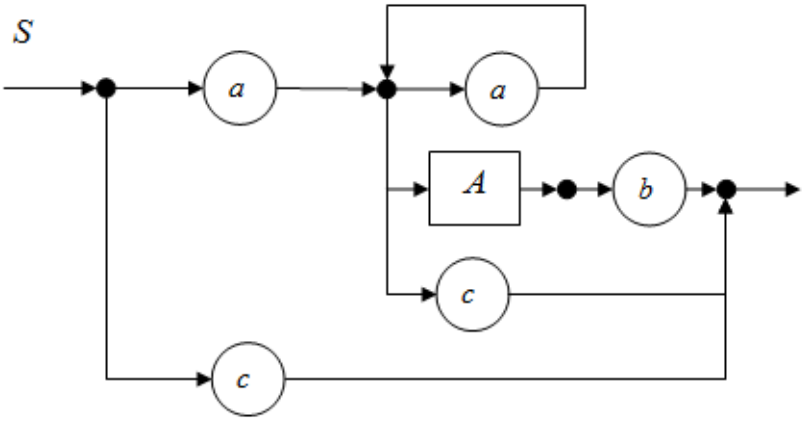

Рис. 8. Псевдодетерминированная компонента S

Рассмотрим еще один пример. Пусть дана псевдодетерминированная СД, представленная на рис. 6.

В этой СД компонента S имеет конфликт «переход – переход» в узле 1, так как ВЫБОР $(1, A) = \{a, c\}$ и пересекается с ВЫБОР$(1, a)$ и ВЫБОР$(1, c)$. Включаем в множество M компоненту S $(M = \{S\})$. Компонента A не принадлежит множеству M, поэтому можно приступить к устранению конфликта в компоненте S. Определим характеристику узла 1: $\chi = \{a, c, A\}$. Включим ее в множество $M\chi$ и обнулим соответствующий ей счетчик $(c = 0)$. Значение счетчика не превышает количества улов в компоненте $(c < 2)$, поэтому вершину с нетерминалом A заменяем соответствую-

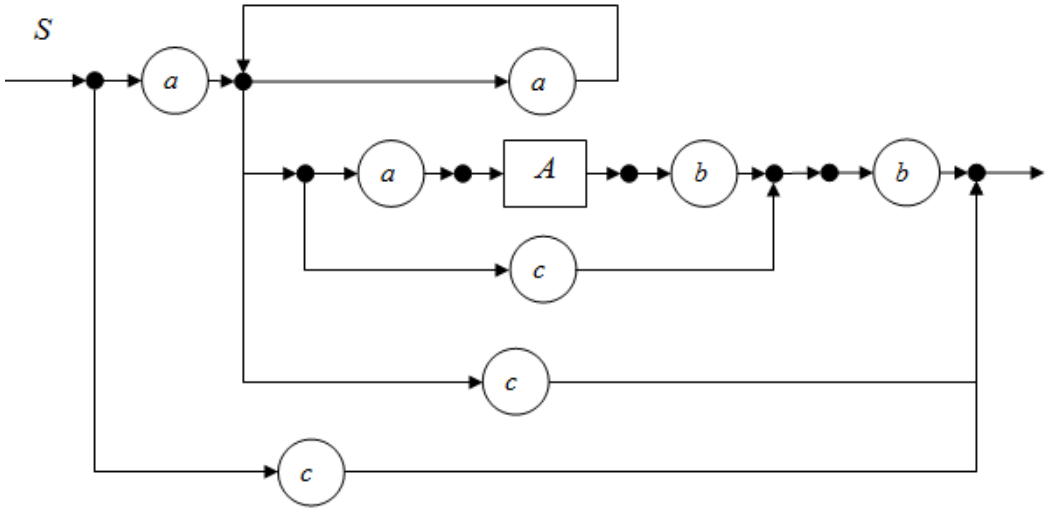

Рис. 9. Компонента S

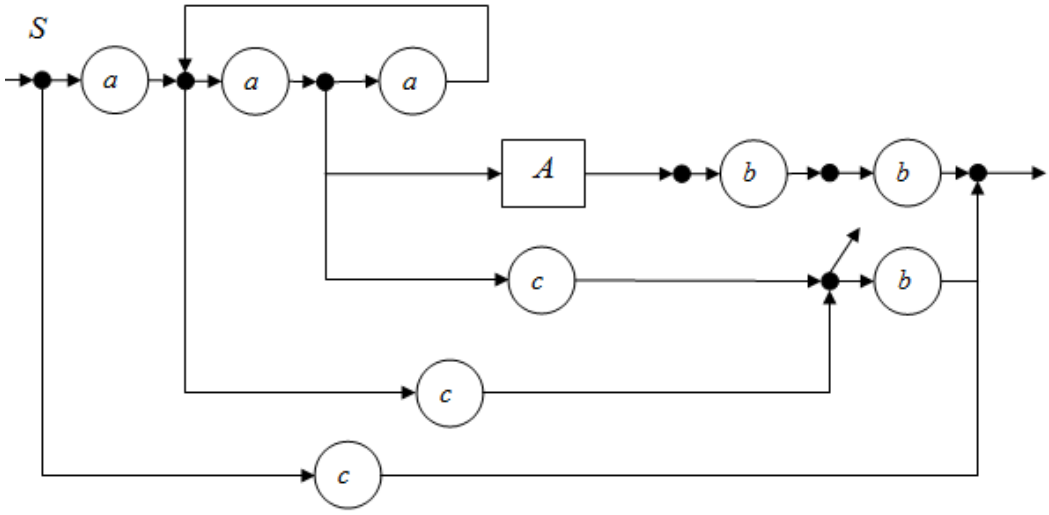

Рис. 10. Псевдодетерминированная компонента S

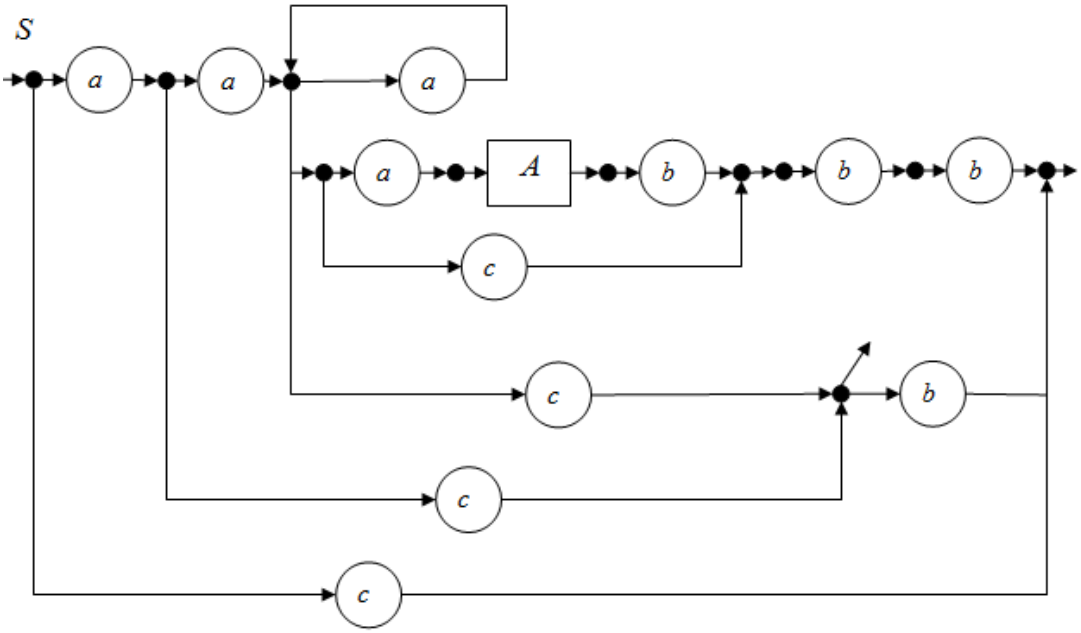

Рис. 11. Компонента S

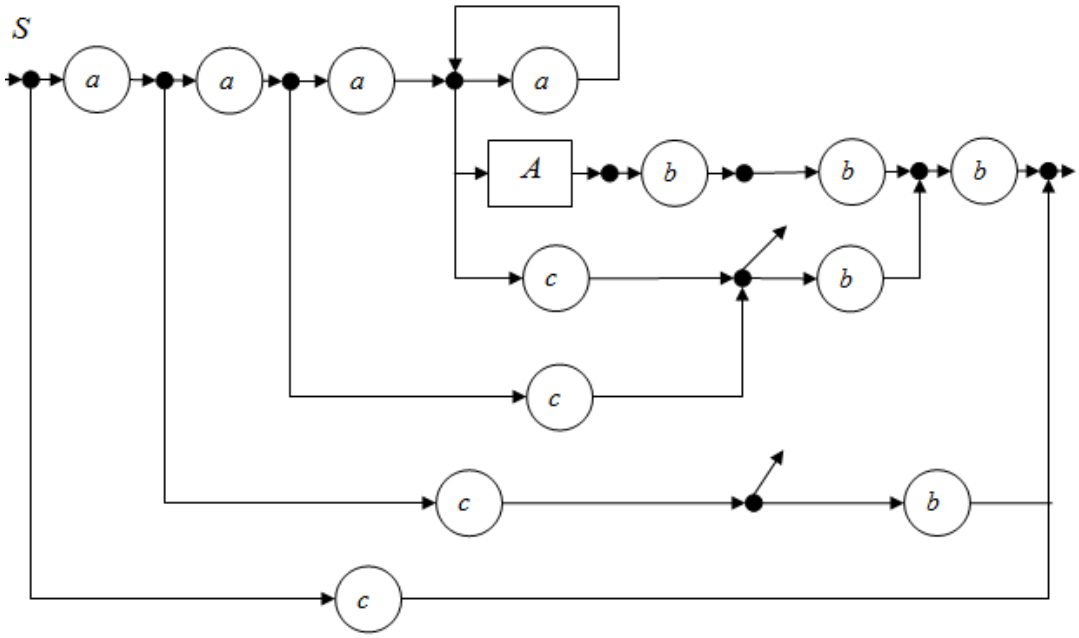

Рис. 12. Псевдодетерминированная компонента S

щей компонентой и получаем компоненту S, представленную на рис. 7.

После преобразования ее в псевдодетерминированную, получим компоненту S, изображенную на рис. 8.

Увеличиваем счетчик на единицу $(c = 1)$. В этой компоненте есть узел с конфликтом и характеристикой $\chi = \{a, c, A\}$, значение счетчика этой характеристики меньше двух, поэтому вершину с нетерминалом A заменяем соответствующей компонентой и получаем компоненту S, представленную на рис. 9.

После преобразования ее в псевдодетерминированную, получим компоненту S, изображенную на рис. 10.

Увеличиваем счетчик на единицу $(c = 2)$. В этой компоненте есть узел с конфликтом и характеристикой $\chi = \{a, c, A\}$, значение счетчика этой характеристики не больше двух (количества узлов в исходной компоненте S), поэтому вершину с нетерминалом A заменяем соответствующей компонентой и получаем компоненту S, представленную на рис. 11.

После преобразования ее в псевдодетерминированную, получим компоненту S, изображенную на рис. 12.

Увеличиваем счетчик на единицу $(c = 3)$. В этой компоненте есть узел с конфликтом и характеристикой $\chi = \{a, c, A\}$, значение счетчика этой характеристики превысило количество узлов в исходной компоненте S, поэтому попытку устранить конфликт считаем неудачной.

ЗАКЛЮЧЕНИЕ

В работе определены конфликты типа «переход-переход» и «переход-выход», из-за которых СД не являются детерминированными. Предложен способ устранения конфликтов типа «переход-переход» и алгоритм, который либо устраняет эти конфликты, либо останавливается с сообщением о невозможности устранить конфликты предложенным способом. Приведен пример СД, в которой конфликт «переход-переход» устраняется предложенным способом, и пример СД, в которой конфликт не может быть устранен. Определение невозможности устранения конфликта типа «переход-переход» представляет собой длительный процесс эквивалентных преобразований, увеличивающий на каждом шаге размер СД и, следовательно, сложность преобразований. Дальнейшим развитием работы может быть поиск более рациональных, более «ранних» условий обнаружения невозможности устранения конфликтов типа «переход-переход» и поиск способов и алгоритмов устранения конфликтов типа «переход-выход». Это позволит расширить класс

СД, которые можно использовать при проектировании эффективных обработчиков формальных языков.

СПИСОК ЛИТЕРАТУРЫ

1. *Йенсен К., Вирт Н.* Паскаль. Руководство для пользователя и описание языка. – М. : Финансы и статистика, 1982.

2. *Легалов А. И., Швец Д. А., Легалов И. А.* Формальные языки и трансляторы. – Красноярск : Сибирский федеральный университет, 2007.

3. *Свердлов С. З.* Языки программирования и методы трансляции. – СПб. : Питер, 2007.

4. *Карпов Ю. Г.* Теория и технология программирования. Основы построения трансляторов. – СПб. : БХВ-Петербург, 2005.

5. *Рязанов Ю. Д.* Анализ синтаксических диаграмм и синтез программ-распознавателей линейной сложности / Ю. Д. Рязанов, М. Н. Севальнева // Научные ведомости БелГУ. Сер. История. Политология. Экономика. Информатика. – 2013. № 8 (151). – Вып. 26/1. – С. 128–136.

6. *Рязанов Ю. Д.* Графовый способ анализа синтаксических диаграмм / Ю. Д. Рязанов, П. В. Крамаренко // Научный электронный архив. URL: http://econf.rae.ru/article/8214 (дата обращения: 20.11.2014)

7. *Рязанов Ю. Д.* Синтез распознавателей с магазинной памятью по детерминированным синтаксическим диаграммам / Ю. Д. Рязанов // Вестник Воронежского ун-та. Серия: Системный анализ и информационные технологии. – 2014. – № 1. – С. 138–145.

8. *Поляков В. М.* Алгоритм построения нерекурсивных программ-распознавателей линейной сложности по детерминированным синтаксическим диаграммам / В. М. Поляков, Ю. Д. Рязанов // Вестник БГТУ им. В.Г. Шухова, № 6, 2013. – С. 194–199.

9. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. – М. : Мир, 1978.

10. *Рязанов Ю. Д.* Псевдодетерминированные синтаксические диаграммы / Ю.Д. Рязанов, М.Н. Севальнева // Прикладная математика, управление и информатика: сборник трудов Междунар. молодеж. конф., Белгород, 3–5 октября 2012 г. : в 2 т., Белгород : ИД «Белгород», 2012. – Т 2. – С. 546–553.

Рязанов Юрий Дмитриевич – доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем, Белгородский государственный технологический университет им. В. Г. Шухова.
Тел.: +7 (910) 325-73-75,
E-mail: Ryazanov.iurij@yandex.ru.

Ryazanov Yury Dmitrievich – associate professor of the department of software computer and automated systems, BSTU after V.G. Shukhov.
Tel.: +7 (910) 325-73-75,
E-mail: Ryazanov.iurij@yandex.ru.