

СПОСОБ УСТРАНЕНИЯ КОНФЛИКТОВ ТИПА «ПЕРЕХОД-ВЫХОД» В СИНТАКСИЧЕСКИХ ДИАГРАММАХ

Ю. Д. Рязанов

Белгородский государственный технологический университет им. В. Г. Шухова

Поступила в редакцию 18.09.2015 г.

Аннотация. Рассматривается задача эквивалентного преобразования синтаксических диаграмм в детерминированные синтаксические диаграммы. Предложен способ и алгоритм устранения конфликтов типа «переход-выход». Это позволит расширить класс синтаксических диаграмм, которые можно преобразовать в детерминированные синтаксические диаграммы.

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 suggests a method and algorithm of resolving the «transition-exit» type of conflicts. This method expanded the class of syntax diagrams, which can be transformed to deterministic syntax diagrams.

Keywords: formal language, syntax diagram, equivalent transformations, deterministic syntax diagram.

ВВЕДЕНИЕ

Синтаксические диаграммы представляют собой наглядный, интуитивно понятный, графический способ задания синтаксиса языка, используется для документирования языков программирования [1, 2] и в проектировании трансляторов [3–7]. Для построения трансляторов линейной сложности [8, 9] используются детерминированные синтаксические диаграммы (ДСД). 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 кото-

рой показана в [8]). В работе [10] определены причины (конфликты типа «переход-переход» и «переход-выход»), по которым СД не являются детерминированными и описан способ разрешения конфликтов типа «переход-переход».

В этой статье предложен способ разрешения конфликтов типа «переход-выход». Это позволит расширить класс исходных СД, которые можно преобразовать в детерминированные синтаксические диаграммы.

ОСНОВНЫЕ ПОНЯТИЯ

На рис. 1 приведен пример синтаксической диаграммы, которая представляет собой ориентированный несвязный граф. В ней главными буквами обозначены нетерминалы, прописными – терминалы, числами – узлы.

Каждому нетерминалу соответствует связанная 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 компо-

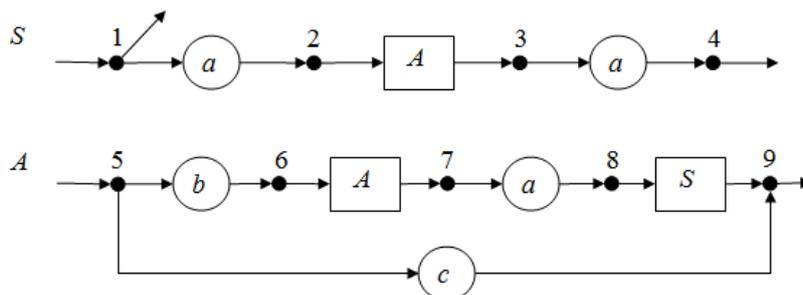


Рис. 1. Синтаксическая диаграмма

ненты 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 выходны дуги, называются заключительными.

СД называется псевдодетерминированной (ПСД) [11], если все ее компоненты псевдодетерминированные. Компонента СД псевдодетерминированная, если:

- 1) содержит только один начальный узел;
- 2) не имеет дуг, соединяющих узлы;
- 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 [11].

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, 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 ($\text{ВЫБОР}(e)$). 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 [8, 12] описаны алгоритмы вычисления множеств выбора дуг, выходящих из узлов, и определения принадлежности СД классу ДСД.

Если узел u заключительный и множества выбора выходной дуги ($u, \text{выход}$) и дуги (u, v) пересекаются ($\text{ВЫБОР}(u, \text{выход}) \cap \text{ВЫБОР}(u, v) \neq \emptyset$) и терминал $x \in \text{ВЫБОР}(u, \text{выход}) \cap \text{ВЫБОР}(u, v)$, то будем говорить, что в узле u имеет место конфликт типа «переход-выход» по терминалу x [10]. В ДСД конфликты типа «переход-выход» отсутствуют. В данной работе предлагается способ устранения конфликтов типа «переход-выход» и алгоритм, который либо устраняет конфликты, либо останавливается с сообщением о невозможности устранить конфликты этим способом. Устранение конфликтов типа «переход-выход» является одним из этапов преобразования недетерминированной СД в ДСД.

УСТРАНЕНИЕ КОНФЛИКТОВ ТИПА «ПЕРЕХОД-ВЫХОД»

Устранение конфликтов «переход-выход» выполняется в ПСД, в которой отсутствуют конфликты «переход-переход» [10]. Предлагаемый способ устранения конфликтов «переход-выход» заключается в следующем:

1. Определить множество M терминалов, по которым имеет место конфликт «переход – выход». Если $M = \emptyset$, то конфликтов «переход-выход» нет.

2. Последовательно выполнять попытки устранения конфликтов по терминалам множества M . Если $x \in M$ и попытка устранения конфликта по терминалу x уже выполнялась, то попытку считать неудачной. Если все попытки удачные, выполнить п. 1, иначе исключить все конфликты «переход-выход» данным способом нельзя.

Попытка устранения конфликта по терминалу x выполняется следующим образом:

1. Сформировать множество узлов M^U с конфликтами по терминалу x .

2. Сформировать множество нетерминалов M^N следующим образом:

2.1. $M^N := \emptyset$.

2.2. Для каждого узла $u \in M^U$ выполнить:

2.2.1. Если узел u принадлежит компоненте A , то $M^A := \{A\}$.

2.2.2. Если в СД есть компонента X , в которой дуга из вершины с нетерминалом B идет в заключительный узел и $B \in M^A$, то $M^A := M^A \cup \{X\}$, т. е. нетерминал X добавить в множество M^A .

2.2.3. Повторять п. 2.2.2, пока множество M^A растет.

2.2.4. $M^N := M^N \cup M^A$.

Достаточным условием конфликта «переход-выход» по терминалу x является наличие в СД узла u , в который входит дуга из вершины с нетерминалом X ($X \in M^N$) и выходит дуга e , такая, что $x \in \text{ВЫБОР}(e)$. Частным случаем условия конфликта является фрагмент СД, представленный на рис. 2. Предлагаемый способ разрешения конфликта по терминалу x заключается в устранении в СД таких фрагментов.

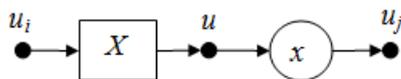


Рис. 2. Фрагмент синтаксической диаграммы

3. Для каждого фрагмента СД (рис. 2), такого, что компонента X не содержит достаточного условия конфликта «переход-выход» по терминалу x , выполнить:

3.1. Заменить вершину с нетерминалом X компонентой X .

3.2. Преобразовать СД в ПСД, используя алгоритм, описанный в работе [9]. Если в результате преобразований возникнут конфликты типа «переход-переход», то попытаться устранить их, например, используя алгоритм, описанный в [10]. Если попытка окажется неудачной, то неудачной считать и попытку устранения конфликта «переход-выход».

4. Если в СД есть фрагмент (рис. 2) и из узла u выходит более одной дуги, то попытку разрешения конфликтов считать неудачной, конец алгоритма. Для каждого фрагмента (рис. 2), в котором в узел u входит более одной дуги, выполнить:

4.1. В СД добавить узел u' , все дуги, входящие в узел u , за исключением дуги, выходящей из вершины с нетерминалом X , перенаправить в u' .

4.2. В СД добавить вершину v с терминалом x и добавить дуги (u', v) и (v, u_j) . В результате в узел u будет входить только одна дуга.

5. Для каждого фрагмента СД (рис. 2), в котором в узел u входит только одна дуга и выходит только одна дуга, выполнить:

5.1. Заменить фрагмент (рис. 2) на фрагмент (рис. 3), где X' – новый нетерминал.

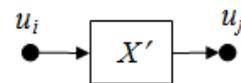


Рис. 3. Фрагмент синтаксической диаграммы

5.2. Построить компоненту X' следующим образом:

5.2.1. Копию компоненты X назвать X' .

5.2.2. Добавить новый узел u' .

5.2.3. Для каждого заключительного узла u компоненты X' выполнить: добавить вершину v с терминалом x и дуги (u, v) и (v, u') .

5.2.4. Заключительным узлом компоненты X' считать только узел u' .

5.3. Компоненту X' преобразовать в псевдодетерминированную. Если в результате преобразований возникнут конфликты типа «переход-переход», то попытаться устранить их, например, используя алгоритм, описан-

ный в [10]. Если попытка окажется неудачной, то неудачной считать и попытку устранения конфликта «переход-выход».

6. Если в результате выполнения п.п. 1–5 будет получена СД без конфликтов «переход-выход» по терминалу x , то попытку устранения конфликта по терминалу x считать удачной, иначе – неудачной.

Рассмотрим пример устранения конфликтов «переход-выход». Пусть дана ПСД без конфликтов «переход-переход» (рис. 1).

В этой СД имеет место конфликт «переход-выход» в узле 1 по терминалу a . Множество $M^N = \{S, A\}$, так как узел 1 принадлежит компоненте S , а в компоненте A из вершины с нетерминалом S идет дуга в заключительный узел 4. Компоненты S и A содержат фрагменты вокруг узлов 3 и 7, которые являются достаточными условиями для конфликта «переход-выход» по терминалу a . Пункт 3 алгоритма не выполняем, так как компонента

A содержит достаточное условие для конфликта. Пункт 4 не выполняем, так как в узлы 3 и 7 входит по одной дуге и из этих узлов по одной дуге выходит. После выполнения пункта 5 получим СД, представленную на рис. 4.

В этой СД, как видно, компонента A стала лишней (в СД нет вершины с нетерминалом A) и ее исключим из СД. В компоненте A' вокруг узла 11 имеется фрагмент, который является достаточным условием конфликта. Компонента S не содержит фрагмента, являющегося достаточным условием конфликта, поэтому вершину с нетерминалом S в компоненте A' заменяем на компоненту S и преобразуем компоненту A' в псевдодетерминированную. В результате получим детерминированную СД (рис. 5).

Рассмотрим еще один пример. Пусть дана ПСД без конфликтов «переход-переход» (рис. 6).

В этой СД имеет место конфликт «переход-выход» в узле 7 по терминалу a . Множе-

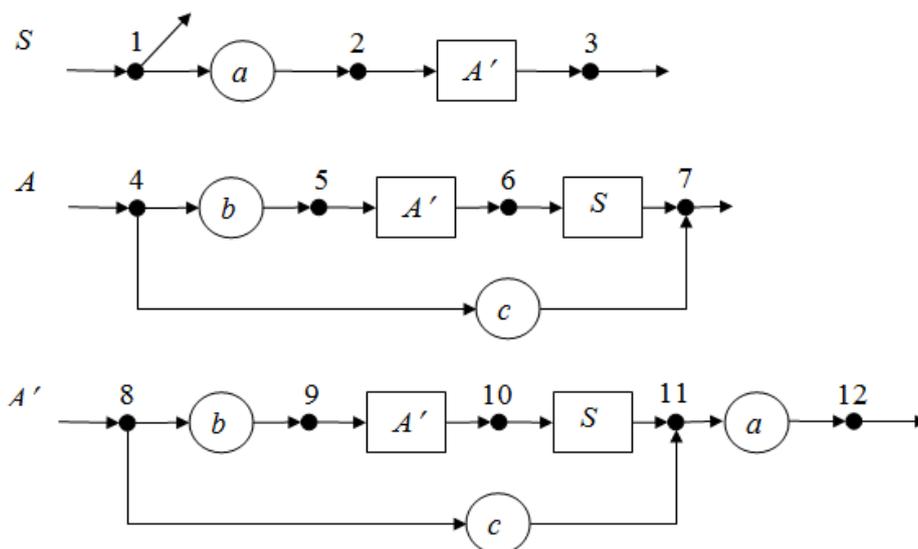


Рис. 4. Синтаксическая диаграмма после выполнения пункта 5 алгоритма

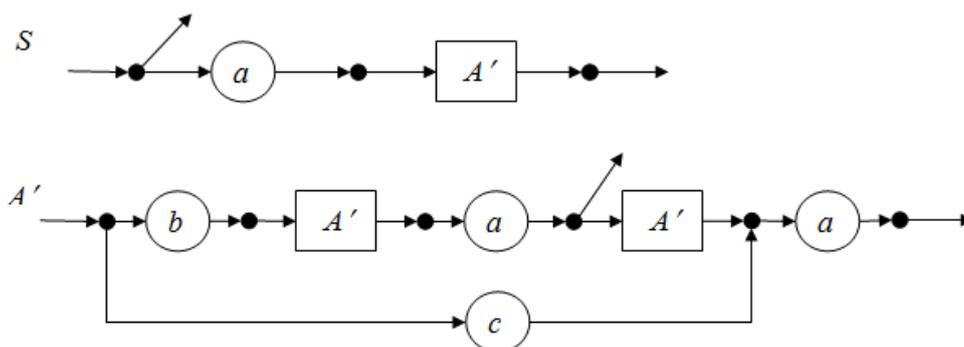


Рис. 5. Детерминированная синтаксическая диаграмма

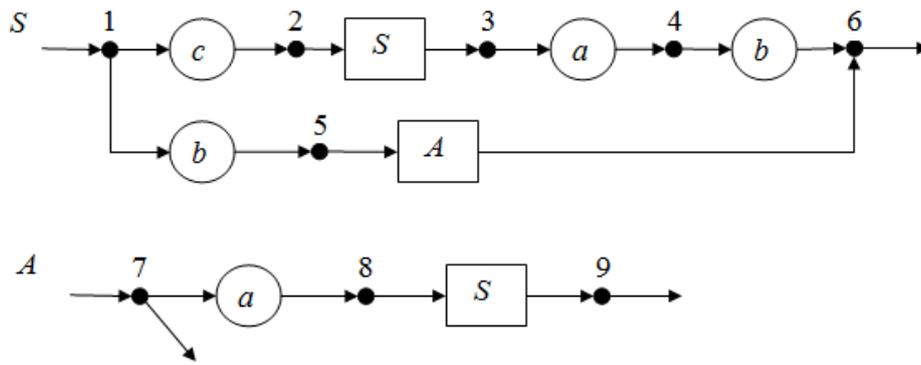


Рис. 6. Синтаксическая диаграмма

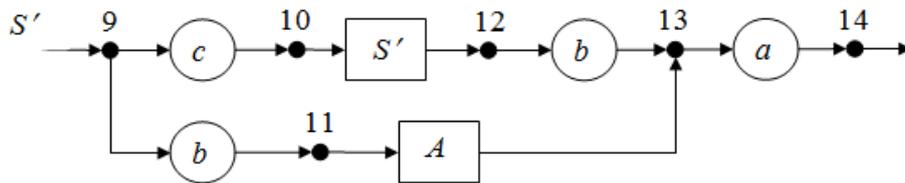
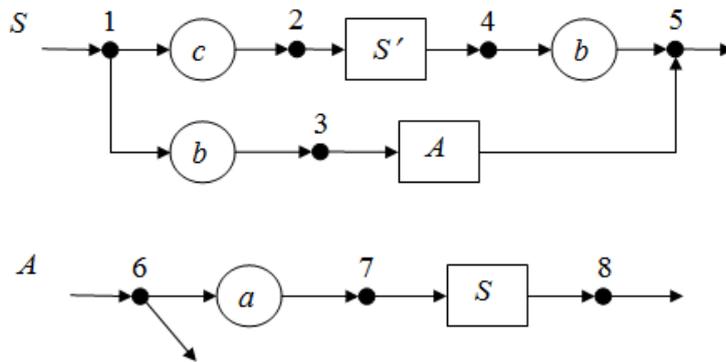


Рис. 7. Синтаксическая диаграмма после выполнения пункта 5 алгоритма

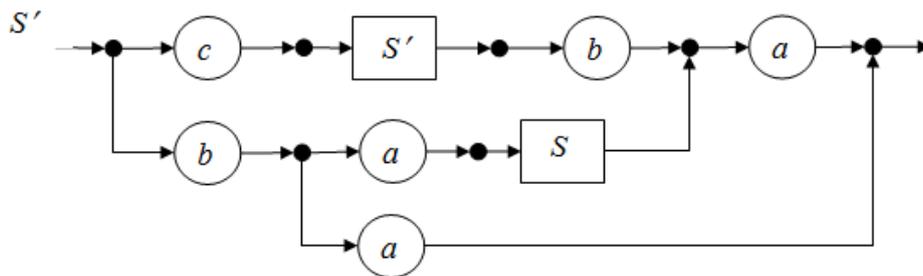


Рис. 8. Компонента S'

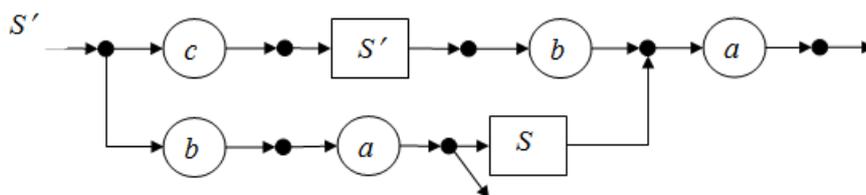


Рис. 9. Псевдодетерминированная компонента S'

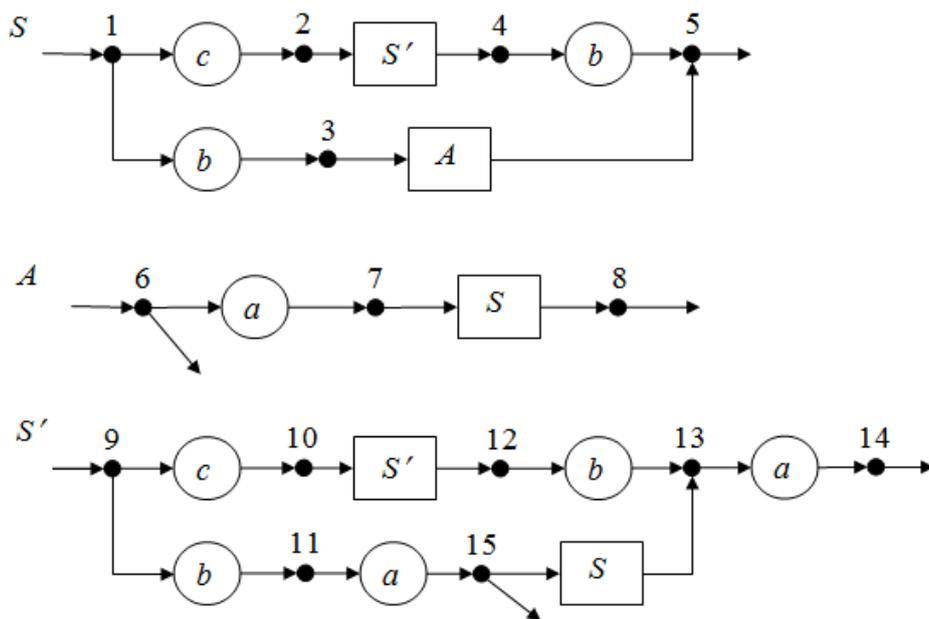


Рис. 10. Псевдодетерминированная синтаксическая диаграмма

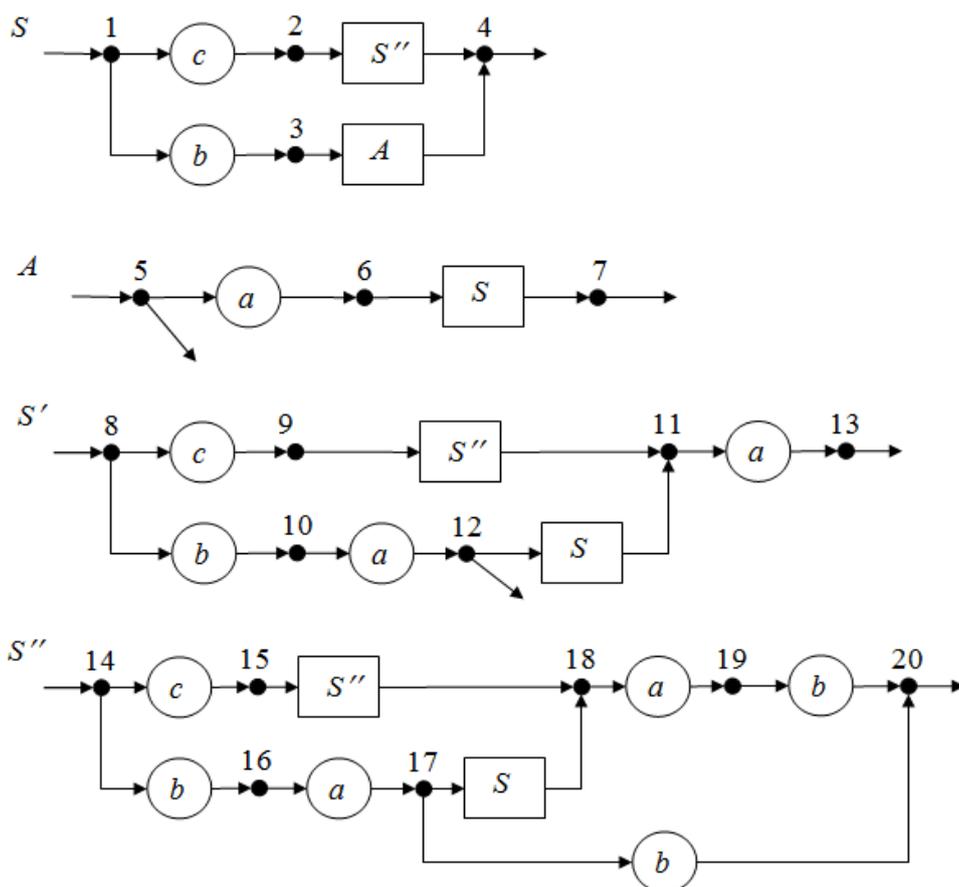
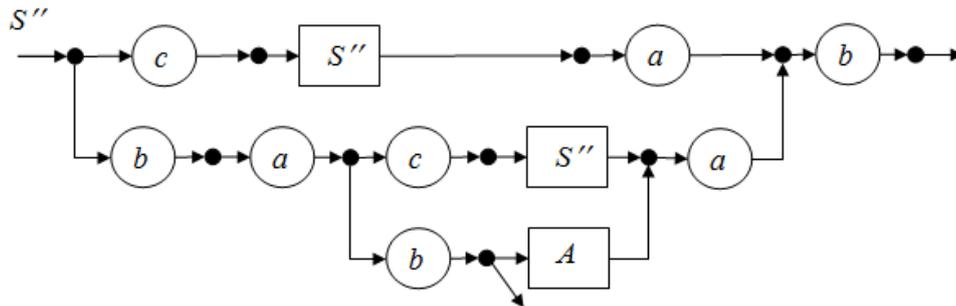


Рис. 11. Синтаксическая диаграмма после выполнения пункта 5 алгоритма

ство $M^N = \{A, S\}$, так как узел 7 принадлежит компоненте A , а в компоненте S из вершины с нетерминалом A идет дуга в заключительный узел 6. В компоненте S вокруг узла 3 имеется фрагмент, который является доста-

точным условием конфликта «переход-выход» по терминалу a . Пункт 3 алгоритма не выполняем, так как компонента S содержит достаточное условие для конфликта. Пункт 4 не выполняем, так как в узел 3 входит одна

Рис. 12. Псевдодетерминированная компонента S''

дуга и одна дуга выходит. Выполнение пункта 5 даст СД, представленную на рис. 7.

В этой СД в компоненте S' вокруг узла 13 имеется фрагмент, который является достаточным условием конфликта «переход-выход» по терминалу a . Компонента A не содержит фрагмента, который является достаточным условием конфликта, поэтому в компоненте S' заменим вершину с нетерминалом A на компоненту A и полученную компоненту S' (рис. 8) преобразуем в псевдодетерминированную (рис. 9).

В результате получим ПСД (рис. 10), в которой нет конфликтов «переход-выход» по терминалу a , но есть конфликт «переход-выход» в узле 15 по терминалу b .

Для устранения конфликта по терминалу b сформируем множество $M^N = \{S'\}$ (компонента S' содержит узел с конфликтом). Компоненты S и S' содержат фрагменты вокруг узлов 4 и 12, являющиеся достаточным условием конфликта. В результате выполнения пункта 5 получим ПСД, представленную на рис. 11.

В этой СД в узле 17 имеет место конфликт «переход-переход». Для его устранения вершину с нетерминалом S в компоненте S'' заменим компонентой S и компоненту S'' преобразуем в псевдодетерминированную (рис. 12).

В полученной СД возник конфликт «переход-выход» по терминалу a в компоненте A . Так как конфликт по терминалу a уже устраняли, попытку устранения конфликта считаем неудачной и преобразовать СД (рис. 6) в детерминированную предложенным способом нельзя.

ЗАКЛЮЧЕНИЕ

В работе предложен способ устранения конфликтов типа «переход-выход» и алгоритм, который либо устраняет эти конфликты, либо останавливается с сообщением о невозможности устранить конфликты предложенным способом. Приведен пример СД, в которой конфликт «переход-выход» устраняется предложенным способом, и пример СД, в которой конфликт не может быть устранен. Устранение конфликтов типа «переход-выход» используется для преобразования недетерминированных СД в детерминированные. Предложенный способ позволяет расширить класс СД, которые можно преобразовать в детерминированные и использовать при проектировании эффективных обработчиков формальных языков.

СПИСОК ЛИТЕРАТУРЫ

1. Jensen K., Wirth N. Pascal User Manual and Report, Springer-Verlag, New York, 1975. – p. 167.
2. Jensen K., Wirth N. Pascal Standard Iso – Jackson Libri, 1996. – p. 290.
3. Легалов А. И. Основы разработки трансляторов. URL: <http://www.softcraft.ru/translat> (дата обращения 22.04.2015).
4. Легалов А. И., Швеиц Д. А., Легалов И. А. Формальные языки и трансляторы. – Красноярск : Сибирский федеральный университет, 2007. – 213 с.
5. Карпов Ю. Г. Теория и технология программирования. Основы построения трансляторов. – СПб. : БХВ-Петербург, 2005. – 272 с.

6. Свердлов С. З. Языки программирования и методы трансляции. СПб. : Питер, 2007. – 638 с.
7. Мартыненко Б. К. Синтаксические диаграммы Н. Вирта и граф-схемы в Syntax-технологии // Компьютерные инструменты в образовании. – 2014, № 2, С. 3–19.
8. Рязанов Ю. Д., Севальнева М. Н. Анализ синтаксических диаграмм и синтез программ-распознавателей линейной сложности // Научные ведомости БелГУ. Сер. История. Политология. Экономика. Информатика. – 2013, № 8 (151). – Вып. 26/1. – С. 128–136.
9. Поляков В. М., Рязанов Ю. Д. Алгоритм построения нерекурсивных программ-распознавателей линейной сложности по детерминированным синтаксическим диаграммам // Вестник БГТУ им. В. Г. Шухова. – 2013, № 6. – С. 194–199.
10. Рязанов Ю. Д. Преобразование недетерминированных синтаксических диаграмм в детерминированные // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2015. – № 1. – С. 139–147.
11. Рязанов Ю. Д., Севальнева М. Н. Псевдодетерминированные синтаксические диаграммы // 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.
12. Рязанов Ю. Д., Крамаренко П. В. Графовый способ анализа синтаксических диаграмм // Научный электронный архив. URL: <http://econfr.ae.ru/article/8214> (дата обращения: 20.11.2014)

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.