УДК 004.55

ПОСТРОЕНИЕ, ОПТИМИЗАЦИЯ И МОДИФИКАЦИЯ ПРОЦЕССОВ

В.В. Девятков

МГТУ им. Н.Э. Баумана, Москва
e-mail: deviatkov@iu3.bmstu.ru

Предложены методы и алгоритмы построения, оптимизации и модификации последовательных детерминированных процессов на основе анализа детерминированности переходов непосредственно по процессным выражениям.

Ключевые слова: процесс, процессные выражения, построение, оптимизация, модификация.

CONSTRUCTION, OPTIMIZATION, AND MODIFICATION OF PROCESSES

V.V. Devyatkov

Bauman Moscow State Technical University, Moscow
e-mail: deviatkov@iu3.bmstu.ru

Methods and algorithms are proposed for construction, optimization, and modification of consecutive deterministic processes based on the analysis of determinacy of transfers directly from process expressions.

Keywords: process, process expressions, construction, optimization, modification.

Построение, оптимизация и модификация процессов в современном информационном мире в силу широты их использования в различных областях человеческой деятельности по-прежнему является актуальной задачей, над решением которой продолжают работать ученые всего мира. Для этого используются различные модели и языки описания процессов. В настоящей работе развивается подход к решению этой задачи, базирующийся на ранних работах автора [1], опубликованных в конце 1970-х гг. на русском языке, и поэтому не нашедших отклика в современных англоязычных источниках, посвященных, по существу, той же тематике. В частности, для описания и последующего применения процессов популярными в настоящее время являются различные модификации так называемого пи-исчисления [2], многие проблемы внутри которого были решены в упоминаемой работе автора [1] еще до появления этого исчисления. Важнейшими проблемами при использовании процессных моделей являются построение, модификация и оптимизация процессов, рассматриваемые в настоящей работе с новых позиций, как симбиоз ранних и последних достижений в этой области, сложившейся к настоящему времени терминологии и новых веяний. Построение или описание процессов – это трудноформализуемая процедура, поскольку в значительной мере зависит от квалификации, способностей и умения человека. Однако в любом случае человеку надо дать возможность делать это как можно проще и естественнее и предоставить для этого алгоритмы.

Раздел 1 настоящей статьи содержит основные необходимые процессные понятия. При небольших размерностях решаемых задач удобно пользоваться совсем простым языком описания процесса. Чем проще язык, тем, как правило, проще понимать и осуществлять преобразования в этом языке и легче излагать сущность этих преобразований. Одним из главных преобразований являются эквивалентные преобразования, позволяющие перейти от более простого в определенном смысле описания процесса к более сложному, но более компактному и эквивалентному. Изложению сущности этих преобразований посвящен раздел 2 статьи. Оптимизации процессов посвящен раздел 3. Получив исходное описание процесса на том или ином языке, часто требуется его модифицировать или изменить. Этому вопросу посвящен раздел 4 статьи.

1. Процессные выражения. Каждый процесс имеет алфавит восприятий и реакций $Act = \{a_1, a_2, \ldots, a_m\}$. Каждый символ a этого алфавита именует некоторый объект, получаемый (воспринимаемый) процессом из внешней среды (восприятие процесса), выдаваемый процессом во внешнюю среду (внешняя реакция процесса), или объект, используемый процессом для внутренних нужд (внутренняя реакция процесса). Процессы действуют, воспринимая, порождая для внутреннего употребления или выдавая наружу объекты с соответствующими именами. Для того чтобы различать типы действий, будем использовать следующие обозначения: $?a$ — для восприятий, $!a$ — для внешних реакций, ba или просто b – для внутренних реакций. Внутренняя реакция может трактоваться по-разному. Она может быть скрытым процессом, структура которого нас не интересует. И тогда внутренняя реакция – это просто имя или метка места, где соответствующий процесс вызывается. Это имя места можно рассматривать также как задержку на выполнение некоторого скрытого процесса или состояние процесса, которому внутренняя реакция принадлежит и в которое процесс перешел после достижения этого места. В зависимости от уровня абстракции бывает целесообразно вообще не употреблять внутренних реакций или, наоборот, вводить внутреннюю реакцию в определенном месте.

Нитью a^* будем называть кортеж (конечный или бесконечный) действий: $a^* = a_0 a_1 a_2 \ldots a_{m-2} a_{m-1}$. Выполнением нити называется последовательность выполнения действий в порядке их записи в нити слева направо, т.е. осуществление в порядке слева направо по порядку восприятия или реакции. Символом e обозначим пустое действие. Нить, состоящая из единственного пустого действия, называется ***пустой нитью***.

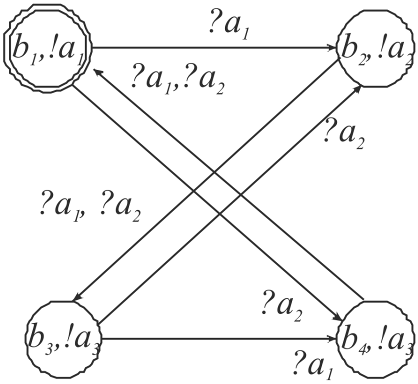

Рис. 1. Граф переходов процесса P

Процессом P называется множество нитей S, которые он может выполнять. Поведением процесса P называется порядок выполнения множества этих нитей.

Популярным языком представления процессных выражений является язык графов переходов. Для построения графа переходов процесса считается, что после каждого его восприятия, в том числе пустого, процесс осуществляет внутреннюю реакцию или, как говорят, переходит во внутреннее состояние ожидания следующего восприятия. Находясь в этом внутреннем состоянии, процесс может порождать внешнюю реакцию, в том числе пустую. В графе переходов каждое состояние процесса изображается кружочком, внутрь которого помещается символ этого состояния. Каждому восприятию соответствует стрелка, соединяющая состояния этого перехода. Начальное состояние выделяется двойным кружком. На рис. 1 показан граф переходов некоторого процесса P.

На рис. 2 приведен пример процесса, заданного процессными выражениями

$$P \triangleq ?e!b_1!a_1(P^1\,|P^2)\,,$$
$$P^1 \triangleq ?a_1b_2!a_2(?a_1b_3!a_3\,|?a_2b_3!a_3)\;(?a_1b_4!a_3\,|?a_2b_2!a_2)\,,$$
$$P^2 \triangleq ?a_2b_4!a_3(?a_1b_1!a_1\,|?a_2b_1!a_1)\,.$$

Граф переходов процесса позволяет компактно описывать небольшое множество нитей, в том числе бесконечных. В этих условиях естественным кажется описывать процесс непосредственно его графом переходов. Однако при большой размерности графа такое описание становится громоздким и ненаглядным. Альтернативой этому

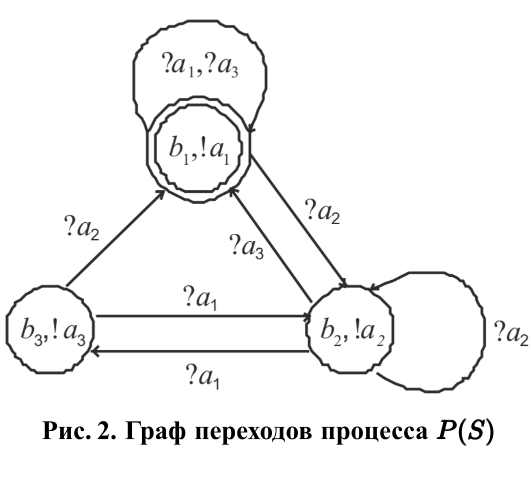

Рис. 2. Граф переходов процесса $P(S)$

является использование адекватных графу процессных выражений, позволяющих легко переходить от графа к этим выражениям и наоборот. Воспользуемся для этого известным из теории конечных автоматов фактом, заключающимся в том, что каждый граф переходов задает две функции: рекурсивную функцию переходов $b_j = f(b_i, ?a)$ и функцию выходов $!a = \phi(b_j)$. Обе эти функции могут быть заданы в виде канонических процессных выражений, адекватных графу переходов процесса. Канонические процессные выражения, задающие функцию переходов, будем называть внутренними, а функцию выхода — внешними.

Так для графа переходов (см. рис. 1) каноническими процессными выражениями, адекватными этому графу, будут следующие:

$$
\begin{aligned}
&b_1 = ?e,\\
&b_1 = b_4?a_1,\\
&b_1 = b_4?a_2,\\
&b_2 = b_1?a_1,\\
&b_2 = b_3?a_2,\\
&b_3 = b_2?a_1,\\
&b_3 = b_2?a_2,\\
&b_4 = b_1?a_2,\\
&b_4 = b_3?a_1,\\
&!a_1 = b_1,\\
&!a_2 = b_2,\\
&!a_3 = b_3 \,|b_4\ .
\end{aligned}
$$

В более компактном виде эти канонические процессные выражения могут быть представлены следующим образом:

$$
\begin{aligned}
&b_1 = ?e\,|b_4?a_1\,|b_4?a_2 = ?e\,|b_4\,|(a_1\,|a_2)\ ,\\
&b_2 = b_1?a_1\,|b_3?a_2\ ,\\
&b_3 = b_2?a_1\,|b_2?a_2 = b_2(?a_1\,|?a_2),\\
&b_4 = b_1?a_2\,|b_3?a_1\ ,\\
&!a_1 = b_1,\\
&!a_2 = b_2,\\
&!a_3 = b_3\,|b_4\ .
\end{aligned}
$$

Очевидно, что если исходным описанием процесса являются канонические процессные выражения, то переход от этих выражений к графу переходов процесса и наоборот очевиден. Канонические процессные выражения могут быть рекурсивными. В исходном представлении каждое внутреннее процессное выражение вида $b_i = b_j?a$ задает один переход из состояния b_j в состояние b_i в результате восприятия $?a$.

2. Построение графа переходов процесса по множеству нитей. Обозначим $?A$ множество всех восприятий некоторого процесса P, включая пустое восприятие $?e$; $!A$ — конечное множество всех внешних реакций процесса P, включая пустую внешнюю реакцию $!e$.

Обозначим S множество всех нитей, выполняемых процессом P таких, что $S \subseteq ?A^* \times !A$, где $?A^*$ — множество всех нитей $?a^*$ в алфавите $?A$, $?a^*$ — нить, состоящая только из восприятий (нить восприятий) процесса P, φ^* — функция на множестве $?A^*$, которая ставит в соответствие каждой нити $?a^* \in ?A^*$ внешнюю реакцию из множества $!A$. Процесс P, выполняющий множество нитей $?a^*\varphi^*(?a^*) \in S$, будем обозначать как $P(S)$. Значение функции φ^* на тех нитях множества $?A^*$, на которых эта функция не определена и на тех нитях, которые этому множеству не принадлежат, считается равным $!e$. Множество нитей $?a^*$ таких, что $\{?a^* \in ?A^* | ?a^*\varphi^*(?a^*) \in S\}$, будем обозначать $?S$.

Будем считать, что множество $?S$удовлетворяет следующим условиям.

Условие полноты. Для любой нити $?a^* \in ?S$ всякое ее начало также принадлежит $?S$.

Условие непротиворечивости. Не существует ни одной пары нитей $?a_1^*,\ ?a_2^* \in ?S$ таких, что $?a_1^* = ?a_2^*,\ \varphi^*(?a_1^*) \neq \varphi^*(?a_2^*)$.

Введем на множестве $?A^*$ отношение R, определяемое следующим образом: $?a_1^* R ?a_2^*$, если $\varphi^*(?a_1^* ?a^*) = \varphi^*(?a_2^* ?a^*)$ для всех $?a^*$ таких, что $a_1^* ?a^* \varphi^*(a_1^* ?a^*) \in S$, $a_1^* ?a^* \varphi^*(a_2^* ?a^*) \in S$.

Отношение R является отношением эквивалентности [1] и разбивает множество $?A^*$ на классы эквивалентности $Q_1, Q_2, \ldots, Q_{k+1}$, где Q_1 — класс эквивалентности отношения R, которому принадлежит восприятие $?e$. То, что отношение R является отношением эквивалентности, объясняется, прежде всего, полной определенностью функции φ^*, в частности за счет введения значения этой функции, равного $!e$, на тех нитях, на которых она не определена.

Множество классов эквивалентности может быть использовано для построения процесса, выполняющего множество нитей S. Это возможно в силу ряда свойств разбиения множества $?A^*$ на классы эквивалентности. Сформулируем эти свойства без доказательства, которое содержится в работе [1].

Свойство однозначной продолжаемости. Если выполняются условия полноты и непротиворечивости, то для любого восприятия $?a^*$ и любого класса такого, что $1 \leq r \leq k+1$, существует единственный класс Q_t такой, что $1 \leq t \leq k+1$ и $Q_r ?a \subseteq Q_t$, где $Q_r ?a$ — множество всех нитей $?a^* ?a$ таких, что нить $?a^* \in Q_r$.

Свойство связности. Если множество $?S$ удовлетворяет условию полноты, то процесс, который выполняет это множество нитей, связный.

Свойство доопределения. Если множество нитей S не совпадает с множеством нитей $?A^* \times !A$, т.е. $S \subset ?A^* \times !A$, то все нити $(?A^* \times \times !A) \backslash P(S)$ можно поместить в единственный класс. Для определенности будем считать, что этот единственный класс нитей есть Q_{k+1}. Из этого свойства также следует, что пересечение класса Q_{k+1} с множеством $?S$ дает пустое множество ($Q_{k+1} \cap ?S = \varnothing$), а все остальные классы $Q_r, 1 \leq r \leq k$, включаются в множество $?S$ ($Q_r \subseteq ?S$).

Построение графа переходов процесса $P(S)$, выполняющего заданное множество нитей S, может быть осуществлено [1] следующим образом. Классы эквивалентности $Q_1, Q_2, \ldots, Q_k, Q_{k+1}$ отношения R взаимно однозначно сопоставляются с внутренними реакциями (внутренними состояниями) $b_1, b_2, \ldots, b_k, b_{k+1}$ процесса $P(S)$. Процесс переходит из одного внутреннего состояния b_i, $1 \leq i \leq k+1$, в другое b_j, $1 \leq j \leq k+1$, в результате восприятия $?a$, если $Q_i?a \subseteq Q_j$.

Начальным состоянием процесса $P(S)$ является состояние b_1, соответствующее классу эквивалентности Q_1, содержащему пустую последовательность $?e$. Вследствие свойства связности граф переходов процесса $P(S)$ будем связным.

Соглашение о том, что значение функции φ^* на тех нитях, на которых эта функция не определена, считается равным $!e$ и называется доопределением функции φ^*. Принятие значения функции φ^* на тех нитях, на которых она не определена, равным $!e$ не означает, что существует внешняя реакция $!e$, а свидетельствует о том, что она на всех нитях, на которых была не определена, после доопределения будет иметь одно и то же значение, равное некоторой условной внешней реакции.

Описанный способ доопределения не единственный. Пока ограничимся рассмотрением еще одного способа доопределения функции φ^*, которым будем пользоваться в примерах.

Если для некоторой пары нитей $?a_1^*$, $?a_2^*$, принадлежащих множеству S, $\varphi^*(?a_1^*?a^*) = \varphi^*(?a_2^*?a^*)$ для всех нитей $?a^*$ таких, что $?a_1^*?a^*$, $?a_2^*?a^* \in ?A^*$, определена только одна из функций $\varphi^*(?a_1^*?a^*)$ и $\varphi^*(?a_2^*?a^*)$, то значение функции $\varphi^*(?a_1^*?a^*)$ для тех $?a_1^*?a^*$, на которых она не определена, принимается равным значению функции $\varphi^*(?a_2^*?a^*)$, если последняя определена, и наоборот. Если обе функции $\varphi^*(?a_1^*?a^*)$ и $\varphi^*(?a_2^*?a^*)$ не определены, то их значение принимается равным $!e$. Из определения отношения R следует, что если такое доопределение на нитях $?a_1^*?a^*$ и $?a_2^*?a^*$ выполнено, то после этого нити $?a_1^*$ и $?a_2^*$ будут находиться в отношении R. Подобное доопределение неоднозначно, так как при наличии даже трех нитей $?a_1^*$, $?a_2^*$, $?a_3^*$, для которых попарно ($?a_1^*$, $?a_2^*$ и $?a_2^*$, $?a_3^*$) возможно доопределение по описанному правилу, может оказаться, что после доопределения функции φ^* на нитях $?a_1^*?a^*$ и $?a_2^*?a^*$ доопределение функции φ^* на нитях

$?a_2^*?a^*$, $?a_3^*?a^*$ становится невозможным из-за появления неравенства $\varphi^*(?a_2^*?a^*) \neq \varphi^*(?a_3^*?a^*)$ (хотя бы для одной нити $?a^*$) в результате доопределения функции φ^* на нити $\varphi^*(?a_2^*?a^*)$.

Пример 1. Задано следующее множество S нитей:

$$S = \{?e!a_1, ?a_1!a_1, ?a_1?a_1!a_1, ?a_1?a_2!a_2, ?a_1?a_3!a_1, ?a_2!a_2,$$
$$?a_2?a_1!a_1,\ ?a_2?a_2!a_1, ?a_2?a_3!a_2, ?a_2?a_1?a_1!a_2, ?a_2?a_1!a_1, ?a_3!a_1\}.$$

$$?S = \{?e, ?a_1, ?a_1?a_1, ?a_1?a_2, ?a_1?a_3, ?a_2, ?a_2?a_1, ?a_2$$
$$?a_2, ?a_2?a_3, ?a_2?a_1?a_1, ?a_2?a_1, ?a_3\}.$$

Множество $?S$ удовлетворяет условиям полноты и непротиворечивости. Поэтому по нему может быть построен граф переходов процесса $P(S)$.

Пользуясь вторым из описанных доопределений функции φ^*, получаем

$$?eR?a_1, ?eR?a_1?a_1, ?eR?a_2?a_3, ?eR?a_2?a_1?a_2, ?eR?a_3,$$

$$Q_1 = \{?e, ?a_1, ?a_1?a_1, ?a_2?a_3, ?a_2?a_1?a_2, ?a_3\},$$

$$?a_2R?a_1?a_2, ?a_2R?a_2?a_2, ?a_2R?a_2?a_1?a_1,$$

$$Q_2 = \{?a_2, ?a_1?a_2, ?a_2?a_2, ?a_2?a_1?a_1\}.$$

В классы эквивалентности Q_1 и Q_2 не вошла только одна нить $?a_2?a_1$, которая образует последний класс $Q_3 = \{?a_2?a_1\}$. Сопоставим классы эквивалентности Q_1, Q_2, Q_3 соответственно с внутренними реакциями b_1, b_2, b_3 некоторого процесса $P(S)$. Поскольку нити $?a_1, ?a_1?a_1$ множества $Q_1?a_1$ принадлежат классу Q_1, то из состояния b_1 в него же ведет дуга, помеченная восприятием $?a_1$, т.е. процесс $P(S)$, находящийся в состоянии b_1, при восприятии $?a_1$ остается в том же состоянии. Остальные нити множества $Q_1?a_1$ не рассматриваются вследствие введенного ранее соглашения о состоянии b_{k+1}. Далее поскольку входные нити $?a_2, ?a_1?a_2$ множества $Q_1?a_2$ принадлежат классу Q_2, то из состояния b_1 в состояние b_2 ведет дуга, помеченная восприятием $?a_2$ и т.д. В результате получаем граф переходов процесса $P(S)$ (см. рис. 2). Начальным состоянием этого процесса является состояние b_1. Обычно тройка $(b_i, ?a, b_j)$ называется переходом.

Нахождение классов эквивалентности $Q_1, Q_2, \ldots, Q_{k+1}$, на которые разбивается множество $?S$ по отношению R и первые k из которых отождествляются соответственно с состояниями $b_1, b_2, \ldots, b_k$ графа переходов процесса $P(S)$, а также дальнейшее построение самого графа переходов процесса $P(S)$ может быть выполнено, как это будет сделано в примере 2, по следующему алгоритму.

Алгоритм 1:

а) принять $i = 0$, объявить множество $S' = ?S$;

б) объявить множество $Q'_i = \varnothing$;

в) если существует хотя бы одна последовательность $?a^* \in ?S$ такая, что $Q'_i R ?a^*$ ($Q'_i R ?a^*$ означает, что для каждой нити $?a^*_1 \in Q'_i$ имеет место $?a^*_1 R ?a^*$, причем если $Q'_i = \varnothing$, то всегда имеет место $Q'_i R ?a^*$), то перейти к следующему пункту. В противном случае перейти к п. "д";

г) считать Q'_i равным $Q'_i \cup ?a^*$, а S' равным $S' \backslash ?a^*$, и перейти к выполнению п. "в" с новыми Q'_i и S';

д) если $S' = \varnothing$, то перейти к пункту е), в противном случае считать $Q_i = Q'_i$, принять i равным $i + 1$ и перейти к п. "б" с новым i;

е) отождествить полученные классы эквивалентности $Q_1, \ldots, Q_k$ с внутренними состояниями $b_1, b_2, \ldots, b_k$ процесса $P(S)$. Для каждого восприятия $?a \in ?A$ и каждого класса эквивалентности Q_i найти, если он существует, класс Q_j такой, что $Q_i ?a \subseteq Q_j$. Провести на графе переходов процесса $P(S)$ дугу (b_i, b_j). Дополнить полученный граф переходов до графа отметкой вершин внешними реакциями. Перейти к следующему пункту;

ж) конец.

3. Оптимизация канонических процессных выражений. Нетрудно показать, что граф переходов процесса $P(S)$, построенный по алгоритму 1 в случае, когда функция $\varphi^*(?a^*)$ определена на всех нитях $?a^* \in ?S$, имеет минимальное число состяний или, как говорят, он минимален и, следовательно, минимальным будет число внутренних и внешних канонических процессных выржений. В противном случае это число оптимально.

Алгоритм 1 имеет существенный недостаток: он требует проверки отношения R для каждой пары нитей множества $?S$. Трудоемкость такой проверки при числе нитей, равном m, и максимальной длине нитей, равной ζ, оценивается $O(m^2\zeta)$.

Эту оценку можно улучшить. Из определения отношения R непосредственно следует, что две нити $?a^*_1$, $?a^*_2$ множества $?S$ тогда и только тогда находятся в отношении R, когда $\varphi^*(?a^*_1) = \varphi^*(?a^*_2)$. Следовательно, если в одно множество включать только те нити множества $?S$, для каждой пары $?a^*_1$, $?a^*_2$ которых имеет место $\varphi^*(?a^*_1) = \varphi^*(?a^*_2)$, то множество $?S$ разобьется на конечное число подмножеств (классов), которые будем обозначать $Q^{(0)}_1, Q^{(0)}_2, \ldots, Q^{(0)}_{k_0}$.

Пример 2. Пусть

$$S = \{?a_1!a_1, ?a_1?a_1!a_2, ?a_1?a_2!a_1, ?a_2?a_1!a_2, ?e!a_3,$$
$$?a_1?a_2?a_1!a_2, ?a_1?a_2?a_2!a_1, ?a_2?a_2?a_1!a_1, ?a_1?a_1?a_1?a_1!a_2,$$
$$?a_1?a_1?a_2?a_1!a_2, ?a_2!a_3, ?a_2?a_2!a_3,$$
$$?a_1?a_1?a_1!a_3, ?a_1?a_1?a_2!a_3\}.$$

Для этого множества будет три класса:

$$Q_1^{(0)} = \{?e, ?a_2, ?a_2?a_2, ?a_1?a_1?a_1, ?a_2?a_1?a_2\},$$
$$Q_2^{(0)} = \{?a_1, ?a_1?a_2, ?a_1?a_2?a_2, ?a_2?a_2?a_1\},$$
$$Q_3^{(0)} = \{?a_1?a_1, ?a_2?a_1, ?a_1?a_2?a_1, ?a_1?a_1?a_1?a_1, ?a_1?a_1?a_2?a_1\}.$$

Сопоставим этим классам соответственно внутренние реакции $b_1^{(0)}, b_2^{(0)}, b_3^{(0)}$ некоторого процесса $P^{(0)}$, построив для него функции перехода и выхода точно так же, как для процесса $P(S)$, но используя классы $Q_1^{(0)}, Q_2^{(0)}, Q_3^{(0)}$. Тогда получим процесс, граф переходов которого показан на рис. 3. Этот процесс является недетерминированным, так как из состояния $b_1^{(0)}$ исходят две дуги, помеченные одним и тем же восприятием $?a_1$ (переход по таким дугам из одного и того же состояния осуществляется недетерминированно). Если с помощью алгоритма 1 для того же множества S построить классы эквивалентности, то их будет четыре:

$$Q_1 = \{?e, ?a_2?a_2\},$$
$$Q_2 = \{?a_1, ?a_1?a_2, ?a_1?a_2?a_2, ?a_2?a_2?a_1\},$$
$$Q_3 = \{?a_1?a_1, ?a_2?a_1, ?a_1?a_2?a_1, ?a_1?a_1?a_1?a_1, ?a_1?a_1?a_2?a_1\},$$
$$Q_4 = \{?a_2, ?a_1?a_1?a_1, ?a_1?a_1?a_2\}.$$

Сравнивая Q_1, Q_2, Q_3, Q_4 и $Q_1^{(0)}, Q_2^{(0)}, Q_3^{(0)}$, видим, что $Q_1^{(0)} = Q_1 \cup Q_4$, $Q_2^{(0)} = Q_2, Q_3^{(0)} = Q_3$, т.е. для того чтобы по классам $Q_i^{(0)}$ построить граф переходов детерминированного процесса $P(S)$ (рис. 4), достаточно разбить множество $Q_i^{(0)}$ на два непересекающихся подмножества Q_1 и Q_4. Эта процедура соответствует замене состояния $b_1^{(0)}$ графа переходов недетерминированного процесса $P^{(0)}$ (см. рис. 3) двумя со-

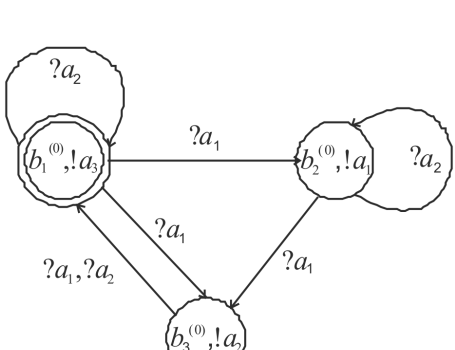

Рис. 3. Граф переходов недетерминированного процесса $P^{(0)}$

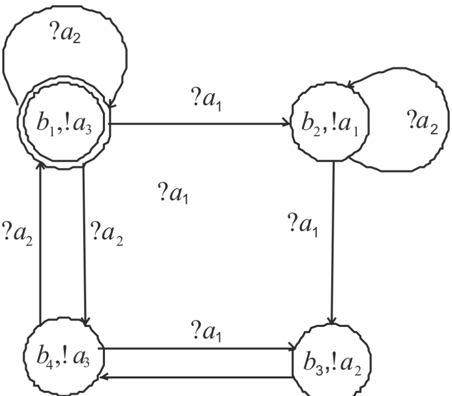

Рис. 4. Граф переходов детерминированного процесса $P^{(1)} = P(S)$

стояниями b_1 и b_4 и превращению его тем самым в детерминированный процесс $P(S)$.

В общем случае под детерминизацией будем понимать итеративную процедуру построения процесса $P^{(i)}$ по процессу $P^{(i-1)}$ с помощью разбиения некоторых классов $Q^{(i-1)}$, соответствующих состояниям процесса $P^{(i-1)}$, на непересекающиеся подклассы $Q^{(i)}$, соответствующие состояниям процесса $P^{(i)}$.

Разобьем класс $Q_1^{(0)} = \{?e, ?a_2, ?a_2?a_2, ?a_1?a_1, ?a_2?a_1?a_2\}$ процесса $P^{(0)}$ на подклассы по следующему принципу: любые две нити $?a_1^*, ?a_2^* \in Q_1^{(0)}$ будем помещать в один и тот же подкласс, если для них имеет место $\varphi^*(?a_1^*?a) = \varphi^*(?a_2^*?a)$ для всех допустимых восприятий $?a$.

Класс $Q_1^{(0)}$ в этом случае можно разбить на два подкласса

$$\{?e, ?a_2a?_2\}, \{?a_2, ?a_1?a_1?a_1, ?a_1?a_1?a_2\},$$

так как

$$\begin{aligned}
&\varphi^*(?e?a_1) = \varphi^*(?a_2?a_2?a_1) = !a_1, \varphi^*(?e?a_2) = \varphi^*(?a_2?a_2?a_2) = !a_3,\\
&\varphi^*(?a_2?a_1) = \varphi^*(?a_1?a_1?a_1?a_1) = \varphi^*(?a_1?a_1?a_2?a_1) = !a_2,\\
&\varphi^*(?a_2?a_2) = \varphi^*(?a_1?a_1?a_1?a_2) = \varphi^*(?a_1?a_1?a_2?a_2) = !a_3.
\end{aligned}$$

Введем новые обозначения классов:

$$\begin{aligned}
&Q_1^{(1)} = \{?e, ?a_2?a_2\},\\
&Q_2^{(1)} = \{?a_1?a_2, ?a_1?a_2?a_2, ?a_2?a_2?a_1\},\\
&Q_3^{(1)} = \{?a_1?a_1, ?a_1?a_2, ?a_1?a_2?a_1, ?a_1?a_1?a_1?a_1, ?a_1?a_1?a_2?a_1\},\\
&Q_4^{(1)} = \{?a_1, ?a_1?a_1?a_1, ?a_1?a_1?a_2\}.
\end{aligned}$$

Эти классы совпадают соответственно с классами Q_1, Q_2, Q_3, Q_4, полученными по алгоритму 1, и, следовательно, процесс $P^{(1)}$, который можно построить по этим множествам, с точностью до обозначения внутренних состояний будет совпадать с процессом $P(S)$ на рис. 2.

Настоящим примером был проиллюстрирован подход к процедуре детерминизации в целях построения детерминированного процесса. Перейдем к более строгой формулировке этой процедуры в виде алгоритма.

Рассмотрим отношение $E^{(i)}$ на множестве $?S$: $?a_1^*E^{(i)}?a_2^*$, если $?a_1^*, ?a_2^* \in ?S, \varphi^*(?a_1^*?a^*) = \varphi^*(?a_2^*?a^*)$ для всех $?a^*$ длиной $l(?a^*) \le i$ [1] таких, что $?a_1^*?a^*, ?a_2^*?a^* \in ?S$.

Из определения отношения $E^{(i)}$ следует, что отношение $E^{(0)}$ разбивает множество нитей $?S$ из примера 1, рассмотренного ранее, на классы $Q_1^{(0)}, Q_2^{(0)}, Q_3^{(0)}$. Действительно, отношение $E^{(0)}$ проверяется для продолжений $?a^*$ таких, что $l(?a^*) = 0$. Тогда по определению $E^{(i)}$ для любых двух нитей $?a_1^*,\ ?a_2^* \in ?S$ имеет место $?a_1^*E^{(0)}?a_2^*$, если $\varphi^*(?a_1^*) = \varphi^*(?a_2^*)$.

В то же время, если i выбрано равным или большим длине ζ самой длинной нити из множества $?S$, то из определения отношения $E^{(i)}$ следует, что отношение $E^{(\zeta)}$ совпадает с отношением R, а отношение $E^{(i+1)}$ разбивает множество $?S$ на классы, каждый из которых является подклассом отношения $E^{(i)}$, т.е. для каждого класса $Q_j^{(i+1)}$ существует класс Q_h^i такой, что $Q_j^{(i+1)} \subseteq Q_h^{(i)}$.

В общем случае граф переходов процесса $P^{(i)}$ строится аналогично графу переходов процесса $P(S)$. Состояниям $b_1^{(i)}, b_2^{(i)}, \ldots, b_{k_i}^{(i)}$ этого процесса соответствуют классы $Q_1^{(i)}, Q_2^{(i)}, \ldots, Q_{k_i}^{(i)}$. Функцией переходов процесса $P^{(i)}$ является функция $f^{(i)}(?a, b_j^{(i)}) = \{b_h^{(i)} | ?a^* \in Q_j^{(i)}, ?a^*?a \in Q_h^{(i)}\}$, а функцией выходов — функция $\varphi^{(i)}(?a, b_j^{(i)}) = = \{\varphi^*(?a^*?a) | ?a^* \in Q_j^{(i)}\}$.

Начальным состоянием процесса $P^{(i)}$ является состояние $b_1^{(i)}$, соответствующее классу $Q_1^{(i)}$, содержащему пустое восприятие $?e$. Процесс $P^{(i)}$ в общем случае является недетерминированным, так как из одного и того же состояния $b_j^{(i)}$ по одному и тому же восприятию возможен переход в несколько различных состояний $b_h^{(i)}$ и, как следствие, есть возможность выдачи различных внешних реакций. Поскольку отношение $E^{(\zeta)}$ совпадает с отношением R, то при выполнении условий полноты и непротиворечивости множества S процесс $P^{(\zeta)}$ совпадает с детерминированным процессом $P(S)$.

Сформулируем алгоритм построения процесса $P(S)$, реализующего заданное множество S нитей, с использованием отношения $E^{(i)}$.

Алгоритм 2:

а) принять $i = 0$;

б) разбить множество $?S$ на классы $Q_1^{(i)}, Q_2^{(i)}, \ldots, Q_{k_i}^{(i)}$ по отношению $E^{(i)}$, где $Q_1^{(i)}$ является классом, содержащим пустую последовательность $?e$;

в) отождествить классы $Q_1^{(i)}, Q_2^{(i)}, \ldots, Q_{k_i}^{(i)}$ с состояниями $b_1^{(i)}, b_2^{(i)}, \ldots, b_{k_i}^{(i)}$ процесса $P^{(i)}$;

г) проверить, является ли процесс $P^{(i)}$ детерминированным. Если он детерминированный, то перейти к п. “е”, в противном случае перейти к п. “д”;

д) разбить каждый класс $Q_1^{(i)}, Q_2^{(i)}, \ldots, Q_{k_i}^{(i)}$ по отношению $E^{(i+1)}$ на подклассы. Считать множество этих подклассов новым множеством классов $Q_1^{(i+1)}, Q_2^{(i+1)}, \ldots, Q_{k_{i+1}}^{(i+1)}$, где $Q_1^{(i+1)}$является классом, содержащим пустую последовательность e. Принять $i = i + 1$ и перейти к п. “в”;

е) считать процесс $P^{(i)}$ процессом $P(S)$. Конец.

Очевидно, что трудоемкость алгоритма 2 не больше, чем трудоемкость алгоритма 1, но практически может быть существенно меньше, если число недетерминированных состояний невелико или быстро падает.

В случае задания процесса множеством рекурсивных канонических процессных выражений множество нитей S задается неявно, множество состояний процесса вводится непосредственно при описании и может быть далеко не оптимальным. Алгоритм 2 может быть легко модифицирован для работы непосредственно с каноническими процессными выражениями, позволяя оптимизировать процесс, описываемый таким образом. Идея этой модификации проста. Алгоритм 2 последовательно разбивает классы $Q_1^{(i)}, Q_2^{(i)}, \ldots, Q_{k_i}^{(i)}$ множества $?S$, с которыми сопоставляются состояния $b_1^{(i)}, b_2^{(i)}, \ldots, b_{k_i}^{(i)}$ процесса $P^{(i)}$, на подклассы $Q_1^{(i+1)}, Q_2^{(i+1)}, \ldots, Q_{k_{i+1}}^{(i+1)}$, с которыми сопоставляются состояния $b_1^{(i+1)}, b_2^{(i+1)}, \ldots, b_{k_{i+1}}^{(i+1)}$ процесса $P^{(i+1)}$. Это осуществляется до тех пор, пока очередной процесс не станет детерминированным. После этого в множестве $?S$ уже нет никакой необходимости — получен оптимальный процесс. Он оптимален, поскольку нельзя объединить в одно состояние ни одной пары его состояний, так как в этом случае получим недетерминированный процесс. Если же описание процесса, например на языке канонических процессных выражений, неоптимальное, то какие-то состояния процесса могут быть объединены в одно, поскольку они эквивалентны, т.е. при объединении этих состояний в одно процесс остается детерминированным. Проверка же на детерминированность может осуществляться аналогично тому, как это делается в алгоритме 2, но не по множеству нитей $?S$, которого у нас в явном виде нет, а путем использования процессных выражений, которые неявно это множество задают. Сформулируем этот алгоритм.

Алгоритм 3:

а) принять $r = 1, \quad s = 0$;

б) сформировать процессное выражение $b_r^{(s)} = b_{i_1} | b_{i_2} | \ldots | b_{i_l}$ из всех состояний b_{i_j}, для которых внешняя реакция $!a$ во внешних процессных выражениях $!a = b_{i_j}$ одна и та же;

в) если имеются состояния, не вошедшие ни в одно процессное выражение $b_r^{(s)}$, то принять $r = r + 1$ и перейти к п. “б”. В противном случае перейти к п. “г”;

г) получить процессные выражения $B_k^{(s)}, \quad k = 1, 2, \ldots, r$, подстановкой в процессное выражение $b_k^{(s)} = b_{i_1} | b_{i_2} | \ldots | b_{i_l}$ вместо каждого состояния b_{i_j} правой части соответствующего процессного выражения b_{i_j}. Принять $j = 1,\ A_j^{(s)} = \varnothing$;

д) поместить $b_j^{(s)}$ в множество $A_j^{(s)}$;

ж) найти все $B_k^{(s)}$, $k = 1, 2, \ldots, r$, для которых $b_j^{(s)} \cap B_k^{(s)} \neq \varnothing$, $k = 1, 2, \ldots, r$, для каждого $b_j^{(s)} \in A_j^{(s)}$;

з) если в п. "ж" для некоторого $b_j^{(s)}$ найдены $B_a^{(s)}, \ldots,\ B_t^{(s)}$ такие, что $b_j^{(s)} \cap B_a^{(s)} \neq \varnothing, \ldots, b_j^{(s)} \cap B_t^{(s)} \neq \varnothing$, число которых не менее двух и среди них существуют пары $B_g^{(s)}$, $B_h^{(s)}$ такие, что $b_j^{(s)} \cap B_g^{(s)} = b_{m_1}?a_{m_1}|b_{m_2}?a_{m_2}|\ldots|b_{m_k}?a_{m_k}$ $b_j^{(s)} \cap B_h^{(s)} = b_{n_1}?a_{n_1}|b_{n_2}?a_{n_2}|\ldots|b_{n_l}?a_{n_l}$ и хотя бы для одной пары $?a_m$, $?a_n$ имеет место $?a_m = ?a_n$, то разбить процесс $b_j^{(s)}$ на подпроцессы $b_{j_1}^{(s)}$, $b_{j_2}^{(s)}, \ldots$, в каждый из которых входят только те состояния $b_m = b_m?a_m, b_n = b_n?a_n$, для которых нет ни одной пары $b_m?a_m, b_n?a_n$ такой, что $?a_m = ?a_n$. Заменить в множестве $A_j^{(s)}$ процесс $b_j^{(s)}$ полученными подпроцессами $b_{j_1}^{(s)}$, $b_{j_2}^{(s)}, \ldots$. Перейти, к следующему пункту. Если процесс $b_j^{(s)}$ не был разбит на подпроцессы, то также перейти к следующему пункту;

и) принять $j = j + 1$. Если $j \leq r$, то перейти к п. "д". В противном случае перейти к п. "к";

к) если суммарное число процессов в множествах $A_1^{(s)}, \ldots, A_r^{(s)}$ равно r, то перейти к пункту л). В противном случае перенумеровать все подпроцессы множеств $A_1^{(s)}, \ldots, A_r^{(s)}$ числами $1, 2, \ldots, r_1$, принять $r = r_1$, присвоить им обозначения $b_j^{(s)}$, где $s = s + 1$, $j = 1, 2, \ldots,\ r = r_1$ и перейти к п. "г";

л) конец. Процессные выражения $B_1^{(s)}, \ldots, B_r^{(s)}, !a_1^{(s)}, \ldots,\ !a_p^{(s)}$ являются оптимальными каноническими.

Пример 3. Имеем следующие канонические процессные выражения:

$$
\begin{aligned}
b_1 &= ?e,\\
b_2 &= b_1?a_1,\\
b_3 &= b_1?a_2,\\
b_4 &= b_1?a_2,\\
b_5 &= b_4?a_3,\\
b_6 &= b_1?a_2,\\
b_7 &= b_6?a_1,\\
b_8 &= b_1?a_2,\\
b_9 &= b_8?a_2,\\
b_{10} &= b_1?a_2,\\
b_{11} &= b_{10}?a_1,
\end{aligned}
$$

$$
\begin{aligned}
b_{12} &= b_{11}?a_3,\\
b_{13} &= b_1?a_2,\\
b_{14} &= b_{13}?a_1,\\
b_{15} &= b_{14}?a_1,\\
b_{16} &= b_1?a_2,\\
b_{17} &= b_{16}?a_1,\\
b_{18} &= b_{17}?a_2,
\end{aligned}
$$

$$
\begin{aligned}
!a_1 &= b_3\,|b_4\,|b_5\,|b_6\ \ |b_8\,|b_{10}\,|b_{12}\,|b_{13}\,|b_{15}\,|b_{16},\\
!a_2 &= b_1\,|b_2\,|b_7\,|b_9\ \ |b_{11}\,|b_{14}\,|b_{17}\,|b_{18}.
\end{aligned}
$$

Согласно п. “г”, “д” при $r = 1, s = 0$ будем иметь

$$
\begin{aligned}
b_1^{(0)} &= \{b_1, b_2, b_7, b_9, b_{11}, b_{14}, b_{17}, b_{18}\},\\
b_2^{(0)} &= \{b_3, b_4, b_5, b_6, b_8, b_{10}, b_{12}, b_{13}, b_{15}, b_{16}\}.
\end{aligned}
$$

По п. “з” получим

$$
\begin{aligned}
B_1^{(0)} &= b_1?a_1\,|b_6?a_1\,|b_{10}?a_1\,|b_{13}?a_1\,|b_{16}?a_1\,|b_8?a_2\,|b_{17}?a_2 =\\
&\quad (b_1\,|b_6\,|b_{10}\,|b_{13}\,|b_{16})\ ?a_1(b_8\,|b_{17})?a_2,\\
B_2^{(0)} &= b_{14}?a_1\,|b_1?a_2\,|b_4?a_3\,|b_{11}?a_3 =\\
&\quad b_{14}?a_1\,|b_1\,?a_2\,|(\ b_4\,|b_{11})\,?a_3
\end{aligned}
$$

Согласно п. “з” поместим состояние $b_1^{(0)}$ в множество $A_1^{(0)} = \{b_1^{(0)}\}$.

По п. “ж” вычислим $b_1^{(0)} \cap B_1^{(0)} = b_{17}?a_2\,|b_1\,?a_1$, $b_1^{(0)} \cap B_2^{(0)} = b_{11}?a_2\,|b_{14}?a_2\ |b_1?a_1$.

Поскольку $b_1^{(0)} \cap B_1^{(0)} \cap B_2^{(0)} = (b_1\,|b_{17})?a_1\,|(b_1\,|b_{14})?a_1$, то, согласно п. “к” состояния b_1 и b_{14}, b_1 и b_{17} должны входить в разные состояния $b_1^{(0)}$. Заменим в множестве $A_1^{(0)}$ состояние $b_1^{(0)}$ на два состояния: $b_{1,1}^{(0)} = b_1\,|b_2\,|b_9\,|b_{18}$ и $b_{1,2}^{(0)} = b_7\,|b_{11}\,|b_{14}\,|b_{17}$. В результате получим множество $A_1^{(0)} = \{b_{1,1}^{(0)}, b_{1,2}^{(0)}\}$. Состояния $b_2, b_7, b_9, b_{11}, b_{18}$ распределены между состояниями $b_{1,1}^{(0)}, b_{1,2}^{(0)}$ произвольным образом, что соответствует некоторому доопределению функции φ^*, описанному ранее.

Далее после п. “и” снова возвращаемся к п. “з”, по которому получим множество $A_2^{(0)} = \{b_2^{(0)}\}$.

По п. “з” вычислим $b_2^{(0)} \cap B_1^{(0)} = (b_6\,|b_8)?a_2\,|(b_{10}\,|b_{13}\,|b_{16}\,)?a_1$, $b_2^{(0)} \cap B_2^{(0)} = b_4?a_3$. Нетрудно видеть, что $b_2^{(0)} \cap B_1^{(0)} \cap B_2^{(0)} = \varnothing$. Следовательно, согласно п. “к”, множество $A_2^{(0)}$не должно расширяться. Поскольку все состояния рассмотрены, то можно переходить к п. “к”. Согласно этому пункту вследствие увеличения суммарного числа функций в множествах $A_1^{(0)}, A_2^{(0)}$ переобозначим состояния этих множеств следующим образом:

$$A_1^{(0)} = \{b_1^{(1)} = b_{1,1}^{(0)}, b_3^{(1)} = b_{1,2}^{(0)}\}, A_2^{(0)} = \{b_2^{(1)} = b_2^{(0)}\}, \text{ т.е.}$$

$$b_1^{(1)} = \{b_1, b_2, b_9, b_{18}\},$$

$$b_2^{(1)} = \{b_3, b_4, b_5, b_6, b_8, b_{10}, b_{12}, b_{13}, b_{15}, b_{16}\},$$

$$b_3^{(1)} = \{b_7, b_{11}, b_{14}, b_{17}\}.$$

Переходя снова к п. “ж” алгоритма 3, вычислим процессы

$$B_1^{(1)} = b_1?a_1(b_8\,|b_{17})?a_2,$$

$$B_2^{(1)} = b_{14}?a_1\,|b_1?a_2\,|(b_4\,|b_{11})?a_3, \ .$$

$$B_3^{(1)} = (b_6\,|b_{13}\,|b_{10}\,|b_{16})?a_1.$$

Если провести еще один цикл вычислений, то можно убедиться, что число состояний $b_i^{(2)}$совпадает с числом состояний $b_i^{(1)}$. Процессные выражения $B_1^{(1)}$, $B_2^{(1)}$, $B_3^{(1)}$, $!a_1$, $!a_2$ могут быть преобразованы подстановкой вместо каждого состояния b_l того состояния $b_i^{(1)}$, в которое оно входит. В результате получим следующие процессные выражения:

$$\begin{aligned}
&b_1^{(1)} = b_1^{(1)}a_1 \left|(b_2^{(1)} \left|b_3^{(1)}\right.)?a_2,\right. \\
&b_2^{(1)} = b_3^{(1)}?a_1 \left|b_1^{(1)}?a_2\ (b_2^{(1)} \left|b_3^{(1)}\right.)a_3,\right. \\
&b_3^{(1)} = b_2^1?a_{(1)}, \\
&!a_1 = b_2^{(1)}, \\
&!a_2 = b_1^{(1)} \left|b_3^{(1)}\right. .
\end{aligned}$$

4. Модификация процесса. При практической реализации процессов достаточно распространенной является следующая ситуация. Процесс $P(S)$ уже построен. Требуется построить новый процесс $P(S')$, который с точки зрения нового создателя то ли тот же самый, то ли похож на него, то ли радикально отличен. Он этого не знает, но хотел бы воспользоваться уже построенным процессом.

Если $S = S'$, то заново строить процесс $P(S')$ нет необходимости. Если $S' \subset S$, то та часть процесса $P(S)$, которая выполняет нити S', может быть использована как процесс $P(S')$. Если $S \cap S' \neq \varnothing$, то та часть процесса $P(S)$, которая выполняет нити множества $S \cap S'$, может быть использована для построения процесса $P'(S')$, но должна быть дополнена частью, выполняющей множество нитей $S' \backslash (S \cap S')$. Если $S \cap S' = \varnothing$, то процесс $P(S')$ должен быть построен заново.

Тривиальный путь анализа соотношения множеств нитей S и S' — это прямое их сравнение. Если мощности множеств S и S' равны соответственно m и n, а длина последовательностей в среднем равна k, то очевидно, что число операций, необходимых для сравнения, оценивается экспоненциальной величиной $\mathrm{O}(kmn)$, которая быстро растет с ростом всех составляющих. Кроме того, как уже отмечалось, только сравнительно простые процессы целесообразно задавать процессными выражениями, напрямую представляющими конечные множества конечных нитей S и S'. Поэтому будем полагать, что процессы $P(S)$ и $P(S')$ задаются каноническими процессными выражениями, позволяющими описывать бесконечные множества нитей S и S'.

Предположим, что процессы $P(S)$ и $P(S')$ заданы соответственно каноническими процессными выражениями $b_i^{(S)}$, $i = 1, \ldots, n_S$, $!a_j^{(S)}$, $j = 1, \ldots, p_S$, и $b_i^{S'}$, $i = 1, \ldots, n_{S'}$, $!a_j^{S'}$, $j = 1, \ldots, p_{S'}$.

Рассмотрим следующий алгоритм.

Алгоритм 4:

а) принять $r = 1$, $s = 0$;

б) сформировать процессное выражение $\dot{b}_r^{(s)} = b_{i_1} | b_{i_2} | \ldots | b_{i_l}$ из всех состояний обоих процессов, для которых внешняя реакция $!a$ во внешних процессных выражениях $!a = b_{i_j}$ одна и та же;

в) если имеются состояния, не вошедшие ни в одно процессное выражение $b_r^{(s)}$, то принять $r = r + 1$ и перейти к п. “б”. В противном случае перейти к следующему п. “г”;

г) получить процессные выражения $\dot{B}_k^{(s)}$, $k = 1, 2, \ldots, r$, подстановкой в процессное выражение $\dot{b}_k^{(s)} = b_{i_1} | b_{i_2} | \ldots | b_{i_l}$ вместо каждого состояния b_{i_j} правой части соответствующего процессного выражения b_{i_j}. Принять $j = 1$, $A_j^{(s)} = \varnothing$;

д) поместить $\dot{b}_j^{(s)}$ в множество $A_j^{(s)}$;

ж) найти все $\dot{B}_k^{(s)}$, $k = 1, 2, \ldots, r$, для которых $\dot{b}_j^{(s)} \cap \dot{B}_k^{(s)} \neq \varnothing$, $k = 1, 2, \ldots, r$, для каждого $\dot{b}_j^{(s)} \in A_j^{(s)}$;

з) если в п. “ж” для некоторого $\dot{b}_j^{(s)}$ найдены $\dot{B}_a^{(s)}, \ldots, \dot{B}_t^{(s)}$ такие, что $\dot{b}_j^{(s)} \cap \dot{B}_a^{(s)} \neq \varnothing, \ldots, \dot{b}_j^{(s)} \cap \dot{B}_t^{(s)} \neq \varnothing$, число которых не менее двух, и среди них существуют пары $\dot{B}_g^{(s)}, \dot{B}_h^{(s)}$ такие, что $\dot{b}_j^{(s)} \cap \dot{B}_g^{(s)} = b_{m_1}?a_{m_1} | b_{m_2}?a_{m_2} | \ldots | b_{m_k}?a_{m_k}$ $\dot{b}_j^{(s)} \cap \dot{B}_h^{(s)} = = b_{n_1}?a_{n_1} | b_{n_2}?a_{n_2} | \ldots | b_{n_l}\ ?a_{n_l}$ и хотя бы для одной пары $?a_m, ?a_n$ имеет место $?a_m = ?a_n$, то разбить процесс $\dot{b}_j^{(s)}$ на подпроцессы $\dot{b}_{j_1}^{(s)}, \dot{b}_{j_2}^{(s)}, \ldots$, в каждый из которых входят только те состояния $b_m = ?a_m b_m, b_n = ?a_n b_n$, для которых нет ни одной пары $b_m?a_m, b_n?a_n$ такой, что $?a_m = ?a_n$. Заменить в множестве $A_j^{(s)}$ процесс $\dot{b}_j^{(s)}$ полученными подпроцессами

$\dot{b}_{j_1}^{(s)}, \dot{b}_{j_2}^{(s)}, \ldots$. Перейти, к п. "и". Если процесс $b_j^{(s)}$ не был разбит на подпроцессы, то также перейти к п. "и";

и) принять $j = j + 1$. Если $j \leq r$, то перейти к п. "д". В противном случае перейти к п. "к";

к) если суммарное число процессов в множествах $A_1^{(s)}, \ldots, A_r^{(s)}$ равно r, то перейти к п. "л". В противном случае перенумеровать все подпроцессы множеств $A_1^{(s)}, \ldots, A_r^{(s)}$ числами $1, 2, \ldots, r_1$, принять $r = r_1$, присвоить им обозначения $\dot{b}_j^{(s)}$, где $s = s + 1$, $j = 1, 2, \ldots$, $r = r_1$, и перейти к п. "г";

л) конец. Процессные выражения $\dot{B}_1^{(s)}, \ldots, \dot{B}_r^{(s)}, !a_1^{(s)}, \ldots, !a_p^{(s)}$ являются композицией процессов $P(S)$ и $P(S')$.

Пример 4. Процесс $P(S)$ задан каноническими процессными выражениями:

$$b_1^{(S)} = b_1^{(S)} a_1 \left| b_2^{(S)} ?a_2 \right| b_3^{(S)} ?a_2 \,,$$

$$b_2^{(S)} = b_3^{(S)} ?a_1 \left| b_1^{(S)} ?a_2 \right| b_2^{(S)} ?a_3 \left| b_3^{(S)} ?a_3 \right.,$$

$$b_3^{(S)} = b_2^{(S)} ?a_1,$$

$$!a_1 = b_2^{(S)},$$

$$!a_2 = b_1^{(S)} \left| b_3^{(S)} \right. .$$

Процесс $P(S')$ задан следующими каноническими процессными выражениями:

$$b_1^{(S')} = b_3^{(S')} ?a_2 \left| b_1^{(S')} ?a_1 \right| b_1^{(S')} ?a_3 \,,$$

$$b_2^{(S')} = b_1^{(S')} ?a_2 \left| b_2^{(S')} ?a_1 \right. ,$$

$$b_3^{(S')} = b_2^{(S')} ?a_3 \left| b_3^{(S')} ?a_1 \right| b_3^{(S')} ?a_3 ,$$

$$!a_1 = b_1^{(S')} \left| b_3^{(S')} \right. ,$$

$$!a_2 = b_2^{(S')}.$$

Получим

$$!a_1 = b_2^{(S)} \left| b_1^{(S')} \right| b_3^{(S')} \,,$$

$$!a_2 = b_1^{(S)} \left| b_3^{(S)} \right| b_2^{(S')} \,,$$

$$\dot{b}_1^{(0)} = b_2^{(S)} \left| b_1^{(S')} \right| b_3^{(S')} \,,$$

$$\dot{b}_2^{(0)} = b_1^{(S)} \left| b_3^{(S)} \right| b_2^{(S')} \,.$$

$$\dot{B}_1^{(0)}=(b_3^{(S)}\left|b_1^{(S')}\right|b_3^{(S')})?a_1\left|(b_1^{(S)}\right|b_3^{(S')})?a_2\,|(\,b_2^{(S)}\left|b_3^{(S)}\right|(b_1^{(S')}\left|b_2^{(S')}\right|b_3^{(S')})\,?a_3,$$

$$\dot{B}_2^{(0)}=(b_1^{(S)}\left|b_2^{(S)}\right|b_2^{(S')})?a_1\left|(b_2^{(S)}\left|b_3^{(S)}\right|b_1^{(S')})?a_2\right. ,$$

$$\dot{b}_1^{(0)}\cap\dot{B}_1^{(0)}=(b_1^{(S')}\left|b_3^{(S')})?a_1\right|b_3^{(S')}?a_2\left|(b_2^{(S)}\left|b_1^{(S')}\right|b_3^{(S')})?a_3\right. ,$$

$$\dot{b}_1^{(0)}\cap\dot{B}_2^{(0)}=b_2^{(S)}?a_1\left|(b_2^{(S)}\left|b_1^{(S')})?a_2\right|b_2^{(S)}?a_3\right. ,$$

$$\dot{b}_2^{(0)}\cap\dot{B}_1^{(0)}=b_1^{(S)}?a_1\left|(b_2^{(S')}\left|b_3^{(S')})\,?a_3,\right.\right.$$

$$\dot{b}_2^{(0)}\cap\dot{B}_2^{(0)}=(b_1^{(S)}\left|b_2^{(S')})\,?a_1\right|b_3^{(S)}?a_2.$$

Состояния $b_2^{(S)}$ и $b_1^{(S')}, b_3^{(S')}$, $b_3^{(S')}$ и $b_2^{(S)}$, $b_1^{(S')}$ должны входить в разные состояния $\dot{b}_1^{(0)}$, а состояния $b_1^{(S)}, b_2^{(S')}$ — в разные состояния $\dot{b}_2^{(0)}$, т.е.

$$\dot{b}_{1,1}^{(0)}=b_1^{(S')},$$

$$\dot{b}_{1,2}^{(0)}=b_3^{(S')},$$

$$\dot{b}_{1,3}^{(0)}=b_2^{(S)},$$

$$\dot{b}_{2,1}^{(0)}=b_1^{(S)}\left|b_3^{(S)}\right. ,$$

$$\dot{b}_{2,2}^{(0)}=b_2^{(S')}.$$

После перенумерации получим следующие выражения:

$$\dot{b}_1^{(1)}=\dot{b}_{1,1}^{(0)}=b_1^{(S')},$$

$$\dot{b}_2^{(1)}=\dot{b}_{1,2}^{(0)}=b_3^{(S')},$$

$$\dot{b}_3^{(1)}=\dot{b}_{1,3}^{(0)}=b_2^{(S)},$$

$$\dot{b}_4^{(1)}=\dot{b}_{2,1}^{(0)}=b_1^{(S)}\left|b_3^{(S)}\right.,$$

$$\dot{b}_5^{(1)}=\dot{b}_{2,2}^{(0)}=b_2^{(S')}.$$

Используя эти выражения, получаем процессные выражения, задающие детерминированный процесс:

$$\dot{b}_1^{(1)}=\dot{b}_1^{(1)}?a_1\left|\dot{b}_2^{(1)}?a_2\right|\dot{b}_1^{(1)}?a_3\ ,$$

$$\dot{b}_2^{(1)}=\dot{b}_2^{(1)}?a_1\left|\dot{b}_2^{(1)}?a_3\right|\dot{b}_4^{(1)}?a_3\ ,$$

$$\dot{b}_3^{(1)}=\dot{b}_5^{(1)}?a_1\left|\dot{b}_4^{(1)}?a_2\right|\dot{b}_3^{(1)}?a_3\left|\dot{b}_5^{(1)}?a_3\right.\ ,$$

$$\dot{b}_4^{(1)} = \dot{b}_4^{(1)} ?a_1 \left| \dot{b}_3^{(1)} ?a_2 \right| \dot{b}_5^{(1)} ?a_2 \left| \dot{b}_1^{(1)} ?a_2 \right| \dot{b}_4^{(1)} ?a_1 ,$$

$$\dot{b}_5^{(1)} = \dot{b}_3^{(1)} ?a_1 ,$$

$$!a_1 = \dot{b}_3^{(1)} \left| \dot{b}_1^{(1)} \right| \dot{b}_2^{(1)} ,$$

$$!a_2 = \dot{b}_4^{(1)} \left| \dot{b}_5^{(1)} \right. .$$

Относительно алгоритма 4 справедливы следующие утверждения, принципы доказательства которых можно найти в работе [1]:

если $S' \subseteq S$, то $P(S \cup S') = P(S)$;

если $S \subseteq S'$, то $P(S \cup S') = P(S')$;

если $S' \neq S$ и $S \cap S' \neq \varnothing$,

то $P(S \cup S') = P(S \backslash (S \cap S')) \,|P\ (S' \backslash (S \cap S'))|P\ (S \cap S')$;

если $S' \neq S$ и $S \cap S' = \varnothing$, то $P(S \cup S') = P(S) \,|P(S')$.

Несмотря на очевидность этих утверждений, выводы, которые можно сделать на их основе, имеют большое значение для практики. Построение процесса $P(S \cup S')$ можно рассматривать как расширение процесса $P(S)$. Процесс расширения может выполняться многократно и при каждом очередном расширении может быть неизвестно, выполняет ли уже процесс $P(S)$ множество S' или его часть. Использование алгоритма 4 для расширения процесса $P(S)$ дает ответ на вопрос, реализует ли процесс $P(S)$ это множество. Если реализует, то никакого расширения процесса $P(S)$ не произойдет, он останется прежним.

Построение процесса $P(S \cup S')$ можно также рассматривать как расширение не всего процесса $P(S)$, а некоторой его части (подпроцесса). В этом случае точнее говорить не о расширении, а об использовании процесса $P(S)$ для выполнения нитей множества $S \cap S'$. Если процесс $P(S)$ выполняет множество нитей $S \cap S'$, то его расширение потребуется только для реализации множества $S' \backslash (S \cap S')$. Расширение процесса $P(S)$ можно интерпретировать так же, как внесение в него изменений. Эта процедура может носить итеративный диалоговый характер, применяя, например, алгоритм 4 последовательно для каждой нити множества S'.

В заключение раздела отметим, что если процесс $P(S)$ является не полностью определенным, то требуются изменения в п. “б”, ориентированные на определенную стратегию доопределения.

Заключение. Рассмотрены алгоритмы построения процессного графа переходов и эквивалентных ему канонических процессных выражений по множеству нитей, оптимизация канонических процессных выражений и их модификация. Дальнейшему рассмотрению подлежат вопросы композиции и декомпозиции взаимодействующих и невзаимодействующих процессов, проверки правильности процессов, преобразования процессов в целях удовлетворения определенным критериям качества, вопросы фаззификации и дефаззификации процессов, вопросы мобильности процессов и ряд других.

СПИСОК ЛИТЕРАТУРЫ

1. Гаврилов М. А., Девятков В. В., Пупырев Е. И. Логическое проектирование дискретных автоматов (языки, методы, алгоритмы). – М.: Наука, 1977. – 350 с.
2. Milner R. A calculus of communicating systems. Springer-Verlag, New York, 1980. – 171 p.

Статья поступила в редакцию 7.09.2012