

## О АЛГОРИТМЕ ПЕРЕВОДА ДОКУМЕНТОВ MSC В СЕТИ ПЕТРИ

*А.В. Чугаенко, С.Л. Кривый*

Институт кибернетики им. В.М. Глушкова НАН Украины,  
03680, Киев, проспект Академика Глушкова, 40.  
Тел.: (+38) (044) 526 7418, (+38) (044) 522 2539.  
avch@avch.org.ua, krivoi@i.com.ua

Описывается алгоритм перевода документа MSC в трассово эквивалентную ему сеть Петри. Полученная таким образом сеть Петри может использоваться для анализа свойств системы, представленной исходным документом MSC. Алгоритм является частью инструментария для автоматического анализа и верификации документов MSC.

The article describes the algorithm for translation MSC document to Petri Net, which is trace-equivalent to the original MSC. Petri Net, obtained by this algorithm, can be used for analysis of properties of the system, described by MSC document. Before mentioned algorithm is a part of the toolset for automatic verification and analysis of MSC documents.

### Введение

Верификация программных систем является одной из наиболее актуальных практических проблем, возникающих при разработке сложного программного обеспечения. В процессе разработки программной системы необходимо стремиться обнаруживать и исправлять ошибки на стадии проектирования, поскольку их обнаружение на более поздних этапах (кодирования и тестирования) становится существенно дороже и может привести к необходимости полной переработке системы. В настоящее время разработано довольно много формальных методов анализа и верификации программных систем, но, как правило, они требуют представления верифицируемой системы в каком-либо из специальных языков (таких как ESTELLE [1], RAISE [2] и т.п.). Это усложняет работу с такими инструментами и повышает вероятность внесения ошибок на дополнительном этапе ручного перевода проекта на такие языки с целью верификации. В данной работе рассматривается язык MSC [3] (Message Sequence Charts), широко использующийся в проектировании программных систем (в частности, в телекоммуникационной сфере). В работах [4, 5] показана возможность автоматического перевода диаграмм MSC в сети Петри, что позволяет автоматизировать процесс верификации MSC-продуктов. При этом для сетей Петри известно большое количество методов верификации и анализа свойств системы, моделируемой этой сетью.

Настоящая работа является дальнейшим развитием данного подхода и в ней представлен новый алгоритм перевода документов MSC в сети Петри. Предлагаемый алгоритм покрывает большую часть конструкций языка MSC, что делает его пригодным для практического применения к описанию реальных систем. Вводится понятие трассовой эквивалентности между документом MSC и соответствующей сетью Петри. Полученная в результате применения алгоритма, трассово эквивалентная исходному документу MSC сеть Петри, является ординарной и пригодна для анализа как статических так и динамических свойств исследуемой системы.

### 1. Предварительные сведения

**1.1. Сети Петри.** Для анализа и верификации документов MSC будут использоваться сети Петри [6], как достаточно удобный и мощный формализм. Этот формализм пригоден к исследованию параллельных и распределенных систем, поэтому с его помощью можно моделировать широкий класс реальных систем. Рассмотрим некоторые базовые определения, необходимые в дальнейшем.

**Определение 1.** Сетью будем называть тройку  $(P, T, F)$ , где  $P$  – непустое множество элементов сети, называемых местами,  $T$  – непустое множество элементов сети, называемых переходами,

$F \subseteq P \times T \cup T \times P$  – отношение инцидентности, и для  $(P, T, F)$  выполнены такие условия:

A1)  $P \cap T = \emptyset$  (множества мест и переходов не пересекаются);

A2)  $(F \neq \emptyset) \wedge (\forall x \in P \cup T, \exists y \in P \cup T : xFy \vee yFx)$  (любой элемент сети инцидентен хотя бы одному элементу другого типа).

Графически такая сеть изображается в виде двудольного графа, места которого составляют одну часть (графически изображаются как  $\bigcirc$ ), а переходы – вторую (графически изображаются как  $\square$ ). При этом условие A1) означает двудольность этого графа, а условие A2) означает, что любой элемент сети инцидентен хотя бы одному элементу другого типа.

Если место  $p$  (переход  $t$ ) связано дугой с переходом  $t'$  (местом  $p'$ ), то  $p(t)$  называют входным местом (переходом) для  $t'(p')$  соответственно. Аналогично определяется понятие выходного места (перехода).

A3) если для произвольного элемента сети  $x \in X$  обозначить  $\cdot x$  множество его входных элементов  $\langle y | yFx \rangle$ , а  $x \cdot$  – множество его выходных элементов  $\langle y | xFy \rangle$ , то  $\forall p_1, p_2 \in P : (\cdot p_1 = \cdot p_2) \wedge (p_1 \cdot = p_2 \cdot) \Rightarrow (p_1 = p_2)$  (сеть не содержит пары мест, которые инцидентны одному и тому же множеству переходов).

**Определение 2.** Сетью Петри (СП) будем называть набор  $N = (P, T, F, W, M_0)$ , где  $(P, T, F)$  – конечная сеть (множество  $X = P \cup T$  конечно), а  $W : F \rightarrow N / \{0\}$  и  $M_0 : P \rightarrow N$  – две функции, называемые соответственно кратностью дуг и начальной разметкой.

Функционирование СП происходит путем срабатывания переходов. Условием срабатывания перехода является наличие фишек во всех его входных местах, которые при срабатывании переносятся в его выходные места в соответствии с отношением  $F$ . Начальное заполнение мест СП фишками выполняется с помощью начальной разметки.

СП, у которой кратность любой дуги равна 1, называется ординарной. Заметим, что ординарные сети Петри и их свойства наиболее подробно изучены.

**Определение 3.** Помеченной СП называется пара  $(N, \Sigma)$ , где  $N$  – сеть Петри,  $\Sigma : T \rightarrow A$  – помечающая функция, над некоторым алфавитом  $A$ .

**Определение 4.** Пусть  $(X, R)$  – частично упорядоченное множество с порядком  $R$ . Отношение  $R$  называется доминированием, если для любых  $x, y \in X$  выполняется условие:

Если  $(xRy) \wedge (x \neq y)$ , то не найдётся  $z \in X$  такое, что  $(xRz) \wedge (zRy)$ .

**Определение 5.** Помеченную сеть Петри и документ MSC будем называть трассово эквивалентными тогда и только тогда, когда каждой трассе в документе MSC можно сопоставить слово из языка, порождаемого сетью Петри, и наоборот.

**1.2. MSC.** Диаграммы MSC [3] являются широко распространённым формализмом для описания различных программных и аппаратных систем, в частности, телекоммуникационных протоколов. В сочетании с жёсткой стандартизацией и наличием формальной семантики это делает их подходящим кандидатом на использование в качестве входного языка для анализа проектируемых систем.

Язык диаграмм MSC предназначен для описания взаимодействия между независимыми сущностями, обменивающимися сообщениями и включает в себя большое количество разнообразных элементов; в данном алгоритме рассматривается только их ограниченное подмножество, состоящее из условий (condition), сущностей (instances), сообщений (messages), обобщённого упорядочения (generic ordering), создания и удаления сущности (create instance, stop instance), шлюзов сообщений и упорядочения (message gate, order gate), инлайнов (inlines). Далее приводится краткое описание этих элементов.

Документ MSC состоит из двух видов диаграмм – MSC и HMSC (High-level MSC).

Диаграмма MSC описывает обмен сообщениями между сущностями и реализует часть поведения описываемой системы. Диаграмма HMSC описывает обмен управлением между остальными диаграммами MSC и связывает их между собой. Сформированный таким образом объект, называемый документом MSC, описывает полное поведение моделируемой системы (в соответствии с выбранным при моделировании уровнем абстракции).

Диаграмма MSC включает такие объекты:

- сущность – базовое понятие MSC, представляет собой объект или экземпляр объекта моделируемой системы. Вертикальная линия, отходящая от сущности представляет собой отображение течения локального времени для этой сущности (сверху вниз);
- сообщение отображает акт обмена информацией между сущностями. Стрелка на изображении сообщения показывает направление передачи информации. Сообщение осуществляет синхронизацию сущностей в том смысле, что приём сообщения всегда происходит не раньше его передачи;
- обобщённое упорядочение позволяет установить порядок между независимыми событиями в пределах одной диаграммы. Порядок в этом случае определяется направлением стрелки (если присутствует) или направлением сверху вниз в противном случае;

- создание сущности – это особый тип сообщения, передача которого приводит к созданию новой сущности. Так же, как для обычного сообщения, при создании сущности ей можно передавать дополнительную информацию;
- удаление сущности прекращает её существование (в пределах данной диаграммы). Никакие сообщения не могут быть переданы или приняты сущностью после её удаления;
- условия используются для ограничения возможных трасс для данной диаграммы. Кроме этого, условия выполняют синхронизацию сущностей в том смысле, что если две сущности разделяют одно и то же условие, то любые обмены сообщениями между ними, начатые до (после) наступления условия должны быть закончены соответственно до (после) этого условия. Неформально это можно записать как “на графическом представлении диаграммы MSC для сущностей, разделяющих одно и то же условие, стрелка, соответствующая сообщению, не должна пересекать символ условия”. Данная особенность условий используется в алгоритме для объединения диаграмм;
- шлюзы сообщений служат для передачи сообщений между различными диаграммами в пределах одного документа MSC;
- шлюзы упорядочения служат для установления порядка между событиями, находящимися в разных диаграммах в пределах одного документа MSC;
- инлайны инкапсулируют части диаграммы MSC, задавая для них собственный порядок выполнения – альтернативный (alt), исключение (exc), циклический (loop), необязательный (opt) или параллельный (par) – следующим образом:
  - alt.– выполняется одна из заданных альтернатив, заданных в инлайне;
  - exc.– выполняется либо тело инлайна либо оставшаяся часть диаграммы;
  - loop – тело инлайна выполняется в цикле;
  - opt – тело инлайна выполняется один раз или пропускается вообще;
  - par – заданные в инлайне альтернативы выполняются параллельно, т. е. с произвольным интерливингом. Легко видеть, что exc и opt являются частными случаями alt.
- Диаграмма HMSC состоит из следующих объектов:
  - начальная точка – с этой точки начинается выполнение документа MSC. В каждой HMSC присутствует ровно одна начальная точка;
  - конечная точка – при попадании в эту точку выполнение документа MSC завершается. В одной HMSC могут присутствовать несколько конечных точек или не присутствовать ни одной;
  - точка альтернативного ветвления – при попадании в данную точку выполнение документа продолжается в одной из альтернативных веток управления;
  - точка объединения – в данной точке альтернативные ветки управления объединяются;
  - конструкция параллельного исполнения, – после попадания в эту конструкцию, выполнение продолжается параллельно в нескольких ветках управления. Все параллельные ветки должны быть закончены до выхода из данной конструкции;
  - ссылка на MSC – обозначает одну из присутствующих в документе. При попадании на ссылку, выполнение документа продолжается внутри указанной диаграммы;
  - цикл – позволяет выполнить указанную ссылку на MSC несколько раз;
  - условия – ограничивает возможность выбора ветки управления выполнением условия.

Каждое выполнение документа MSC формирует трассу MSC, которую можно представить как последовательность объектов, которые встречались во время выполнения документа. Параллельное выполнение в данном случае обозначает интерливинг в трассе объектов, встречающихся на параллельных ветках выполнения. Под структурно (или статически) возможными трассами будем понимать трассы, возможные в данном документе, если все находящиеся в нём условия принять истинными. Легко видеть, что множество трасс в данном документе является подмножеством множества структурно возможных трасс данного документа.

## 2. Описание алгоритма

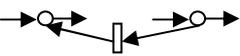
Алгоритм построения сети Петри, (структурно) трассово эквивалентной заданному документу MSC, состоит из двух этапов.

**Первый этап:** построение графа трасс. На первом этапе алгоритма мы строим так называемый граф трасс, который описывает структурно возможные в исходном документе MSC трассы. Построение графа трасс состоит из таких шагов:

- 1) если HMSC отсутствует, она восстанавливается согласно нижеприведённому алгоритму, иначе используется имеющаяся;
- 2) начальная точка HMSC переводится в узел “start”;

- 3) конечная точка HMSC переводится в узел “end”;
- 4) циклы, оформленные как loop, переводятся в свободную петлю (“free loop”);
- 5) точки альтернативного ветвления переводятся в узлы “alt-in”;
- 6) точки объединения переводятся в узлы “alt-out”;
- 7) конструкция параллельного исполнения переводится в пару узлов “par-in” и “par-out”, между которыми находится содержимое конструкции;
- 8) рёбра графа соответствуют линиям HMSC;
- 9) ссылки на MSC разворачиваются в соответствующие графы по правилам, приведённым далее (10 – 18);
- 10) в начале каждой MSC находится узел “mhc-in” в конце – “mhc-out”. Данная пара узлов формирует зону синхронизации. Все дуги, соответствующие линиям HMSC входящим в ссылку на данную MSC или исходящим из неё, соединяются непосредственно с этими узлами;
- 11) узлы, соответствующие начальным событиям MSC соединяются с её узлом “mhc-in”, конечным – с её узлом “mhc-out”;
- 12) инлайны “exc” и “opt” преобразуются в соответствующим им “alt”;
- 13) инлайны типа “alt” преобразуются в пару узлов “alt-in” и “alt-out”, каждая из альтернатив внутри конструкции образует свою зону синхронизации;
- 14) инлайны типа “par” преобразуются в пару узлов “par-in” и “par-out”, каждая из альтернатив внутри конструкции образует свою зону синхронизации;
- 15) инлайны типа “loop” преобразуются в пару узлов “loop-in” и “loop-out”;
- 16) все дуги, соответствующие линиям MSC входящим в инлайн или исходящим из него, соединяются непосредственно с этими in/out узлами;
- 17) части MSC, находящиеся внутри инлайна, преобразуются по правилам, общим для MSC за исключением замены узлов “mhc-in”/“mhc-out” на соответствующую инлайну пару;
- 18) остальные обрабатываемые события MSC переводятся в узлы графа;
- 19) в граф добавляются дуги, соответствующие отношению порядка, явно заданному в исходной MSC;
- 20) в пределах каждой зоны синхронизации удаляются дуги, не соответствующие отношению доминирования (при этом вложенная зона упорядочения рассматривается как одно событие в наружной зоне);
- 21) находится соответствие между шлюзами разных MSC и в граф добавляются дуги, соответствующие прошедшим через эти шлюзы сообщениям и отношениям упорядочения.

**Второй этап:** построение сети Петри. На втором этапе по графу трасс строится трассово эквивалентная ему сеть Петри. Построение графа сети Петри состоит из таких шагов:

- 1) узел “start” заменяется на , переход помечается как  $\lambda$ ;
- 2) узел “end” заменяется на , переход помечается как  $\lambda$ ;
- 3) узел “alt-in” заменяется на , количество выходных дуг равно числу альтернатив;
- 4) узел “alt-out” заменяется на , количество входных дуг равно числу альтернатив;
- 5) узел “par-in” заменяется на , количество выходных дуг равно числу альтернатив. Переход помечается как  $\lambda$ ;
- 6) узел “par-out” заменяется на , количество входных дуг равно числу альтернатив. Переход помечается как  $\lambda$ ;
- 7) узлы “mhc-in”, “mhc-out” заменяются переходами, помечаются как  $\lambda$ ;
- 8) пара узлов “loop-in” и “loop-out” заменяется конструкцией . Переход помечается как  $\lambda$ ;
- 9) остальные узлы заменяются переходами и помечаются названиями соответствующих узлов исходной MSC;
- 10) узлы соединяются в соответствии в дугами в исходном графе. Если дуга соединяет переход с переходом, в неё вставляется место.

### 3. Восстановление HMSC по начальным и конечным условиям.

В случае отсутствия в документе HMSC, её можно восстановить с помощью метода, приведённого в [4]. В каждой из диаграмм MSC находят условия, раньше которых (или позже которых) нет других событий кроме, возможно, других условий. Такие условия называем соединяющими (JC – Join Conditions). JC называется стартовым, если до него раньше не было событий, кроме других переходов, и конечным, если после него нет таких событий.

Строим HMSC таким образом:

каждая MSC представляется ссылкой;

все MSC, имеющие одинаковые стартовые JC, объединяются конструкцией “alt”;

все MSC, имеющие одинаковые конечные JC, объединяются конструкцией “join”;

если две MSC имеют одна стартовое, а другая конечное JC с одинаковым именем, данные диаграммы соединяются дугой в направлении диаграммы, содержащей стартовое условие к конечному;

если диаграмма имеет конечное JC не соединённое ни с чем, оно соединяется с конечной точкой HMSC;

начальная точка HMSC выбирается следующим образом:

задаётся пользователем явно;

если присутствует ровно одна MSC со стартовым JC не соединённым ни с какой другой диаграммой, стартовая точка соединяется с этой MSC;

если присутствует больше одной MSC со стартовым JC не соединённым ни с какой другой диаграммой, стартовая точка соединяется конструкцией “alt”, соединённой со всеми этими MSC;

если MSC со стартовым JC не соединённым ни с какой другой диаграммой отсутствует, стартовая точка соединяется со случайно выбранной MSC.

### 4. Пример перевода документа MSC в сеть Петри

Реальные системы, описанные документами MSC и сгенерированные из них сети Петри, как правило, слишком громоздки для использования в работе, поэтому рассмотрим действие описанного алгоритма на упрощённом примере:

Пусть исходный документ MSC имеет вид (рис. 1)

Данный документ состоит из двух диаграмм MSC и одной HMSC, в которых используются условия, передача сообщений а также конструкции параллельного выполнения (par) в msc1 и альтернативного выполнения (alt) в msc2 соответственно.

После применения шагов 1)–19) алгоритма, получаем граф, показанный на рис. 2. Зоны синхронизации в данном графе обведены штрихпунктирной линией и помечены цифрами 1–6. После удаления в каждой из этих зон дуг, не соответствующих отношению доминирования, получаем граф трасс в его окончательном виде (рис. 3).

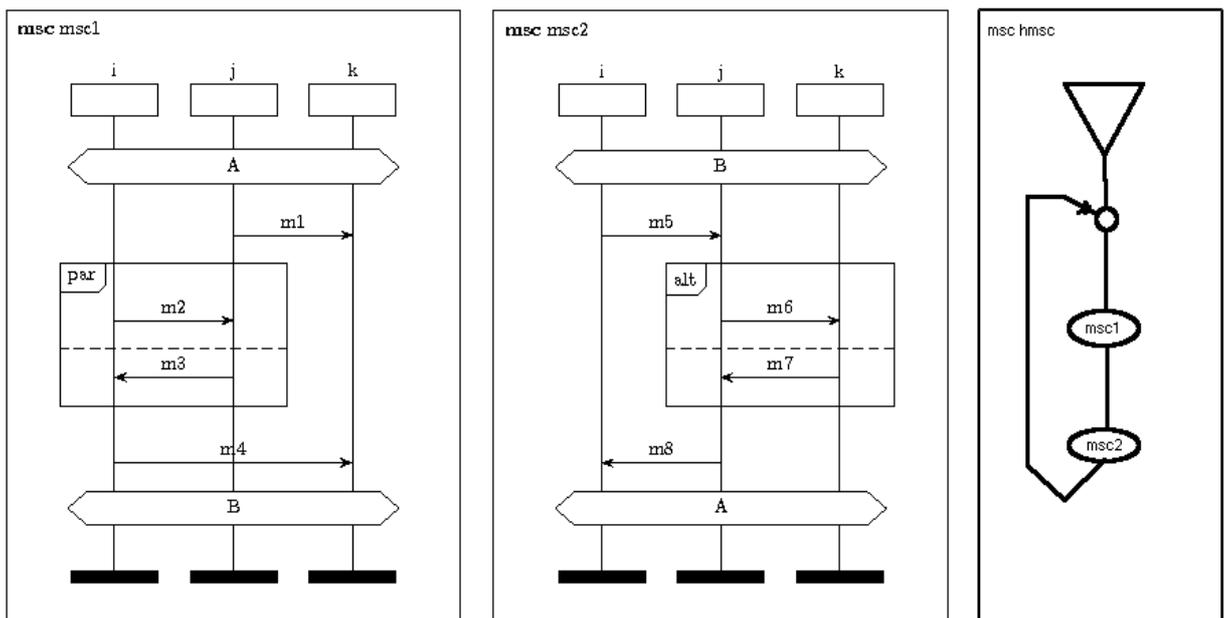


Рис. 1

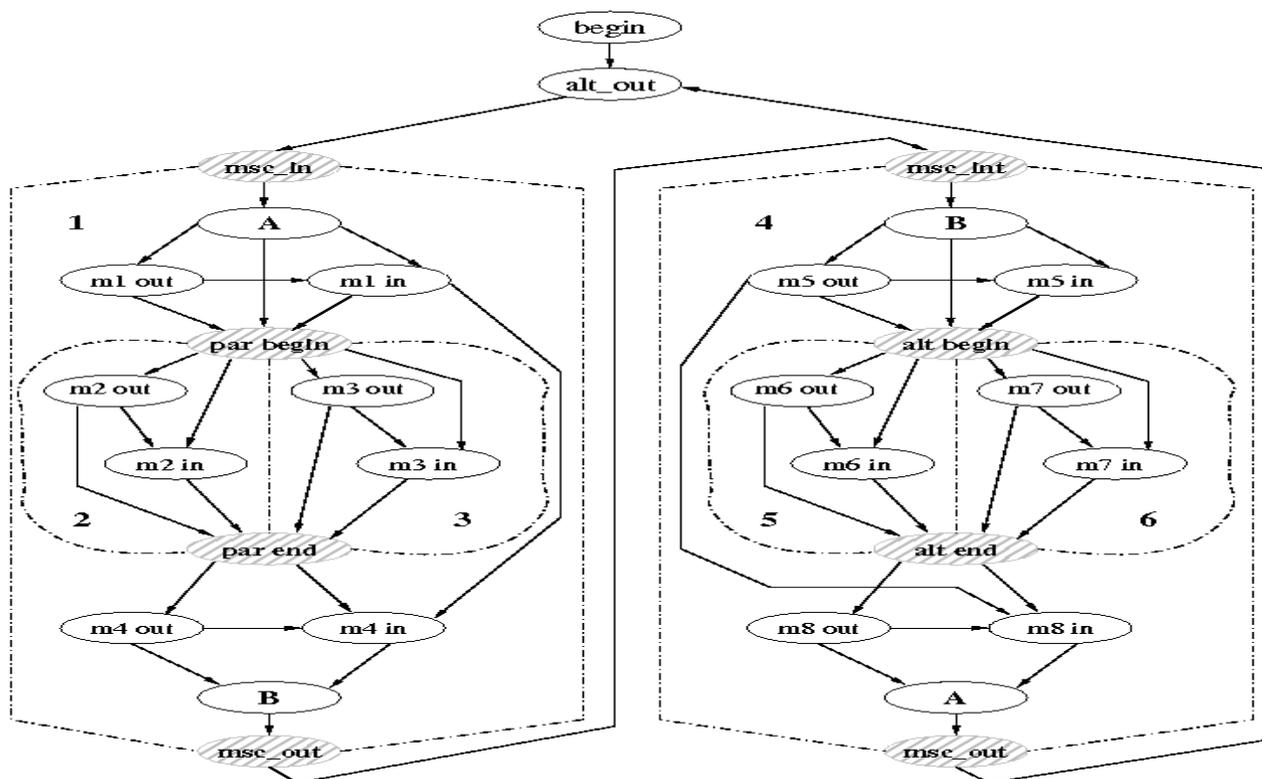


Рис. 2

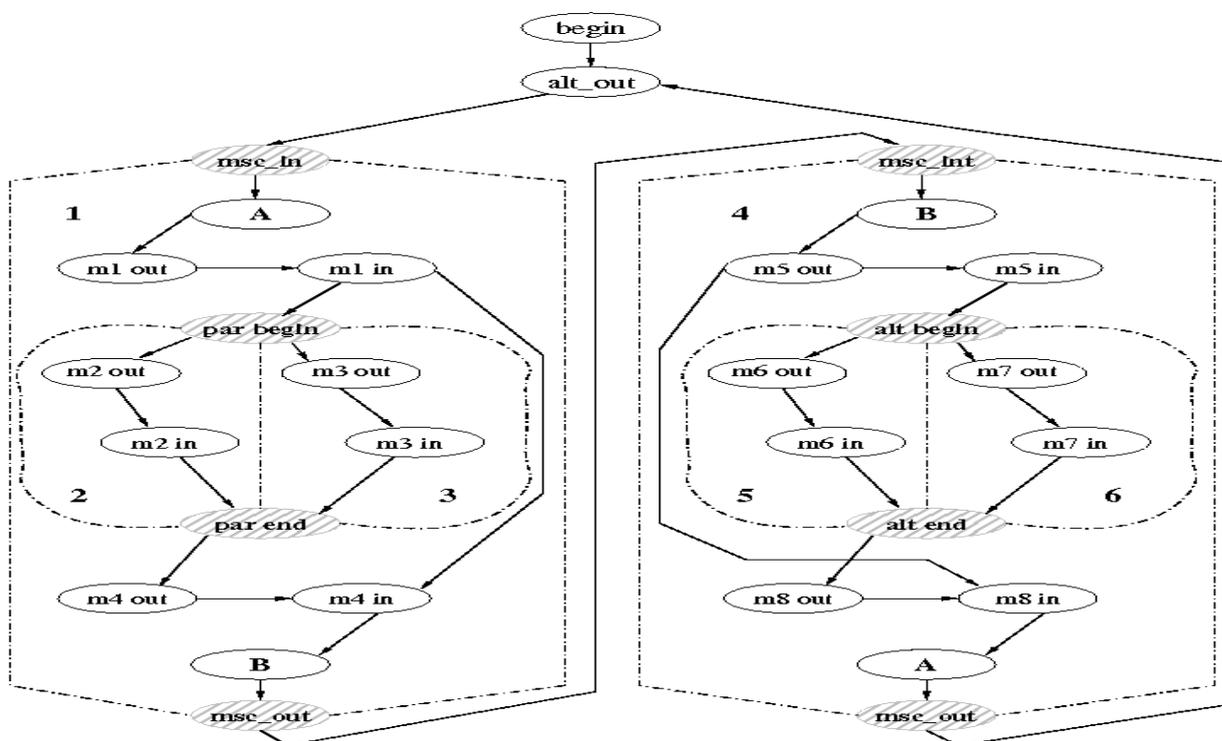


Рис. 3

Далее выполним шаги, по построению на основе полученного графа сети Петри, которая показана на рис. 4. В скобках рядом с переходами приведены значения их функции разметки (если она не равна  $\lambda$ ).

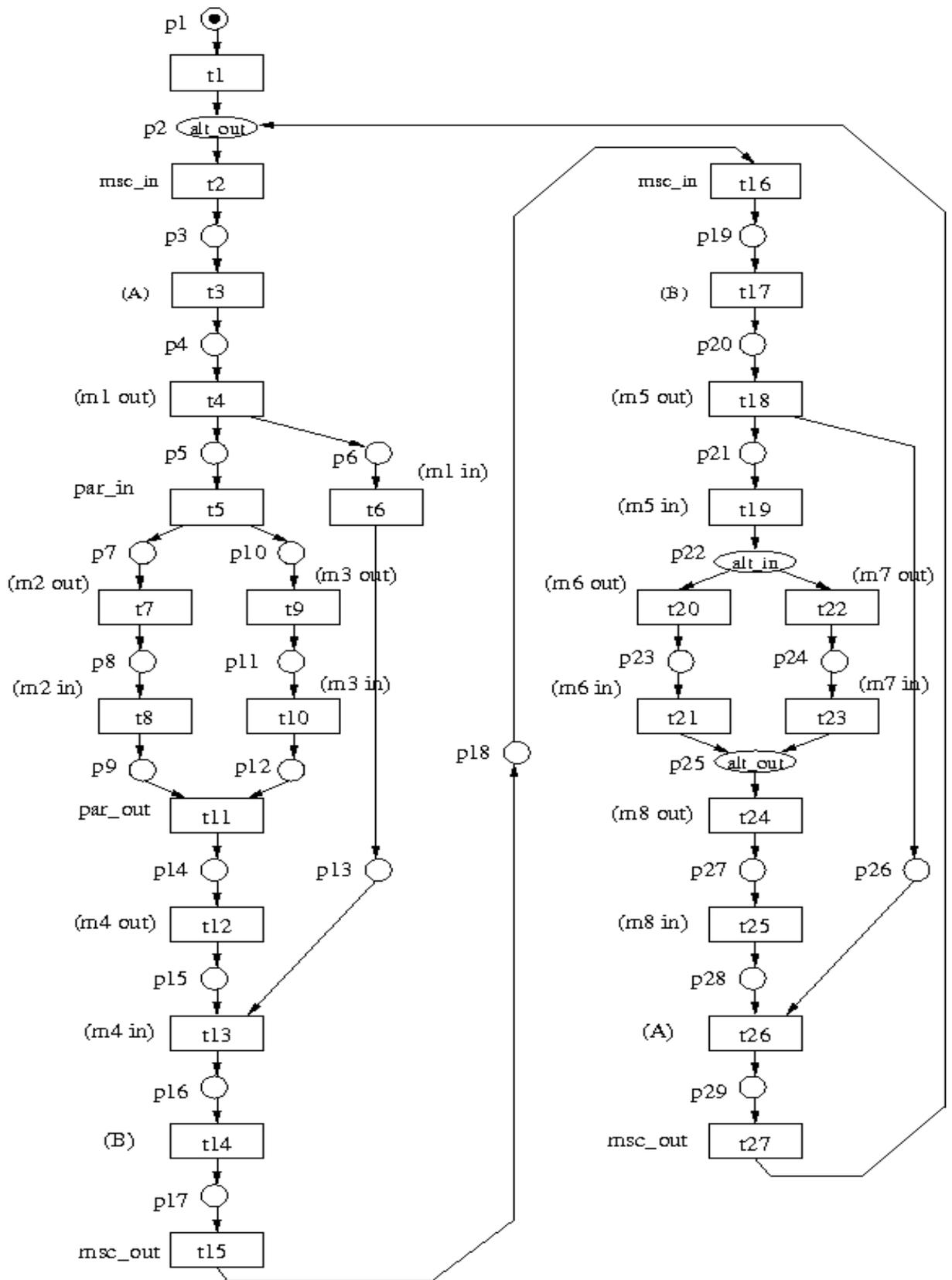


Рис. 4

### Заключение

Описан алгоритм перевода документа MSC в трассово эквивалентную ему сеть Петри. Основными достоинствами данного алгоритма является то, что он охватывает большую часть конструкций языка MSC, в результате его применения получается ординарная сеть Петри. Последнее преимущество позволяет применить

большое количество методов анализа сетей Петри, применяемых к ординарным СП. Кроме того, данные преимущества дают возможность использовать его на практике для анализа реальных систем. В частности, с помощью данного алгоритма можно находить дедлоки, проверять живость системы и её ограниченность, а также осуществлять проверку свойств справедливости и корректности.

1. *International Organization for Standardization (ISO). Information processing systems – open systems interconnections – Estelle – a formal description technique based on an extended state transition model. ISO 9074, 1989.*
2. *George C., Haxthausen A. E., Hughes S., Milne R., Prehn S., Storbank J. Pedersen. The RAISE Development Method. Prentice Hall Int., 1995.*
3. *ITU-TS Recommendation Z.120: Message Sequence Chart (MSC). ITU-TS, Geneva, 2000.*
4. *Матвеева Л.С. Аналіз та верифікація MSC-систем за допомогою мереж Петрі. Дис. канд. фіз-мат. наук. – К.: 2005. – 198 с.*
5. *Круузі, S., Matveyeva, L. Algorithm of Translation of MSC-specified System into Petri Net. Fundamenta Informaticae, Vol.79. P. 1 – 15, 2007.*
6. *Котов В.Е. Сети Петри. – М: Наука., 1984. – 160 с.*