Описание и преобразование управляющих процессов.

Сети Петри и их модификация.

<u>Основная задача начального этапа проектирования УА</u> – выбор формализованного языка.

<u>Основные понятия</u> – базис сетей Петри:

1. **событие;**
2. **условие.**

Сеть Петри – структура УП

↓

это последовательность процедур

Условия → событие

Состояние системы – это множество условий

Событие → новые условия →

→ изменение состояния системы

События – множество переходов

$$T=\{t_0, t_1, \ldots, t_r\}$$

Условия – множество позиций

$$A=\{a_0, a_1, \ldots, a_f\}$$

I – входная функция

связь T и A

O – выходная функция

I – отображает $t_v (v=\overline{0,r})$ в мн-во позиций $I(t_v)$ – **входные позиции перехода**

O – отображает t_v в мн-во позиций $O(t_v)$ – **выходные позиции перехода**

a_μ - входная позиция t_v, если $a_\mu \in I(t_v)$

a_μ - выходная позиция t_v, если $a_\mu \in O(t_v)$

Сеть Петри – N = (A, T, I, O)

Пример:

$A = \{a_0, a_1, a_2, a_3, a_4\}$

$T = \{t_0, t_1, t_2, t_3, t_4\}$

$I(t_0) = a_0$ $I(t_1) = a_1$

$I(t_2) = a_2$ $I(t_3) = a_3$

$I(t_4) = a_4$

$O(t_0) = a_1$ $O(t_1) = a_2$

$O(t_2) = a_3$ $O(t_3) = a_4$

I – матрица следования

O – матрица предшествования

Графическое представление сети Петри

Типы вершин:

1. позиции – «O»
2. переходы – «–»

if (a_μ - вход для t_v), then (дуга $a_\mu \rightarrow t_v$)

if (a_μ - выход для t_v), then (дуга $t_v \rightarrow a_\mu$)

↓

G = (V, W) – ориентированный двудольный мультиграф, где

V – множество вершин

W – множество направленных дуг

$V = A \cup T$ $A \cap T = \varnothing$

позиция – условие

↓

Выполнение условия – маркировка позиции

(метка – «точка» в позиции)

↓

ʘ

↓

Если несколько точек – то «**емкость условия**»

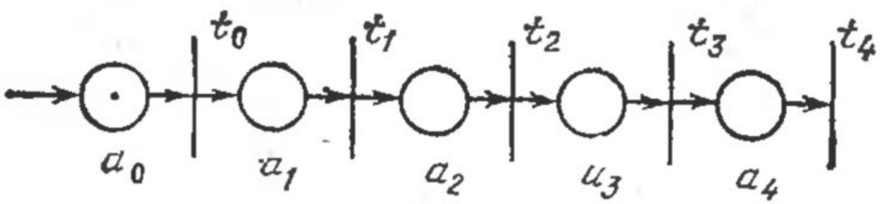
f-вектор маркировки сети Петри.

N = (A, T, I, O, M_0), где

M_0 – вектор начальной маркировки

Пример:

M_0 = (1, 0, 0, 0, 0)



распределение меток в позициях

↓

порядок выполнения сети

↑ *- зависит от*

последовательности реализации переходов

переход реализуется если он активен, т.е.

число меток во вх. позиц. => числу дуг, соединяющих ее с эти переходом

Разрешающие метки

реализация активного перехода

↓

замена маркировки сети

M

на

M’ (непосредственно достижимая из M)

Достоинства языка сети Петри:

1. позволяет описывать параллельные процессы;
2. имеет средства для задания конфликтных состояний.

q

ω > q

Выполнение сети → связанные последовательности:

1. реализуемых переходов
2. маркировок M_0, M_1, M_2, …

Безопасная сеть Петри.

1. запрещено наличие кратных дуг между позициями и переходами;
2. вектор маркировки может содержать лишь 0 и 1;
3. реализация активного перехода возможна, если ни 1 из его выходных позиций не содержит меток – число меток в любой позиции не больше 1;
4. конечное число состояний – 2f при f позициях.

Ограниченная сеть Петри.

k → k-безопасная позиция или k-ограниченная

k’ >= k – k’-безопасной

k_{max}

Ограничение оригинальной сети Петри – моделирование примитивных событий.

это **сеть позиция-переход**

↓

автоматная сеть

↓

маркированный граф

сети с предикатами на переходах

↓

расширение ее описательных возможностей

Введение позиции времени в сети Петри.

1. **Временные сети**: переход – t;
2. **Тайм-аутные сети**: переход – a и b.

Тайм-аутные сети Петри.

0<=a<=b

q

(q+a) (q+b)

Помеченные сети Петри.

метка – цвет

1 позиция – несколько цветов

Численные сети Петри.

1. метки любой природы и величины;
2. условия активизация и результата реализации независимы;
3. при реализации переходов изменяется маркировка входных и выходных позиций и содержимое памяти данных

Использование дуг разных типов в сети Петри.

Существуют:

1. ***Простые дуги:***
   1.1. ***активизирующая***;
   1.2. ***сдерживающая***;
   1.3. ***входная***;
   1.4. ***выходная***;
2. ***Составные дуги***:
   2.1. ***активизирующая входная***;
   2.2. ***сдерживающая выходная***.

Управляющие процессы и их формализованное описание.

Простейший линейный последовательный процесс – оригинальная сеть Петри.

A_i – процедуры (i = 0 – k)

операторные функциональные блоки – ОФБ

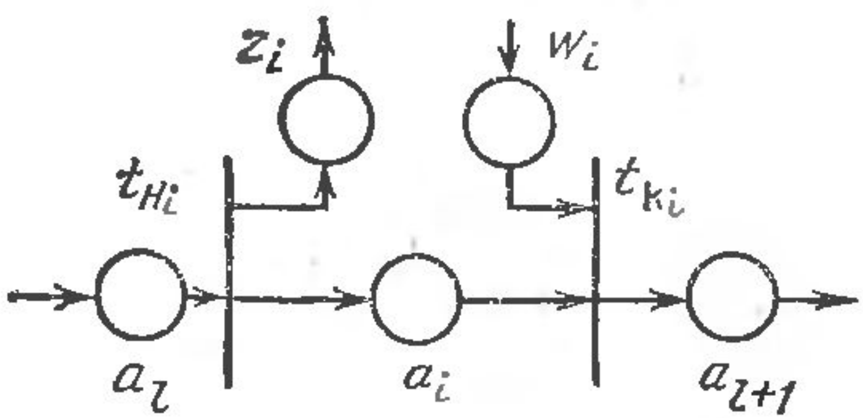
Процедура – переход сети Петри – t_i (i = 0 – k)

a_j (j = 0 – f) – позиции

Фазы выполнения процедуры:

1. начало;
2. выполнение;
3. Окончание.

Подсеть Петри для процедуры A_i.



где:

t_{Hi} и t_{Ki} – переходы

z_i и ω_i - внешние позиции

t_{Hi} – начало процедуры A_i

метки в z_i – включение ОФБ$_i$

метки в a_i – выполнение A_i

метки в ω_i – окончание ОФБ$_i$

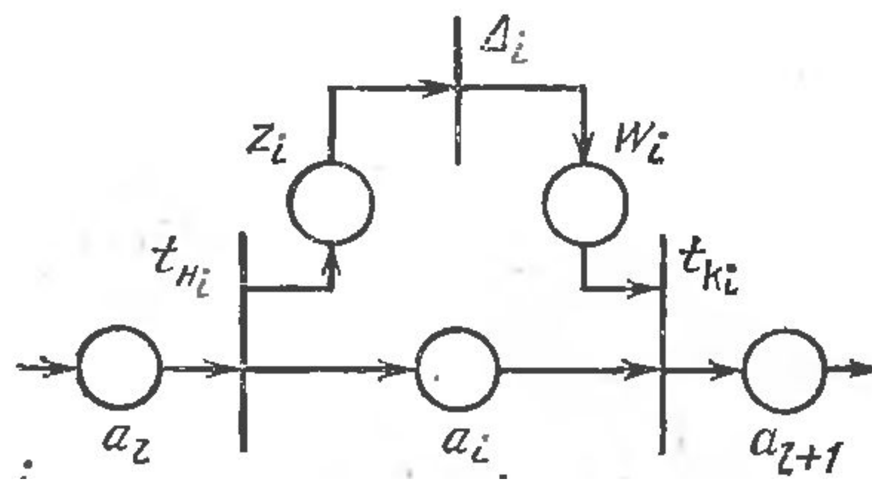
↓

срабатывание перехода t_{Ki}

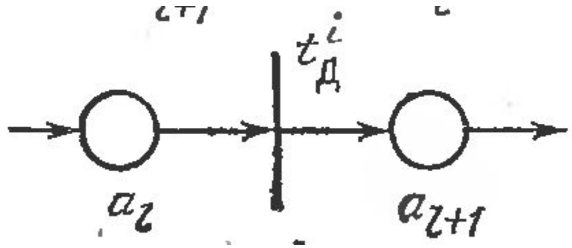
↓

метки в a_{i+1} – завершение A_i

Δ_i – непримитивный переход этой же сети Петри



Если выполнение процедуры – неделимое событие, то:
фрагмент с t_{Hi}, t_{Ki}, Δ_i и z_i, a_i,ω_i – на $t^i_д$

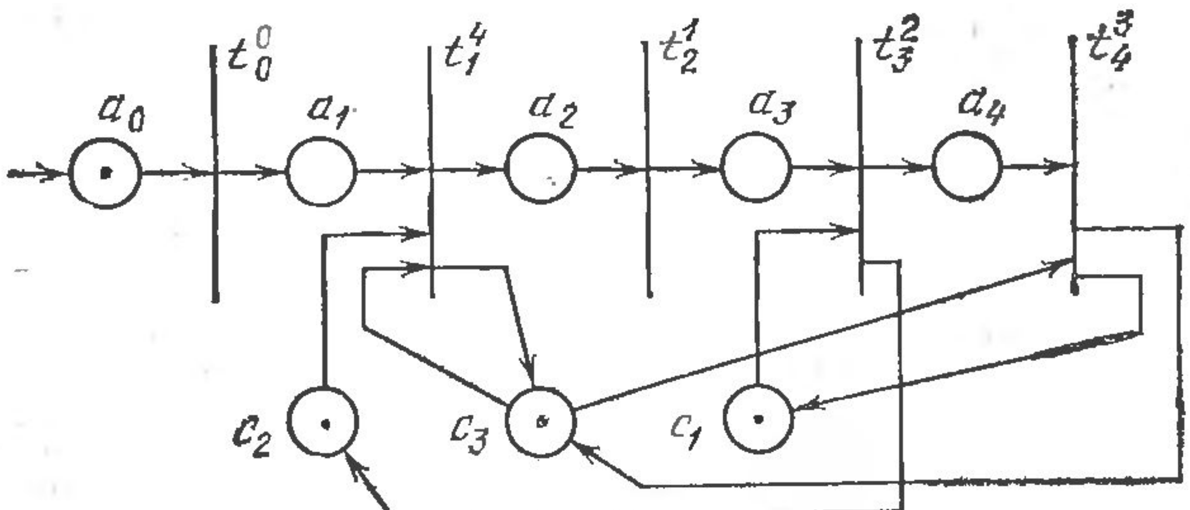


Это **длительный переход**.
У него есть **время выполнения**.

Функциональные ресурсы **(ФР)**
Собственный ФР
Разделяемый ФР
Пример:

C_i (i = 0 – l) – разделяемые ресурсы
q – число экземпляров i-го ФР
↓
q – кратность ресурса C_i – C_i^q
↓
его могут использовать α <= q процедур
при q=1 - у ресурса 2 состояния
q+1
внутренние или собственные ресурсы

Процедуры A_i линейного процесса:
1. $\{C_в^i\}$ – множество ФР – уже владеет;
2. $\{C_з^i\}$ – множество ФР – запрашивает;
3. $\{C_о^i\}$ – множество ФР – освобождает.



Процесс из 5-и последовательно выполняемых процедур A_i при следующем распределении 3-х ФР C_j:
A_1({C_2}, {-}, {-});
A_2({C_2}, {C_1}, {C_2});
A_3({C_1}, {C_3}, {C_1, C_3});
A_4({-}, {C_2, C_3}, {C_3}).
C_j – ресурсные внутренние позиции
$T_д^i$- длительные переходы
a_μ - основные внутренние позиции

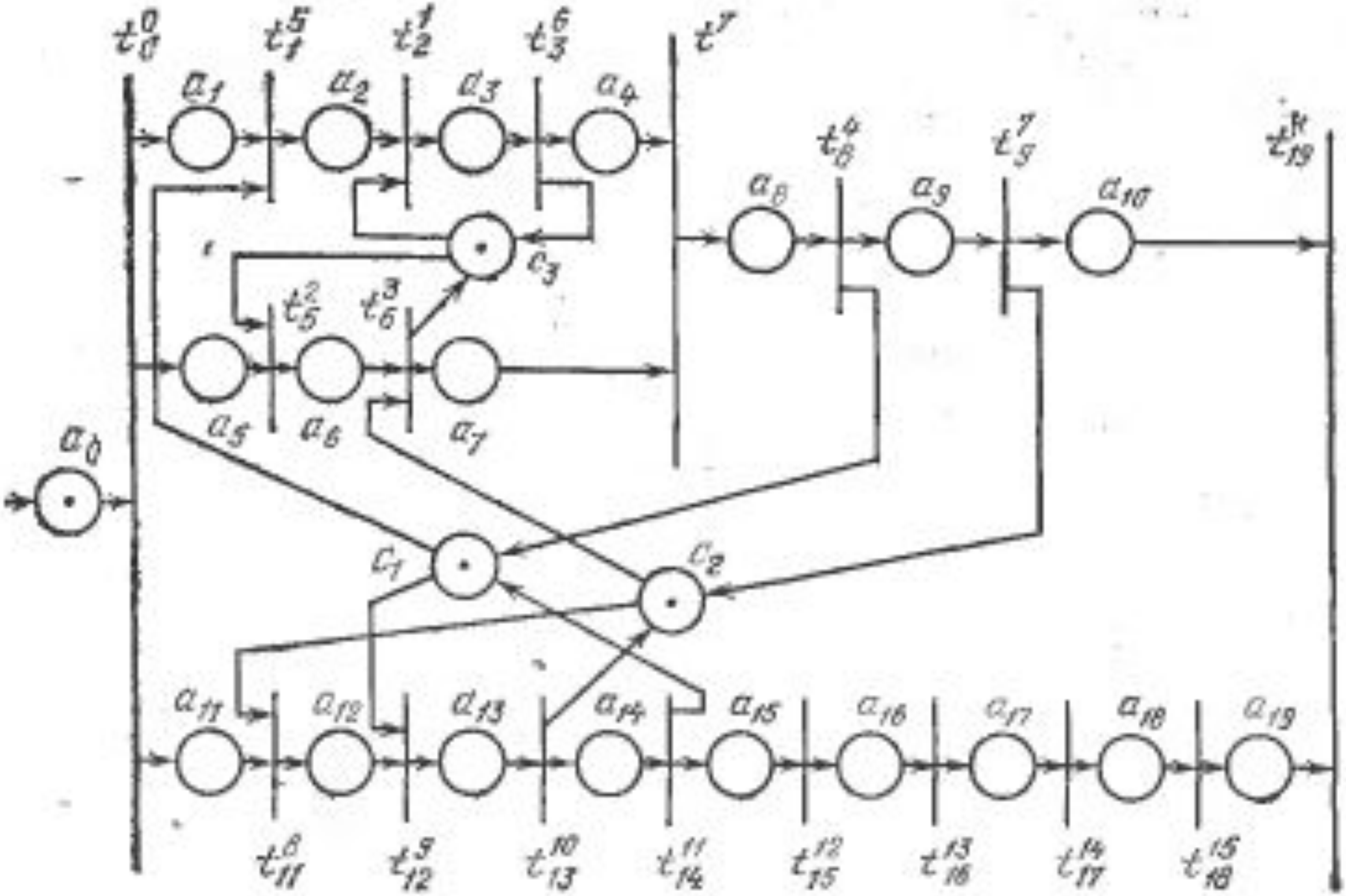
Пример:

Если для A_i – $\{C_в{}^i\}=C_1$, $\{C_з{}^i\}=C_3$, C_4 и $\{C_о{}^i\}=C_1$, C_4,
то $A_i(\{C_1\}, \{C_3, C_4\}, \{C_1, C_4\})$
$\{C_з{}^i\}\cap\{C_в{}^i\}=\varnothing$
Иногда: $\{C_в{}^i\}=\varnothing$ и $\{C_з{}^i\}=\{C_о{}^i\}$

Особенности описания параллельного линейного процесса в сети Петри.

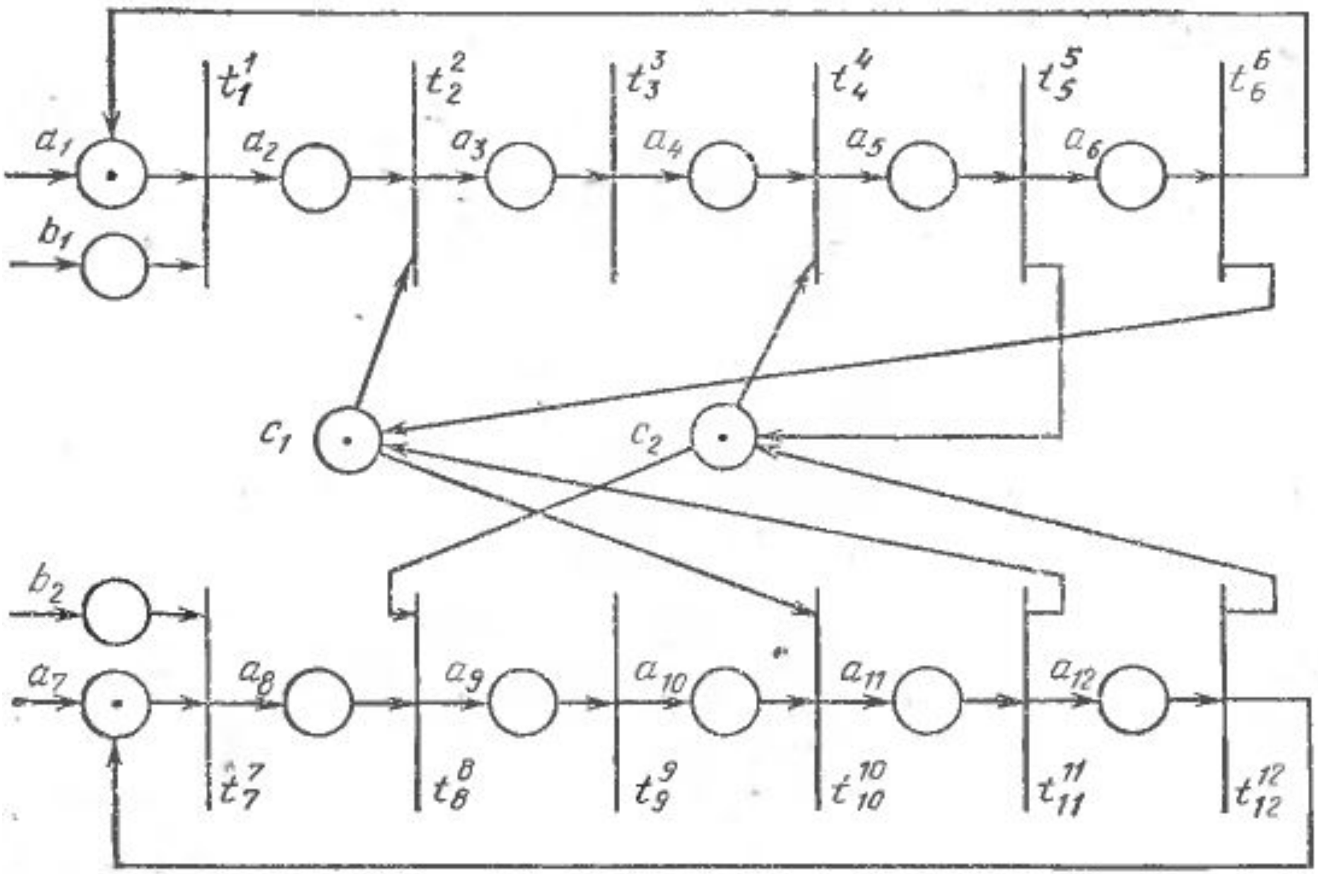
1. длительные переходы – процедуры;
2. t_R – переходы распараллеливания;
3. t_S – переходы соединения;
4. наличие элементарных подпроцессов;
5. собственные ФР подпроцесса

Пример:



$A_1(\{C_1\}, \{C_3\}, \{—\})$; $A_2(\{—\}, \{C_3\}, \{—\})$; $A_3(\{C_3\}, \{C_2\}, \{C_3\})$; $A_4(\{C_1, C_2\}, \{—\}, \{C_1\})$; $A_5(\{—\}, \{C_1\}, \{—\})$; $A_6(\{C_1, C_3\}, \{—\}, \{C_3\})$; $A_7(\{C_2\}, \{—\}, \{C_2\})$; $A_8(\{—\}, \{C_2\}, \{—\})$; $A_9(\{C_2\}, \{C_1\}, \{—\})$; $A_{10}(\{C_1, C_2\}, \{—\}, \{C_2\})$; $A_{11}(\{C_1\}, \{—\}, \{C_1\})$.

Пример:



Особенности описания разветвленного процесса в сети Петри.

1. позиции альтернативного разветвления;
2. позиции альтернативного соединения;
3. набор значений логических условий в конфликтных переходах альтернативного разветвления;

Логические ресурсы системы – ЛР.

D_i ($i = 1 – m$) – ЛР

в ЛР D_s проверяется p_s – условие

Внутренние ЛР

A_i ($\{P_1^i\}$, $\{P_2^i\}$)

Пример:

A_i ($\{\bar{p}_1, p_2\}$, $\{\bar{p}_2, p_3\}$)

p_s – $\{P_2^i\}$ – изменяется $A_i \rightarrow D_s$ – занято

p_s – $\{P_1^i\}$ – не изменяется $A_i \rightarrow D_s$ – не занято

Описание ЛР в сети Петри.

D_s:
- d_s – наличие метки – нет монополии
- d_s^1 – наличие метки – $p_s = 1$
- d_s^0 – наличие метки – $p_s = 0$

Пример 1:

A_i зависит от ЛУ ($p_s \epsilon D_s$)

и изменяет его (p_s)

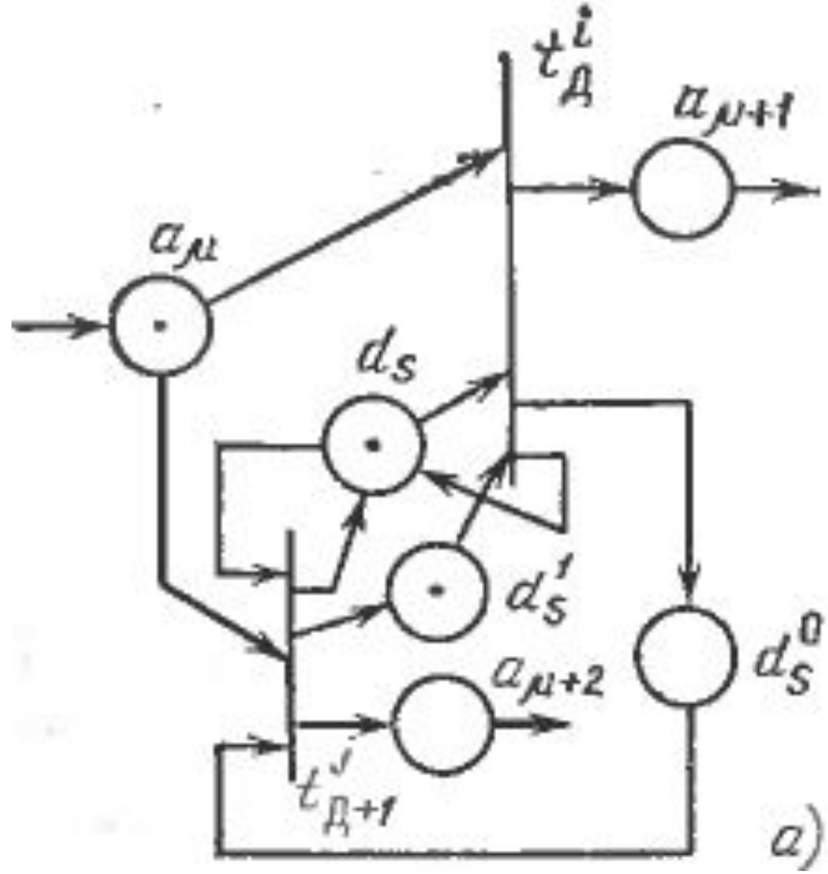
A_i ($\{\bar{p}_s, p_s\}$) и A_j ($\{\bar{p}_s, p_s\}$)

входные позиции для $t_Д^i$ ($t_Д^j$):

a_μ, d_s и d_s^1 (d_s и d_s^0)

выходные позиции для $t_Д^i$ ($t_Д^j$):

$a_{\mu+1}$($a_{\mu+2}$), d_s и d_s^0 (d_s и d_s^1)



Пример 2:

A_i не зависит от p_s, но меняет его.

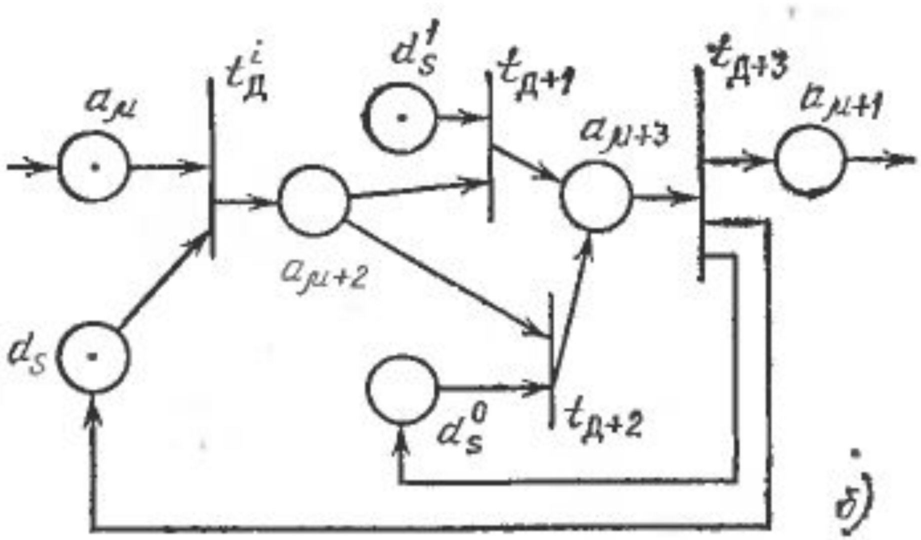
входные позиции $t_д^i$:

a_μ, d_s

Т.к. p_s не проверяется в начале, то:

1. удаляется метка из d_s^0 (или d_s^1)
2. помещается метка в d_s^0 (или d_s^1)

если после A_i $p_s = 0$ (или 1)



Пример 3:

A_i зависит от p_s, но не меняет его.

↓

новый тип дуг – **неизменяющиеся**.

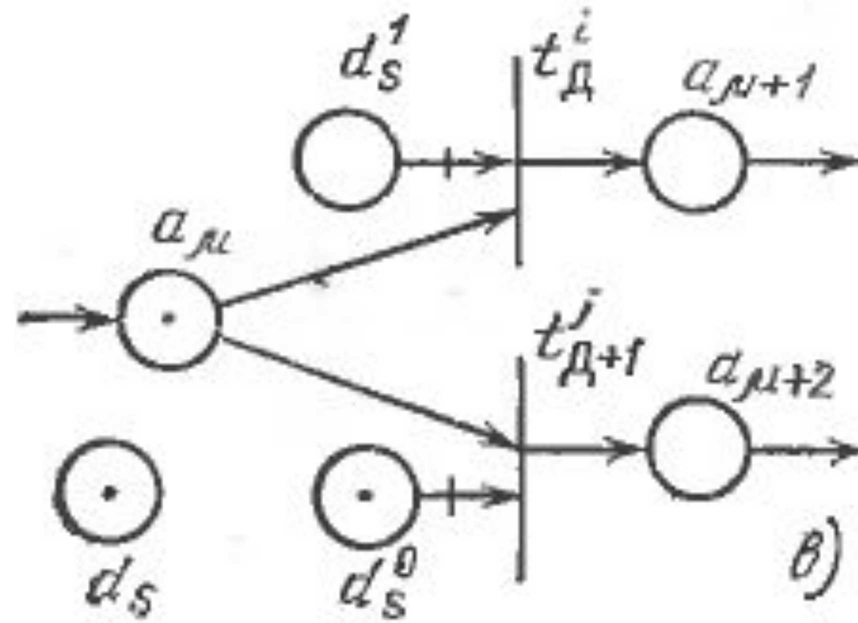
t_v с a_μ неизменяющейся дугой, то

в a_μ должна быть метка, но она **не удаляется**

Если A_i ($\{p_s\}$, $\{-\}$), то d_s^1 с $t_д^i$ неизменяющейся дугой

Если A_i ($\{\cancel{p}_s\}$, $\{-\}$), то d_s^0 с $t_д^j$ неизменяющейся дугой

d_s не используется



Введение **сдерживающих** (**тормозящих**) **дуг**.

Если t_v с a_μ - тормозящей дугой, то:

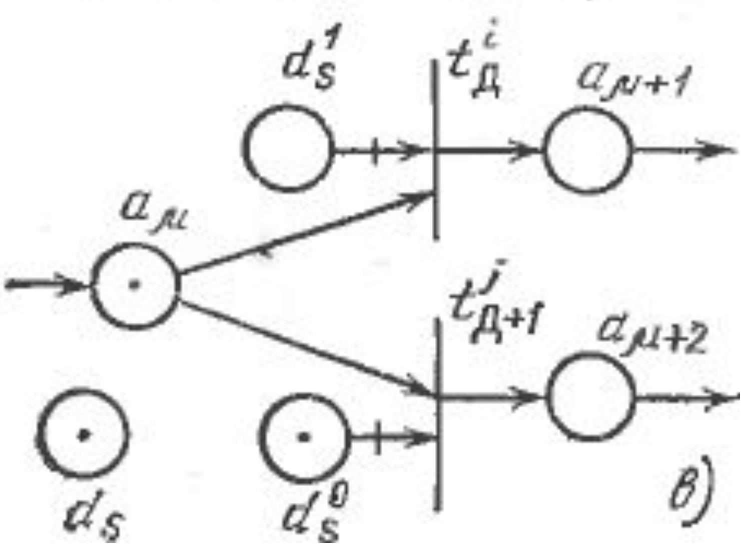
1. a_μ не должна содержать метки
2. D_s 2-мя позициями:

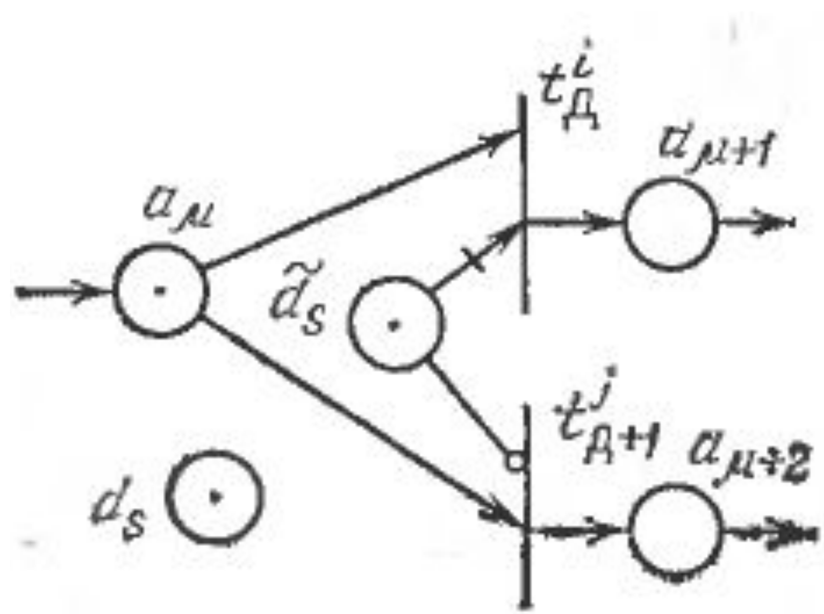
   а) d_s

   б) $\tilde{d}_s$ – содержит метку, если p_s=1

Пример 4:

A_i ({p_s}, {-}) из примера 3.





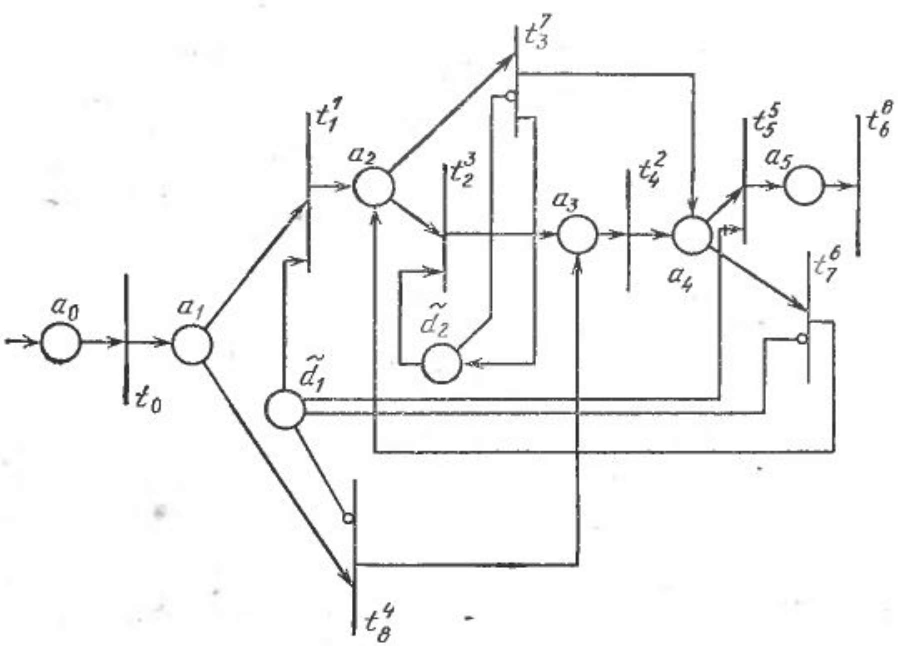
Пример 5:

Разветвленный последовательный процесс:

1. Все A_i используют собственные ФР
2. $A_1, A_3, A_4, A_5, A_6, A_7$ – зависят от p_1 и p_2
3. A_1, A_3, A_7 – меняют p_j

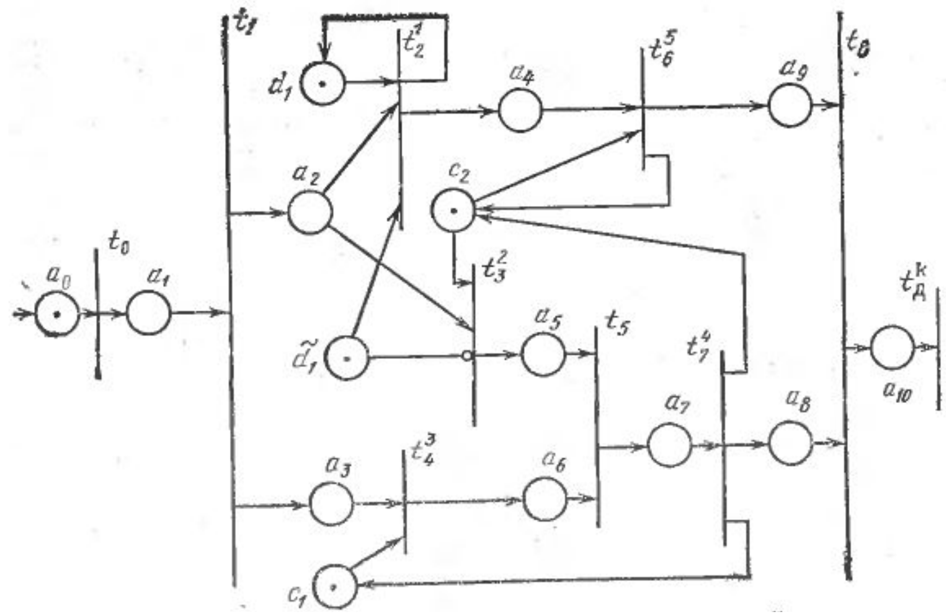
$A_1(\{p_1\},\{\overline{p}_1\})$; $A_3(\{p_2\},\{\overline{p}_2\})$; $A_4(\{\overline{p}_1\},\{-\})$;

$A_5(\{p_1\},\{-\})$; $A_6(\{\overline{p}_1\},\{-\})$; $A_7(\{\overline{p}_2\},\{p_2\})$



Пример 6:

УП с альтернативными и параллельными участками.



Обобщенная сеть Петри для описания неавтономного управляющего процесса.

Автономный УП

Неавтономный УП

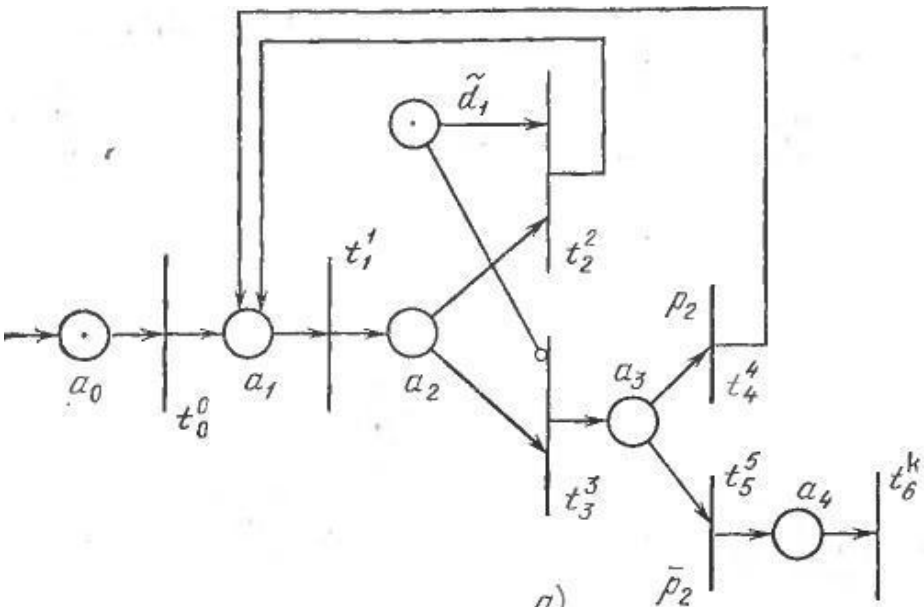
Описание неавтономного процесса:

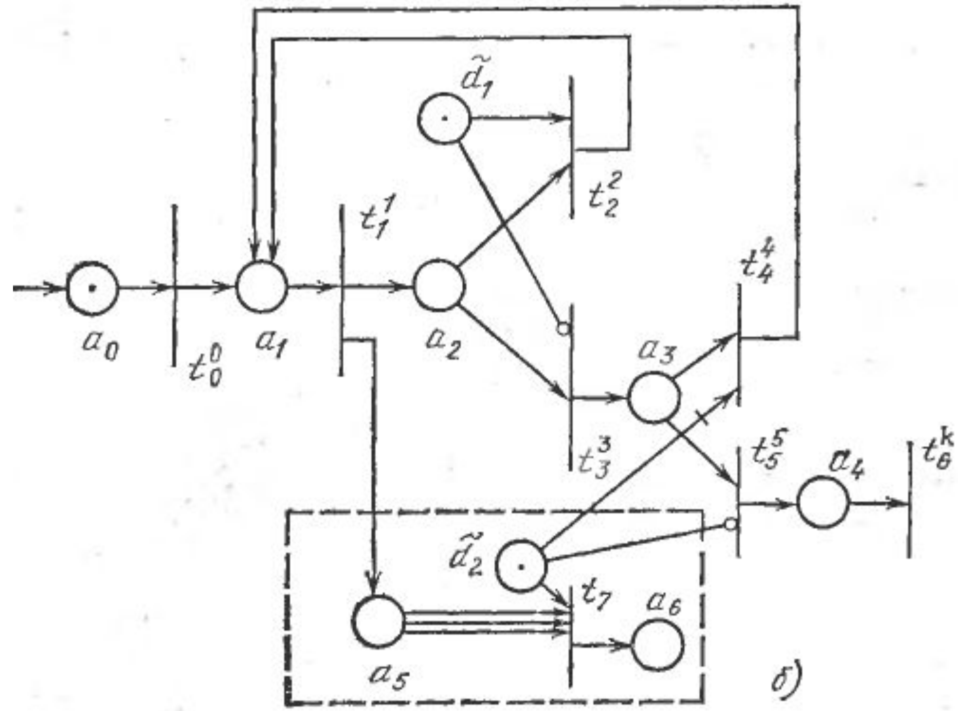
1. внеш. ЛУ (p_u) ↔ внеш. позиция h_u – метка есть, если p_u=1; нет при p_u=0
2. внеш. ЛУ ϵ $\{P_1\}$
3. есть внутренние и внешние ЛУ
4. если A_i выполняется при p_u=1 (0), то h_u соединяется с $t_д^{\,i}$ сдерживающей дугой
5. не включается позиция состояния внешнего ЛР
6. развитие процесса – зависит от начальной маркировки внутренних позиций и текущей маркировки внешних входных позиций
7. замена внешних входных позиций на предикаты, зависящие от внешних ЛУ

Если не определено влияние A_i на значение p_s:

1. возможное изменение p_s – это безразличное значение (p_s) в $\{P_2^{\,i}\}$
2. позиция состояния D_s - в описании параллельного процесса
3. на время выполнения $t_д^{\,i}$ метка из d_s удаляется
4. позиция d_s аналогична внешней позиции

Пример:





ФР – собственные

ЛР D_1 – внутренний

ЛР D_2 – изменяется A_1 → изменяется p_2

Задано: $A_2(\{p_1\},\{\overline{p_1}\})$

$A_3(\{p_1\},\{-\})$

$A_4(\{p_2\},\{-\})$

$A_5(\{p_2\},\{-\})$

ЛР D_2 – счетчик → позиция d_2 - внутренняя

k – константа для сравнения

k-кратная дуга между a_5 и t_7

Пример:

Одни и те же ресурсы запрашиваются разными параллельными подпроцессами.

Для этого:

в d_s n меток в начальной маркировке – n – максимальное число продпроцессов, немонопольно владеющих D_s.

↓

d_s – входная и выходная позиция для n переходов

дуга кратности n соединяет переход и позицию d_s при монопольном владении D_s

---

$П_1$ и $П_2$ немонопольно владеют D_1 при A_3 или A_4 и A_7

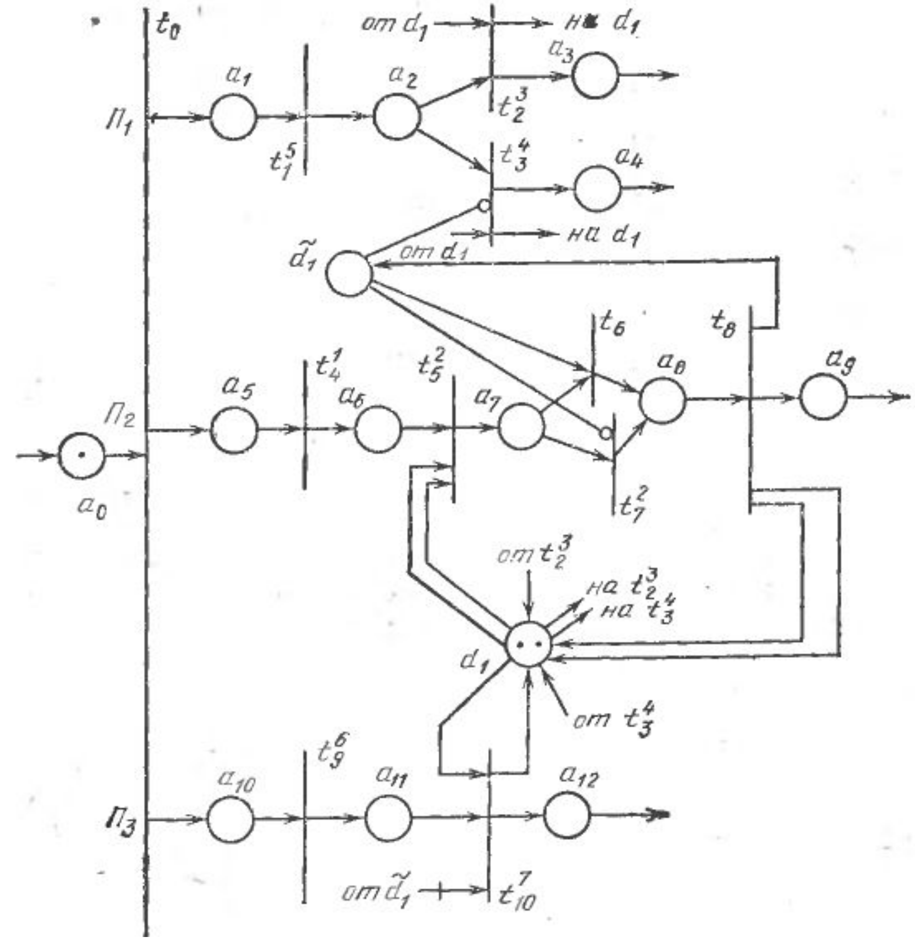
$A_3(\{p_1\},\{-\})$ $A_4(\{p_1\},\{-\})$

$A_7(\{p_1\},\{-\})$ $A_2(\{-\},\{p_2\})$

взаимодействие параллельных подпроцессов – 2-е метки в d_1 и 2-кратные дуги к t_2^5

↓

одновременно t_2^5 и t_{10}^7



t_2^5 – удаляет обе маркировки из d_1 – монопольное использование D_1

маркировка $\tilde{d}_1$ не изменяется при

Граф обобщенной сети Петри содержит:

1. длительные переходы
2. примитивные переходы
3. основные внутренние позиции
4. ресурсные внутренние позиции
5. основные дуги
6. неизменяющие дуги заданной
7. сдерживающие дуги кратности
8. длительный переход – это процедура
9. предикаты у $t_д^i$, если A_i зависит от внешних ЛУ
10. примитивные переходы – переходы распараллеливания и соединения – задание структуры процесса
11. маркировка a_μ (основные) и c_j, d_s, $\tilde{d}_s$ (внутренние ресурсные) – полное состояние УП
12. дуги – последовательность выполнения процедур и их взаимодействие с ФР и ЛР.

Свойства:

Временных сетей с переходами, помеченными предикатами и операциями, и дугами разных типов.

Особенность:

1. в описание процесса вводятся используемые им ресурс

2. учитывается влияние процедур процесса на состояние ресурсов

Получение правильного управляющего процесса.

Граф достижимых маркировок сети Петри.

Недопустимые – <u>тупиковые</u> состояния.

<u>Причины возникновения</u> тупиковых состояний.

<u>Методы анализа</u> сетей Петри.

<u>Дерево достижимых состояний сетей Петри</u>.

$M_0 \quad t_l \quad M_1$

ω – бесконечное число меток

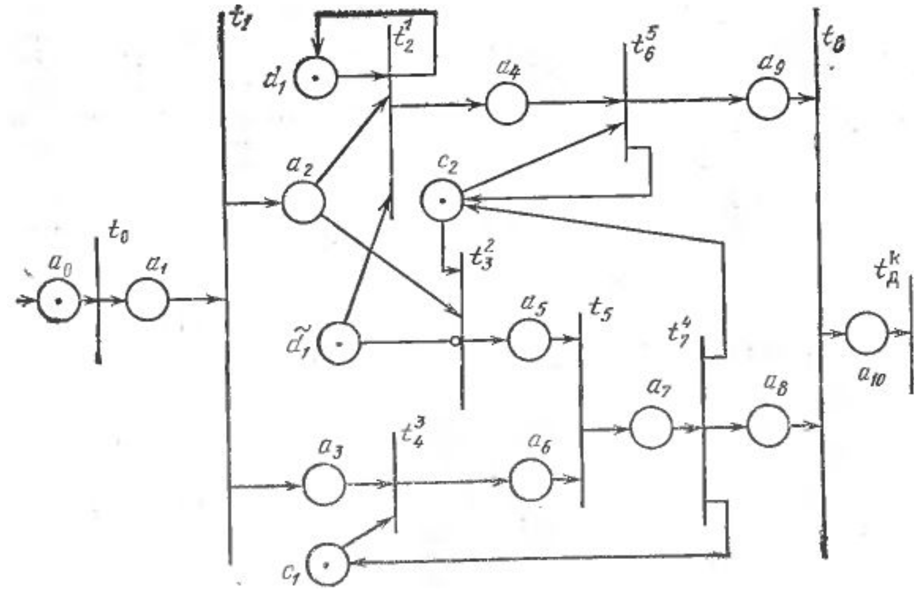
<u>Неограниченные и ограниченные сети Петри.</u>

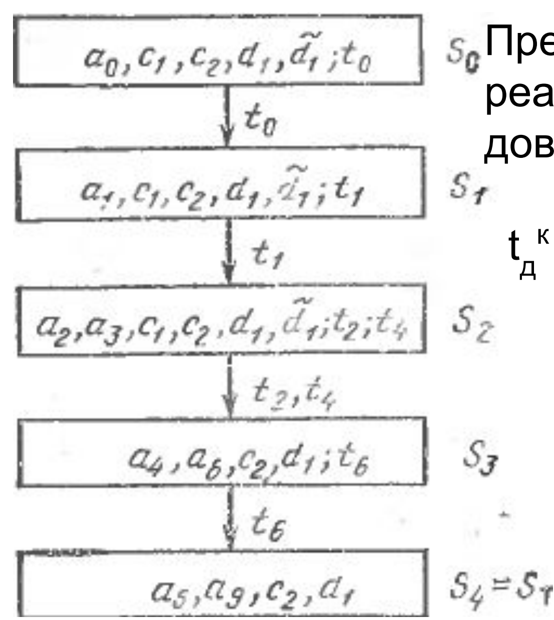
<u>Описание графа достижимых маркировок:</u>

G_N
M_i
a_μ
S_i

<u>Влияние структуры процесса на наличие тупиковых состояний.</u>

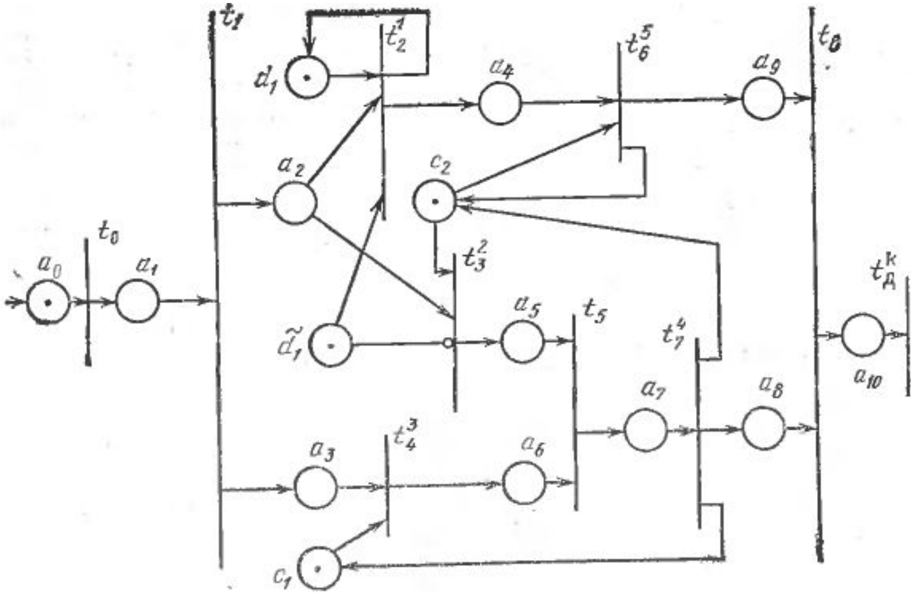
<u>Пример:</u>

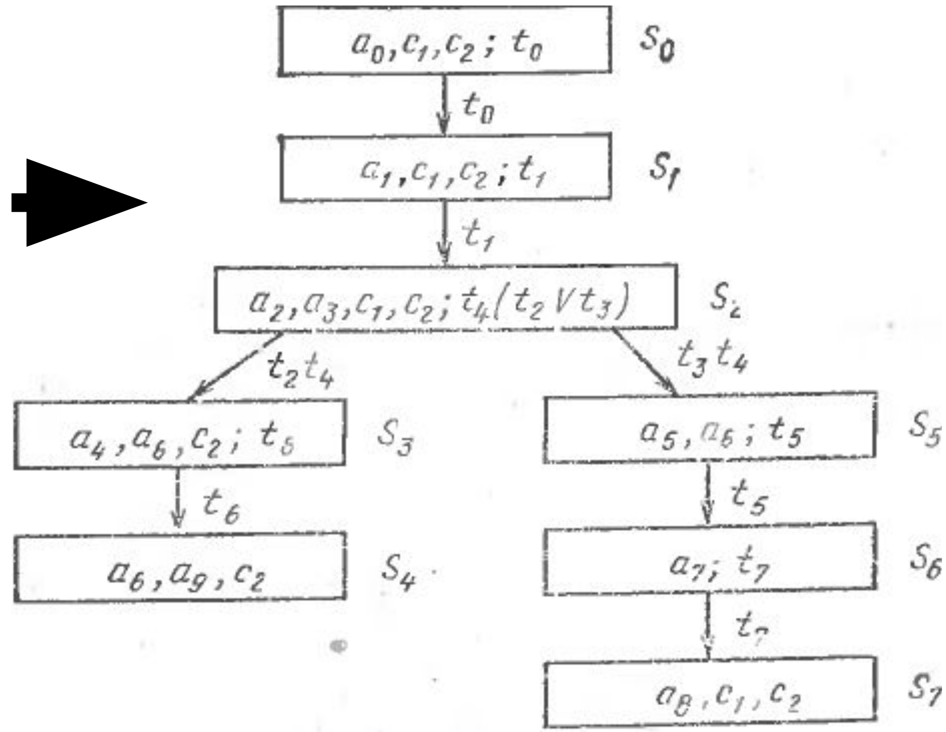




Предположение – время реализации всех переходов одинаково.

$t_Д^к$ при p=1 в S_4 - тупик





Для p=0 в начальной маркировке, т. е. в $\tilde{d}_s$ нет метки – вместо t_2 будет активизирован t_3.

Реализация активизированных переходов завершается одновременно.

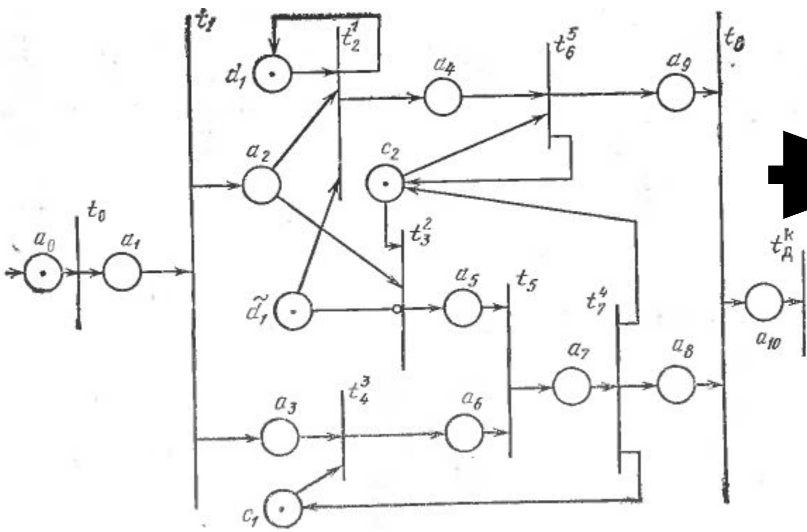
левая ветвь – p=1

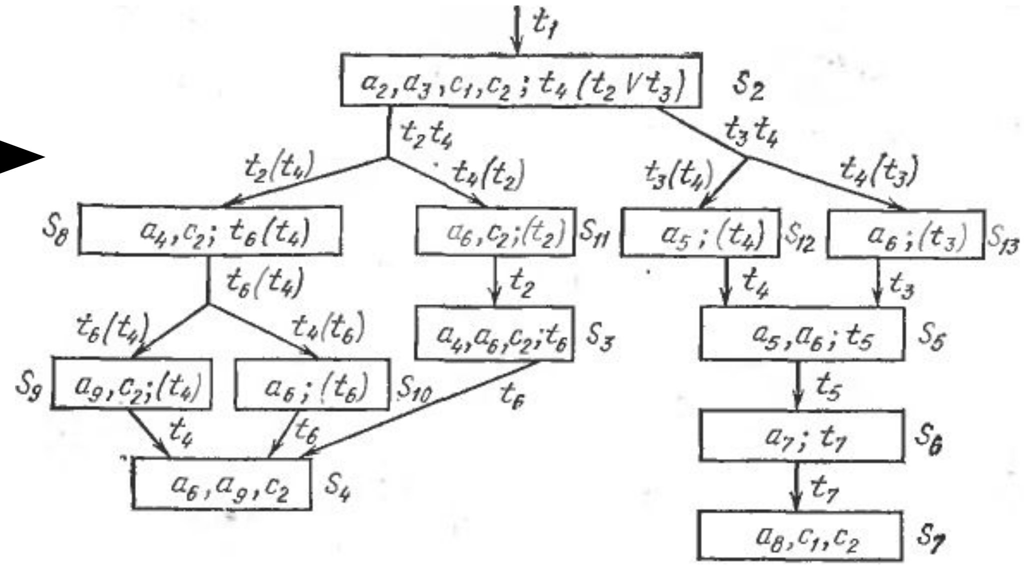
правая ветвь – p=0

Это **<u>граф статических состояний процесса.</u>**

S_4 и S_7 – тупиковые

Граф, содержащий статические и промежуточные состояния.





Это **динамический граф.**

S_8 a_3 a_6 t_4

S_4 и S_7 – тупиковые

Исходящие дуги – переходы, переходящие в стадию реализации.

Входящие дуги – переходы, закончившие реализацию.

В скобках – переходы, продолжающие реализацию.

↓

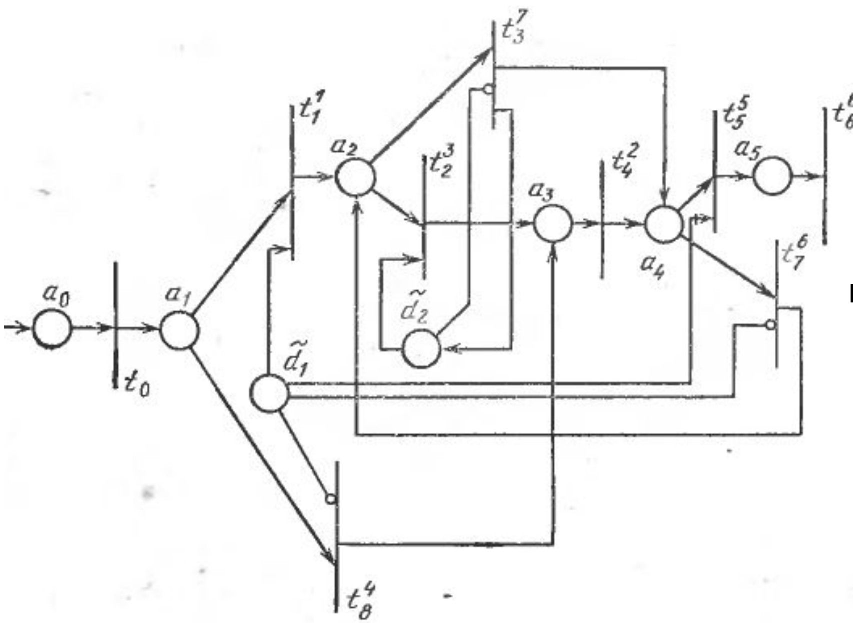
Неустойчивые состояния.

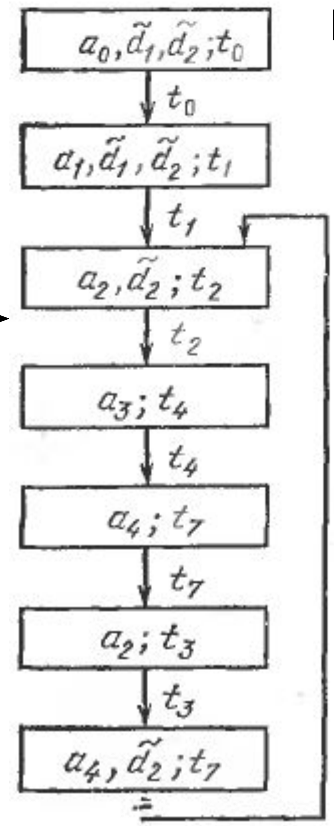
Причина – недопустимая структура процесса.

Требования к правильной структуре процесса.

<u>Другая причина</u> недостижимости конечного состояния – **циклы**.

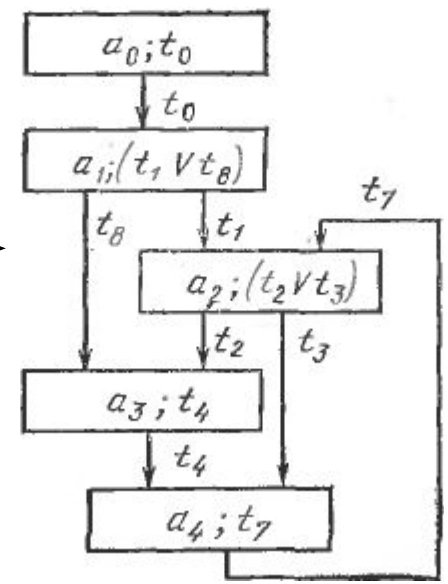
<u>Пример:</u>



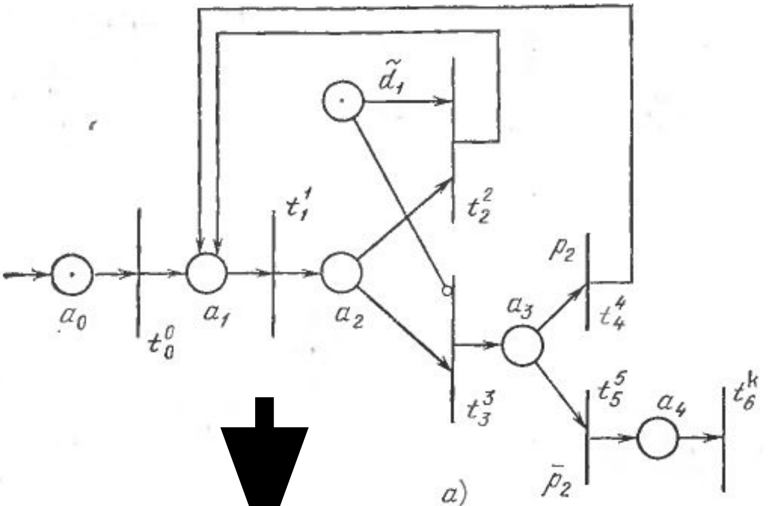


Для фиксированной начальной позиции d_1 и d_2.

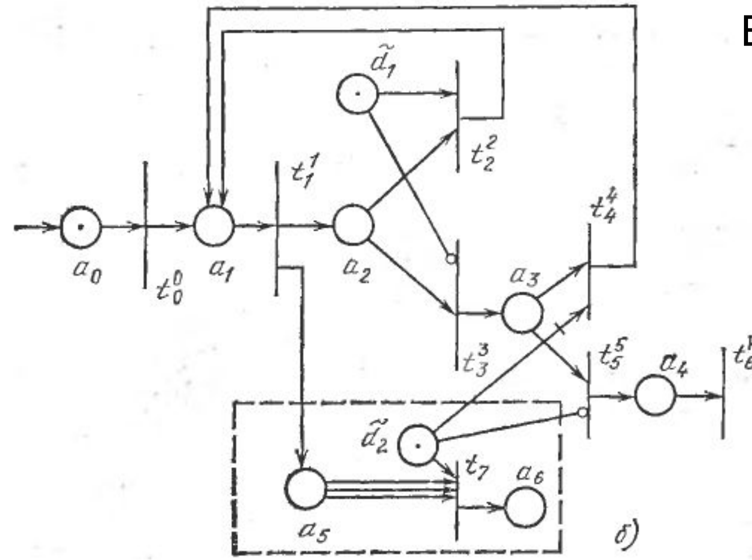
Полный граф достижимости

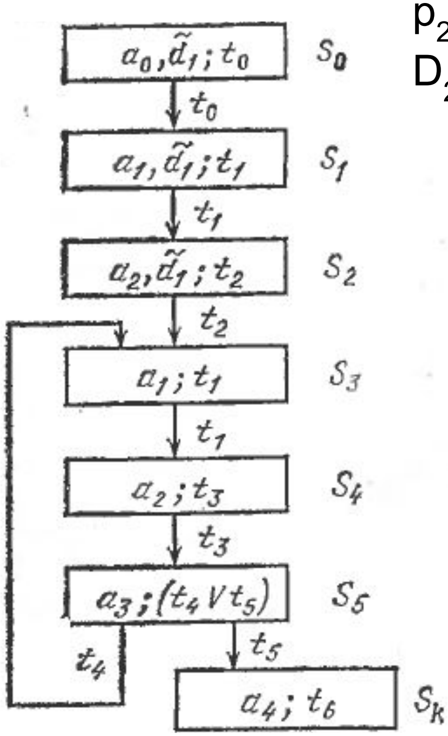


Пример:



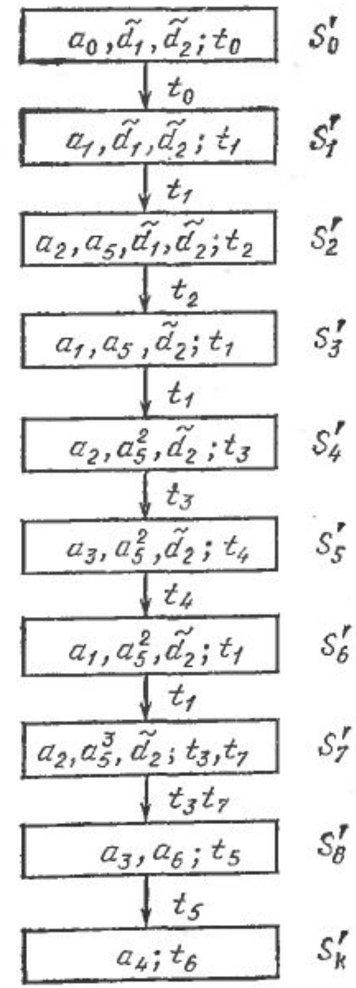
Есть информация о D_2 и его взаимодействии с УП





p_2=1

D_2 – внешний ЛР



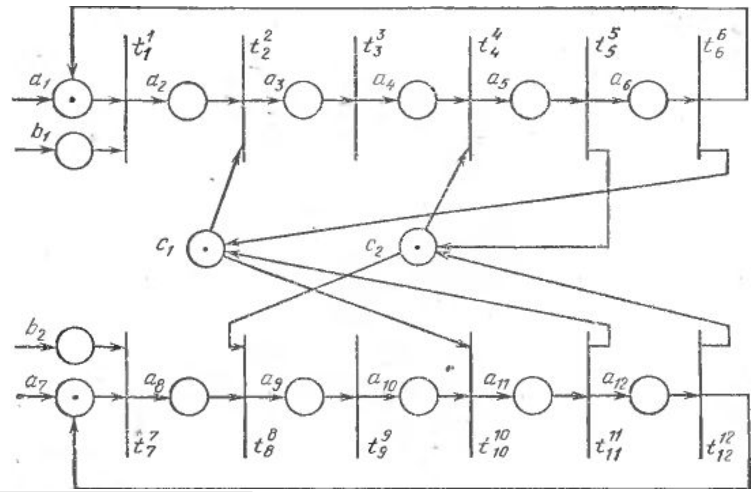
Тупиковые состояния, вызываемые разделением функциональных ресурсов.

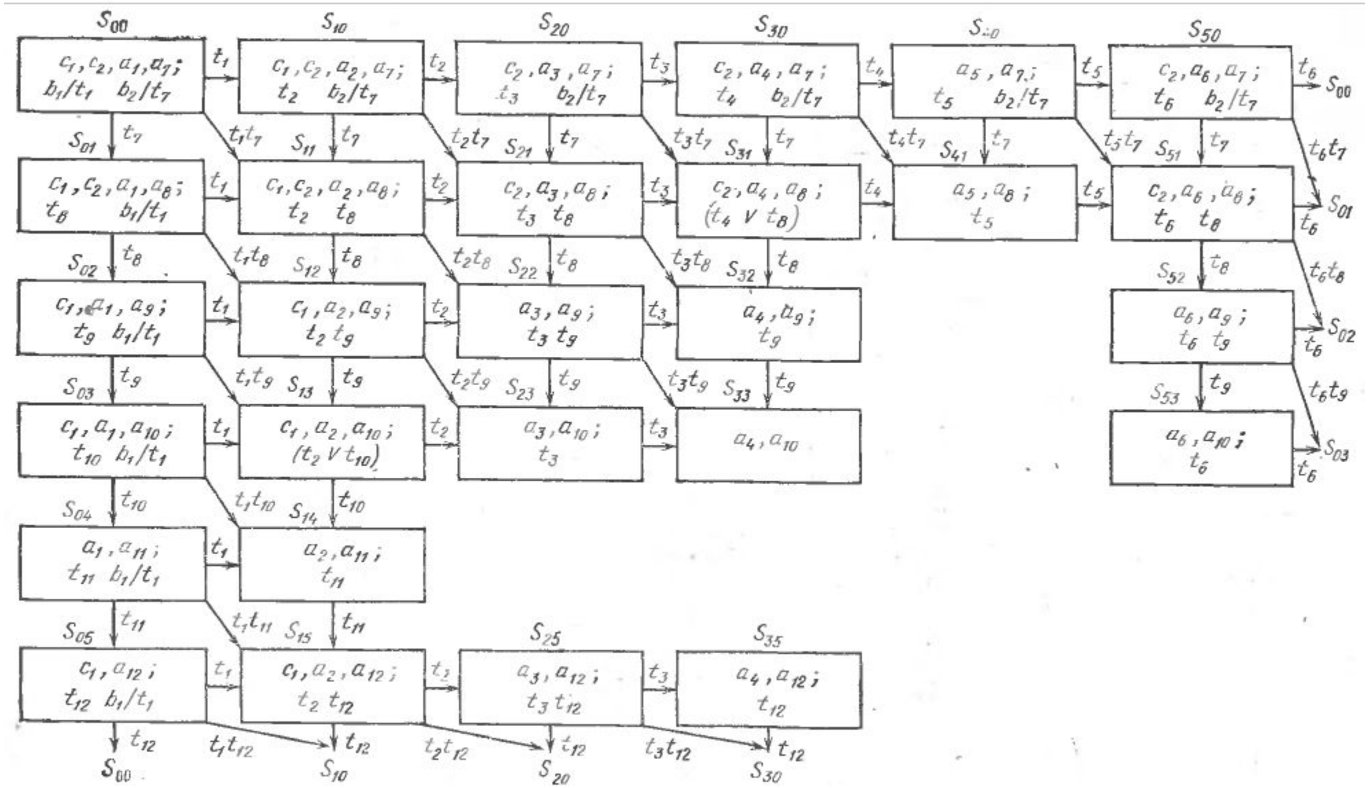
Пример:

$П_1$ и $П_2$ – асинхронные циклические процессы

C_1 и C_2 – разделяемые ФР

b_1 и b_2 – внешние входные позиции





$П_1$ – по горизонтали

$П_2$ – по вертикали

S_{ij} – вершины, состояния, где i – номер в $П_1$, а j – в $П_2$

Классификация состояний в графе достижимых маркировок сети Петри.

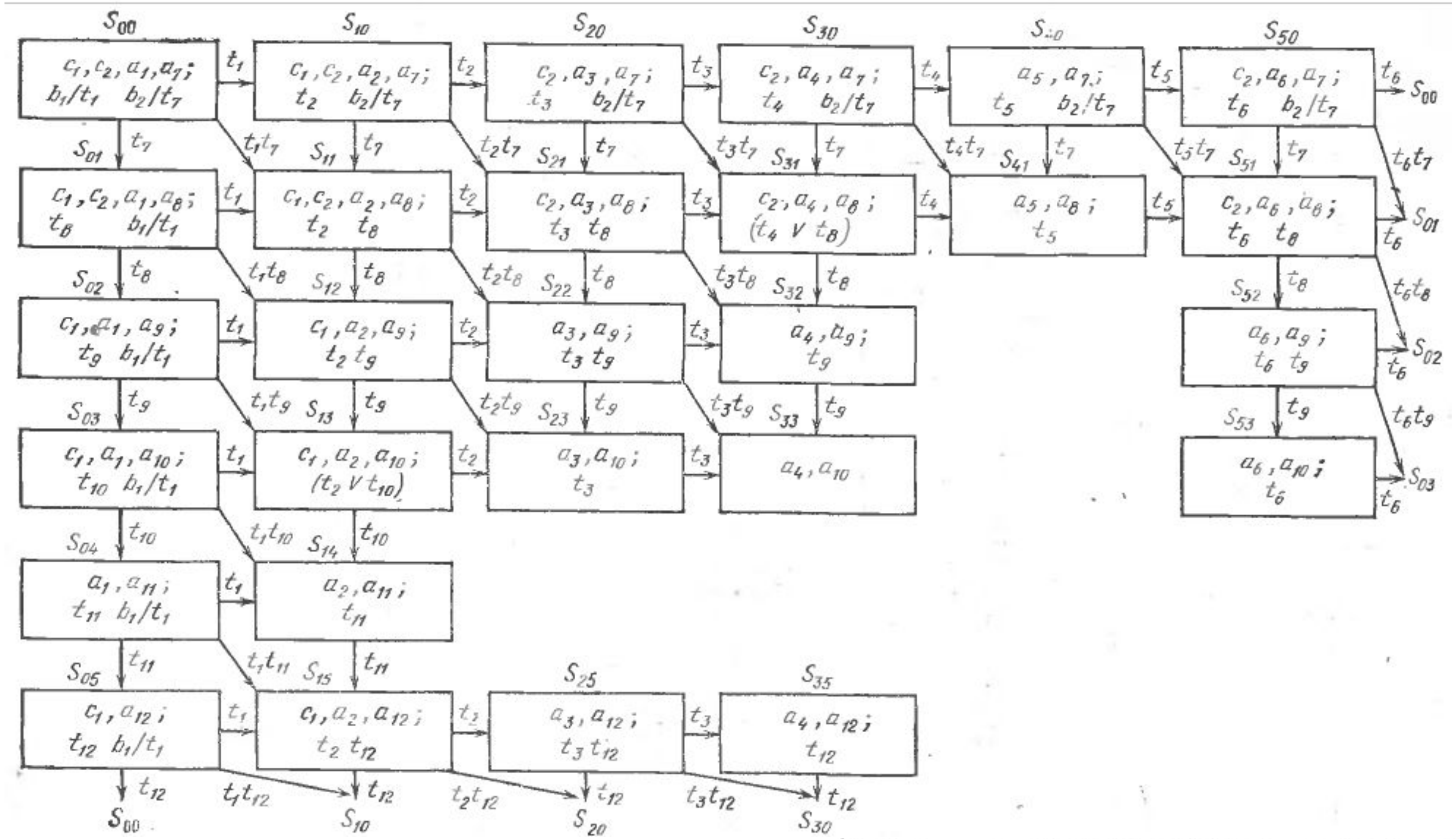
1. Состояние блокировки – $S_б$:
   a_μ t_i
2. Состояние взаимной блокировки – $S_{в.б}$
3. Состояние полной взаимной блокировки – $S_{п.в.б}$
4. Тупиковое состояние – $S_т$ – это $S_{в.б}$ и $S_{п.в.б}$
5. Предтупиковое состояние – $S_{п.т}$
   $Q_з\{S_т, S_{п.т}\}$ – множество запрещенных состояний
6. Опасное состояние - $S_{оп}$, если:
   $S_v \xrightarrow{ребро} S_u$
   и
   $S_v \in Q_з$, а $S_u \notin Q_з$
   $Q_{оп}$ – множество опасных состояний
7. Безопасное состояние
8. Состояние конфликта – $S_{кн}$

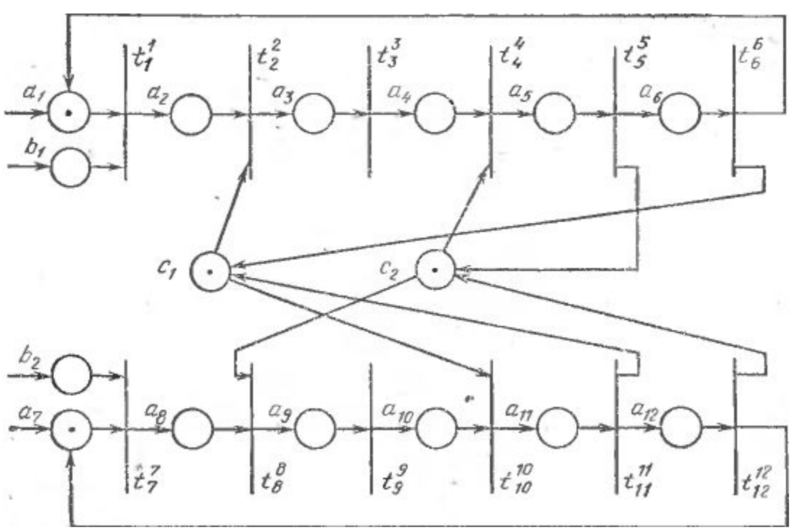
Опасные отрезки пути в графе

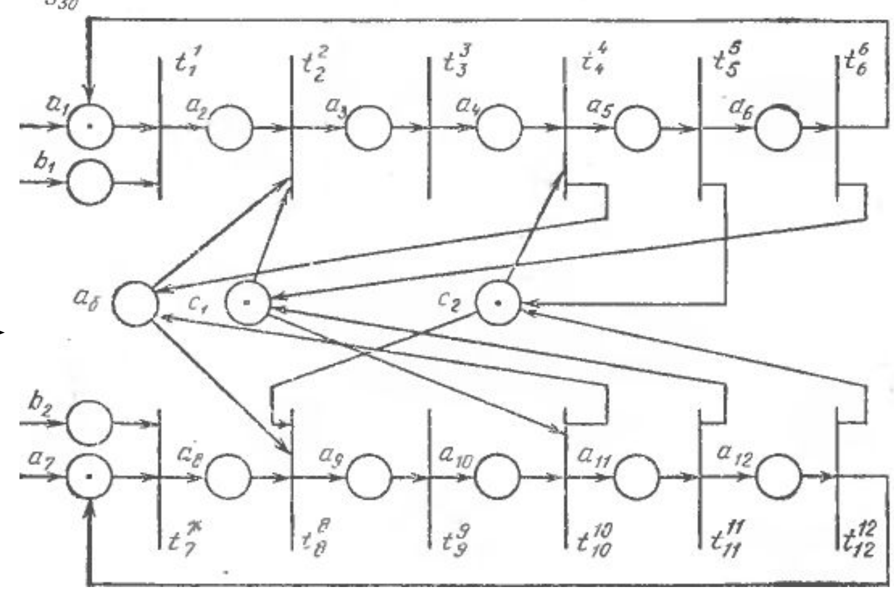
Корень опасных отрезков – $S_{к.оп}$

Дополнительная блокирующая позиция – $a_б$

Пример:


S00
c1, c2, a1, a7; b1/t1 b2/t7
S10
c1, c2, a2, a7; t2 b2/t7
S20
c2, a3, a7; t3 b2/t7
S30
c2, a4, a7; t4 b2/t7
a5, a7; t5 b2/t7
S50
c2, a6, a7; t6 b2/t7
S01
c1, c2, a1, a8; t8 b1/t1
S11
c1, c2, a2, a8; t2 t8
S21
c2, a3, a8; t3 t8
S31
c2, a4, a8; (t4 V t8)
S41
a5, a8; t5
S51
c2, a6, a8; t6 t8
S02
c1, a1, a9; t9 b1/t1
S12
c1, a2, a9; t2 t9
S22
a3, a9; t3 t9
S32
a4, a9; t9
S52
a6, a9; t6 t9
S03
c1, a1, a10; t10 b1/t1
S13
c1, a2, a10; (t2 V t10)
S23
a3, a10; t3
S33
a4, a10
S53
a6, a10; t6
S04
a1, a11; t11 b1/t1
S14
a2, a11; t11
S05
c1, a12; t12 b1/t1
S15
c1, a2, a12; t2 t12
S25
a3, a12; t3 t12
S35
a4, a12; t12


a1
b1
a2
a3
a4
a5
a6
c1
c2
b2
a7
a8
a9
a10
a11
a12


a1
b1
a2
a3
a4
a5
a6
a0
c1
c2
b2
a7
a8
a9
a10
a11
a12