

# Сети Петри

Определение Сеть Петри  $S$  является четверкой,  $S = (P, T, I, O)$ .

$P = \{p_i\}$  – конечное множество позиций

$T = \{t_i\}$  – конечное множество переходов

Множество позиций и множество переходов не пересекаются.

$I$  - отображает переход  $t_i$  в множество позиций,

$I(t_i) \text{ -- } I: T \rightarrow P_\infty$  - является входной функцией – отображением из переходов в комплекты позиций.

$O$ -отображает переход  $t_j$  в множество позиций

$O(t_j) \text{ -- } O: T \rightarrow P_\infty$  - выходная функция – отображение из переходов в комплект позиций.

# Сети Петри

Определение Граф  $G$  сети Петри есть двудольный ориентированный мультиграф,  $G=(V, A)$ , где

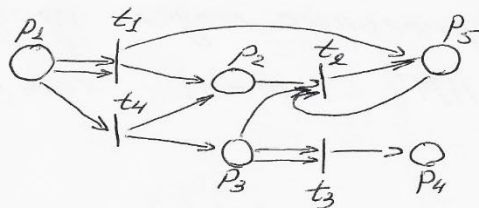
$V = \{v_i\}$  – множество вершин

$A = \{a_{i,j}\}$  – комплект направленных дуг,  $a_{i,j} = (v_j, v_i)$ , где  $v_j, v_i \in V$ .

Множество  $V$  разбито на два непересекающихся подмножества  $P$  и  $T$ , таких, что  $V = P \cup T$ ,  $P \cap T = \emptyset$ , и для любой направленной дуги  $a_{i,j} \in A$ , если  $a_{i,j} = (v_j, v_i)$ , тогда либо  $v_j \in P$  и  $v_i \in T$ , либо  $v_j \in T$  и  $v_i \in P$ .

- . Дуга, направленная от позиции  $p_i$  к переходу  $t_j$ , определяет позицию, которая является входом перехода. Кратные входы в переход указываются кратными дугами из входных позиций в переход. Выходная позиция определяется дугой от перехода к позиции. Кратные выходы также представлены кратными дугами. При большом числе дуг – показывается одна дуга и указывается только их количество.

Рис



$$C\pi = (P, T, \bar{I}, O)$$

$$P = (p_1, p_2, p_3, p_4, p_5)$$

$$T = (t_1, t_2, t_3, t_4)$$

$$\bar{I}(t_1) = (2 \cdot p_1)$$

$$O(t_1) = (p_2, p_5)$$

$$\bar{I}(t_2) = (p_2, p_3, p_5)$$

$$O(t_2) = (p_5)$$

$$\bar{I}(t_3) = (2 \cdot p_3)$$

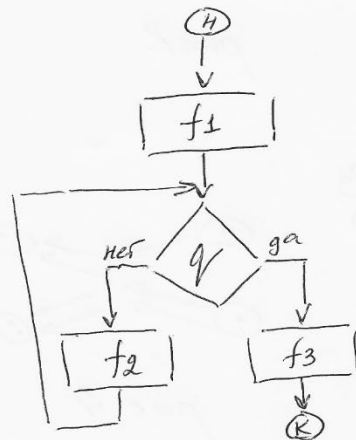
$$O(t_3) = (p_4)$$

$$\bar{I}(t_4) = (p_1)$$

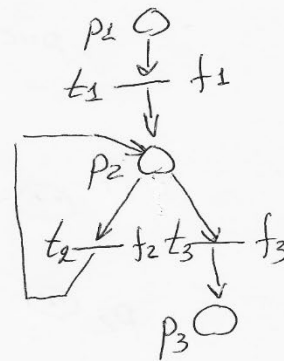
$$O(t_4) = (p_2, p_3)$$

Рис

Формальные модели программ.  
 Схема ПРГ  $m = n!$  Сеть Пету

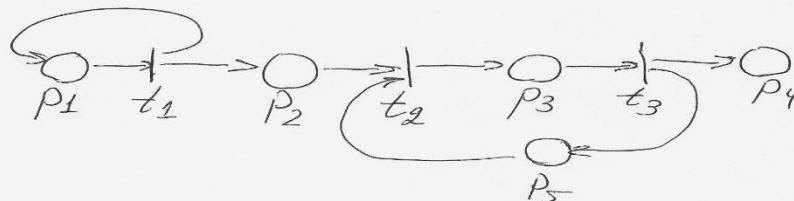

$$\begin{aligned} f_1: & \quad in(n) \\ & \quad m := 1 \\ f_2: & \quad m := m \times n \\ & \quad n := n - 1 \\ f_3: & \quad out(m) \\ q: & \quad if(n \leq 0) \end{aligned}$$

Сеть Петри



Рис

Модель СМО.



$t_1$  - генерация заявки в  
Источнике

$t_2$  - начало обслуживания  
заявки

$t_3$  - завершение обслужи-  
вания

# Маркированная СП

Определение Маркировка  $\mu$  сети Петри  $S = (P, T, I, O)$  есть функция, отображающая множество позиций  $P$  в множество неотрицательных целых чисел  $N$ :  $\mu: P \rightarrow N$ .

Маркировка  $\mu$  может быть также определена как вектор  $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ , где  $n = |P|$  и каждое  $\mu_i \in N$ ,  $i=1, \dots, n$ .

Вектор  $\mu$  определяет для каждой позиции  $p_i$  СП количество меток (фишек) в этой позиции. Состояние СП задается ее маркировкой  $\mu$ .

**На графе сети Петри фишки изображаются маленькой точкой в кружке, который представляет позицию сети Петри.**

Определение Маркированная сеть Петри  $M = (S, \mu)$  есть совокупность сети Петри  $S = (P, T, I, O)$  и маркировки  $\mu$  и может быть записана в виде  $M = (P, T, I, O, \mu)$

# Выполнение СП

Сеть Петри выполняется посредством **запусков** переходов.

Переход может запускаться только в том случае, когда он ***разрешен***.

Переход называется разрешенным, если **каждая из его входных позиций** имеет число фишек не менее числа дуг из позиции в переход.

**Определение** Переход  $t_j \in T$  в маркированной сети Петри  $C = (P, T, I, O)$  с маркировкой  $\mu$  разрешен, если для всех  $p_i \in P$   $\mu(p_i) \geq \#(p_i, I(t_j))$ .

На каждом **шаге выполнения** СП запускается **только один** из разрешенных переходов.

Запуск перехода заменяет маркировку  $\mu$  СП на новую маркировку  $\mu'$ :

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j)).$$

Алгоритм формирования новой маркировки.

1. Из **каждой входной позиции** запускаемого перехода **удаляется** количество фишек равное количеству дуг, соединяющих эту позицию и переход .
2. В **каждую выходную позицию** запускаемого перехода **добавляется** количество фишек равное количеству дуг, соединяющих эту позицию и переход .



# Множество достижимости

- Определение Для сети Петри  $S = (P, T, I, O)$  с маркировкой  $\mu_0$  маркировка  $\mu_1$  называется непосредственно достижимой из  $\mu_0$ , если существует переход  $t_j \in T$ , такой что  $\mu_1 = \delta(\mu_0, t_j)$ .

Можно распространить это понятие на определение множества достижимых маркировок данной маркированной СП. Если  $\mu_1$  непосредственно достижима из  $\mu_0$ , а  $\mu_2$  - из  $\mu_1$ , то говорят, что  $\mu_2$  достижима из  $\mu_0$ .

- Определение Множество достижимости  $R(S, \mu)$  СП  $S = (P, T, I, O)$  с маркировкой  $\mu$  есть множество всех маркировок, достижимых из  $\mu$ .

Маркировка  $\mu'$  принадлежит  $R(S, \mu)$ , если существует какая-либо последовательность запусков переходов, изменяющих  $\mu$  на  $\mu'$ .

## **Задача**

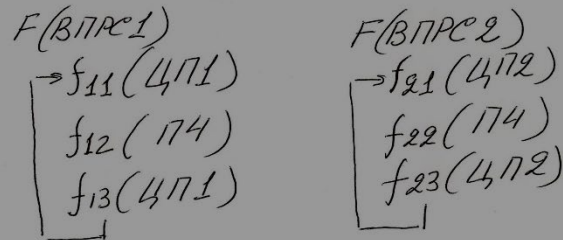
Изделие состоит из 3-х блоков, для изготовления которых используются 4-ре материала. Процесс изготовления начинается с заказа материалов, после получения которых начинается изготовление блоков. По мере готовности любых 2-х блоков начинается сборка изделия, которая завершается подключением по готовности третьего блока. Так же по готовности третьего блока формируется заявка в отдел доставки, где изготавливается упаковка, комплект документов и по готовности изделия осуществляется доставка заказчику и завершается весь процесс.

# Взаимодействие ВПРС

## Модель "Исключение КУ"

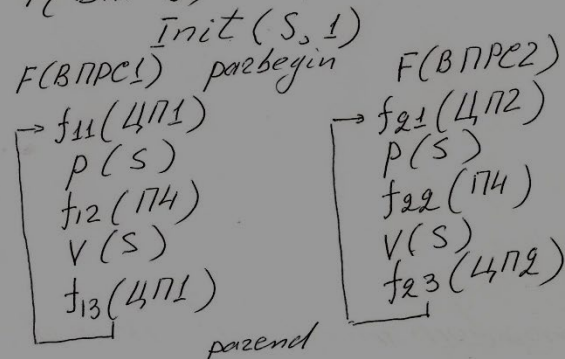
1. Ресурсы ВС:

ЦП1, ЦП2 - индивидуальные РП  
П4 - ОРРВ



2. Семафор:  $S$   
 $S = \begin{cases} 0 - \text{П4 занята} \\ 1 - \text{П4 свободна} \end{cases}$

3.  $F(\text{ВПРС0})$



# Взаимодействие ВПРС

В рамках модели «Исключение критических ресурсов» рассмотреть

Задачу взаимодействия:

**ВПРС** - три ВПРС по структуре соответствуют представленным на предыдущем слайде

**ВС:**

- Три ПРЦ – индивидуальные РЕС
- Две ПЧ (эквивалентные по функциям) - ОРРВ

Необходимо:

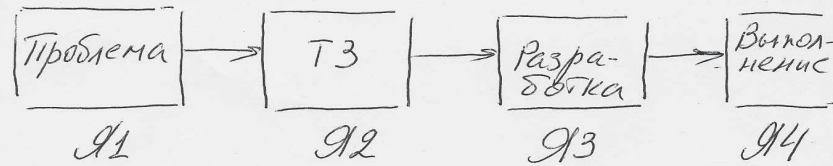
1. Написать ПРГ ВПРС с учетом разделения использования РЕС
2. Построить граф сети ПЕТРИ и доказать корректность взаимодействий ВПРС

# ЛИТЕРАТУРА

Питерсон Дж.

Теория сетей Петри и моделирование систем. М, Мир, 1984

рис



А1 - Специалист проблемной области

А2 - Системный аналитик

А3 - Разработчик ПО.

А4 - Код, выполнение на БМ

А<sub>i</sub> → А<sub>j</sub>  
транслятор

Транслятор:

Компиляция  
Интерпретация

## ***ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ***

*Предметно (проблемно)-ориентированный язык* (Domain Specific Language, DSL) – это язык программирования или моделирования, созданный для использования в рамках конкретной предметной области.

Примеры DSL:

HTML - программирование сайтов

SQL - программирование действий с реляционной БД

LISP - язык обработки списков

GPSS - язык разработки имитационных моделей

ФОРТРАН - переводчик формул. Вычислительные задачи

1с - язык разработки учетных систем автоматизации деятельности

проблемно-ориентированный язык для формирования полисов страхования

проблемно-ориентированный язык для вычисления зарплаты

проблемно-ориентированный язык для формирования счетов

## **Достоинства.**

- Проблемно-ориентированные языки позволяют выражать решения (разрабатывать программные системы) на уровне абстракций проблемной области. Это позволяет специалистам предметной области разрабатывать и развивать программные системы.
- Промышленная разработка систем – сокращение затрат на разработку и сопровождение

## **Недостатки.**

- Затраты на разработку проблемно-ориентированного языка (трансляторы, средства поддержки пользователей)
- Затраты на изучение нового языка с ограниченной областью применения
- Возможные потери эффективности на этапе исполнения по сравнению с программным обеспечением на базе универсальных языков.



Проблемно-ориентированные языки по реализации подразделяются на:

- Внешние.
- Внутренние.

Внешние DSL - предполагает разработку информационных объектов, операторов манипулирования, трансляторов и средств поддержки программирования, т.е. полноценная разработка языка программирования.

Внутренние DSL - написаны на базовом языке и являются лишь дополнением базового языка программирования, поэтому остается нерешенной проблема адекватного представления объектов предметной области средствами универсального языка программирования.

Преимущество внешнего DSL для использования программистами-непрофессионалами состоит в том, что есть возможность отказаться от сложности базового языка, и создать DSL, который будет понятен и удобен пользователям.