

Естественные модели параллельных вычислений

Лекция 5 :: Сети Петри

Ершов Н.М.

ershovnm@gmail.com

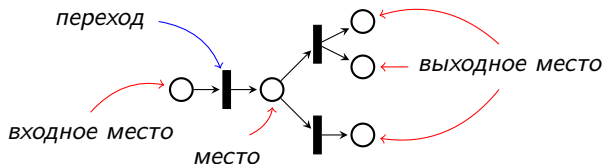
*Проект комиссии Президента РФ
по модернизации и техническому развитию экономики России
«Создание системы подготовки высококвалифицированных кадров в
области суперкомпьютерных технологий и специализированного
программного обеспечения»*

- Сети Петри представляют собой простое и удобное средство для моделирования разнообразных распределенных систем и процессов.
- Эта модель была придумана немецким ученым Карлом Петри в 1939 году для описания химических процессов.
- Официальная история сетей Петри началась в 1962 году, когда Карл Петри защитил диссертацию «Kommunikation mit Automaten», в которой им и было введено понятие сети Петри.



Структура сети Петри: места и переходы

- Сеть Петри представляет собой *двудольный ориентированный граф*, содержащий вершины двух типов — *места* (обозначаются кружками) и *переходы* (обозначаются прямоугольниками).
- Любая дуга ведет либо от вершины–места в вершину–переход, либо наоборот.
- Дуги, соединяющие два места или два перехода, запрещены.
- Места, у которых нет входящих дуг, называются *входными*.
- Места, у которых нет исходящих дуг, называются *выходными*.



Структура сети Петри: метки

- Каждое место сети Петри может содержать ноль или более *меток* (англ. tokens). Распределение меток по местам сети называется ее *разметкой*. Работа сети начинается с начальной разметки.
- Метки могут переноситься с одного места на другое, используя следующую схему:
 - ▶ переход является *активным*, если каждое его входное место содержит по крайней мере одну метку (более точно — по одной метке на каждую входящую в этот переход дугу);
 - ▶ активный переход может *сработать*, при срабатывании переход поглощает по одной метке с каждого своего входного места и размещает по одной метке на каждое свое выходное место (по одной метке на каждую исходящую дугу);
 - ▶ в каждый момент времени для срабатывания из всех активных переходов недетерминированным образом выбирается один, если активных переходов нет, то работа сети на этом завершается.

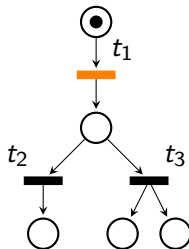
Последовательность срабатываний

- Припишем каждому переходу сети Петри некоторый уникальный символ (например, пронумеруем их).
- Последовательность символов σ , в которой i -ый символ равен символу перехода, сработавшему на i -ом шаге работы сети, называется последовательностью срабатываний сети Петри.
- Последовательность срабатываний однозначно определяет последовательность разметок μ_i ($i = 0, 1, 2, \dots$), где μ_0 является начальной разметкой.
- Тот факт, что после срабатывания t -го перехода разметка μ преобразуется в разметку μ' , будем обозначать кратко

$$\mu \xrightarrow{t} \mu'.$$

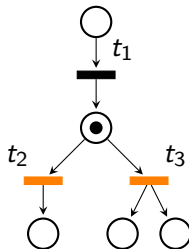
Пример работы сети Петри

- На рисунке показан пример работы сети Петри для последовательности срабатываний $\sigma = [t_1, t_3]$.
- Активные переходы в каждом случае выделены оранжевым цветом.
- Заметим, что для той же сети имеется еще только одна возможная последовательность срабатываний $\sigma = [t_1, t_2]$.



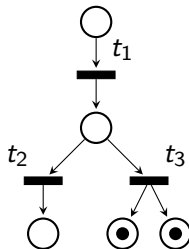
Пример работы сети Петри

- На рисунке показан пример работы сети Петри для последовательности срабатываний $\sigma = [t_1, t_3]$.
- Активные переходы в каждом случае выделены оранжевым цветом.
- Заметим, что для той же сети имеется еще только одна возможная последовательность срабатываний $\sigma = [t_1, t_2]$.



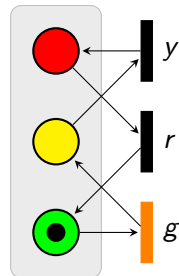
Пример работы сети Петри

- На рисунке показан пример работы сети Петри для последовательности срабатываний $\sigma = [t_1, t_3]$.
- Активные переходы в каждом случае выделены оранжевым цветом.
- Заметим, что для той же сети имеется еще только одна возможная последовательность срабатываний $\sigma = [t_1, t_2]$.



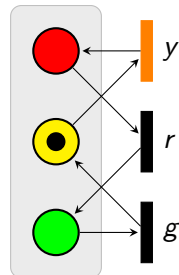
Моделирование работы светофора

- На рисунке приведен пример моделирования сетью Петри простого светофора, в котором цвета переключаются в следующем порядке (*зеленый, желтый, красный*), причем продолжительность каждого сигнала одинакова (и равна одному такту работы сети Петри).
- Особенностью этой сети является то, что ее работа никогда не завершается.



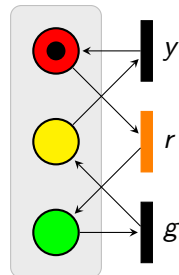
Моделирование работы светофора

- На рисунке приведен пример моделирования сетью Петри простого светофора, в котором цвета переключаются в следующем порядке (*зеленый, желтый, красный*), причем продолжительность каждого сигнала одинакова (и равна одному такту работы сети Петри).
- Особенностью этой сети является то, что ее работа никогда не завершается.



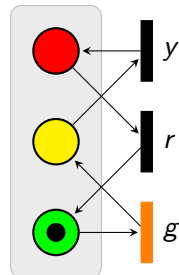
Моделирование работы светофора

- На рисунке приведен пример моделирования сетью Петри простого светофора, в котором цвета переключаются в следующем порядке (*зеленый, желтый, красный*), причем продолжительность каждого сигнала одинакова (и равна одному такту работы сети Петри).
- Особенностью этой сети является то, что ее работа никогда не завершается.



Моделирование работы светофора

- На рисунке приведен пример моделирования сетью Петри простого светофора, в котором цвета переключаются в следующем порядке (*зеленый, желтый, красный*), причем продолжительность каждого сигнала одинакова (и равна одному такту работы сети Петри).
- Особенностью этой сети является то, что ее работа никогда не завершается.



Формальное определение сети Петри

Формальным образом сеть Петри определяется как четверка

$$\langle S, T, W, \mu_0 \rangle,$$

где

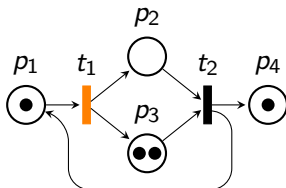
- S — конечное множество мест ($|S| = n$);
- T — конечное множество переходов ($|T| = m, S \cap T = \emptyset$);
- $W : (S \times T) \cup (T \times S) \rightarrow Z_+$ — мультимножество дуг (символом Z_+ будем обозначать множество неотрицательных целых чисел);
- $\mu_0 : S \rightarrow Z_+$ — начальная разметка сети.

Матричная форма сети Петри

- Сеть Петри можно задать и в компактной векторно-матричной форме.
- В этом случае структура сети Петри (т. е. ее граф) описывается двумя матрицами W^+ и W^- размера $n \times m$, определяющими наборы дуг, ведущих от мест к переходам (матрица W^-) и обратно (W^+):
 - ▶ w_{ik}^- = числу дуг, ведущих из i -го места в k -ый переход;
 - ▶ w_{ik}^+ = числу дуг, ведущих в i -ое место из k -го перехода.
- Из матриц W^+ и W^- составляется еще одна матрица $W = W^+ - W^-$, которая используется для вычисления новой разметки сети после применения к ней заданной последовательности срабатываний.
- Начальная разметка сети задается целочисленным вектором μ_0 длины n .

Пример матричного описания сети Петри

- Например, для следующей сети Петри



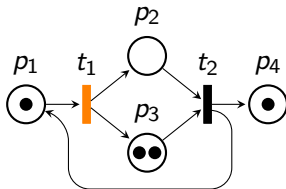
матрицы W^+ , W^- и W равны

$$W^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad W^+ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad W = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}.$$

- Разметка этой сети определяется вектором $\mu = [1, 0, 2, 1]$.

Определение активных переходов

- Для разметки μ переход $t_k \in T$ является активным, если выполняется условие $w_k^- \leq \mu$, где w_k^- обозначает k -ый столбец матрицы W^- , а сравнение двух векторов выполняется поэлементно.
- Например, для следующей сети Петри



выполняются неравенства $w_1^- \leq \mu$: $[1, 0, 0, 0] \leq [1, 0, 2, 1]$ и $w_2^- \not\leq \mu$: $[0, 1, 1, 0] \not\leq [1, 0, 2, 1]$.

- Значит первый переход является активным, а второй — нет.

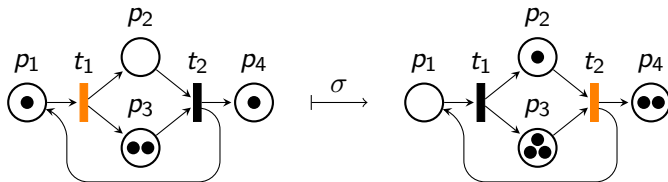
Вычисление новой разметки

- Пусть для некоторой сети Петри задана корректная последовательность срабатываний σ .
- Обозначим через $v(\sigma)$ вектор, k -ый элемент которого равен числу вхождений символа t_k в σ .
- Например, для сети с двумя переходами t_1 и t_2 и последовательности $\sigma = (t_1, t_2, t_1)$ вектор $v = [2, 1]$.
- Тогда, векторное представление разметки μ' , полученной путем применения последовательности σ к разметке μ , может быть найдено по следующей формуле:

$$\mu' = \mu + Wv(\sigma).$$

Вычисление новой разметки

- Например, применение последовательности $\sigma = (t_1, t_2, t_1)$ к сети Петри (левая часть рисунка)



приведет к разметке (правая часть рисунка), которая может быть вычислена следующим образом:

$$\mu' = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \\ 2 \end{bmatrix}.$$

Безопасные сети Петри

- Отдельные элементы сети Петри (места и переходы) могут обладать различными свойствами, на основе которых сначала определяются свойства самих сетей, а затем строится и их классификация.
- Простейшим свойством места является количество меток, которые могут в нем располагаться. Если в любой достижимой разметке заданной сети Петри число меток в заданном месте будет не более одной (0 или 1), то такое место называется безопасным.
- Сеть Петри называется *безопасной*, если все ее места безопасны.
- В безопасных сетях состояние каждого места описывается всего одним битом, поэтому такие сети могут быть легко реализованы аппаратно, используя те или иные виды переключателей.
- Кстати, оригинальный вариант определения сети Петри, данный самим Петри, подразумевал, что сеть является безопасной.

Ограниченные сети Петри

- Для большинства приложений требование безопасности сети является чересчур строгим. Его можно ослабить, разрешив каждому месту хранить некоторое ограниченное число меток.
- Более строго, место называется k -ограниченным, если в любой достижимой разметке в данном месте будет не более k меток. Очевидно, что 1-ограниченное место является безопасным.
- Место называется ограниченным, если существует такое k , что это место является k -ограниченным.
- Наконец, сеть Петри является k -ограниченной, если любое его место является k -ограниченным, и просто *ограниченной*, если все его места ограничены.
- Ограниченные сети также допускают эффективную аппаратную реализацию, в которой каждое место представляется уже счетчиком (например, регистром) некоторой заданной емкости.

Консервативные сети Петри

- Сеть называется *консервативной*, если число меток в любой достижимой разметке сохраняется одним и тем же.
- Такая модель используется, например, в тех случаях, когда метки представляют собой некоторые ресурсы системы, которые не уничтожаются и не создаются в процессе работы этой системы.
- Ресурсы могут переходить от одной части системы к другой, но их суммарное количество в процессе работы системы не изменяется.
- Нетрудно показать, что если сеть является недетерминированной, то любой переход (из срабатывающих хотя бы в одной достижимой разметке) должен иметь одинаковое число входных и выходных дуг — сколько переход выбрал меток, столько он должен их и поставить.

- В стандартной интерпретации сетей Петри каждый переход трактуется как некоторый процесс, который берет свои входные данные из входных мест, обрабатывает их некоторым образом и помещает результат в свои выходные места.
- Однако, природа этих процессов, в частности, их продолжительность, никак не конкретизируется.
- Хотя сети Петри обладают необходимым потенциалом для моделирования параллельно происходящих процессов, очевидно, что говорить о параллельности процессов имеет смысл только тогда, когда они имеют некую продолжительность.

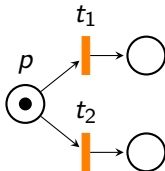
Моделирование последовательных процессов

- В простейшем варианте сеть Петри может рассматриваться как последовательное устройство, работа которого заключается в реализации некоторой последовательности срабатываний.
- Например, если в модели предполагается непрерывное время, а продолжительность срабатывания перехода является нулевой, то вероятность одновременного срабатывания двух переходов может считаться равной нулю. В этом случае важным оказывается как раз порядок (последовательность) срабатывания переходов.
- Последовательностная интерпретация применяется также, если все процессы обрабатываются одним и тем же устройством.
- Пример: однопроцессорный компьютер, выполняющий сразу несколько различных процессов. Эти процессы логически могут быть не связанными друг с другом (и рассматриваться пользователем как параллельные), однако, реализуются они в системе строго последовательно один за другим.

- В стандартной модели сети Петри с параллельным выполнением процессов предполагается, что время срабатывания всех переходов является одинаковым (например, равным 1) и срабатывание любого перехода производится в целочисленный момент времени.
- Таким образом, время в такой модели оказывается *дискретным*, его отсчет обычно ведется с нулевого момента (начальная разметка).
- Для того, чтобы описать параллельное поведение сети Петри, необходимо определить понятие конфликта.

Конфликты в сетях Петри

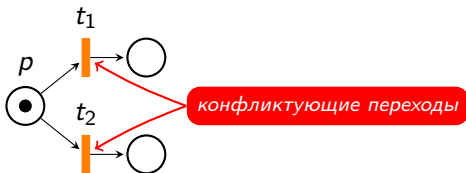
- *Конфликтом* в сети Петри называется ситуация, когда несколько активных переходов претендуют на одну и ту же метку.



- При последовательном срабатывании переходов конфликты никак не учитываются, однако при параллельной интерпретации требуется некоторый способ их разрешения.
- Стандартная схема: из всех активных в данный момент времени переходов выбирается некоторое их бесконфликтное подмножество — все эти переходы срабатывают одновременно.

Конфликты в сетях Петри

- *Конфликтом* в сети Петри называется ситуация, когда несколько активных переходов претендуют на одну и ту же метку.



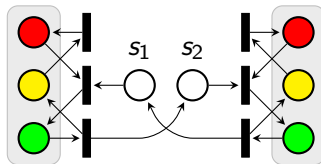
- При последовательном срабатывании переходов конфликты никак не учитываются, однако при параллельной интерпретации требуется некоторый способ их разрешения.
- Стандартная схема: из всех активных в данный момент времени переходов выбирается некоторое их бесконфликтное подмножество — все эти переходы срабатывают одновременно.

Принцип максимального параллелизма

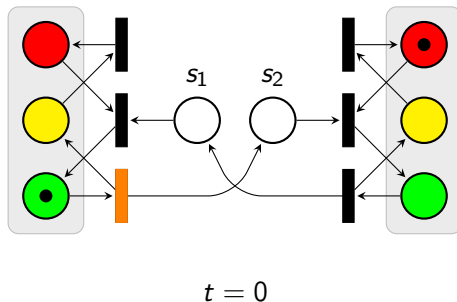
- Недетерминизм сети, заключающийся в выборе переходов для срабатывания, можно существенно понизить, если ввести следующее правило, называемое *принципом максимального параллелизма*: на каждом шаге для срабатывания выбирается такое множество A' множества A активных переходов, что 1) никакие два перехода из A' не конфликтуют; 2) для любого невыбранного перехода из множества $A \setminus A'$ имеется хотя бы один конфликтующий переход в A' .
- Другими словами, выбирается такой набор переходов, который не может быть расширен бесконфликтно никакими другими переходами.
- Будет ли такой набор наибольшим (по числу переходов) не является важным, главное, чтобы он был не расширяемым.

Моделирование синхронных процессов

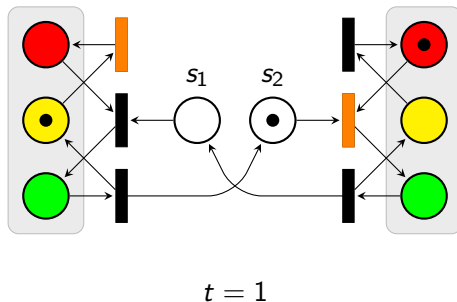
- С помощью сетей с максимальным параллелизмом оказывается возможным моделировать системы, в которых необходима синхронизация выполнения отдельных процессов.
- В качестве примера рассмотрим сеть Петри, моделирующую систему из двух светофоров, работающих согласованно: когда на первом горит красный свет, на втором горит зеленый, и наоборот.
- Для построения такой модели можно взять два светофора, рассмотренных нами выше, и связать их с помощью двух дополнительных «безопасных» мест s_1 и s_2 .



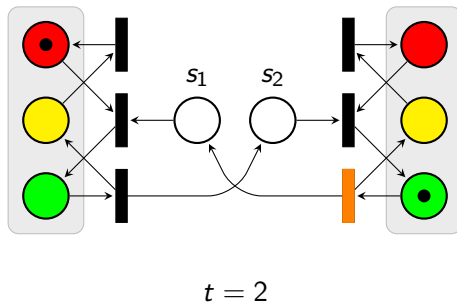
Модель двух светофоров



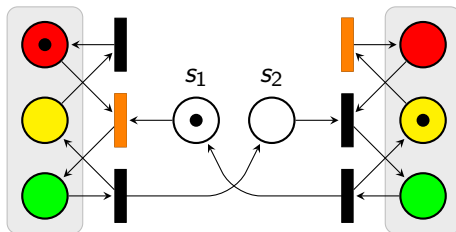
Модель двух светофоров



Модель двух светофоров

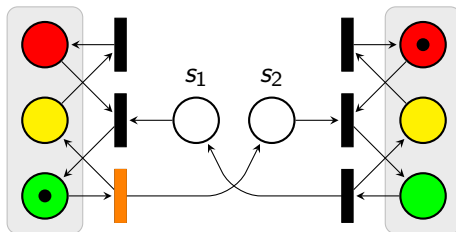


Модель двух светофоров



$t = 3$

Модель двух светофоров

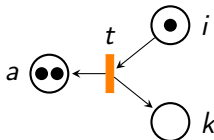


$t = 4$

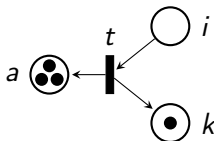
- Известно, что классические сети Петри с полностью недетерминированным поведением не являются полными по Тьюрингу. Однако, сети с максимальным параллелизмом таким свойством обладают.
- Чтобы показать алгоритмическую универсальность этого типа сетей, покажем, как с помощью сети Петри смоделировать работу счетчиковой машины Минского.
- Напомним, что счетчиковая машина Минского состоит из конечного набора именованных регистров, в которых хранятся целые неотрицательные числа. Программа для такой машины состоит из пронумерованной последовательности команд трех типов:
 - ▶ *Inc* a, k — инкремент;
 - ▶ *Dec* a, k, l — условный декремент;
 - ▶ *Halt* — останов.

- Сеть Петри, моделирующая выполнение программы для машины Минского, содержит три типа мест:
 - ▶ первые используются для представления регистров и имеют соответствующие этим регистрам имена;
 - ▶ вторые однозначно представляют команды и обозначаются соответствующими номерами;
 - ▶ третьи являются вспомогательными.
- Число меток, находящихся в месте a , соответствует числу, хранимому в регистре a .
- Текущая команда представляется местом с одной меткой, все остальные места—команды меток не имеют.

- Команда $i : Inc\ a, k$ представляется с помощью перехода t , имеющего одно входное место с номером i и два выходных — с меткой a и номером k .
- Как только в i -ом месте появится метка, сразу же будет активирован переход t , его срабатывание приведет к добавлению еще одной метки в место a и размещению метки в k -ом месте, что приведет на следующем шаге к активации уже k -ой команды.

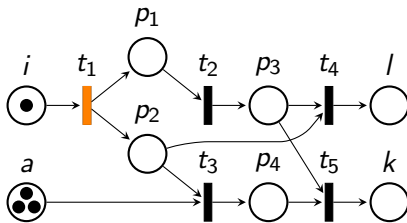


- Команда $i : Inc\ a, k$ представляется с помощью перехода t , имеющего одно входное место с номером i и два выходных — с меткой a и номером k .
- Как только в i -ом месте появится метка, сразу же будет активирован переход t , его срабатывание приведет к добавлению еще одной метки в место a и размещению метки в k -ом месте, что приведет на следующем шаге к активации уже k -ой команды.



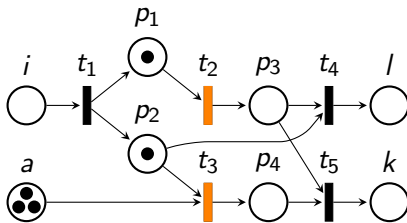
Условный декремент

- Команда условного вычитания $i: Dec\ a, k, l$ реализуется существенно сложнее.
- Если в регистре a находится число $n > 0$, то будет выполнена следующая последовательность переходов $\sigma = (t_1, \{t_2, t_3\}, t_5)$, которая приведет к уменьшению на единицу числа меток в a и появлению одной метки в k -ом месте (вычитание единицы и переход).



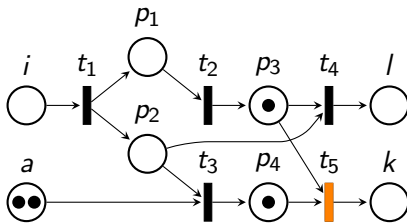
Условный декремент

- Команда условного вычитания $i: Dec\ a, k, l$ реализуется существенно сложнее.
- Если в регистре a находится число $n > 0$, то будет выполнена следующая последовательность переходов $\sigma = (t_1, \{t_2, t_3\}, t_5)$, которая приведет к уменьшению на единицу числа меток в a и появлению одной метки в k -ом месте (вычитание единицы и переход).



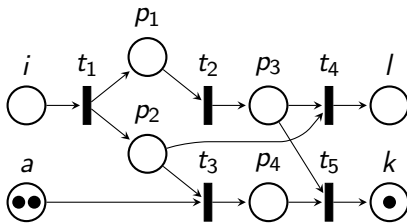
Условный декремент

- Команда условного вычитания $i: Dec\ a, k, l$ реализуется существенно сложнее.
- Если в регистре a находится число $n > 0$, то будет выполнена следующая последовательность переходов $\sigma = (t_1, \{t_2, t_3\}, t_5)$, которая приведет к уменьшению на единицу числа меток в a и появлению одной метки в k -ом месте (вычитание единицы и переход).



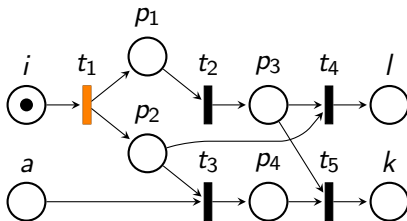
Условный декремент

- Команда условного вычитания $i: Dec\ a, k, l$ реализуется существенно сложнее.
- Если в регистре a находится число $n > 0$, то будет выполнена следующая последовательность переходов $\sigma = (t_1, \{t_2, t_3\}, t_5)$, которая приведет к уменьшению на единицу числа меток в a и появлению одной метки в k -ом месте (вычитание единицы и переход).



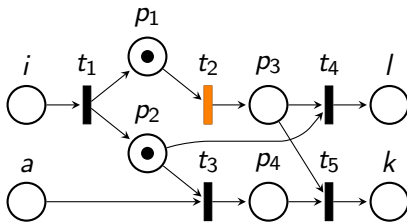
Условный декремент

- Если же регистр a пустой, то последовательность срабатывания переходов будет другой: $\sigma = (t_1, t_2, t_4)$, ее выполнение приведет к появлению метки в l -ом месте (переход к новой команде).

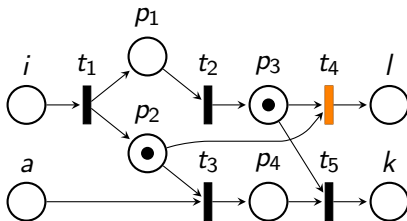


Условный декремент

- Если же регистр a пустой, то последовательность срабатывания переходов будет другой: $\sigma = (t_1, t_2, t_4)$, ее выполнение приведет к появлению метки в l -ом месте (переход к новой команде).

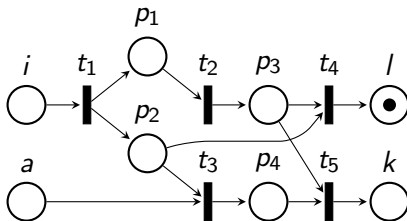


- Если же регистр a пустой, то последовательность срабатывания переходов будет другой: $\sigma = (t_1, t_2, t_4)$, ее выполнение приведет к появлению метки в l -ом месте (переход к новой команде).



Условный декремент

- Если же регистр a пустой, то последовательность срабатывания переходов будет другой: $\sigma = (t_1, t_2, t_4)$, ее выполнение приведет к появлению метки в l -ом месте (переход к новой команде).



- Проще всего устроена реализация команды $i: Halt$, которая представляется переходом t с одним входным i -ым местом и без выходных мест.
- Это значит, что активация этой команды приведет к поглощению переходом t метки из места i , после чего выполнение программы будет завершено, т. к. ни одна команда больше не сможет активироваться.



- Проще всего устроена реализация команды $i: Halt$, которая представляется переходом t с одним входным i -ым местом и без выходных мест.
- Это значит, что активация этой команды приведет к поглощению переходом t метки из места i , после чего выполнение программы будет завершено, т. к. ни одна команда больше не сможет активироваться.

