Сети Петри – наиболее удачный из существующих математический аппарат для моделирования, анализа, синтеза и проектирования самых разных дискретных систем с параллельно протекающими процессами.

*Определение.* Сетью Петри называется четвёрка элементов

*C = (P, T, I ,O)*, (3.2.1)

где

*P = { p1, p2,…,pn }, n > 0* (3.2.2)

множество позиций (конечное),

*T = { t1, t2,…,tm }, m > 0* (3.2.3)

множество переходов (конечное),

*I: T → P* (3.2.4)

функция входов (отображение множества переходов во входные позиции),

O: T → P (3.2.5)

функция выходов (отображение множества переходов в выходные позиции).

Если *pi*  *I (tj)* , то *pi* – входная позиция *j* - го перехода, если *pi* *I (tj)* , то *pi* – выходная позиция  *j* - го перехода.

Для наглядного представления сетей Петри используются графы.

Граф сети Петри есть двудольный ориентированный мультиграф

*G = (V,**),*  (3.2.6)

где  *V = P U T* , причём *P ∩ T = Ø*.

Исходя из графического представления сети Петри, её можно определить и так:

*C = (P, T, A)*, (3.2.7)

где *А* – матрица инцидентности графа сети.

Определим понятие маркированной сети Петри – оно является ключевым для любой сети.

Маркировка *μ*  сети Петри  *C = (P, T, I, O)* есть функция:

*N = μ(P), N* ***N***, (3.2.8)

отображающая множество позиций на множество натуральных чисел. Маркировку можно также определить как вектор:

*μ = {μ1, μ2,…, μn} ,*  (3.2.9)

где *n = │P │*, а *μi*  ***N***. Между этими определениями есть связь:

*μi = μ (pi)* (3.2.10)

На графе маркировка отображается ссответствующим числом точек в каждой позиции. Точки называются маркерами или фишками. Если фишек много (больше трёх), то их количество отображается числом.

Таким образом, маркированная сеть Петри представляет собой пятёрку элементов:

*M = (P, T, I, O, μ)*. (3.2.11)

Пример простейшей сети Петри:

*p1*

▪▪▪

*t1* *p3*

*p2*

Рисунок 3.2.1 – Пример сети Петри

Правила работы с сетями Петри.

Сеть Петри выполняется посредством запуска переходов. Переход может быть запущен в том случае, когда он разрешён. Переход является разрешённым, если каждая из его входных позиций содержит число фишек не меньшее, чем число дуг из неё в данный переход.

Процедура запуска состоит в удалении из каждой входной позиции перехода числа фишек, равного числу дуг из неё, и в выставлении в каждой выходной позиции числа фишек, равного числу дуг, входящему в неё.

Проиллюстрируем сказанное на примере уже нарисованной сети Петри. Запустим в ней переход *t1* – он является разрешённым:

*p1*

▪

*t1* *p3*

▪

*p2*

Рисунок 3.2.2 – Пример запуска перехода сети Петри

Пространство состояний и поведенческие свойства сетей Петри.

Пусть имеется маркированная сеть Петри:

*M = (P, T, I, O, μ)* (3.2.12)

У неё *n* позиций. В каждой позиции не более *N* фишек. Тогда пространство сотояний есть множество всех возможных маркировок сети. Определим *δ* – функцию следующего состояния.

Если переход *tj* разрешён при текущей маркировке *μ* , то следующая маркировка *μ’*  определится так:

*μ’ = δ (μ, tj)*  (3.2.13)

Если переход *tj* не разрешён, то *δ* не определена.

Пусть *{tj0, tj1,…, tjs}* – последовательность запущенных переходов. Тогда ей будет соответствовать последовательность *{μ0, μ1,…,μs+1}*, то есть

*μk+1 = δ(μk, tjk)* (3.2.14)

На основании последнего равенства можно определить понятие непосредственно достижимой маркировки. Для сети *C = (P, T, I ,O)* маркировка *μ’* называется непосредственно достижимой из *μ* , если существует такой переход *tj*  *T*, при котором

*μ' = δ(μ , tj)* (3.2.15)

Можно распространить это понятие на множество достижимых из данной маркировок. Определим множество достижимых из *μ* маркировок *R(C, μ)* следующим образом:

во - первых,  *μ*  *R(C, μ)*;

во - вторых, если *μ’*  *R(C, μ)*, *μ’ = δ(μ , tj)*  и *μ’’ = δ(μ’, tk)*, то и *μ’’*  *R(C, μ)*.

На основе введённых понятий можно сформулировать ряд свойств сети Петри, характеризующих её в процессе смены маркировок – назовём их поведенческими свойствами сети Петри. Определим наиболее важные из них.

1. Достижимость данной маркировки. Пусть имеется некоторая маркировка *μ*, отличная от начальной. Тогда возникает вопрос достижимости: можно ли путём запуска определённой поледовательности переходов перейти из начальной в заданную маркировку.
2. Ограниченность. Сеть Петри называется *k*- ограниченной, если при любой маркировке количество фишек в любой из позиций не превышает *k*. В частности, сеть называется безопасной, если *k* равно 1. Кроме того, сеть называется однородной, если в ней отсутствуют петли и одинарной (простой), если в ней нет кратных дуг.
3. Активность. Сеть Петри называется активной, если независимо от дотигнутой из *μ0*  маркировки существует последовательность запусков, приводящая к запуску этого перехода.

Реально вводят понятия нескольких уровней активности для конкретных переходов. Переход *tj*  *T*  называется:

а) пассивным (*L0-* активным), если он никогда не может быть запущен;

б) *L1-* активным, если он может быть запущен последовательностью переходов из *μ0* хотя бы один раз;

в) *L2-* активным, если для любого числа *K* существует последовательность запусков переходов из *μ0* , при которой данный переход может сработать *K* и более раз;

г) *L3-* активным, если он является *L2-* активным при  *K → ∞*.

1. Обратимость. Сеть Петри обратима, если для любой маркировки  *μ*  *R(C, μ0)* маркировка *μ0* достижима из *μ*.
2. Покрываемость. Маркировка *μ* покрываема, если существует другая маркировка *μ’*  *R(C, μ0)* такая, что в каждой позиции *μ’* фишек не меньше, чем в позициях маркировки *μ*.
3. Устойчивость. Сеть Петри называется устойчивой, если для любых двух разрешённых переходов срабатывание одного из них не приводит к запрещению срабатывания другого.

Существуют два основных метода анализа сетей Петри: матричные и основанные на построении дерева покрываемости.

Первая группа методов основана на матричном представлении маркировок и последовательностей запуска переходов. Для этого определим две матрицы размерности количество позиций  количество переходов, связанные со структурой сети. Первая матрица называется матрицей входов:

***D*** *– [i, j] = # (pi , I(tj))*, (3.2.16)

каждый её элемент равен числу фишек, уходящих из *j*- й позиции при запуске *i*- го перехода. Вторая матрица называется матрицей выходов:

***D*** *+ [i, j] = # (pi , O(tj))*, (3.2.17)

каждый её элемент равен числу фишек, приходящих в *j*- ю позицию при запуске *i*- го перехода. Определим единичный вектор  *e[j]* размерности *m*, содержащий нули во всех позициях кроме той, которая соответствует запускаемому в данный момент переходу. Очевидно, что переход разрешён, если *μ ≥ e[j]·****D*** *–*. Тогда результат запуска *j*- го перехода можно описать так:

*μ’ = μ + e[j]·****D***, (3.2.18)

где ***D =*** *(****D +*** –***D*** *–)*  – матрица изменений. Тогда все сформулированные ранее проблемы сети Петри легко интерпретируются матричными уравнениями вида

*μ = μ0 + σ·****D***, (3.2.19)

где *μ* – исследуемая маркировка,  *σ* – вектор, компоненты которого показывают, сколько раз срабатывает каждый переход.

Хотя данный метод достаточно прост, он не лишён некоторых недостатков. А именно, его применение даёт лишь необходимые условия существования какого- либо свойства, иными словами, может гарантировать лишь его отсутствие, а о присутствии мы сможем говорить с уверенностью, только проанализировав дерево покрываемости (смены) маркировок.

Дерево маркировок сети – это связанный граф, в вершинах которого находятся маркировки, которых мы достигли в результате последовательного запуска разрешённых переходов, а на дугах, соединяющих вершины – зпускаемые переходы. Путь от корня к каждой маркировке отражает последовательность запусков, приведшую к ней. Корнем дерева является начальная маркировка. При неограниченном накапливании фишек в позиции на дереве образуется петля, а в маркировке на месте, соответствующем зациклившейся позиции, ставится *ω* – символ бесконечно большого числа.

Ясно, что этот метод хотя и требует утомительного перебора всех возможных маркировок сети, но зато по уже готовому дереву достаточно легко анализировать проблемы достижимости, покрываемости, активности, обратимости сети.

Описав поведенческие свойства и методы анализа, можно перейти непосредственно к анализу конкретной сети Петри.

Для матричного анализа сети найдём её матрицу изменений.

 (3.3.1)

 (3.3.2)

Исходная сеть в виде графа:

*p1 p6*

▪ ▪

*t1* ▪ *p4* *t4*

*p2 p7*

*t2* ▪ *p5* *t5*

*p3 p8*

*t3 t6*

Рисунок 3.3.1 – Исходная сеть Петри

Матрицу изменений найдём как разность между (3.3.2) и (3.3.1):

 (3.3.3)

Таким образом, получив матрицу изменений, можно записать матричное уравнение смены маркировок вида (3.2.19). Вектор начальной маркировки определим так:

*μ0 = (10011100)* (3.3.4)

Составим дерево покрываемости маркировок сети.

*(10011100) ‘Новая’*

*t1 t4*

*‘Новая’ ‘Новая’*

*(01001100) (10010010)*

*t2 t4 t1 t5*

*(00100100) (01000010) (01000010) (10000001)*

*‘Новая’ ‘Тупик’ ‘Тупик’ ‘Новая’*

*t3 t6*

*(10011100) ‘Старая’ (10011100) ‘Старая’*

Рисунок 3.3.1 – Дерево покрываемости маркировок

Дерево покрываемости удобно оформить в виде графа. При этом более наглядно видны зацикливающиеся переходы, тупиковые маркировки никакими дополнительными пояснениями снабжать не требуется – отсутствие дуг, исходящих из данной маркировки, говорит само за себя. При достижении старой маркировки её не нужно заново наносить на граф – достаточно соединить дугой предыдущую маркировку и уже существующую “старую”.

Граф покрываемости сети выглядит следующим образом:

*μ0*

*t3 t6*

*10011100*

*00100100 t1 t4 10000001*

*t2 t5*

*01001100 10010010*

*t4 t1*

*01000010*

Рисунок 3.3.2 – Граф покрываемости маркировок сети Петри

Проанализируем сеть двумя методами – матричным и графическим и сравним полученные результаты.

Вопрос достижимости какой- либо маркировки легче всего решается, глядя на граф покрываемости. Действительно, возьмём для примера две маркировки: *μ1 = (01000010)* и *μ2 = (00100010)*. Первая из них достижима, и возможны два пути прихода к ней: *t1 , t4*  или *t4 , t1* . Однако они не единственны, перед вторым запуском перехода возможно бесконечное число раз запустить для первого случая последовательность *t2 , t3 ,* для второго случая –  *t5 , t6* . Вторая маркировка явно недостижима, так как её нет на графе.

С помощью матриц этот вопрос решается следующим образом. Составляем уравнение вида (3.2.19), в котором вместо *σ* ставим неизвестный вектор *x* той же размерности, а вместо  *μ* – интересующую нас маркировку *μ1*. В итоге получаем систему из 8 уравнений относительно 6 неизвестных компонент вектора *x*.

 (3.3.5)

Проанализировав данную систему, видим, что пятое уравнение является следствием из третьего и шестого, шестое – из седьмого и восьмого, первое – из второго и третьего. Из (1) и (4) следует, что *x5 = 0*, *x6 = 0*, из (7) следует, что *x4 = 1*. Первые три уравнения в (3.3.5) являются линейно зависимыми, поэтому за свободное неизвестное примем *x1*. Тогда получаем решение в виде *x1 = {y y-1 y-1 1 0 0}*, где *y* – любое целое число. Полученное решение говорит о достижимости маркировки *μ1* и указывает, какие из переходов и сколько раз должны быть для этого запущены.

Сравнив оба способа решения, сразу можно увидеть недостатки второго. Во- первых, решение (3.3.5) не указывает, в какой именно последовательности должны быть запущены указанные переходы. Во- вторых, глядя на матрицу изменений, мы не можем судить о наличии в сети петель. Кроме того, полученное матричное решение не даёт, вообще говоря, гарантий своей реализуемости – оно является лишь необходимым условием достижимости. Однако, не получив решения, можно говорить о недостижимости маркировки.

Действительно, записав (3.2.19) для *μ2*, получаем систему:

 (3.3.6)

Система является несовместной, так как после вычитания третьего уравнения из шестого получаем уравнение, противоречащее пятому. Поэтому можно сделать вывод о недостижимости *μ2*, совпадающий с полученным из графа покрываемости маркировок.

Исходя из графа (3.3.2), можно заключить, что сеть является безопасной. Действительно, ни в одной из позиций на маркировках не накапливается больше одной фишки. Это говорит о том, что реальный процесс, описываемый сетью, протекает без конфликтов. Однако о полном отсутствии конфликтов говорить пока рано. Данный вывод невозможно получить из матричного уравнения, так как он является обобщением, сделанным на основе знания всех возможных маркировок, получающихся в сети.

Данная сеть является активной – в ней каждый переход может сработать хотя бы один раз. Проанализируем уровни активности отдельных переходов. Переходы *t1* и *t4* являются *L1*- активными, так как они в худшем случае (то есть при получения тупиковой маркировки) могут сработать хотя бы один раз. Переходы *t2*, *t3*, *t5* и *t6* являются *L2*- активными, так как они могут сработать любое наперёд заданное число раз и даже больше.

Отсюда можно сделать вывод о том, что данная сеть не является бесконфликтной – у неё есть тупиковое состояние.

Можно также сказать, что сеть является обратимой. Этот вывод можно получить и матричным путём – решив уравнение

*x·****D*** *= 0* (3.3.7)

Получаем систему

 (3.3.8)

Данная система имеет 2 решения: *{y y y 0 0 0}* и *{0 0 0 y y y}*, где *y* – любое. Действительно, запуская любое число раз последовательности *t1 t2 t3*  или *t4 t5 t6* , каждый раз мы возвращаемся к исходной маркировке.

Из графа (3.3.2) также следует, что ни одна из маркировок сети не является покрываемой. Действительно, ни для одной маркировки не существует другой такой, для которой в каждой позиции было бы не меньше фишек, чем в исходной.

Можно сказать, что данная сеть не является устойчивой. У неё есть тупик, и, кроме того, непосредственно перед переходом в тупиковое состояние всегда существуют два разрешённых перехода. Запуская ‘неправильный’ переход, мы запрещаем оба – и оказываемся в тупике. Такое свойство сети говорит о наличии потенциально возможных конфликтов.

Па основании графа (3.3.2) можно выписать множество достижимых из *μ0* маркировок:

 (3.3.9)

Для моделирования сети была написана программа на языке *Turbo Pascal*. Она отображает состояние сети и разрешённые в каждый момент переходы. Для выбора запускаемого перехода используется мышь.

Выводы по разделу

В данном разделе быа проанализирована и смоделирована сеть Петри, которая служит моделью функционирования двух производственных процессов, связанных двумя общими ресурсами. В результате можно сделать вывод о принципиальном наличии в системе тупиковой ситуации, которая возникает при попытке одновременного запуска обоих процессов на выполнение. Чтобы не возникало тупика, необходимо каждый из процессов доводить до завершения, и не запускать другой процесс, пока не окончены все три цикла первого. Всё вышесказанное полностью подтверждается написанной программой, моделирующей все описанные ситуации, возникающие в сети.

ЛИТЕРАТУРА

1. Г.Корн, Т.Корн Справочник по математике для научных работников и инженеров.– М.: Наука, 1984. –831 с.
2. В.Брауэр Введение в теорию конечных автоматов.– М.: Радио и связь, 1987. –392 с.
3. Фаронов В.В. Турбо Паскаль 7.0: практика программирования. – М.: Нолидж, 1997. –432 с.