Использование аппарата сетей Петри для моделирования поведения вычислительных структур.

Часть 1. Изучение способов анализа СП-моделей.

Цель работы: Изучение матричных способов представления сетей Петри (СП) и методов исследования СП-моделей на основе матричных уравнений и дерева достижимых разметок (ДДР).

Постановка задачи

Сеть Петри это двудольный, ориентированный мультиграф $N = (P, T, F, H, \mu_0)$, где P – конечное непустое множество элементов, называемых позициями; T – конечное непустое множество элементов, называемых переходами; $F{:}PxT\to\{0,1,2,...\}$ и $H{:}TxP\to\{0,1,2,...\}$ функции инцидентности; $\mu_0 :P\to\{0,1,2,...\}$ – начальная разметка.

СП обычно представляют в виде геометрического объекта. При этом позиции изображаются кружками, переходы – черточками или прямоугольниками. Дуга проводится от позиции p_i к переходу t_j в том случае, если $F(p_i,t_j)=>1$, а от перехода t_j к позиции p_i если $H(t_j ,p_i)>=1$.

Если $F(p_i,t_j)=>1$, то позицию p_i называют входной к переходу t_j, а переход t_j выходным к позиции p_i. Множество входных позиций к переходу t_j определяется выражением $pre(t_j)=\{p{:}F(p,t_j)>=1\}$, а множество выходных переходов к позиции p_i выражением $post(p_i)=\{t{:}F(p_i,t)=>1\}$. Аналогично определяются множества входных переходов и выходных позиций.

При функционировании СП переходит от одной разметки к другой. Переход t может сработать при разметке μ, если $\forall p\in pre(t){:}\ \mu(p)-F(p,t)>=0$, где $\mu(p)$ число меток в позиции p. В результате срабатывания перехода t новая разметка μ' возникает в соответствии со следующим правилом:

$$\forall p\in (pre(t)\cup post(t)){:}\ \mu'(p)= \mu(p)–F(p,t)+H(t,p).$$

В этом случае говорят, что разметка μ' достижима из разметки μ, а μ предшествует μ' . Данный факт обозначается следующим образом: $\mu\to^{t}\mu'$.Сеть останавливается, если при некоторой разметке не может сработать ни один из ее переходов. Такая разметка называется тупиковой. Таким образом, СП моделирует некоторую структуру и динамику ее функционирования.

Если разметка μ' достижима из разметки μ в результате срабатывания последовательности переходов $r=t_1,t_2,...,t_k$, где: r – слово из множества T^* всех слов в алфавите T , то это обозначается как $\mu\to^{t1}\mu'''\to^{t2}... \to\mu''\to^{tk}\mu'$ или $\mu\to^{r}\mu'$. Разметка μ достижима в сети N, если $\mu_0\to\mu$. Множество всех достижимых разметок в сети N обозначим через $R(N)$. Переход t достижим от разметки $\mu\,(\mu\to t)$ в сети N, если:

$$\exists\mu '\exists\mu ''(\{\mu ', \mu ''\}\in R(N)\&\ \mu–\mu '){:}\ \mu '\to^{T}\mu ''.$$

Переход t достижим в сети N, если $\mu_0 \rightarrow t$. В приложении к моделированию систем проблема достижимости разметки μ' из μ_0 интерпретируется как возможность достижения определенного состояния системы.

Переход t **живой**, если:

$$\forall \mu \in R(N):\ \mu \rightarrow t.$$

СП **живая**, если:

$$\forall t\ \forall \mu\ (t \in T \& \mu \in R(N)):\ \mu\ (p) \rightarrow t.$$

Позиция p **ограничена**, если существует такое наперед заданное K, для которого

$$\forall\ \mu \in R(N):\ \mu\ (P) <= K.$$

СП **ограничена**, если:

$$\forall p\ \forall \mu\ (p \in P \& \mu \in R(N)):\ \mu\ (p) <= K.$$

Если $\kappa=1$, то СП называется **безопасной.**

Для моделей реальных ВС анализ на ограниченность (безопасность) позволит проверить возможность функционирования системы в некотором стационарном режиме без перехода заданного предела по числу объектов в отдельных подсистемах. При анализе на это свойство, в частности, могут быть выявлены требования к внутренним буферным накопителям в ВС. В настоящее время наиболее широко используются два метода анализа СП: построение дерева достижимых разметок и решение матричных уравнений, построенных на основе уравнения СП.

Дерево достижимых разметок представляет собой ориентированный граф, множество вершин которого образовано множеством $R(N)$, причем из вершины μ в вершину μ' ведет дуга, помеченная символом перехода t , если и только если $\mu \rightarrow {}^t\mu'$. В общем случае дерево достижимых разметок может иметь бесконечное число вершин. Для превращения дерева в полезный инструмент анализа опишем алгоритм построения конечного дерева достижимости.

Каждая i-я вершина дерева связывается с расширенной разметкой $\mu(i)$. В расширенной разметке число меток в позиции может быть либо неотрицательным целым, либо бесконечно большим. Бесконечное число меток обозначим символом ω. Каждая вершина классифицируется или как граничная, терминальная, дублирующая вершина, или как внутренняя. Граничными являются вершины, которые еще не обработаны алгоритмом. После обработки граничные вершины становятся либо терминальными, либо дублирующими, либо внутренними. Алгоритм начинает свою работу с определения начальной разметки. До тех пор, пока имеются граничные вершины, они обрабатываются алгоритмом.

Пусть x – граничная вершина, которую необходимо обработать, и с которой связана разметка $\mu\ (X)$.

1. Если в дереве имеется другая вершина y, не являющаяся граничной, и с ней связана разметка $\mu(y)=\mu(x)$, то вершина x дублируется.

2. Если для разметки $\mu(x)$ на один из переходов неразрешим, т.е. $\mu(x)$ тупиковая разметка, то x терминальная вершина.

3. Для любого перехода t_j, из множества T разрешенного в разметке $\mu(x)$, создать новую вершину z дерева достижимости. Разметка $\mu(z)$, связанная с этой вершиной, определяется для каждой позиции p_i следующим образом:

а) если $\mu(x)_i=\omega$, то $\mu(z)_i=\omega$;

б) если на пути от корневой вершины к x существует вершина y такая, что

$\mu(y)\to^{tj}\mu(x)$, $\mu(y)<\mu(x)$ и $\mu(y)_i<\mu(x)_i$, то $\mu(z)_i=\omega$;

в) в противном случае $\mu(z)_i=\mu(x)_i$.

Дуга, помеченная t_j, направлена от вершины x к вершине z. Вершина x переопределяется как внутренняя, вершина z становится граничной. Когда все вершины дерева становятся терминальными, дублирующими или внутренними, алгоритм останавливается.

С помощью матричных методов можно показать, что если СП живая и ограниченная, то она должна быть последовательной и инвариантной. Данные свойства недостаточны для утверждения живости и ограниченности СП. Однако их полезно проверить исходя из матриц инцидентности, так как если одно из этих свойств не подтверждается, то можно заключить, что описываемая система содержит некоторые недоработки.

Введем в рассмотрение матрицу C, которая получается следующим образом:

$C=H-F^T$, где F^T – транспонированная матрица F.

Пусть размерность C равна n x m , где m и n – мощности множеств P и T.

Рассмотрим матричные уравнения:

$$yC=0, \quad (1)$$

$$Cx=0, \quad (2)$$

где y и x – векторы, размерность которых равна n и m соответственно.

Вектор y, удовлетворяющий решению уравнения (1), все элементы которого положительны, называется p-цепью; p-цепь, все элементы которой больше нуля, называется полной p-цепью. Аналогично на основе уравнения (2) определяются понятия t-цепи и полной t-цепи. СП, для которой существует полная p-цепь, называется инвариантной. СП, для которой существует полная t-цепь, называется последовательной.

Лабораторное задание

1. Выбрать структуру СП в соответствии с номером варианта из приложения 1.
2. Описать заданную СП-модель с помощью матриц $F,H,$ μ_0.
3. Провести исследование СП-модели на основе матричных методов. Сделать заключение о живости и безопасности сети.
4. Провести исследование СП-модели путем построения дерева достижимых разметок (ДДР.
5. На основе проведенных исследований оценить корректность СП-модели и предложить варианты устранения недостатков в случае их обнаружения. Допустимо добавлять новые элементы и ограниченно видоизменять топологию сети. Полученная модель должна отвечать требованиям живости и безопасности.
6. Провести исследование полученной сети с помощью матричных методов и ДДР.

Часть 2. Построение и исследование моделей вычислительных структур.

Цель работы: Изучение методов использования иерархических сетей Петри при анализе многоуровневых вычислительных структур.

Постановка задачи

Существует три различных метода, с помощью которых может быть разработана многоуровневая ВС. Первый метод (сверху вниз) заключается в том, что сначала разрабатывается самый высокий уровень, затем уровень, находящийся под ним, и т.д., пока не будет достигнут уровень, который может быть интерпретирован аппаратными средствами. Второй метод (снизу вверх) является прямой противоположностью методу "сверху вниз". При его использовании первым разрабатывается уровень, наиболее близкий к аппаратуре, затем уровень, примыкающий к нему сверху, и т.д. до тех пор, пока не будет достигнут самый высокий уровень. При использовании третьего метода (с промежуточного уровня) проектирование начинается с одного из промежуточных уровней, а затем процесс разработки распространяется одновременно вверх и вниз.

Сети Петри с успехом могут применяться при использовании любого метода. Возможны два пути практического применения СП при проектировании и анализе систем. Первый путь заключается в использовании СП-моделей в качестве вспомогательного инструмента анализа. В этом случае построенная структура моделируется сетью Петри и модель анализируется. Любые трудности, встречающиеся при анализе, указывают на изъяны в проекте. Для их исправления необходимо модифицировать проект. Модифицированный проект затем снова моделируется и анализируется. Этот цикл повторяется до тех пор, пока проводимый анализ не приведет к успеху. Второй путь заключается в том, что весь процесс проектирования и определения характеристик ВС проводится в терминах сетей Петри.

Ниже представлены варианты ВС, назначение которых заключается в вводе, обработке и выводе информации. Предлагаемые структуры состоят из процессорных элементов (ПЭ), которые могут соединяться последовательно и параллельно, и каналов ввода-вывода, которые состоят из подканалов. Последовательное соединение $ПЭ_i$ и $ПЭ_j$ обозначается как – $(ПЭ_i{-}ПЭ_j)$, параллельное соединение $ПЭ_i$ и $ПЭ_j$ как – $(ПЭi||ПЭj)$.

Лабораторное задание

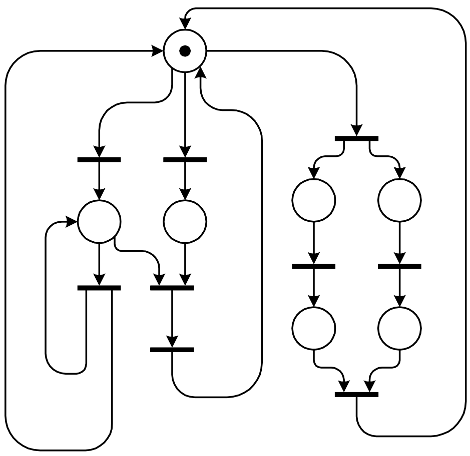
1. Выбрать вычислительную структуру в соответствии с номером варианта из Приложения 2.
2. Разработать СП-модель в соответствии с ее словесным описанием.
3. Провести анализ полученной СП-модели при помощи матричных методов и дерева достижимых разметок.
4. На основе исследования сделать выводы о корректности модели, предложить варианты устранения недостатков в случае их обнаружения.

Контрольные вопросы

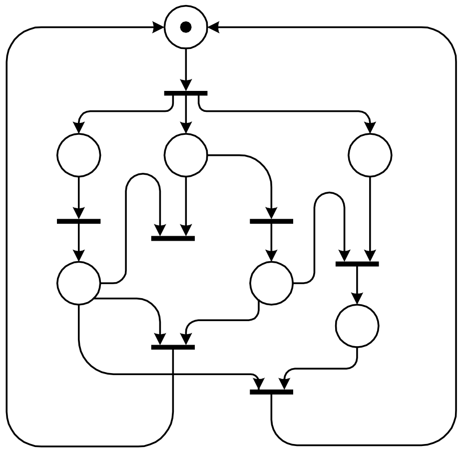
1. Что такое СП и с помощью каких параметров она задается?
2. Что такое живость, безопасность, ограниченность и достижимость СП?
3. Как интерпретируются для моделируемой ВС живость, ограниченность и достижимость СП?
4. Как выглядит уравнение состояния СП?
5. В чем заключаются матричные методы исследования СП-моделей?
6. Что такое полная p-цепь и полная t-цепь?
7. Что такое дерево достижимых разметок?
8. Какие приемы использованы в алгоритме построения дерева достижимых разметок для ограничения дерева?
9. Какие свойства СП исследуются в процессе анализа?
10. Какова интерпретация позиций и переходов при описании СП вычислительных структур?
11. Как можно доказать корректность иерархической СП-модели?
12. Как определяется степень детализации иерархической СП-модели ВС?
13. Какие Вы знаете пути практического применения СП при проектировании и анализе ВС?
14. Какие методы проектирования многоуровневых ВС Вам известны? В чем достоинства и недостатки данных методов?

Приложение 1. Варианты заданий к части 1

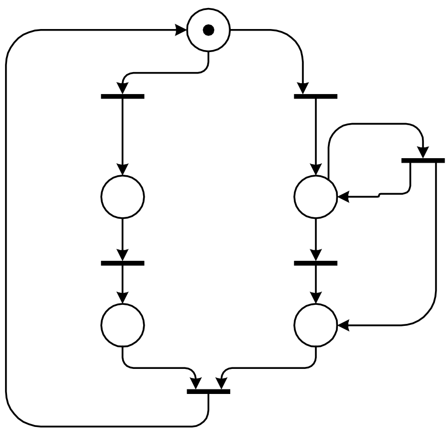
1



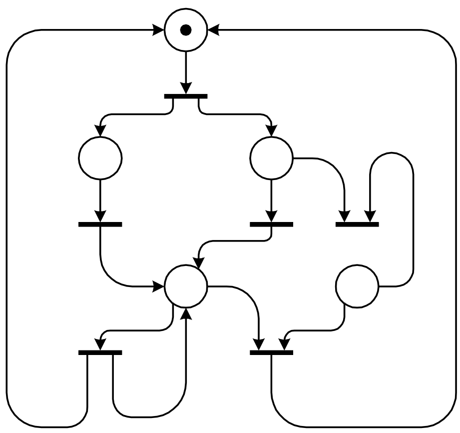
2



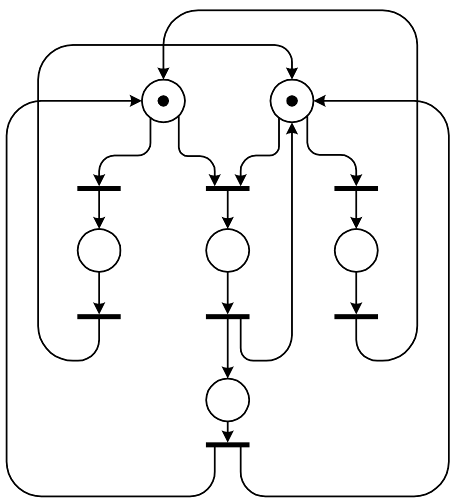
3



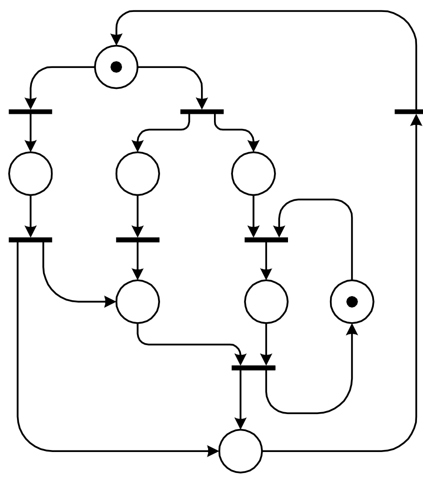
4



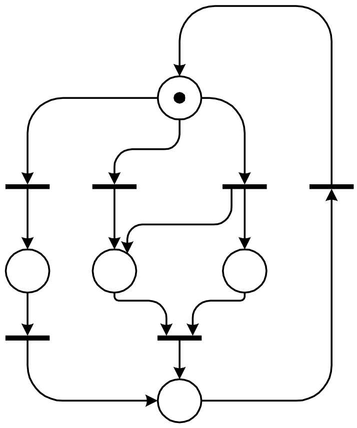
5



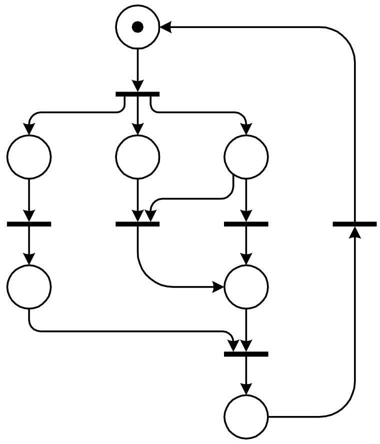
6



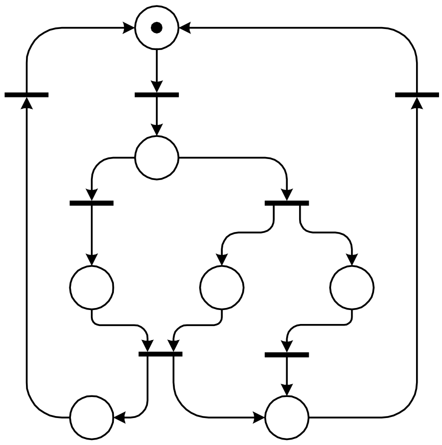
7



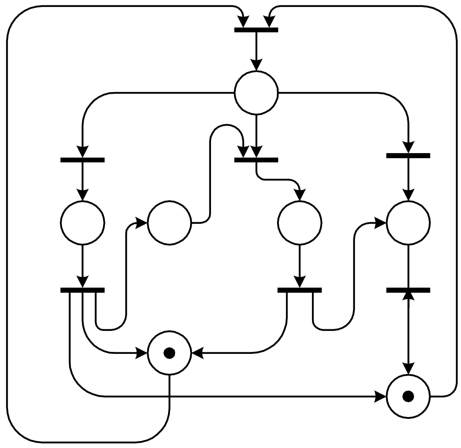
8



9



10



Приложение 2. Варианты заданий к части 2.

1.	Дана вычислительная структура, которая состоит из двух независимых подканалов *ПКВ1*, который вводит данные, и *ПКВ2*, который выводит данные. Обработка данных ведется на конвейерном процессоре, состоящем из трех процессорных элементов. Если работает процессор, то ввод данных запрещен.
2.	Дана вычислительная структура, которая включает канал ввода-вывода, состоящий из подканалов *ПКВ1, ПКВ2, ПКВ3*, и параллельный процессор, состоящий из трех процессорных элементов *ПЭ1*, *ПЭ2*, *ПЭ3*. Ввод данных выполняют подканалы *ПКВ1* и *ПКВ2*, вывод - подканал *ПКВ2*. Подканал *ПКВ3* управляет передачей данных в процессорные элементы: *ПЭ1* занимает подканал *ПКВ3* на все время обработки данных, *ПЭ2* – только на время ввода и вывода, *ПЭ3* – только на время вывода.
3.	Даны вычислительные структуры *ВС1* и *ВС2*. *ВС1* имеет параллельный процессор, состоящий из двух процессорных элементов. *ВС2* имеет конвейерный процессор, также состоящий из двух процессорных элементов. Канал ввода-вывода включает два подканала *ПКВ1* и *ПКВ2*. Ввод и обработка данных в *ВС1* производится под управлением подканала *ПКВ1*, а в *ВС2* – под управлением подканала *ПКВ2*. Вывод данных из *ВС1* и *ВС2* требует занятия канала ввода-вывода полностью.
4.	Даны вычислительные структуры *ВС1* и *ВС2*, которые имеют соответственно параллельный *(ПЭ1‖ПЭ2‖ПЭ3)* и последовательный *(ПЭ1−ПЭ2)* процессоры. Обработка данных в процессорах *ВС1* и *ВС2* начинается одновременно. Канал ввода-вывода имеет один подканал и выполняет ввод и вывод данных в каждой вычислительной структуре.
5.	Даны вычислительные структуры *ВС1*, *ВС2*, *ВС3* и канал ввода-вывода, состоящий из подканалов *ПКВ1*, *ПКВ2*, *ПКВ3*. *ВС1* выполняет ввод данных с использованием подканалов *ПКВ1* и *ПКВ2*. *ВС2* выполняет обработку данных на процессоре со следующей структурой *((ПЭ1‖ПЭ2)−ПЭ3)*. *ВС3* выполняет вывод данных с использованием подканалов *ПКВ2* и *ПКВ3*.
6.	Даны вычислительные структуры *ВС1*, *ВС2*, *ВС3* и канал ввода-вывода, который включает два подканала *ПКВ1* и *ПКВ2*. *ВС1* вводит данные с использованием подканалов *ПКВ1* и *ПКВ2*. *ВС2* выводит данные с использованием подканала *ПКВ2*. Обработка ведется *ВС3*

	на последовательно-параллельном процессоре со структурой *(ПЭ1 (ПЭ2‖ПЭ3))*.
7.	Дана вычислительная структура и канал ввода-вывода, который может использоваться при вводе и выводе данных одновременно. Обработке данных ведется на параллельном процессоре со структурой *(ПЭ1‖ПЭ2‖ПЭ3)*.
8.	Дана конвейерная система, которая включает вычислительные структуры *ВС1*, *ВС2*, *ВС3* и канал ввода-вывода с подканалами *ПКВ1* и *ПКВ2*. *ВС1* и подканал *ПКВ1* вводят данные, *ВС2* и подканал *ПКВ2* выводят данные, *ВС3* выполняет обработку. Обработка ведется на процессоре со структурой *((ПЭ1‖ПЭ2)−ПЭ3−(ПЭ4‖ПЭ5))*.
9.	Дана параллельная система, которая включает вычислительные структуры *ВС1*, *ВС2*, *ВС3* и канал ввода-вывода, который вводит и выводит данные во все структуры синхронно. Каждая вычислительная структура имеет последовательный процессор, состоящий из двух процессорных элементов *ПЭ1* и *ПЭ2*. Условием начала работы *ПЭ2* в *ВС2* является окончание обработки данных в *ВС3*, а условием начала работы *ПЭ2* в *ВС1* является окончание обработки данных в *ВС2*.
10.	Даны вычислительные структуры *ВС1*, *ВС2*, *ВС3* и *ВС4*. Все вычислительные структуры обмениваются данными с одним и тем же буфером. Передача данных осуществляется каналом ввода-вывода, содержащим подканал *ПКВ1*. Процессоры вычислительных структур являются последовательными и состоят из двух процессорных элементов. Обработку данных вычислительные структуры ведут в следующем порядке: *ВС1*, *ВС3*, *ВС4*, *ВС2*.