**Содержание**

1. Задание

2. Описание процесса

3. Построение метамодели «асинхронный процесс»

4. Операции над процессами

4.1 Репозиция

4.2 Редукция

4.3 Композиция

5. Предметная интерпретация асинхронного процесса

5.1 Построение сети Петри

Заключение

**1. Задание**

1. Выбрать вычислительный процесс и на его примере:
	* построить метамодель «асинхронный процесс» и определить свойства исходного процесса на основе анализа метамодели;
	* выполнить операции над процессом: репозиция, редукция, композиция, и оценить полученные результаты с практической точки зрения;
	* построить предметную интерпретацию метамодели на основе сети Петри и сделать вывод о динамических характеристиках исходного процесса.
2. Оформить отчет.

**2. Описание процесса**

В данном РГЗ рассматривается процесс установки модемного соединения и последующей передачи информации. Процесс представлен в упрощённом виде: модем переводится в режим ожидания, либо набора номера, после чего происходит коммутация; после установки связи с помощью модема можно отправлять и получать информацию от сходного устройства. Всеми действиями управляет контроллер, поэтому он присутствует во всех ситуациях процесса. Сначала пользователь либо устанавливает модем в режим ожидания, либо инициирует набор номера, при котором функционирует реле. После этого модему необходимо получить ответ от другого аналогичного устройства. В зависимости от того, принимается или получается информация, возможны два варианта. Информация получается: входящий фрагмент данных сначала демодулируется демодулятором, после чего расшифровывается декодером. Отправка данных: исходящий фрагмент информации кодируется кодером, модулируется модулятором, потом непосредственно отправляется.

**3. Построение метамодели «асинхронный процесс»**

*Компоненты:*

1. C – контроллер

1 – работает;

0 – не работает;

1. MO – модулятор

1 – работает, модулируется закодированный фрагмент информации;

0 – в данный момент бездействует;

1. DEM – демодулятор

1 – работает, демодулируется фрагмент информации;

0 - в данный момент бездействует;

1. K – кодер

1 – функционирует, кодирует фрагмент информации;

0 - в данный момент бездействует;

1. DK – декодер

1 - функционирует, декодирует фрагмент информации;

0 - в данный момент бездействует;

1. R – реле

1 – работает, реле набирает телефонный номер;

0 - в данный момент бездействует;

1. W – режим ожидания

1 – модем находится в режиме ожидания входящего звонка;

0 - модем не находится в режиме ожидания входящего звонка;

1. A – получен «ответ»

1 – удалённый модем отвечает на запросы;

0 - удалённый модем не отвечает на запросы;

1. F –флаг

1 – установлен;

0 – не установлен.

*Ситуации:*

1. Модем готов к работе

C+MO-DEM-K-DK-R-W-A-F+

1. Ожидание входящего звонка

C+MO-DEM-K-DK-R-W+A-F-

1. Состояние набора номера

C+MO-DEM-K-DK-R+W-A-F-

1. Связь установлена

C+MO-DEM-K-DK-R-W-A+F-

1. Поступление фрагмента информации

C+MO-DEM+K-DK-R-W-A+F-

1. Декодирование фрагмента информации

C+MO-DEM-K-DK+R-W-A+F-

1. Кодирование фрагмента информации

C+MO-DEM-K+DK-R-W-A+F-

1. Отправка фрагмента информации

C+MO+DEM-K-DK-R-W-A+F-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C | MO | DEM | K | DK | R | W | A | F |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Граф процесса*

*Инициаторы:*

 - ситуация, при которой функционирует контроллер, что свидетельствует о готовности устройства к выполнению своих основных функций, то есть непосредственно инициирует данный процесс;

- ситуации ожидания и дозвона инициируют модемное соединение и процесс приёма/передачи информации (собственно, основной функции модема).

- ситуация, когда получен ответ от другого модема, непосредственно инициирует процесс приёма/передачи информации (то есть только после получения ответа о готовности обоих устройств возможен процесс передачи информации);

*Результанты:*

- ситуация декодирования информации, то есть получение готового к дальнейшей обработке фрагмента информации, что и является результатом работы устройства модем;

- ситуация отправки фрагмента информации, аналогично является результатом функционирования устройства.

*Классы эквивалентности:*

Начальный класс - ;

Заключительные классы - , .

Траектории -

*Свойства исходного асинхронного процесса:*

1. АП является эффективным, т.к. из инициаторов все траектории ведут в результанты, и каждая из траекторий, приводящая к результантам, начинается в каком-либо инициаторе;
2. АП не является управляемым, так как траектории из инициаторов ведут в различные заключительные классы , , то есть существует неопределённость;

1. АП не является простым, т.к. из инициатора можно попасть в другие инициаторы , что противоречит условию: .

**4. Операции над процессами**

**4.1 Репозиция**

Исходный процесс за один раз может передать или принять только один фрагмент информации. Для того, чтобы процесс был более универсальным и мог передавать/принимать неограниченное количество блоков информации, над процессом можно совершить операцию репозиции, т.е. повторно активизировать процесс.

В результате репозиции получим АП, где , , .

*Объединение графа исходного процесса с репозицией.*

Отношение задаёт траектории переходов от элементов из множества к элементам множества . Так как , но и , то репозиция является частичной. Ввиду относительной простоты процесса дополнительно не оговаривается, в какой момент времени процесс будет остановлен.

Операция репозиции позволила добиться необходимого результата – возобновить процесс с момента выбора приёма/передачи информации и продолжать процесс передачи необходимое число раз.

**4.2 Редукция**

Редукция предусматривает выделение из исходного процесса некоторой части. Операция состоит в сведении данного АП к более простому. В данном случае попытаемся выделить из исходного процесса ту часть, которая выполняет какую-то логически завершённую функцию, конкретно – приём информации, то есть выделим одну из веток процесса, которая несёт конкретную функциональную нагрузку.

Таблица векторов.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C | MO | ***DEM*** | K | ***DK*** | R | W | ***A*** | F |
|  | 1 | 0 | ***0*** | 0 | ***0*** | 0 | 0 | ***0*** | 1 |
|  | 1 | 0 | ***0*** | 0 | ***0*** | 0 | 1 | ***0*** | 0 |
|  | 1 | 0 | ***0*** | 0 | ***0*** | 1 | 0 | ***0*** | 0 |
|  | 1 | 0 | ***0*** | 0 | ***0*** | 0 | 0 | ***1*** | 0 |
|  | 1 | 0 | ***1*** | 0 | ***0*** | 0 | 0 | ***0*** | 0 |
|  | 1 | 0 | ***0*** | 0 | ***1*** | 0 | 0 | ***0*** | 0 |
|  | 1 | 0 | ***0*** | 1 | ***0*** | 0 | 0 | ***0*** | 0 |
|  | 1 | 1 | ***0*** | 0 | ***0*** | 0 | 0 | ***0*** | 0 |

В качестве входной компоненты выберем 3 (демодулятор), 5 (декодер) и 8 (наличие установленной связи) элементы векторов ситуаций. Выбираем такие компоненты специально, чтобы от них отсечь «посторонние» и получить необходимую редукцию (именно эти компоненты отвечают за приём информации).

Выберем различных значений входной компоненты: .

Тогда редукция по :

*Граф редукции.*

В результате редукции удалось выделить требуемый подпроцесс – приём фрагмента информации. Таким образом, выбирая определённые входные компоненты вектором, можно выделять любой подпроцесс в системе, при этом получая более простой по своей структуре процесс.

**4.3 Композиция**

Рассмотрим два АП. Один из них исходный процесс , другой – вновь построенный .

Процесс :

В качестве дополнительного процесса построим вспомогательный процесс, проверяющий наличие и функционирование драйвера модема.

*Компоненты процесса :*

C – контроллер;

D – драйвер;

*Ситуации процесса :*

1. Драйвер функционирует и способен управлять модемом C-D+
2. Модем готов к работе C+D+

|  |  |  |
| --- | --- | --- |
|  | D | C |
|  | 1 | 0 |
|  | 1 | 1 |

*Инициатор: .*

*Результант:* .

*Граф процесса :*

Выделим в исходном процессе первую (контроллер) и девятую (вспомогательный флаг) компоненты в качестве выходных. Выбираем контроллер, так как он является основным показателем работоспособности устройства, и некоторую дополнительную вспомогательную компоненту – флаг, необходимый для распознавания удачной инициализации драйвера (флаг введён в процесс несколько искусственным образом).

Выбираем в исходном процессе ситуации, при которых функционирует котроллер (первая компонента; т.е. все ситуации).

Таким образом, получаем исходный процесс целиком. То есть исходный процесс совпадает со своей редукцией. Это вполне соответствует действительности, так как котроллер функционирует на всех стадиях работы модема и дополнительный процесс (который известен заранее) выполняет функцию предварительной подготовки модема, которая распространяется на всё устройство. Тогда редукция процесса по множеству :

, то есть получаем весь исходный процесс.

Выделим в дополнительном процессе входную компоненту. Эта компонента будет совпадать со всем набором компонент процесса *.*

В качестве выберем все наборы входной компоненты дополнительного процесса. Дополнительный процесс представлен простой линейной структурой, которая не нуждается в дополнительном упрощении.

Сцепление процессов будет происходить по компоненте . То есть выходная компонента процесса соответствует входной компоненте .

Построим композицию исходного процесса и дополнительного .

*Граф композиции:*

Обеспечены условия 1, 2, 3, 4:

1),:

2) выходные компоненты ситуаций из равны входной компоненте ситуации из

3) если в компонента , то

4) если , то

Таким образом, выбрав в качестве дополнительного процесса небольшой подпроцесс, предшествующий основному процессу, построили композицию этих процессов путём выделения входных и выходных компонент соответствующих процессов и последующего их (процессов) сцепления. Исходный процесс не стали упрощать с помощью репозиции, так как по смыслу данной композиции модель должна сохранить свою целостностью. Получившийся процесс представляет собой несколько усложнённый исходный процесс.

**5. Предметная интерпретация асинхронного процесса**

Предметная интерпретация согласована с приложением и зависит от специфики решаемой задачи.

1. Система выполняет те функции, для которых она предназначена;
2. Данная система функционирует эффективно;
3. В исходном АП ошибки и аварийные ситуации возникнуть не могут (именно в рамках рассматриваемого процесса), в редуцированном процессе есть потенциально узкие места – не оговаривается условие остановки процесса;
4. Систему упростить нельзя, т.к. она уже является достаточно упрощённой, процесс не является функционально избыточным, но тем не менее некоторые компоненты можно изменить в сторону упрощения.

**5.1 Построение сети Петри**

Сетью Петри называется пятёрка.

 - конечное непустое количество условий;

 - конечное непустое количество событий;

- функция инцидентности;

- функция инцидентности;

- начальная разметка.

- контроллер работает;

 - происходит модуляция сигнала;

- происходит модуляция сигнала;

- кодирование фрагмента информации;

- декодирование фрагмента информации;

- функционирует реле;

- включен режим ожидания;

- получен ответ от другого модема;

- флаг установлен.

 - переход к режиму ожидания;

 - переход к набору номера;

 - переход к состоянию, когда получен ответ (от режима ожидания);

 - переход к состоянию, когда получен ответ (от режима набора номера);

 - переход к демодуляции входящего сигнала;

 - переход к кодированию исходящего фрагмента информации;

 - переход к декодированию входящего фрагмента информации;

 - переход к модуляции исходящего сигнала.



|  |  |
| --- | --- |
|  |  |

Граф разметок:

100000001

100001000

100000100

100000010

100100000

101000000

110000000

100010000

Свойства сети Петри:

1. *Ограниченность сети.* Места являются ограниченными (существует число достижимой в сети разметки M справедливо ) вся сеть является ограниченной;

1. *Безопасность сети.* Места являются безопасными (разметки M имеет место ) вся сеть является безопасной;

1. *Живость сети.* При переходы являются потенциально живыми в сети, но эти же переходы не являются живыми при любой достижимой в сети разметкесеть не является живой;

1. *Устойчивость сети.* Сеть не является устойчивой, так как существуют переходы, которые, сработав, могут лишить другой переход этой возможности: переходы и взаимно блокируются;

На основе построенной модели была создана сеть Петри для заданного процесса. В качестве ситуаций были взяты компоненты исходного процесса. Между ситуациями были расставлены переходы, а также установлена начальная разметка. После был построен граф разметок, который полностью совпал с исходным графом процесса, что свидетельствует о правильности построения и функционирования сети Петри.

**Заключение**

асинхронный процесс репозиция сеть петри

АП является общей моделью описания динамики поведения параллельно функционирующих систем. Эта модель задаёт допустимые последовательности действий над некоторыми объектами систем, каждой из которых соответствует некоторая траектория АП. АП – модель управляющей структуры системы. АП можно понимать как метамодель, порождающую различные широко используемые динамические модели. Порождение частных моделей использует механизм интерпретации АП.

В данном РГЗ рассматривается процесс функционирования модема. Процесс представляется в сильно упрощенном виде, т.к. целью выполнения задания является не построение сложной модели, близкой к реальной, а изучение основ создания модели «асинхронный процесс», операций над процессом. Результатом является построение модели, с помощью которой можно проанализировать заданный процесс: выделены компоненты и ситуации процесса, построен граф (наглядное представление функционирования процесса), проведены операции над процессом: репозиция – повторное выполнение передачи фрагмента информации; редукция – выделение в исходном процессе подпроцесса по непосредственному приёму информации; композиция – построение последовательной композиции из исходного процесса и вновь созданного подпроцесса). Проведена предметная интерпретация АП: процесс эффективен, но есть узкое место в функционировании системы после репозиции. Описаны составляющие модели «асинхронный процесс», используя понятия модели «сеть Петри». Проведён анализ свойств мест сети Петри на ограниченность и безопасность: сеть ограниченная и безопасная. Проведён анализ свойств переходов сети Петри на живость и устойчивость: сеть не является потенциально живой, и не является устойчивой.