

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»

УДК 004.42

СОГЛАСОВАНО

Доцент департамента  
программной инженерии  
НИУ ВШЭ  
кандидат компьютерных  
наук

УТВЕРЖДАЮ

Академический руководитель  
образовательной программы  
«Программная инженерия»  
старший преподаватель  
департамента программной  
инженерии

Р.А. Нестеров  
«\_\_\_\_\_» 2025 г.

Н.А. Павлов  
«\_\_\_\_\_» 2025 г.

**Отчет  
по исследовательскому курсовому проекту**

на тему "Проверка соответствия декларативных моделей процессов и журналов  
событий"

по направлению подготовки бакалавров 09.03.04 «Программная инженерия»

Выполнил  
студент группы БПИ 238  
образовательной программы  
09.03.04 «Программная инженерия»  
Анненков Алексей Николаевич

10.04.2025

Москва 2025

## РЕФЕРАТ

Отчёт 39 с., 12 источн., 1 прил.

PROCESS MINING, ЖУРНАЛ СОБЫТИЙ, ТРАССА, ДЕКЛАРАТИВНЫЕ МОДЕЛИ, ЗАДАЧА ПРОВЕРКИ СООТВЕТСТВИЯ (CONFORMANCE CHECKING)

Объектом исследования является задача проверки соответствия декларативной модели ациклическим журналам событий.

Цель работы - разработать и протестировать алгоритм проверки соответствия декларативной модели ациклического журнала событий.

В результате работы были описаны методы проверки соответствия декларативной модели с различными бинарными ограничениями и ациклического журнала событий; была разработана библиотека, содержащая алгоритмы, реализующие данные методы. Также были проведены эксперименты, направленные на проверку корректности работы алгоритмов на генерированных данных.

На основании этого были сделаны выводы о целесообразности проведения дальнейших исследований.

## СОДЕРЖАНИЕ

### Содержание

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ . . . . .	4
ВВЕДЕНИЕ . . . . .	5
1 Предварительные сведения . . . . .	7
1.1 Сети Петри . . . . .	7
1.2 Журнал событий . . . . .	10
1.3 Отношения между событиями . . . . .	11
1.4 Conformance checking . . . . .	13
1.5 Декларативные модели . . . . .	16
1.6 Выводы и постановка задачи проверки соответствия . . . . .	18
2 Методы и алгоритмы оценки соответствия декларативной модели журналу событий . . . . .	19
2.1 Ограничение Response . . . . .	19
2.2 Ограничение RespondedExistence . . . . .	25
2.3 Ограничение CoExistence . . . . .	26
2.4 Ограничение NotCoExistence . . . . .	27
2.5 Ограничение Choice . . . . .	29
2.6 Ограничение ExclusiveChoice . . . . .	30
2.7 Коэффициент соответствия декларативной модели и журнала . . . . .	31
3 Программная реализация и проведение экспериментов . . . . .	34
3.1 Детали реализации . . . . .	34
3.2 Проведение экспериментов . . . . .	34
4 Обзор литературы . . . . .	36
ЗАКЛЮЧЕНИЕ . . . . .	37
5 Список литературы . . . . .	38

## **ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ**

Трасса (trace) – последовательность событий.

Журнал событий (event log) – объективные данные, дающие представление о процессе таком, какой он есть; последовательность трасс.

Process mining – дисциплина, направленная на выявление, мониторинг и улучшение процессов при помощи извлечения данных из журнала событий.

Задача проверки соответствия (Conformance Checking) – одна из задач process mining, направленная на проверку степени соотношения существующей модели и имеющегося журнала событий.

Декларативная модель (declare model) – модель, в которой допустимое поведение задается как любое поведение, не запрещенное заданными в модели ограничениями.

## ВВЕДЕНИЕ

В последние годы с развитием компьютерных технологий возрастает интерес к дисциплине process mining, которая объединяет в себе методы двух широких областей: data science – науки о данных, и process science – науки о процессах.

Data science – это междисциплинарная область, конечной целью которой является извлечение из данных какой-либо реальной ценности. Data science включает в себя хранение и извлечение данных, подготовку и преобразование данных, исследование данных, различные виды интеллектуального анализа, а также представление объяснений и прогнозов. Process science – широкая дисциплина, которая комбинирует в себе знания в области менеджмента и информационных технологий с целью улучшения и управления операционными процессами [1].

Однако методы интеллектуального анализа данных, как правило, не учитывают сквозные модели процессов. А подходы, основанные на анализе процессов, зачастую фокусируются на моделировании, а не на изучении данных о событии. Process mining можно рассматривать как связывающее звено между data science и process science.

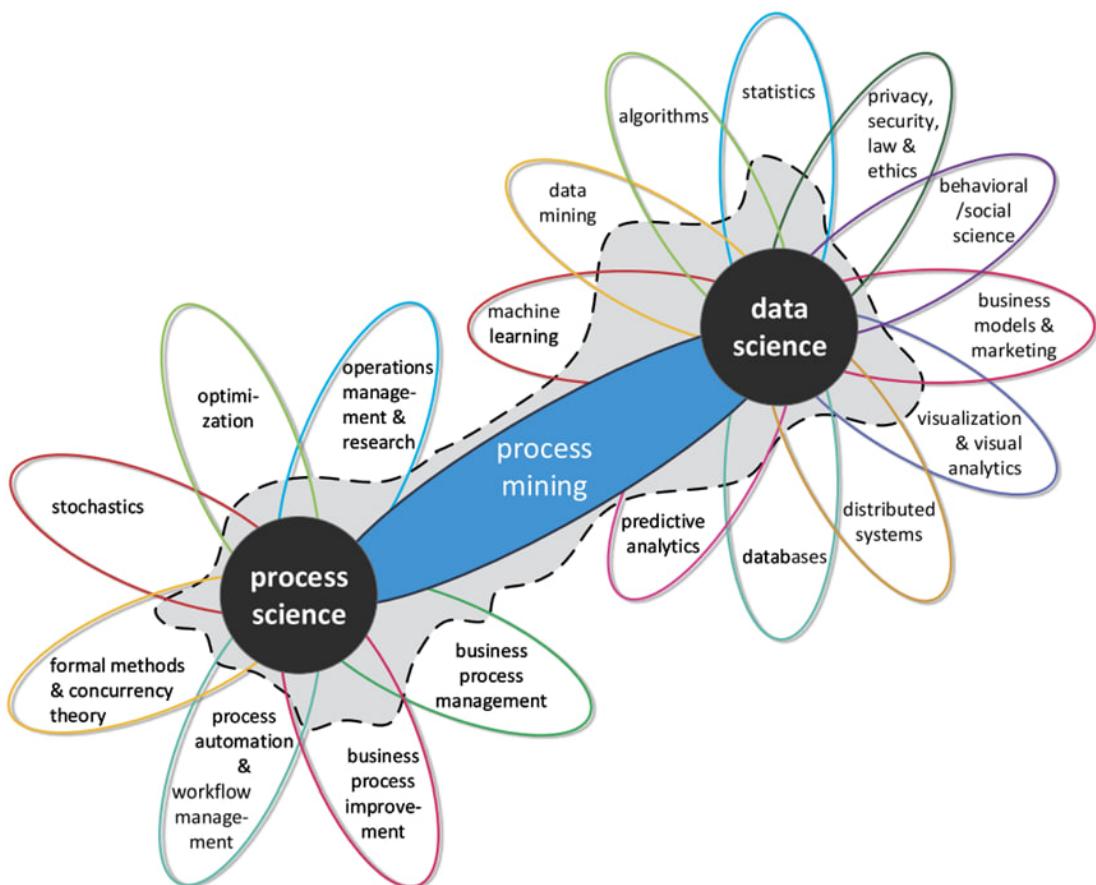


Рисунок 1 - Process mining [1]

Цель дисциплины process mining состоит в автоматическом обнаружении и улучшении реальных процессов путём извлечения данных из журналов событий.

Традиционно системы управления рабочими процессами основаны на идее, что выполнение в процессе одной задачи может повлечь за собой выполнение других задач. Такие системы хорошо описываются императивными моделями (например, основанными на сетях Петри [12]), которые подходят для явного отображения таких шаблонов управления, как последовательность, параллелизм, циклы, выбор вариантов. Хотя такие модели и обеспечивают высокую степень руководства, они описывают лишь небольшое разрешенное подмножество всех поведений, не охватывая при этом все пути, что

может быть важно в случае процессов со слабой структурой. Другими словами, императивные модели больше подходят для контроля, нежели для гибкости. В случаях, когда модель должна быть более гибкой, больше подходят декларативные (declare) модели [7]. Эти модели, вместо перечисления всех возможных поведений процесса, задаются ограничениями, которые определяют запрещенное поведение. Другими словами, в таких моделях разрешено всё то, что не запрещено явно. Такие модели лучше определяют границы процессов, и расширяют поведение, которое может быть зафиксировано с помощью императивных моделей.

Выделяют три подхода, которые применяются в process mining: извлечение (discovery), проверка соответствия (conformance checking) и усовершенствование (enhancement) [1]. В подходе discovery строится модель только на основании данных из журнала событий; в enhancement существующая модель улучшается при помощи извлеченной из журнала событий информации. В рамках данной работы подробно будет рассмотрен подход conformance checking.

В conformance checking существующая модель сравнивается с журналом событий этого же процесса. Цель проверки соответствий – проверить, насколько точно моделируемый процесс соотносится с реальным, фактическим поведением. Conformance checking используется для выявления отклонений и возможных инцидентов и является ценным инструментом для управления рисками.

В подходе проверки соответствий для императивных моделей используются четыре основные метрики, позволяющие описать степень соответствия модели и журнала событий: fitness, precision, generalization и simplicity [4].

Однако, несмотря на существование многих методов проверки соответствий для императивных моделей, малое внимание уделено conformance checking декларативных моделей [9, 10, 11]. Целью этого исследования является изучение методов проверки соответствия декларативных моделей журналам событий, а также разработка алгоритмов численной проверки соответствия.

## 1. Предварительные сведения

Как было показано во введении, цель этого исследования - разработка алгоритмов проверки соответствия журналов событий и декларативных моделей процессов. В настоящем разделе приводятся общие сведения об исследуемых объектах, необходимые для дальнейшего изложения.

### 1.1. Сети Петри

Попробуем смоделировать процесс приготовления пиццы. Для этого: сначала необходимо купить ингредиенты (buy ingredients, bi); затем подготовить тесто (create base, cb); далее, добавить ингредиенты – сыр, помидоры, колбасу (add cheese – ac, add tomato – at, add salami - as), причем последовательность добавления ингредиентов не важна; далее, поставить пиццу запекаться в духовку (bake in oven, bo); после этого, готовую пиццу можно съесть; наконец, необходимо убраться на кухне (clean kitchen, ck).

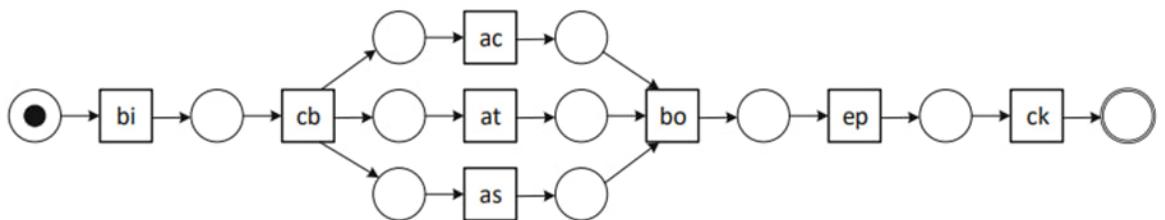


Рисунок 2 - Приготовление пиццы, начальное состояние

Одним из способов моделирования процессов является использование сетей Петри. На рисунке 2 изображена сеть Петри, моделирующая описанный процесс.

Структура сетей Петри включает в себя два вида элементов: состояния и переходы [12].

Состояния – это «пассивные» компоненты сети: они могут хранить, накапливать или показывать объекты. Графически состояния обозначаются кругом.

Переходы – это «активные» компоненты сети: они могут переносить, потреблять или производить объекты, а также транспортировать их. Графически обозначаются прямоугольником.

Состояния и переходы связаны друг с другом направленными дугами, которые обозначаются стрелками. В отличие от переходов и состояний, дуга не моделирует никакой компонент. Она представляет собой некоторое отношение между компонентами – например, логическую связь, или права доступа. Дуга может соединять только состояние и переход – связи типа «состояние-состояние» или «переход-переход» не разрешены.

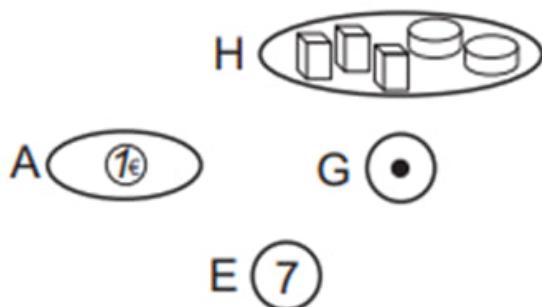


Рисунок 3 - Фишки в состояниях в сети Петри [12]

Разметка – это распределение жетонов по состояниям. Графически разметка представляется символами, выполняющими роль жетонов, в кругах. Символические знаки обозначают элементы реального мира. Эта связь настолько сильна, что символические представления элементов и сами элементы реального мира считаются эквивалентными. Однако для удобства вместо элементов используются абстрактные черные символы – фишкы. Например, в состоянии G на рис.3 находится фишкa.

Переход в сети Петри может сработать тогда и только тогда, когда во всех состояниях, из которых исходит дуга в рассматриваемый переход, находится как минимум одна фишкa. Переход, который может сработать, называется активным. Например, на рис.2 активен только переход bi. При срабатывании перехода он поглощает по одной фишке из всех состояний, которые вели в переход, а также добавляет по фишке в каждое состояние, в которое исходит из перехода дуга. Так, после срабатывания перехода bi сеть Петри будет иметь разметку, как указано на рис.4.1. Далее, сработает переход cb (рис. 4.2) – он поглотит 1 фишку и произведет по фишке в каждое состояние, в которое из него ведет дуга.

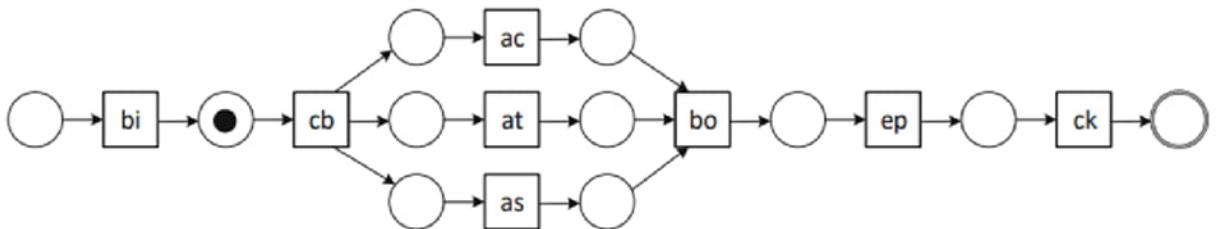


Рисунок 4.1 - Приготовление пиццы, сработал переход bi

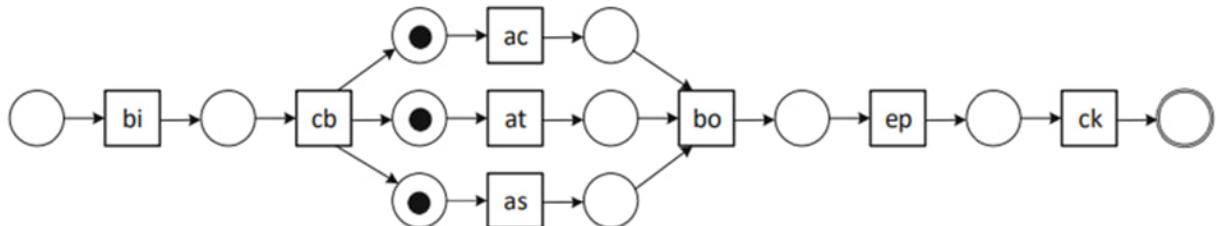


Рисунок 4.2 - Приготовление пиццы, сработал переход cb

В получившейся разметке на рис.4.2 активны три перехода – ac, at, as. Сети Петри позволяют недетерминированное поведение – так, любой из трёх перечисленных переходов может сработать. Предположим, сработал переход at, а после него – переход as.

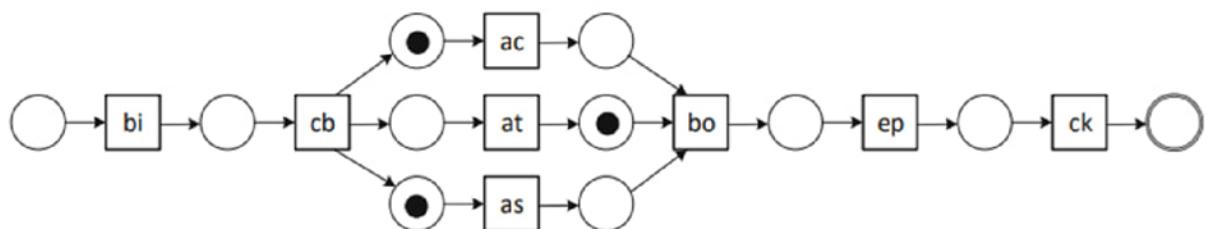


Рисунок 5.1 - Приготовление пиццы, сработал переход at

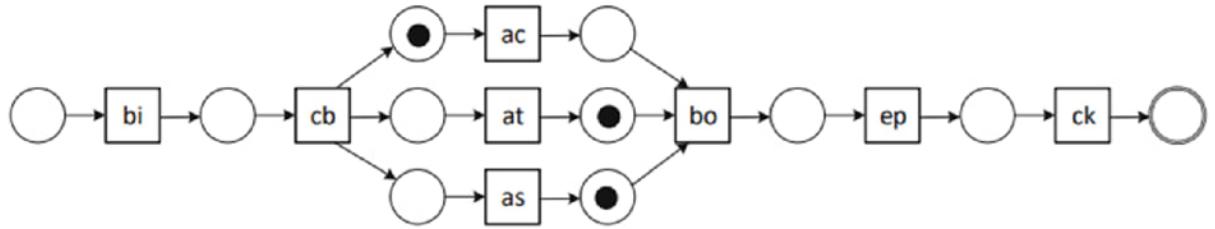


Рисунок 5.2 - Приготовление пиццы, сработал переход as

Заметим, что в полученной на рис.5.2 разметке активен только переход ac, а переход bo не является активным – в верхнем состоянии нет фишк. Сработает единственный активный переход – ac.

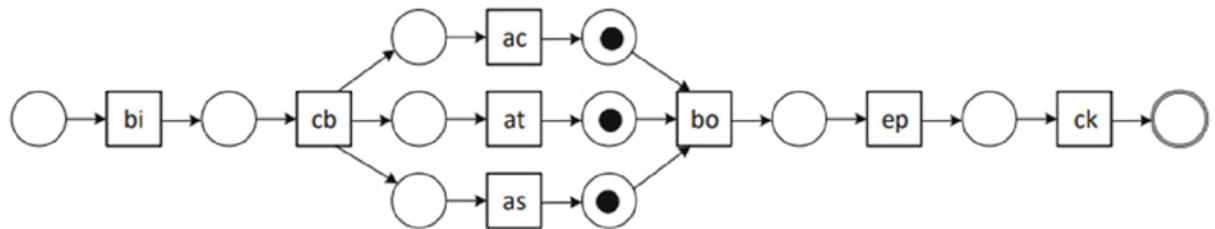


Рисунок 6 - Приготовление пиццы, сработал переход ac

Теперь активным является только переход bo. Сработав, он поглотит 3 фишк, а произведёт одну.

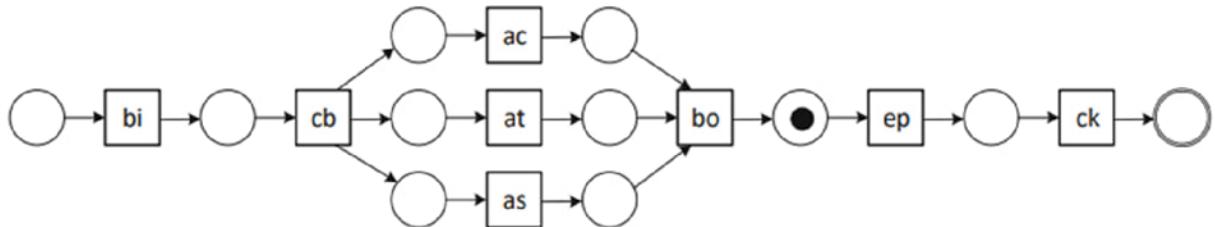


Рисунок 7 - Приготовление пиццы, сработал переход bo

Далее, последовательно сработают переходы ep и ck, после чего выполнение процесса прекратится.

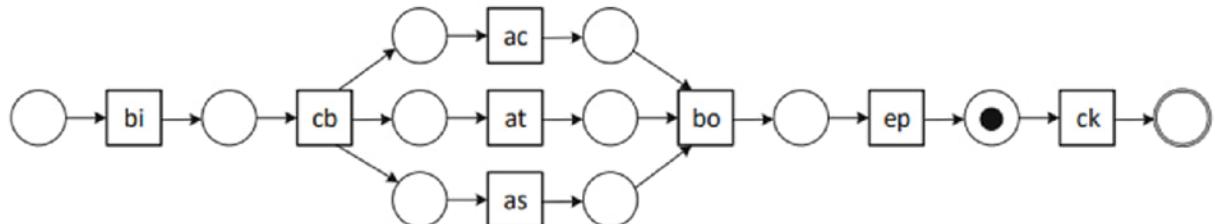


Рисунок 8.1 - Приготовление пиццы, сработал переход ep

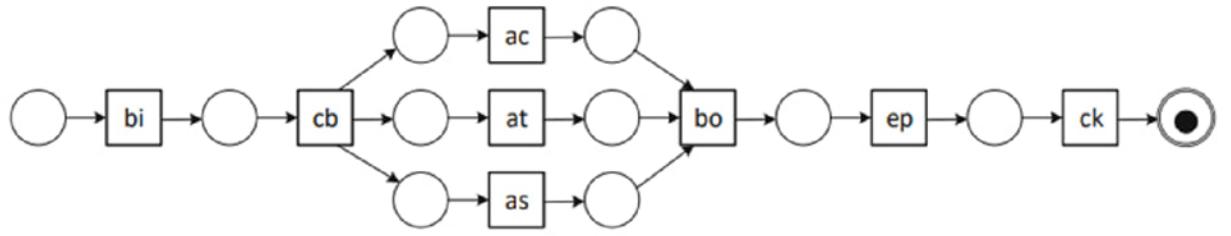


Рисунок 8.2 - Приготовление пиццы, сработал переход ск. Финальное состояние

## 1.2. Журнал событий

Дисциплина process mining направлена на анализ и улучшение процессов путём использования данных о событии. Эти данные могут быть извлечены из журнала событий. Журнал событий – это след, оставленный в системе в результате выполнения процессов. Журнал событий абстрактно можно представить как CSV файл или как реляционную базу данных. На рис.9 изображена часть журнала событий процесса приготовления пиццы. Каждая строка соответствует определенному событию. Каждое событие содержит несколько атрибутов, в данном примере это: case – случай, относится к экземпляру процесса; activity – действие, описывающее выполнение какой-то операции; timestamp – временная метка, указывает на момент происхождения события; resource – человек, обслуживший покупателя; customer – покупатель. В process mining ключевыми атрибутами являются case, activity, timestamp [1].

Case	Activity	Timestamp	Resource	Customer
...	...	...	...	...
pizza-56	buy ingredients (bi)	18:10	Stefano	Valentina
pizza-57	buy ingredients (bi)	18:12	Stefano	Giulia
pizza-57	create base (cb)	18:16	Mario	Giulia
pizza-56	create base (cb)	18:19	Mario	Valentina
pizza-57	add tomato (at)	18:21	Mario	Giulia
pizza-57	add cheese (ac)	18:27	Mario	Giulia
pizza-56	add cheese (ac)	18:34	Mario	Valentina
pizza-56	add tomato (at)	18:44	Mario	Valentina
pizza-56	add salami (as)	18:45	Mario	Valentina
pizza-56	bake in oven (bo)	18:48	Stefano	Valentina
pizza-57	add salami (as)	18:50	Mario	Giulia
pizza-56	eat pizza (ep)	19:10	Valentina	Valentina
pizza-58	buy ingredients (bi)	19:17	Stefano	Laura
pizza-57	bake in oven (bo)	19:23	Stefano	Giulia
pizza-57	eat pizza (ep)	19:27	Giulia	Giulia
pizza-57	clean kitchen (ck)	19:44	Mario	Giulia
pizza-58	create base (cb)	19:48	Mario	Laura
pizza-58	add salami (as)	19:49	Mario	Laura
pizza-58	add tomato (at)	19:55	Mario	Laura
pizza-56	clean kitchen (ck)	20:08	Mario	Valentina
pizza-58	add cheese (ac)	20:13	Mario	Laura
pizza-58	bake in oven (bo)	20:29	Stefano	Laura
pizza-58	eat pizza (ep)	20:48	Laura	Laura
pizza-58	clean kitchen (ck)	20:51	Mario	Laura
...	...	...	...	...

Рисунок 9 - Журнал событий процесса приготовления пиццы

Назовем трассой (trace) последовательность событий, относящихся к одному случаю (case). Сгруппируем все события в мультимножества трасс, количество появлений трассы в журнале событий обозначим верхним индексом.

Рассмотрим следующий журнал событий (рис.10.1). Из него можно синтезировать модель на рис.10.2. Около каждой дуги подписано, сколько раз она была использована.

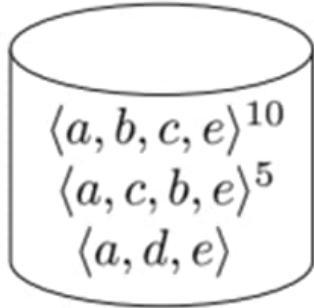


Рисунок 10.1 - Журнал событий

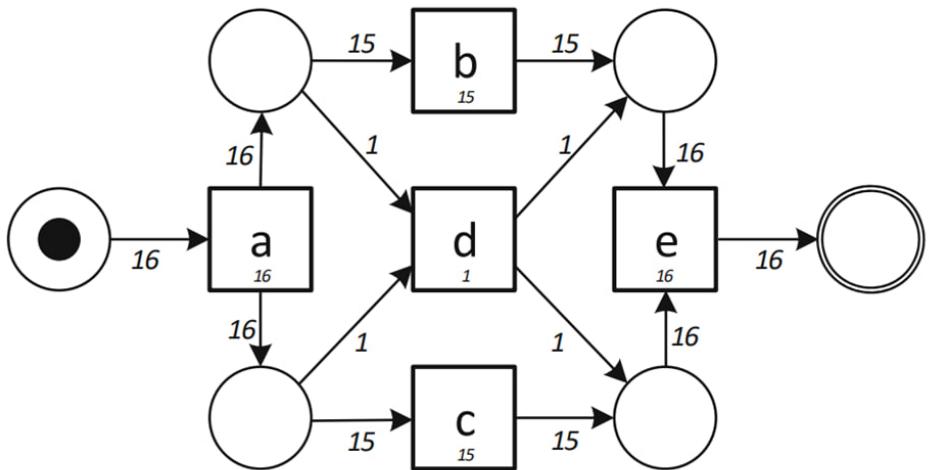


Рисунок 10.2 - Модель А

### 1.3. Отношения между событиями

Зачастую в процессах можно выделить какие-либо закономерности, связанные с порядком событий. Например, в рассмотренном выше примере о приготовлении пиццы покупка продуктов всегда будет производиться до подготовки теста. Завершающим событием будет являться уборка на кухне. А добавление ингредиентов, в свою очередь, может производиться в любом порядке.

Таким образом, события могут состоять в связи - например, покупка продуктов и приготовление пиццы находятся в причинно-следственной связи. Для отображения связи вводятся отношения между событиями. Одно из применений журнала событий - выявление отношений между событиями.

Пусть  $\mathcal{L}$  - журнал событий над множеством событий  $\mathcal{A}$ . Пусть  $a, b \in \mathcal{A}$ . Выделяют 4 основных отношения между событиями, которыми описываются закономерности в журнале событий [6]:

- $a > b$  (события  $a$  и  $b$  находятся в отношении  $>$ ) тогда и только тогда, когда существуют  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$  и  $i \in \{1, \dots, n-1\}$ , такие что  $\sigma \in \mathcal{L}$  и  $t_i = a$  и  $t_{i+1} = b$ ;

- $a \rightarrow b$  тогда и тогда тогда, когда  $a > b$  и  $b \not> a$  (события  $a$  и  $b$  не находятся в отношении  $>$ );
- $a \# b$  тогда и только тогда, когда  $a \not> b$  и  $b \not> a$ ;
- $a \parallel b$  тогда и только тогда, когда  $a > b$  и  $b > a$ .

Рассмотрим журнал событий  $\mathcal{L}_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$ . Выделим для него все вышеперечисленные отношения:

$$>_{\mathcal{L}_1} = \{(a, b), (a, c), (a, e), (b, c), (c, b), (b, d), (c, d), (e, d)\}$$

$$\rightarrow_{\mathcal{L}_1} = \{(a, b), (a, c), (a, e), (b, d), (c, d), (e, d)\}$$

$$\#_{\mathcal{L}_1} = \{(a, a), (a, d), (b, b), (b, e), (c, c), (c, e), (d, a), (d, d), (e, b), (e, c), (e, e)\}$$

$$\parallel_{\mathcal{L}_1} = \{(b, c), (c, b)\}$$

Отношение  $>_{\mathcal{L}_1}$  содержит все пары событий, находящихся в отношении "непосредственно следует". Например,  $a > c$ , потому что  $c$  сразу следует за  $a$  в трассе  $\langle a, c, b, d \rangle$ . Однако  $c \not> a$ , потому что ни в какой трассе  $a$  не следует за  $c$ . Отношение  $\rightarrow$  содержит все пары событий, находящихся в причинно-следственном отношении - так,  $a \rightarrow c$ , потому что иногда  $c$  непосредственно следует за  $a$  и  $a$  никогда не следует за  $c$  непосредственно.  $b \parallel c$  потому что  $b > c$  и  $c > b$  - иногда  $c$  следует за  $b$  и наоборот.  $a \# d$  потому что  $a \not> d$  и  $d \not> a$ .

Для любого журнала событий  $\mathcal{L}$  над множеством событий  $\mathcal{A}$  и для любых  $x, y \in \mathcal{A}$  верно, что  $x \rightarrow_{\mathcal{L}} y$ ,  $y \rightarrow_{\mathcal{L}} x$ ,  $x \#_{\mathcal{L}} y$  или  $x \parallel_{\mathcal{L}} y$ , то есть ровно одно из этих отношений верно. Таким образом, можно составить матрицу отношений журнала событий  $\mathcal{L}$ .

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>a</i>	# <sub><math>L_1</math></sub>	$\rightarrow_{L_1}$	$\rightarrow_{L_1}$	# <sub><math>L_1</math></sub>	$\rightarrow_{L_1}$
<i>b</i>	$\leftarrow_{L_1}$	# <sub><math>L_1</math></sub>	$\parallel_{L_1}$	$\rightarrow_{L_1}$	# <sub><math>L_1</math></sub>
<i>c</i>	$\leftarrow_{L_1}$	$\parallel_{L_1}$	# <sub><math>L_1</math></sub>	$\rightarrow_{L_1}$	# <sub><math>L_1</math></sub>
<i>d</i>	# <sub><math>L_1</math></sub>	$\leftarrow_{L_1}$	$\leftarrow_{L_1}$	# <sub><math>L_1</math></sub>	$\leftarrow_{L_1}$
<i>e</i>	$\leftarrow_{L_1}$	# <sub><math>L_1</math></sub>	# <sub><math>L_1</math></sub>	$\rightarrow_{L_1}$	# <sub><math>L_1</math></sub>

Рисунок 11 - Матрица отношений журнала событий

Как уже было сказано выше, отношения между событиями в журнале могут быть использованы для обнаружения шаблонов отношений в соответствующей модели.

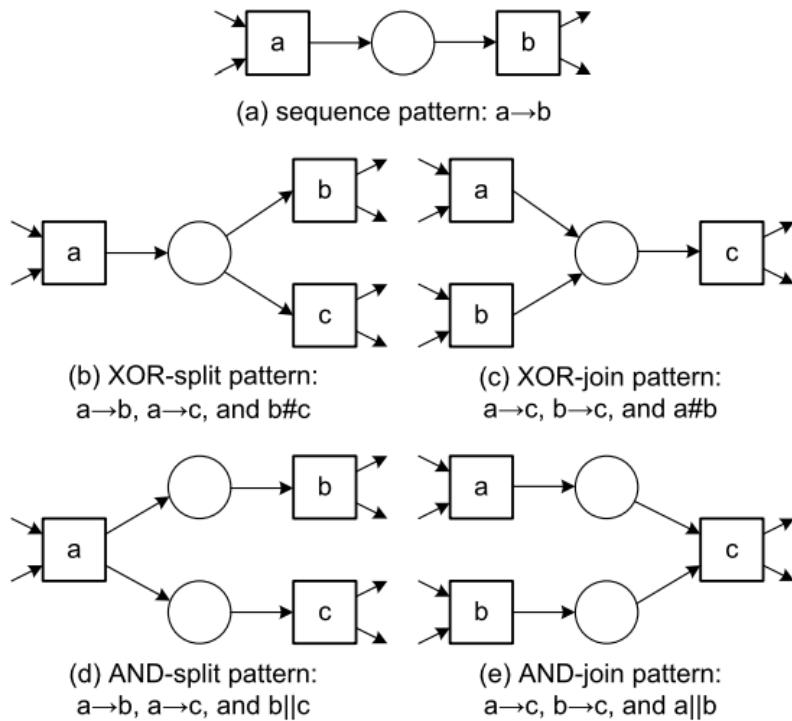


Рисунок 12 - Шаблоны отношений

#### 1.4. Conformance checking

Процесс построения, или синтеза, моделей из журнала событий – одна из задач process mining.

Однако заметим, что на рис.10.2 изображена не единственная модель, которая может быть получена из журнала событий, рис.10.1. Так, примерами других моделей могут послужить модели с рис.13.1 и рис.13.2.

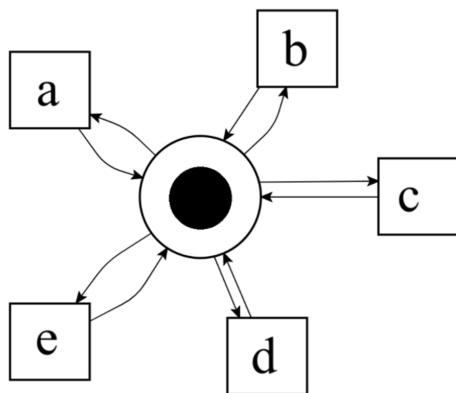


Рисунок 13.1 - Flower model - модель В

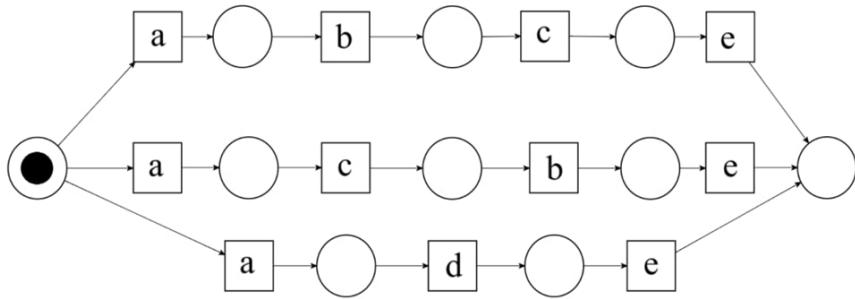


Рисунок 13.2 - Модель С

Однако интуитивно понятно, что модель А описывает процесс лучше. В модели С, например, для каждой трассы создана уникальная последовательность состояний и переходов. В ней первое ветвление является излишним – три перехода можно объединить в один, так как они все отвечают за исполнение одной и той же операции. Модель В (имеющая название «модель-цветок» – flower model), в свою очередь, разрешает любую последовательность из возможных событий. Модель В является слишком общей и не отражает специфику процесса.

Мы столкнулись с проблемой, которую решает проверка соответствия (conformance checking) – необходимо как-то оценить качество полученных или существовавших заранее моделей.

Меру соответствия удобно оценивать численно. Однако можно утверждать, что модель В является слишком общей, а модель С является слишком избыточной в сравнении с моделью А. Заметим также, что модель В не является избыточнее (в плане структуры), а модель С не является более общей, чем модель А. На этом примере видно, что модели В и С хуже А только по некоторым параметрам. Таким образом, становится очевидным необходимость проверки соответствия в нескольких измерениях. Чаще всего выделяют 4 измерения: fitness, precision, generalization и simplicity [4].

Fitness указывает на то, насколько точно модель может воспроизвести поведение, записанное в журнале событий.

Метрика precision нужна для нахождения слишком «общих» моделей – таких моделей, которые могут моделировать поведение, не наблюдаемое в журнале событий.

Метрика generalization оценивает, насколько точно модель сможет описать поведение, которое будет получено в будущем из журнала событий.

Simplicity описывает простоту модели. Так, модели с простой структурой предпочтительнее моделей с избыточными компонентами.

Зачастую значение метрики – коэффициент от 0 до 1, где 1 означает идеальное соответствие измерения, а 0 означает полное несоответствие.

На рис.14 изображены журнал событий, а также несколько моделей, синтезированных разными алгоритмами. У каждой модели были посчитаны 4 метрики. Заметим, что у некоторых моделей есть невидимые переходы (графически обозначаются черным прямоугольником). Такие переходы не наблюдаются в журнале событий.

Trace	#
A B C D E G	6
A B C D F G	38
A B D C E G	12
A B D C F G	26
A B C F G	8
A C B E G	1

Trace	#
A D B C F G	1
A D B C E G	1
A D C B F G	4
A C D B F G	2
A C B F G	1

Рисунок 14.1 - Журнал событий

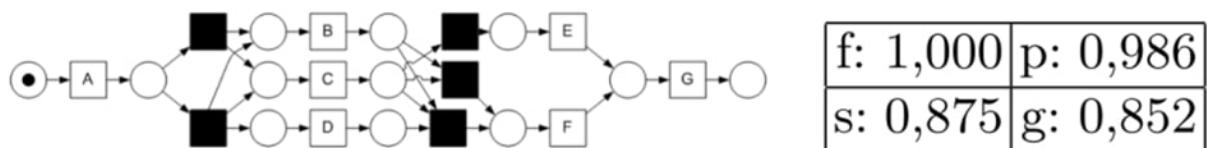
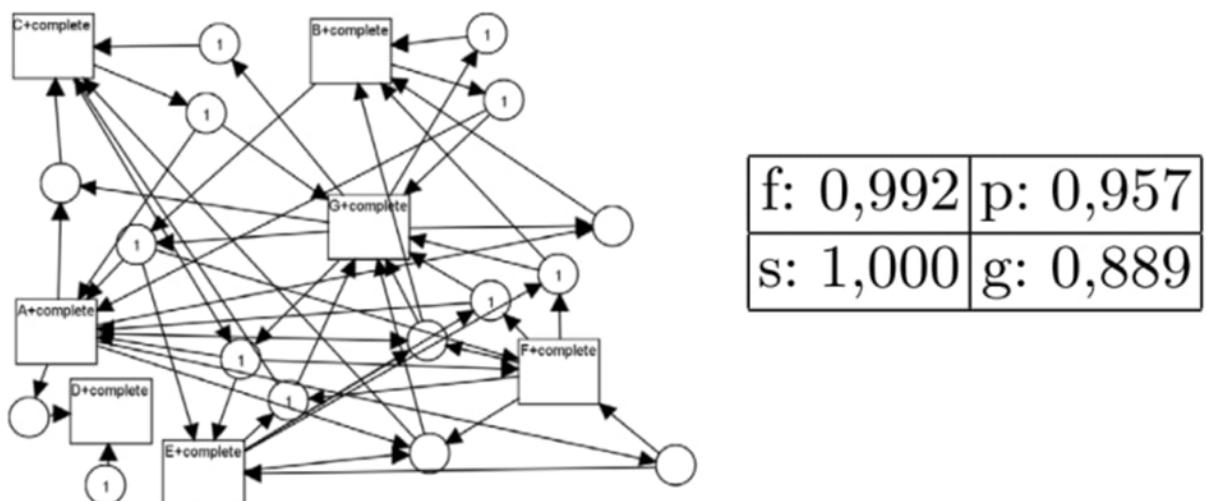
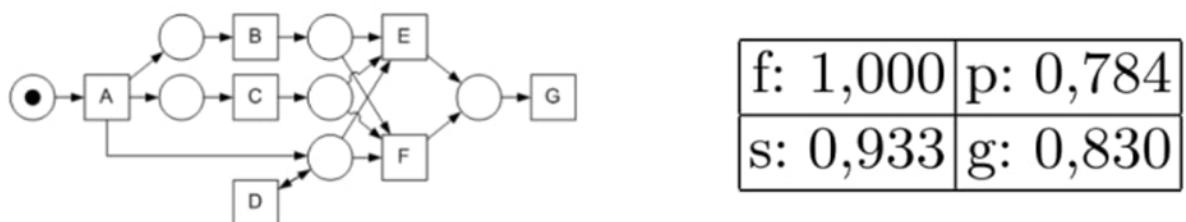
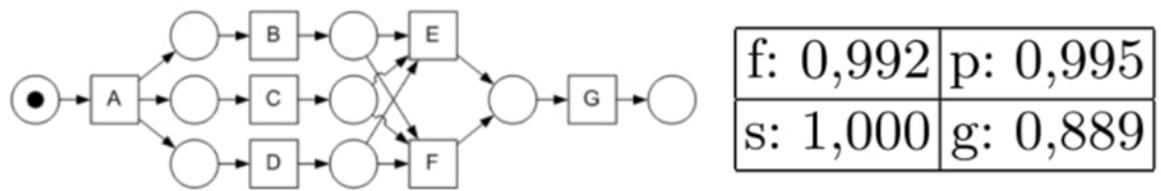


Рисунок 14.2 - Сгенерированные модели и подсчитанные метрики

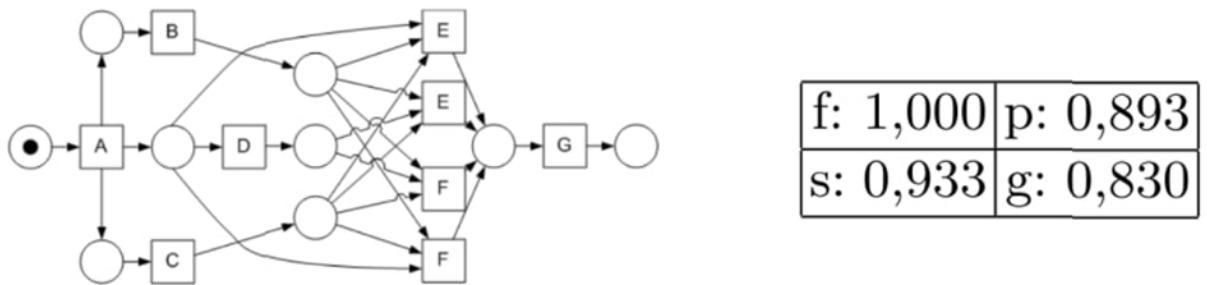
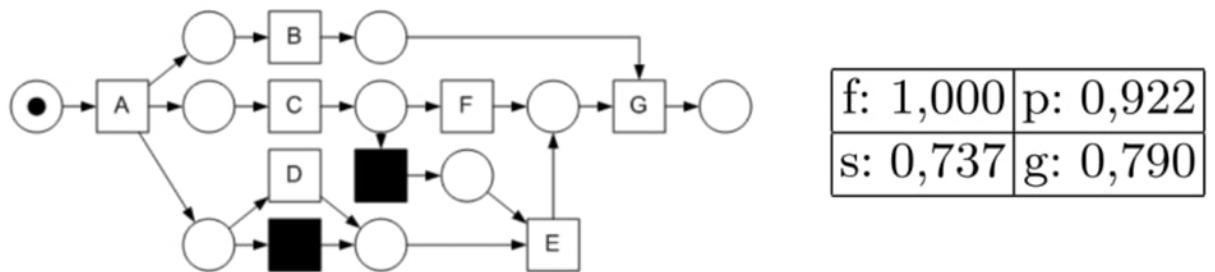
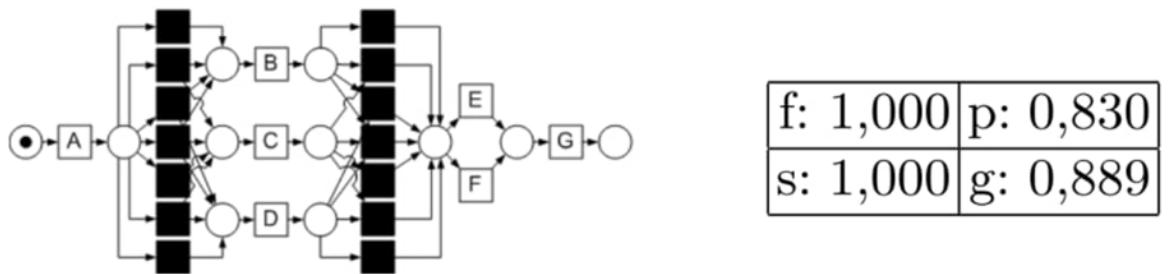


Рисунок 14.3 - Сгенерированные модели и подсчитанные метрики

### 1.5. Декларативные модели

Сети Петри являются императивной моделью – они хорошо контролируют заданное подмножество поведений. Но это подмножество поведений зачастую намного меньше всего множества поведений (рис.15, (b)). В некоторых процессах важна гибкость. Эту гибкость можно было бы задать какими-то границами или рамками, которые были бы достаточно широкими, чтобы фиксировать большую часть процесса. Для таких процессов подходят декларативные (declare) модели [6].

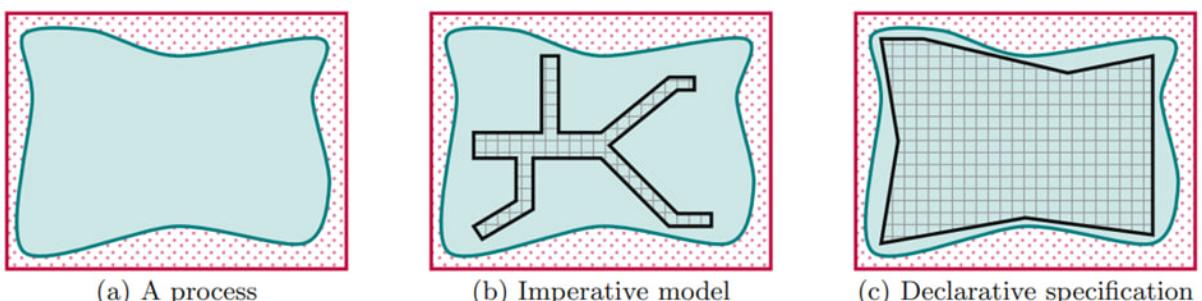


Рисунок 15 - Процесс, императивная и декларативная модели [6]

Декларативная модель задается как множество ограничений. Процесс, соответствующий модели, может «развиваться» любым непротиворечащим ограничениям образом.

Шаблоны ограничений могут быть описаны на языке линейной темпоральной логики (linear temporal logic, LTL).

На рис.16 приведены некоторые шаблоны ограничений. Например, Response(a, b) – ограничение, в котором появление в трассе действия a влечёт за собой появление действия b.

Constraint	Explanation	Examples			Notation	
Existence constraints						
INIT(a)	a is the first to occur	$\checkmark \langle a, c, a, c \rangle$	$\times \langle c, c \rangle$	$\times \langle b, a, c \rangle$		
ATLEASTONE(a)	a occurs at least once	$\checkmark \langle b, c, a, a, c \rangle$	$\times \langle b, c, c \rangle$	$\times \langle c \rangle$		
ATMOSTONE(a)	a occurs at most once	$\checkmark \langle b, c, a, c \rangle$	$\times \langle b, c, a, a, c \rangle$	$\times \langle b, c, a, c, a, a \rangle$		
END(a)	a is the last to occur	$\checkmark \langle b, c, a \rangle$	$\times \langle b, c \rangle$	$\times \langle b, a, c \rangle$		
Relation constraints						
RESPONDEDEXISTENCE(a, b)	If a occurs in the trace, then b occurs as well	$\checkmark \langle b, c, a, a, c \rangle$	$\checkmark \langle b, c, c \rangle$	$\times \langle c, a, a, c \rangle$	$\times \langle a, c, c \rangle$	
RESPONSE(a, b)	If a occurs, then b occurs after a	$\checkmark \langle c, a, a, c, b \rangle$	$\checkmark \langle b, c, c \rangle$	$\times \langle c, a, a, c \rangle$	$\times \langle b, a, c, c \rangle$	
ALTERNATERESPONSE(a, b)	Each time a occurs, then b occurs afterwards, and no other a recurs in between	$\checkmark \langle c, a, c, b \rangle$	$\checkmark \langle a, b, c, a, c, b \rangle$	$\times \langle c, a, a, c, b \rangle$	$\times \langle b, a, c, a, c, b \rangle$	
CHAINRESPONSE(a, b)	Each time a occurs, then b occurs immediately afterwards	$\checkmark \langle c, a, b, b \rangle$	$\checkmark \langle a, b, c, a, b \rangle$	$\times \langle c, a, c, b \rangle$	$\times \langle b, c, a \rangle$	
PRECEDENCE(a, b)	b occurs only if preceded by a	$\checkmark \langle c, a, c, b, b \rangle$	$\checkmark \langle a, c, c \rangle$	$\times \langle c, c, b, b \rangle$	$\times \langle b, a, c, c \rangle$	

Рисунок 16 - Шаблоны ограничений [6]

На рис.17 изображены две модели, описывающие один процесс покупки книги – императивная модель на рис.17.1 и декларативная на рис.17.2.

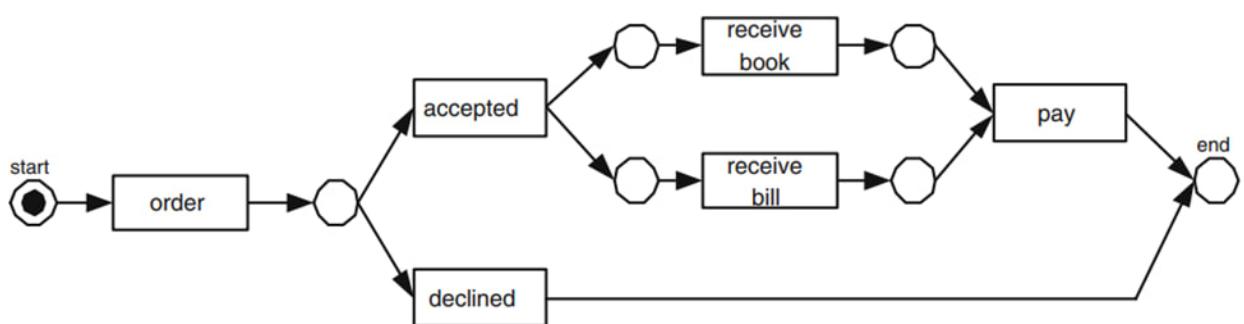


Рисунок 17.1 - Императивная модель

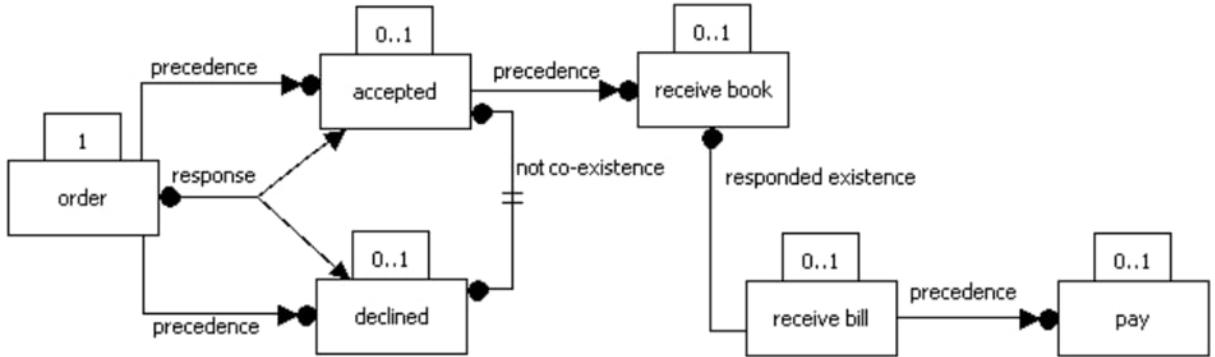


Рисунок 17.2 - Декларативная модель

### 1.6. Выводы и постановка задачи проверки соответствия

В данном разделе были систематизированы ключевые понятия, необходимые для задачи анализа соответствия:

- *журналы событий* как структурированные записи о выполнении процессов,
- *декларативные модели*, задающие ограничения на трассы,
- *сети Петри и отношения между событиями* как инструменты для анализа.

Несмотря на обширные исследования каждого из этих объектов (см., напр., [4, 5]), проблема автоматизированной проверки соответствия между декларативной моделью и журналом событий остаётся недостаточно изученной [9, 10, 11]. В частности, требуются:

1. Способы численной оценки степени соответствия,
2. Эффективные алгоритмы их вычисления.

В следующем разделе представлены методы решения описанных задач.

## 2. Методы и алгоритмы оценки соответствия декларативной модели журналу событий

Как было установлено в предыдущем разделе, существует множество видов отношений между событиями (шаблонов ограничений). В настоящем разделе в подразделах 2.1-2.6 рассматриваются бинарные отношения между событиями. Рассмотренные отношения предполагают ацикличность каждой трассы в журнале событий - любое событие не может появиться более 1 раза в любой трассе. К каждому из рассмотренных шаблонов представлены методы оценки соответствия журнала событий и декларативной модели, содержащей соответствующий шаблон, а также алгоритмы, реализующие описанные методы. В разделе 2.7 содержится метод оценки соответствия декларативной модели, содержащей любые из рассмотренных шаблонов, и журнала событий.

### 2.1. Ограничение Response

**Определение 2.1** (*Ограничение Response декларативной модели*) Пусть  $\mathcal{D}$  - декларативная модель над множеством событий  $\mathcal{A}$ ;  $a, b \in \mathcal{A}$ . Тогда если события  $a$  и  $b$  удовлетворяют ограничению  $\text{Response}(a, b)$ , или  $a <_{\mathcal{D}} b$ , то для любой трассы  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , не противоречащей модели  $\mathcal{D}$ , выполнено следующее:

$$\exists i \in \mathbb{N} : t_i = a \implies \exists j \in \mathbb{N} : i < j \wedge t_j = b$$

Другими словами, после появления в трассе события  $a$  рано или поздно также случится событие  $b$ .

Так, например, в трассе  $\langle a, b, d \rangle$  в отношении Response находятся пары  $(a, b)$ ,  $(b, d)$ ,  $(a, d)$ . А пара  $(d, a)$  не находится в отношении Response.

Нетрудно убедиться, что отношение Response транзитивно. Таким образом, можно составить матрицу  $M_{\mathcal{D}}$  ограничений декларативной модели  $\mathcal{D}$ .

**Определение 2.2** (*Матрица ограничений Response декларативной модели*) Пусть  $\mathcal{D}$  - декларативная модель над множеством событий  $\mathcal{A} = \{a_1, \dots, a_n\}$ , где  $n = |\mathcal{A}|$  - мощность множества событий. Назовем матрицей ограничений декларативной модели матрицу  $M_{\mathcal{D}}$  порядка  $n$ , элементы матрицы будут задаваться следующим образом: в  $i$ -ой строке в  $j$ -ом столбце находится 1, если события  $a_i$  и  $a_j$  связаны отношением Response, и 0 иначе:

$$M_{\mathcal{D}}[i, j] = \begin{cases} 1, & \text{Response}(a_i, a_j) \in \mathcal{D} \\ 0, & \text{Response}(a_i, a_j) \notin \mathcal{D} \end{cases}$$

Пусть  $\mathcal{A}_1 = \{a, b, c, d\}$ , и  $\mathcal{D}_1 = \{a <_{\mathcal{D}} d, d <_{\mathcal{D}} b\}$ . Заметим, что из  $\mathcal{D}$  напрямую следует отношение  $a <_{\mathcal{D}} b$ . Для таких декларативной модели и множества событий матрица ограничений модели будет выглядеть следующим образом:

$$M_{\mathcal{D}_1} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

	a	b	c	d
a	0	1	0	1
b	0	0	0	0
c	0	0	0	0
d	0	1	0	0

Определим аналогичное отношение для трассы.

**Определение 2.3** (*Отношение Response для трассы*) Пусть  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$  - трасса над множеством событий  $\mathcal{A}$ . Пусть  $a, b \in \mathcal{A}$ . Если события  $a$  и  $b$  находятся в отношении  $Response(a, b)$ , или  $a <_{\sigma} b$ , то:

$$\exists i, j \in \mathbb{N} : t_i = a \wedge t_j = b$$

Заметим, что отношение *Response* для трасс также транзитивно.

Определим матрицу  $M_{\sigma}$  отношений трассы  $\sigma$  аналогично матрице  $M_{\mathcal{D}}$  ограничений декларативной модели  $\mathcal{D}$ .

**Определение 2.4** (*Матрица отношений трассы*) Пусть  $\sigma$  - трасса над множеством событий  $\mathcal{A} = \{a_1, \dots, a_n\}$ , где  $n = |\mathcal{A}|$  - мощность множества событий. Назовем матрицей отношений трассы матрицу  $M_{\sigma}$  порядка  $n$ , элементы матрицы будут задаваться следующим образом: в  $i$ -ой строке в  $j$ -ом столбце находится 1, если события  $a_i$  и  $a_j$  связаны отношением *Response*, и 0 иначе:

$$M_{\sigma}[i, j] = \begin{cases} 1, & a_i <_{\sigma} a_j \\ 0, & a_i \not<_{\sigma} a_j \end{cases}$$

Для множества событий  $\mathcal{A}_1$  и трассы  $\sigma_1 = \langle a, d, b, c \rangle$  матрица отношений будет выглядеть следующим образом:

$$M_{\sigma_1} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

	a	b	c	d
a	0	1	1	1
b	0	0	1	0
c	0	0	0	0
d	0	1	1	0

Нетрудно заметить, что матрицы  $M_{\mathcal{D}}$  и  $M_{\sigma}$  схожи не только по структуре, но и "по смыслу". Естественным кажется попытаться наложить эти матрицы друг на друга, сравнить их и получить выводы о соответствии трассы  $\sigma$  ограничениям  $\mathcal{D}$ .

Интуитивно понятно, что если элементы матриц совпадают -  $M_{\mathcal{D}}[i, j] = M_{\sigma}[i, j]$  - то события  $a_i, a_j$  не противоречат ограничениям. Более того, если  $M_{\mathcal{D}}[i, j] = 0$ , то  $M_{\sigma}[i, j]$  может быть равным как 0, так и 1 - на события  $a_i, a_j$  не наложены ограничения, поэтому любое поведение допустимо. Остался только случай  $M_{\mathcal{D}}[i, j] = 1, M_{\sigma}[i, j] = 0$ . Если  $a_i$  встретился в трассе, то ограничение  $a_i <_{\mathcal{D}} a_j$  было нарушено. Однако если  $a_i$  не встретился в трассе, то ограничение  $a_i <_{\mathcal{D}} a_j$  не было нарушено. Это наблюдение показывает необходимость хранения дополнительной информации о трассе, по которой можно понять, встречается ли событие в трассе.

**Определение 2.5** (*Индикаторный массив*) Пусть  $\sigma$  - трасса над множеством событий  $\mathcal{A} = \{a_1, \dots, a_n\}$ , где  $n = |\mathcal{A}|$  - мощность множества событий. Введем индикаторный массив  $I_{\sigma}$  размера  $1 \times n$  для трассы  $\sigma$ . Массив будет хранить информацию о появлении каждого из событий в трассе. Так,  $I_{\sigma}[i] = 1$ , если  $a_i \in \sigma$ , и  $I_{\sigma}[i] = 0$ , если

$a_i \notin \sigma$ .

$$I_\sigma[i] = \begin{cases} 1, & a_i \in \sigma \\ 0, & a_i \notin \sigma \end{cases}$$

Для трассы  $\sigma_1$ , в которой встречаются все события из  $\mathcal{A}_1$ , индикаторный массив  $I_{\sigma_1}$  выглядит следующим образом:

$$I_{\sigma_1} = \begin{array}{|c|c|c|c|} \hline & a & b & c & d \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Теперь введем символ  $\delta$ , позволяющий сравнивать матрицы  $M_D$  и  $M_\sigma$  поэлементно с учетом вышеописанных наблюдений.

**Определение 2.6** (*Символ  $\delta$* ) Пусть  $\mathcal{D}$  и  $\sigma$  - декларативная модель и трасса над множеством событий  $\mathcal{A}$ , и  $M_D, M_\sigma$  - их матрицы ограничений и отношений соответственно. Символ  $\delta[i, j]$ , отвечающий за сравнение элементов  $M_D[i, j]$  и  $M_\sigma[i, j]$ , равен:

1. 0, если ограничение  $a_i < a_j$  есть в декларативной модели, а в трассе встречается событие  $a_i$  и после него не встречается  $a_j$
2. 1, если либо в модели нет ограничения Response на события  $a_i, a_j$ , либо отношение  $a_i <_\sigma a_j$  имеет место в трассе, либо событие  $a_i$  не встретилось в трассе.

$$\delta[i, j] = \begin{cases} 0, & M_D[i, j] = 1 \wedge M_\sigma[i, j] = 0 \wedge I_\sigma[i] = 1 \\ 1, & M_D[i, j] = 0 \vee M_\sigma[i, j] = 1 \vee I_\sigma[i] = 0 \end{cases}$$

Для визуализации объединим символы  $\delta[i, j]$  в матрицу  $\Delta$ . Так, для рассмотренного примера матрица  $\Delta$  будет выглядеть следующим образом:

$$\Delta = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ \hline b & 1 & 1 & 1 & 1 \\ \hline c & 1 & 1 & 1 & 1 \\ \hline d & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

С помощью символа  $\delta$  можно ввести коэффициент соответствия  $\mathcal{F}$  декларативной модели и трассы.

**Определение 2.7** (*Коэффициент соответствия декларативной модели с ограничениями вида Response и трассы*)

Пусть  $\mathcal{D}$  и  $\sigma$  - декларативная модель с ограничениями вида Response и трасса над множеством событий  $\mathcal{A}$ . Пусть  $t$  – количество ограничений в транзитивном замыкании модели  $\mathcal{D}$ , то есть  $t$  является суммой элементов матрицы ограничений  $M_D$ .

Назовем  $\mathcal{F}(\sigma, \mathcal{D})$  коэффициентом соответствия  $\mathcal{D}$  и  $\sigma$ :

$$\mathcal{F}(\sigma, \mathcal{D}) = 1 - \frac{1}{m} \sum_{1 \leq i, j \leq n} (1 - \delta[i, j])$$

По своей сути коэффициент  $\mathcal{F}$  считает долю ограничений, которые были нарушены, и вычитает её из единицы. Такой коэффициент принимает значения от 0 до 1 включительно, и чем больше значение, тем большее степень соответствия. Так, значение 1 отображает идеальное соответствие трассы модели, близкие к 1 значения отображают хорошее соответствие, а значение 0 отображает полное несоответствие.

Для рассмотренных выше трассы  $\sigma_1$  и декларативной модели  $\mathcal{D}_1$ :  $\mathcal{F}(\sigma_1, \mathcal{D}_1) = 1$ . Действительно, трасса удовлетворяет всем ограничениям.

Понятие коэффициента соответствия трассы модели можно обобщить до журнала событий, взяв среднее арифметическое всех трасс в журнале.

**Определение 2.8** (*Коэффициент соответствия декларативной модели с ограничениями вида Response и журнала*)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  - журнал событий и декларативная модель с ограничениями вида Response над множеством событий  $\mathcal{A}$  соответственно; пусть  $k \in \mathbb{R}$  – степень штрафа за нарушение ограничения. Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^k(\sigma, \mathcal{D})$$

Для моделей, где нарушение ограничений является критичным, может потребоваться ввести более строгий штраф за несоответствие. Регулировать степень штрафа можно путём возведения каждого коэффициента в степень  $k$ . Понятно, что при увеличении  $k$  увеличивается и штраф за несоответствие. При этом целесообразно брать  $k \geq 1$ , так как иначе коэффициент будет получаться смешённым к 1.

Пусть  $\mathcal{A}_2 = \{a, b, c, d\}$ ,  $\mathcal{L}_2 = \{\sigma_1 = \langle a, b, d \rangle^{100}, \sigma_2 = \langle a, c, d \rangle^{100}, \sigma_3 = \langle a, b, d, c \rangle^{25}, \sigma_4 = \langle a, b, c, d \rangle^{25}\}$ ,  $\mathcal{D}_2 = \{b < d, c < d\}$ . Найдем коэффициент соответствия  $\mathcal{L}$  и  $\mathcal{D}$  при  $k = 1$ .

$$M_{\mathcal{D}_2} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 0 & 0 & 0 & 0 \\ \hline b & 0 & 0 & 0 & 1 \\ \hline c & 0 & 0 & 0 & 1 \\ \hline d & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad m = 2$$

Для  $\sigma_1 = \langle a, b, d \rangle$ :

$$M_{\sigma_1} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 0 & 1 & 0 & 1 \\ \hline b & 0 & 0 & 0 & 1 \\ \hline c & 0 & 0 & 0 & 0 \\ \hline d & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad I_{\sigma_1} = \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 1 & 1 & 0 & 1 \\ \hline \end{array} \quad \Delta_{\sigma_1} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ \hline b & 1 & 1 & 1 & 1 \\ \hline c & 1 & 1 & 1 & 1 \\ \hline d & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\mathcal{F}(\sigma_1, \mathcal{D}_2) = 1 - \frac{1}{m} \sum_{1 \leq i, j \leq n} (1 - \delta[i, j]) = 1 - \frac{1}{2} \cdot 0 = 1$$

Для  $\sigma_2 = \langle a, c, d \rangle$ :

$$M_{\sigma_2} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 0 & 0 & 1 & 1 \\ \hline b & 0 & 0 & 0 & 0 \\ \hline c & 0 & 0 & 0 & 1 \\ \hline d & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad I_{\sigma_2} = \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 1 & 0 & 1 & 1 \\ \hline \end{array} \quad \Delta_{\sigma_2} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ \hline b & 1 & 1 & 1 & 1 \\ \hline c & 1 & 1 & 1 & 1 \\ \hline d & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\mathcal{F}(\sigma_2, D_2) = 1 - \frac{1}{2} \cdot 0 = 1$$

Для  $\sigma_3 = \langle a, b, d, c \rangle$ :

$$M_{\sigma_3} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 0 & 1 & 1 & 1 \\ \hline b & 0 & 0 & 1 & 1 \\ \hline c & 0 & 0 & 0 & 0 \\ \hline d & 0 & 0 & 1 & 0 \\ \hline \end{array} \quad I_{\sigma_3} = \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 1 & 1 & 1 & 0 \\ \hline \end{array} \quad \Delta_{\sigma_3} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ \hline b & 1 & 1 & 1 & 0 \\ \hline c & 1 & 1 & 1 & 0 \\ \hline d & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\mathcal{F}(\sigma_3, D_2) = 1 - \frac{1}{2} \cdot 1 = \frac{1}{2}$$

Для  $\sigma_4 = \langle a, b, c, d \rangle$ :

$$M_{\sigma_4} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 0 & 1 & 1 & 1 \\ \hline b & 0 & 0 & 1 & 1 \\ \hline c & 0 & 0 & 0 & 1 \\ \hline d & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad I_{\sigma_3} = \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array} \quad \Delta_{\sigma_3} = \begin{array}{|c|c|c|c|c|} \hline & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ \hline b & 1 & 1 & 1 & 1 \\ \hline c & 1 & 1 & 1 & 1 \\ \hline d & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\mathcal{F}(\sigma_4, D_2) = 1 - \frac{1}{2} \cdot 0 = 1$$

Таким образом, коэффициент соответствия:

$$\mathcal{F}(\mathcal{L}_2, \mathcal{D}_2) = \frac{1}{|\mathcal{L}_2|} \sum_{\sigma \in \mathcal{L}_2} \mathcal{F}(\sigma, \mathcal{D}_2) = \frac{1}{250} \cdot (100 \cdot 1 + 100 \cdot 1 + 25 \cdot \frac{1}{2} + 25 \cdot 1) = \frac{19}{20} = 0.95$$

Заметим, что в рассмотренном примере при  $k \rightarrow \infty$  коэффициент  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k) \rightarrow 0.9$ .

Опишем функцию, вычисляющую коэффициент соответствия данных на вход модели и журнала событий. Для этого также опишем другие необходимые функции.

Функция `Follows_Trace_Matrix` - заполнение матрицы отношений трассы.

`Follows_Trace_Matrix(trace, n)`

```

1:  $M \leftarrow \text{Array}[n][n]$             $\triangleright$  Создание заполненного нулями массива размера  $n \times n$ 
2: for  $i \leftarrow 1$  to  $\text{Length}(trace) - 1$  do
3:   for  $j \leftarrow i + 1$  to  $\text{Length}(trace)$  do
4:      $M[\text{trace}[i]][\text{trace}[j]] = 1$ 
5:   end for
6: end for
7: return  $M$ 

```

Сложность алгоритма -  $O(|trace|^2) = O(n^2)$ .

Функция `Response_Declare_Matrix` - заполнение матрицы ограничений Response декларативной модели.

`Response_Declare_Matrix(Declare, n)`

```
1:  $M \leftarrow \text{Array}[n][n]$ 
```

```

2: for  $i \leftarrow 1$  to  $\text{Length}(\text{Declare})$  do                                 $\triangleright$  Заполнение ограничений из модели
3:    $a, b \leftarrow \text{Declare}[i]$ 
4:    $M[a][b] = 1$ 
5: end for
6: for  $i \leftarrow 1$  to  $n$  do                                          $\triangleright$  Заполнение транзитивных ограничений
7:   for  $j \leftarrow 1$  to  $n$  do
8:     if  $M[i][j] = 1$  then
9:       for  $k \leftarrow 1$  to  $n$  do
10:        if  $M[j][k] = 1$  then
11:           $M[i][k] = 1$ 
12:        end if
13:      end for
14:    end if
15:  end for
16: end for
17: return  $M$ 

```

Сложность алгоритма -  $O(n^2)$ .

Функция Indicator\_Array - заполнение индикаторного массива на основе трассы.

Indicator\_Array(trace, n)

```

1:  $I \leftarrow \text{Array}[n]$ 
2: for  $k \leftarrow 1$  to  $\text{Length}(\text{trace})$  do
3:    $I[\text{trace}[k]] \leftarrow 1$ 
4: end for
5: return  $I$ 

```

Сложность алгоритма -  $O(|\text{trace}|) = O(n)$ .

Вычисление  $\Delta$ -матрицы:

Response\_Delta\_Matrix(trace, Declare, n)

```

1:  $T \leftarrow \text{Follows\_Trace\_Matrix}(\text{trace}, n)$ 
2:  $I \leftarrow \text{Indicator\_Array}(\text{trace}, n)$ 
3:  $D \leftarrow \text{Response\_Declare\_Matrix}(\text{Declare}, n)$ 
4:  $M \leftarrow \text{Array}[n][n]$ 
5: for  $i \leftarrow 1$  to  $n$  do
6:   for  $j \leftarrow 1$  to  $n$  do
7:     if  $D[i][j] = 0$  or  $T[i][j] = 1$  or  $I[i] = 0$  then
8:        $M[i][j] \leftarrow 1$ 
9:     end if
10:   end for
11: end for
12: return  $M$ 

```

Сложность алгоритма -  $O(n^2)$ .

Функция Response\_Trace\_Fitness - вычисление коэффициента соответствия декларативной модели с ограничениями вида Response и трассы:

```

Response_Trace_Fitness(trace, Declare, n)
1:  $M \leftarrow Response\_Delta\_Matrix(trace, Declare, n)$ 
2:  $D \leftarrow Response\_Declare\_Matrix(Declare, n)$ 
3:  $S \leftarrow 0$ 
4:  $k \leftarrow 0$ 
5: for  $i \leftarrow 1$  to  $n$  do
6:   for  $j \leftarrow 1$  to  $n$  do
7:      $S \leftarrow S + 1 - M[i][j]$ 
8:      $k \leftarrow k + D[i][j]$ 
9:   end for
10: end for
11:  $c \leftarrow 1 - \frac{S}{k}$ 
12: return  $c$ 
```

Сложность алгоритма –  $O(n^2)$ .

Функция Response\_Log\_Fitness - вычисление коэффициента соответствия декларативной модели с ограничениями вида Response и журнала:

```

Response_Log_Fitness(Log, Declare, n, k)
1:  $S \leftarrow 0$ 
2: for  $i \leftarrow 1$  to  $Length(Log)$  do
3:    $S \leftarrow S + Response\_Trace\_Fitness^k(Log[i], Declare, n)$ 
4: end for
5:  $c \leftarrow \frac{S}{Length(Log)}$ 
6: return  $c$ 
```

Сложность алгоритма –  $O(|Log| \cdot n^2)$ .

## 2.2. Ограничение RespondedExistence

**Определение 2.9** (*Ограничение RespondedExistence декларативной модели*)

Пусть  $\mathcal{D}$  - декларативная модель над множеством событий  $\mathcal{A}$ ;  $a, b \in \mathcal{A}$ . Тогда если события  $a$  и  $b$  удовлетворяют ограничению  $RespondedExistence(a, b)$ , то для любой трассы  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , не противоречащей модели  $\mathcal{D}$ , выполнено следующее:

$$\exists i \in \mathbb{N} : t_i = a \implies \exists j \in \mathbb{N} : t_j = b$$

Таким образом, если события удовлетворяют ограничению  $RespondedExistence$ , то во всех трассах, в которых встречается первое событие, встречается и второе.

Так, в рассмотренном ранее примере трассы  $\sigma_2 = \langle a, c, d \rangle$ ,  $\sigma_3 = \langle a, b, d, c \rangle$ ,  $\sigma_4 = \langle a, b, c, d \rangle$  удовлетворяют ограничению  $RespondedExistence(b, c)$ , а трасса  $\sigma_1 = \langle a, b, d \rangle$  – нет.

В последующих определениях записанное в квадратных скобках выражение является значением логической функции, которая принимает значение 1, если выражение истинно, и 0 иначе. Так,  $[x < y] = 1$ , если  $x < y$ , и  $[x < y] = 0$ , если  $x \geq y$ .

**Определение 2.10** (*Коэффициент соответствия декларативной модели с ограничениями вида RespondedExistence и трассы*)

Пусть  $\mathcal{D}$  и  $\sigma$  – декларативная модель с ограничениями вида RespondedExistence и трасса над множеством событий  $\mathcal{A}$ . Назовем  $\mathcal{F}(\sigma, \mathcal{D})$  коэффициентом соответствия  $\mathcal{D}$  и  $\sigma$ :

$$\mathcal{F}(\sigma, D) = \frac{1}{n} \sum_{(d_1, d_2) \in \mathcal{D}} [I_\sigma[d_1]] \leq I_\sigma[d_2]]$$

**Определение 2.11** (*Коэффициент соответствия декларативной модели с ограничениями вида RespondedExistence и журнала*)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  – журнал событий и декларативная модель с ограничениями вида RespondedExistence над множеством событий  $\mathcal{A}$  соответственно;  $k \in \mathbb{R}$ . Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^k(\sigma, \mathcal{D})$$

RespondedExistence\\_Trace\\_Fitness(trace, Declare, n)

```

1:  $S \leftarrow 0$ 
2:  $IndicatorArray \leftarrow Indicator\_Array(trace, n)$ 
3: for  $i \leftarrow 1$  to  $Length(Declare)$  do
4:    $a, b \leftarrow Declare[i]$ 
5:   if  $IndicatorArray[a] \leq IndicatorArray[b]$  then
6:      $S \leftarrow S + 1$ 
7:   end if
8: end for
9:  $c \leftarrow \frac{S}{len(Declare)}$ 
10: return  $c$ 
```

Сложность алгоритма –  $O(|Declare|) = O(n^2)$ .

RespondedExistence\\_Log\\_Fitness(Log, Declare, n, k)

```

1:  $S \leftarrow 0$ 
2: for  $trace \in Log$  do
3:    $S \leftarrow S + RespondedExistence\_Trace\_Fitness^k(trace, Declare, n)$ 
4: end for
5: return  $\frac{S}{len(Log)}$ 
```

Сложность алгоритма –  $O(|Log| \cdot n^2)$ .

### 2.3. Ограничение CoExistence

**Определение 2.12** (*Ограничение CoExistence декларативной модели*)

Пусть  $\mathcal{D}$  – декларативная модель над множеством событий  $\mathcal{A}$ ;  $a, b \in \mathcal{A}$ . Тогда если события  $a$  и  $b$  удовлетворяют ограничению  $CoExistence(a, b)$ , то для любой трассы  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , не противоречащей модели  $\mathcal{D}$ , выполнено следующее:

$$(\exists i \in \mathbb{N} : t_i = a \implies \exists j \in \mathbb{N} : t_j = b) \wedge (\exists i \in \mathbb{N} : t_i = b \implies \exists j \in \mathbb{N} : t_j = a)$$

Таким образом, события удовлетворяют ограничению *CoExistence*, если они либо оба встречаются, либо оба не встречаются в трассе.

Так, в рассмотренном ранее примере трассы  $\sigma_2 = \langle a, c, d \rangle$ ,  $\sigma_3 = \langle a, b, d, c \rangle$ ,  $\sigma_4 = \langle a, b, c, d \rangle$  удовлетворяют ограничению *CoExistence*( $c, d$ ), а трасса  $\sigma_1 = \langle a, b, d \rangle$  – нет.

**Определение 2.13** (*Коэффициент соответствия декларативной модели с ограничениями вида CoExistence и трассы*)

Пусть  $\mathcal{D}$  и  $\sigma$  – декларативная модель с ограничениями вида *CoExistence* и трасса над множеством событий  $\mathcal{A}$ . Назовем  $\mathcal{F}(\sigma, \mathcal{D})$  коэффициентом соответствия  $\mathcal{D}$  и  $\sigma$ :

$$\mathcal{F}(\sigma, \mathcal{D}) = \frac{1}{n} \sum_{(d_1, d_2) \in \mathcal{D}} [I_\sigma[d_1] = I_\sigma[d_2]]$$

**Определение 2.14** (*Коэффициент соответствия декларативной модели с ограничениями вида CoExistence и журнала*)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  – журнал событий и декларативная модель с ограничениями вида *CoExistence* над множеством событий  $\mathcal{A}$  соответственно;  $k \in \mathbb{R}$ . Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^k(\sigma, \mathcal{D})$$

```

CoExistence_Trace_Fitness(trace, Declare, n)
1:  $S \leftarrow 0$ 
2:  $IndicatorArray \leftarrow Indicator\_Array(trace, n)$ 
3: for  $i \leftarrow 1$  to  $Length(Declare)$  do
4:    $a, b \leftarrow Declare[i]$ 
5:   if  $IndicatorArray[a] = IndicatorArray[b]$  then
6:      $S \leftarrow S + 1$ 
7:   end if
8: end for
9:  $c \leftarrow \frac{S}{len(Declare)}$ 
10: return  $c$ 
```

Сложность алгоритма –  $O(|Declare|) = O(n^2)$ .

```

CoExistence_Log_Fitness(Log, Declare, n, k)
1:  $S \leftarrow 0$ 
2: for  $trace \in Log$  do
3:    $S \leftarrow S + CoExistence\_Trace\_Fitness^k(trace, Declare, n)$ 
4: end for
5: return  $\frac{S}{len(Log)}$ 
```

Сложность алгоритма –  $O(|Log| \cdot n^2)$ .

## 2.4. Ограничение NotCoExistence

**Определение 2.15** (*Ограничение NotCoExistence декларативной модели*)

Пусть  $\mathcal{D}$  – декларативная модель над множеством событий  $\mathcal{A}$ ;  $a, b \in \mathcal{A}$ . Тогда если

события  $a$  и  $b$  удовлетворяют ограничению  $\text{NotCoExistence}(a, b)$ , то для любой трассы  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , не противоречащей модели  $\mathcal{D}$ , выполнено следующее:

$$(\exists i \in \mathbb{N} : t_i = a \implies \nexists j \in \mathbb{N} : t_j = b) \wedge (\exists i \in \mathbb{N} : t_i = b \implies \nexists j \in \mathbb{N} : t_j = a)$$

Другими словами, события, удовлетворяющие ограничению  $\text{NotCoExistence}$ , не могут одновременно находиться в трассе.

Так, в рассмотренном ранее примере трассы  $\sigma_1 = \langle a, b, d \rangle, \sigma_2 = \langle a, c, d \rangle$  удовлетворяют ограничению  $\text{NotCoExistence}(b, c)$ , а трассы  $\sigma_3 = \langle a, b, d, c \rangle, \sigma_4 = \langle a, b, c, d \rangle$  – нет.

**Определение 2.16** (Коэффициент соответствия декларативной модели с ограничениями вида  $\text{NotCoExistence}$  и трассы)

Пусть  $\mathcal{D}$  и  $\sigma$  – декларативная модель с ограничениями вида  $\text{NotCoExistence}$  и трасса над множеством событий  $\mathcal{A}$ . Назовем  $\mathcal{F}(\sigma, \mathcal{D})$  коэффициентом соответствия  $\mathcal{D}$  и  $\sigma$ :

$$\mathcal{F}(\sigma, \mathcal{D}) = \frac{1}{n} \sum_{(d_1, d_2) \in \mathcal{D}} [I_\sigma[d_1] + I_\sigma[d_2] \leq 1]$$

**Определение 2.17** (Коэффициент соответствия декларативной модели с ограничениями вида  $\text{NotCoExistence}$  и журнала)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  – журнал событий и декларативная модель с ограничениями вида  $\text{NotCoExistence}$  над множеством событий  $\mathcal{A}$  соответственно;  $k \in \mathbb{R}$ .

Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^k(\sigma, \mathcal{D})$$

```

NotCoExistence_Trace_Fitness(trace, Declare, n)
1:  $S \leftarrow 0$ 
2:  $IndicatorArray \leftarrow Indicator\_Array(trace, n)$ 
3: for  $i \leftarrow 1$  to  $Length(Declare)$  do
4:    $a, b \leftarrow Declare[i]$ 
5:   if  $IndicatorArray[a] + IndicatorArray[b] \leq 1$  then
6:      $S \leftarrow S + 1$ 
7:   end if
8: end for
9:  $c \leftarrow \frac{S}{len(Declare)}$ 
10: return  $c$ 
```

Сложность алгоритма –  $O(|Declare|) = O(n^2)$ .

```

NotCoExistence_Log_Fitness(Log, Declare, n, k)
1:  $S \leftarrow 0$ 
2: for  $trace \in Log$  do
3:    $S \leftarrow S + NotCoExistence\_Trace\_Fitness^k(trace, Declare, n)$ 
4: end for
5: return  $\frac{S}{len(Log)}$ 
```

Сложность алгоритма –  $O(|Log| \cdot n^2)$ .

## 2.5. Ограничение Choice

**Определение 2.18** (*Ограничение Choice декларативной модели*)

Пусть  $\mathcal{D}$  - декларативная модель над множеством событий  $\mathcal{A}$ ;  $a, b \in \mathcal{A}$ . Тогда если события  $a$  и  $b$  удовлетворяют ограничению  $Choice(a, b)$ , то для любой трассы  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , не противоречащей модели  $\mathcal{D}$ , выполнено следующее:

$$\exists i \in \mathbb{N} : t_i = a \vee \exists j \in \mathbb{N} : t_j = b$$

Таким образом, хотя бы одно из событий, удовлетворяющих ограничению  $Choice$ , встречается в трассе.

Так, в рассмотренном ранее примере все трассы удовлетворяют ограничению  $Choice(b, c)$ .

**Определение 2.19** (*Коэффициент соответствия декларативной модели с ограничениями вида Choice и трассы*)

Пусть  $\mathcal{D}$  и  $\sigma$  - декларативная модель с ограничениями вида  $Choice$  и трасса над множеством событий  $\mathcal{A}$ . Назовем  $\mathcal{F}(\sigma, \mathcal{D})$  коэффициентом соответствия  $\mathcal{D}$  и  $\sigma$ :

$$\mathcal{F}(\sigma, \mathcal{D}) = \frac{1}{n} \sum_{(d_1, d_2) \in \mathcal{D}} [I_\sigma[d_1] + I_\sigma[d_2] \geq 1]$$

**Определение 2.20** (*Коэффициент соответствия декларативной модели с ограничениями вида Choice и журнала*)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  - журнал событий и декларативная модель с ограничениями вида  $Choice$  над множеством событий  $\mathcal{A}$  соответственно;  $k \in \mathbb{R}$ . Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^k(\sigma, \mathcal{D})$$

```

Choice_Trace_Fitness(trace, Declare, n)
1:  $S \leftarrow 0$ 
2:  $IndicatorArray \leftarrow Indicator\_Array(trace, n)$ 
3: for  $i \leftarrow 1$  to  $Length(Declare)$  do
4:    $a, b \leftarrow Declare[i]$ 
5:   if  $IndicatorArray[a] + IndicatorArray[b] \geq 1$  then
6:      $S \leftarrow S + 1$ 
7:   end if
8: end for
9:  $c \leftarrow \frac{S}{len(Declare)}$ 
10: return  $c$ 
```

Сложность алгоритма –  $O(|Declare|) = O(n^2)$ .

```

Choice_Log_Fitness(Log, Declare, n, k)
1:  $S \leftarrow 0$ 
2: for  $trace \in Log$  do
3:    $S \leftarrow S + Choice\_Trace\_Fitness^k(trace, Declare, n)$ 
4: end for
5: return  $\frac{S}{len(Log)}$ 
```

Сложность алгоритма –  $O(|Log| \cdot n^2)$ .

## 2.6. Ограничение ExclusiveChoice

**Определение 2.21** (*Ограничение ExclusiveChoice декларативной модели*)

Пусть  $\mathcal{D}$  - декларативная модель над множеством событий  $\mathcal{A}$ ;  $a, b \in \mathcal{A}$ . Тогда если события  $a$  и  $b$  удовлетворяют ограничению  $\text{ExclusiveChoice}(a, b)$ , то для любой трассы  $\sigma = \langle t_1, t_2, \dots, t_n \rangle$ , не противоречащей модели  $\mathcal{D}$ , выполнено следующее:

$$(\exists i \in \mathbb{N} : t_i = a \wedge \nexists j \in \mathbb{N} : t_j = b) \vee (\exists i \in \mathbb{N} : t_i = b \wedge \nexists j \in \mathbb{N} : t_j = a)$$

Таким образом, ровно одно из событий, удовлетворяющих ограничению  $\text{ExclusiveChoice}$ , встречается в трассе.

Так, в рассмотренном ранее примере только трасса  $\sigma_1 = \langle a, b, d \rangle$  удовлетворяет ограничению  $\text{ExclusiveChoice}(b, c)$ .

**Определение 2.22** (*Коэффициент соответствия декларативной модели с ограничениями вида ExclusiveChoice и трассы*)

Пусть  $\mathcal{D}$  и  $\sigma$  - декларативная модель с ограничениями вида  $\text{ExclusiveChoice}$  и трасса над множеством событий  $\mathcal{A}$ . Назовем  $\mathcal{F}(\sigma, \mathcal{D})$  коэффициентом соответствия  $\mathcal{D}$  и  $\sigma$ :

$$\mathcal{F}(\sigma, \mathcal{D}) = \frac{1}{n} \sum_{(d_1, d_2) \in \mathcal{D}} [I_\sigma[d_1] + I_\sigma[d_2] = 1]$$

**Определение 2.23** (*Коэффициент соответствия декларативной модели с ограничениями вида ExclusiveChoice и журнала*)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  - журнал событий и декларативная модель с ограничениями вида  $\text{ExclusiveChoice}$  над множеством событий  $\mathcal{A}$  соответственно;  $k \in \mathbb{R}$ .

Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^k(\sigma, \mathcal{D})$$

ExclusiveChoice\_Trace\_Fitness(trace, Declare, n)

```

1:  $S \leftarrow 0$ 
2:  $IndicatorArray \leftarrow Indicator\_Array(trace, n)$ 
3: for  $i \leftarrow 1$  to  $Length(Declare)$  do
4:    $a, b \leftarrow Declare[i]$ 
5:   if  $IndicatorArray[a] + IndicatorArray[b] \equiv 0 \pmod{2}$  then
6:      $S \leftarrow S + 1$ 
7:   end if
8: end for
9:  $c \leftarrow \frac{S}{len(Declare)}$ 
10: return  $c$ 
```

Сложность алгоритма –  $O(|Declare|) = O(n^2)$ .

ExclusiveChoice\_Log\_Fitness(Log, Declare, n, k)

```

1:  $S \leftarrow 0$ 
2: for  $trace \in Log$  do
3:    $S \leftarrow S + ExclusiveChoice\_Trace\_Fitness^k(trace, Declare, n)$ 
4: end for
5: return  $\frac{S}{len(Log)}$ 
```

Сложность алгоритма –  $O(|Log| \cdot n^2)$ .

## 2.7. Коэффициент соответствия декларативной модели и журнала

Декларативные модели сочетают в себе различные ограничения, зачастую не ограничиваясь каким-либо одним. Для оценки соответствия журнала модели в целом можно декомпозировать модель: можно рассмотреть каждое ограничение по отдельности, вычислить коэффициент соответствия модели этому ограничению, а затем на основе этого оценить соответствие журнала модели в целом.

**Определение 2.24** (*Коэффициент соответствия декларативной модели и журнала*)

Пусть  $\mathcal{L}$  и  $\mathcal{D}$  - журнал событий и декларативная модель. Пусть  $\mathcal{D} = \bigcup_{i=1}^n \mathcal{D}_i$ , где  $\mathcal{D}_i$  содержит все ограничения какого-то одного вида из  $\mathcal{D}$  и только их.

Пусть  $k = (k_1, \dots, k_n) \in \mathbb{R}^n$ . Назовем  $\mathcal{F}(\mathcal{L}, \mathcal{D}, k)$  коэффициентом соответствия  $\mathcal{L}$  и  $\mathcal{D}$ :

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^n |\mathcal{D}_i| \cdot \mathcal{F}(\mathcal{L}, \mathcal{D}_i, k_i),$$

где  $|\mathcal{D}_i|$  - количество ограничений вида  $i$ ,  $\mathcal{F}(\mathcal{L}, \mathcal{D}_i, k_i)$  - коэффициент соответствия  $\mathcal{L}$  и  $\mathcal{D}_i$ .

Как и все предыдущие коэффициенты, коэффициент соответствия декларативной модели и журнала принимает значения от 0 до 1, причем близкие к 0 значения отображают плохое соответствие, а близкие к 1 – хорошее. При такой оценке задействуются все встречающиеся в модели ограничения, причём различным видам ограничений приписываются веса, пропорциональные доле этих видов ограничений среди всех ограничений.

Рассмотрим пример. Пусть  $\mathcal{A} = \{a, b, c, d\}$ ,  $\mathcal{L} = \{\sigma_1 = \langle a, b, d \rangle^{100}, \sigma_2 = \langle a, c, d \rangle^{100}, \sigma_3 = \langle a, b, c \rangle^{25}, \sigma_4 = \langle a, d \rangle^{25}\}$ ,  $\mathcal{D} = \{\text{Response}\{b < d, c < d\}, \text{ExclusiveChoice}\{(b, c)\}\}$ . Положим  $k_1 = k_2 = 3$ . Найдем коэффициент соответствия модели и журнала.

Здесь  $\mathcal{D}_1 = \text{Response}\{b < d, c < d\}$  - все ограничения вида Response, и  $\mathcal{D}_2 = \text{ExclusiveChoice}\{(b, c)\}$  - все ограничения вида ExclusiveChoice.

Сначала вычислим коэффициент соответствия декларативной модели с ограничениями вида Response и журнала.

	a	b	c	d
a	0	0	0	0
b	0	0	0	1
c	0	0	0	1
d	0	0	0	0

Для  $\sigma_1 = \langle a, b, d \rangle$ :

	a	b	c	d
a	0	1	0	1
b	0	0	0	1
c	0	0	0	0
d	0	0	0	0

a	b	c	d
1	1	0	1
1	1	0	1

	a	b	c	d
a	1	1	1	1
b	1	1	1	1
c	1	1	1	1
d	1	1	1	1

$$\mathcal{F}(\sigma_1, D_1) = 1 - \frac{1}{m} \sum_{1 \leq i, j \leq n} (1 - \delta[i, j]) = 1 - \frac{1}{2} \cdot 0 = 1$$

Для  $\sigma_2 = \langle a, c, d \rangle$  :

	a	b	c	d
a	0	0	1	1
b	0	0	0	0
c	0	0	0	1
d	0	0	0	0

a	b	c	d
1	0	1	1

	a	b	c	d
a	1	1	1	1
b	1	1	1	1
c	1	1	1	1
d	1	1	1	1

$$\mathcal{F}(\sigma_2, \mathcal{D}_1) = 1 - \frac{1}{2} \cdot 0 = 1$$

Для  $\sigma_3 = \langle a, b, c \rangle$  :

	a	b	c	d
a	0	1	1	0
b	0	0	1	0
c	0	0	0	0
d	0	0	0	0

a	b	c	d
1	1	1	0

	a	b	c	d
a	1	1	1	1
b	1	1	1	0
c	1	1	1	0
d	1	1	1	1

$$\mathcal{F}(\sigma_3, \mathcal{D}_1) = 1 - \frac{1}{2} \cdot 2 = 0$$

Для  $\sigma_4 = \langle a, d \rangle$  :

	a	b	c	d
a	0	0	0	1
b	0	0	0	0
c	0	0	0	0
d	0	0	0	0

a	b	c	d
1	0	0	1

	a	b	c	d
a	1	1	1	1
b	1	1	1	1
c	1	1	1	1
d	1	1	1	1

$$\mathcal{F}(\sigma_4, \mathcal{D}_1) = 1 - \frac{1}{2} \cdot 0 = 1$$

Таким образом, коэффициент соответствия:

$$\mathcal{F}(\mathcal{L}, \mathcal{D}_1, k_1) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}(\sigma, \mathcal{D}_1, k_1) = \frac{1}{250} \cdot (100 \cdot 1^3 + 100 \cdot 1^3 + 25 \cdot 0^3 + 25 \cdot 1^3) = \frac{9}{10}$$

Теперь вычислим коэффициент соответствия декларативной модели с ограничениями вида ExclusiveChoice и журнала.

Для  $\sigma_1 = \langle a, b, d \rangle$  :

a	b	c	d
1	1	0	1

$$\mathcal{F}(\sigma_1, \mathcal{D}_2) = \frac{1}{n} \sum_{(d_1, d_2) \in \mathcal{D}} [I_\sigma[d_1] + I_\sigma[d_2] = 1] = 1$$

Для  $\sigma_2 = \langle a, c, d \rangle$  :

a	b	c	d
1	0	1	1

$$\mathcal{F}(\sigma_2, \mathcal{D}_2) = 1$$

Для  $\sigma_3 = \langle a, b, c, d \rangle$  :

a	b	c	d
1	1	1	1

$$\mathcal{F}(\sigma_3, \mathcal{D}_2) = 0$$

Для  $\sigma_4 = \langle a, d \rangle$ :

$$I_{\sigma_4} = \begin{array}{|c|c|c|c|} \hline & a & b & c & d \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array}$$

$$\mathcal{F}(\sigma_4, \mathcal{D}_2) = 0$$

Таким образом, коэффициент соответствия декларативной модели с ограничениями вида ExclusiveChoice и журнала:

$$\mathcal{F}(\mathcal{L}, \mathcal{D}_2, k_2) = \frac{1}{|\mathcal{L}|} \sum_{\sigma \in \mathcal{L}} \mathcal{F}^{k_2}(\sigma, \mathcal{D}_2) = \frac{1}{250} (100 \cdot 1^3 + 100 \cdot 1^3 + 25 \cdot 0^3 + 25 \cdot 0^3) = \frac{4}{5}$$

Теперь можно вычислить коэффициент соответствия декларативной модели и журнала:

$$\mathcal{F}(\mathcal{L}, \mathcal{D}, k) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^n |\mathcal{D}_i| \cdot \mathcal{F}(\mathcal{L}, \mathcal{D}_i, k_i) = \frac{1}{3} (2 \cdot \frac{9}{10} + 1 \cdot \frac{4}{5}) = \frac{13}{15} \approx 0.866$$

Заметим, что сложность алгоритма, вычисляющего коэффициент соответствия всей модели таким образом, будет зависеть от сложностей алгоритмов, вычисляющих различные коэффициент соответствия журнала и различных видов исключений. В частности, в приведенной реализации сложность алгоритма будет составлять  $O(|\mathcal{D}| \cdot n^2)$ . Таким образом, сложность зависит от размера журнала и от мощности множества событий.

### **3. Программная реализация и проведение экспериментов**

В настоящем разделе представлены детали реализации алгоритмов проверки соответствия декларативной модели журналу событий, описанные в предыдущем разделе, а также результаты экспериментов, проведённых над ними.

#### **3.1. Детали реализации**

Описанные алгоритмы были реализованы в виде библиотеки DCC на языке Python (<https://github.com/alekseiannenkov/DeclareConformanceChecking>).

Алгоритмы работают со следующими представлениями данных:

1. События могут являться целыми числами или строками (`Union[int, str]`);
2. Трасса - список событий (`List[Union[int, str]]`);
3. Журнал событий - список трасс (`List[List[Union[int, str]]]`);
4. Каждое отдельное ограничение конкретного вида является кортежем из двух событий (`Tuple[Union[int, str], Union[int, str]]`);
5. Каждый шаблон ограничений является списком всех ограничений одного вида (`List[Tuple[Union[int, str], Union[int, str]]]`);
6. Декларативная модель представлена в виде отдельного класса (`DeclareModel`). Для её инициализации необходимо задать все ограничения, а также передать все события, которые могут встретиться в журнале.

#### **3.2. Проведение экспериментов**

Для проведения экспериментов были сгенерированы множества событий мощности 10, 100, 1000. На их основе генерировались непротиворечивые декларативные модели (то есть такие модели, которым удовлетворяет непустое множество трасс). Так как задача проверки непротиворечивости модели еще не решена, количество ограничений в моделях было небольшим, чтобы можно было вручную проверить их непротиворечивость.

Для каждой такой модели генерировались журналы событий одного из следующих 3 типов:

1. Журнал, в котором все трассы не нарушают ни одно из ограничений (назовём его хорошим);
2. Журнал, в котором все трассы нарушают большую часть ограничений (назовём его плохим);
3. Журнал, в котором все трассы генерировались случайно (назовём его случайным).

Трассы генерировались таким образом, чтобы они не содержали циклов (то есть каждое из событий встречалось в трассе не более одного раза).

Для каждого фиксированного множества событий, декларативной модели и типа журнала генерировался 1 журнал; хорошие и плохие журналы генерировались вручную. Испытания проводились для степеней штрафа 3, 5 и 7.

Таким образом, всего было проведено 27 испытаний. Для каждого из них были подсчитаны коэффициенты соответствия модели и журнала (`Declare_Log_Fitness`). В таблицах ниже приведены результаты проведённых экспериментов.

Таблица 1: Результаты для мощности множества 10

Тип журнала	Степень штрафа ( $k$ )		
	3	5	7
Хороший	0.9014	0.8781	0.8659
Плохой	0.2665	0.2154	0.1975
Случайный	0.5161	0.4727	0.4606

Таблица 2: Результаты для мощности множества 100

Тип журнала	Степень штрафа ( $k$ )		
	3	5	7
Хороший	0.8531	0.7988	0.7535
Плохой	0.2092	0.1382	0.1022
Случайный	0.5396	0.5138	0.5054

Таблица 3: Результаты для мощности множества 500

Тип журнала	Степень штрафа ( $k$ )		
	3	5	7
Хороший	0.8546	0.7995	0.7512
Плохой	0.3729	0.2358	0.1578
Случайный	0.4868	0.3393	0.2511

Из результатов экспериментов видно, что журналы с трассами, удовлетворяющими большинству ограничений, имеют близкие к 1 значения коэффициентов. Журналы, в которых трассы нарушают большинство ограничений, в свою очередь, имеют близкие к 0 значения коэффициентов. А значения коэффициентов случайно генерированных журналов ожидаемо заключены между коэффициентами хороших и плохих журналов.

Более того, так как ни один из коэффициентов не принимает значение 0 или 1, при увеличении  $k$  все коэффициенты монотонно убывают.

Таким образом, алгоритмы работают в соответствии с описанной логикой и показывают хорошую применимость на генерированных моделях и журналах.

#### 4. Обзор литературы

Подробный обзор области process mining представлен в книге [1]. В книге описана мотивировка process mining, а также перечислены наиболее важные проблемы, решаемые рассматриваемой дисциплиной.

Цель дисциплины process mining – автоматическое обнаружение моделей при помощи данных из журнала событий. Однако можно выделить два основных недостатка, свойственных современным алгоритмам process mining: созданные модели получаются слишком большими, «похожими на спагетти», или полученные модели плохо описывают журнал событий или слишком обобщают его. В статье [2] представлен систематический обзор и сравнительная оценка автоматизированных методов обнаружения процессов. Результат работы большинства разобранных алгоритмов – сеть Петри, однако разобраны также алгоритмы, производящие декларативные ограничения. В результате исследования были выявлены алгоритмы, находящие модели с наилучшими показателями по метрикам fitness, precision и simplicity. Также была обоснована необходимость фильтрации журнала событий для более точной работы алгоритмов. Более того, было выявлено отсутствие универсальных метрик fitness и precision как для императивных, так и для декларативных моделей.

Обнаруженная модель не должна ограничиваться одним измерением, а должна быть максимально точной по всем возможным метрикам. Так, в [3] описан полиномиальный алгоритм, гарантирующий нахождение множества fit и sound моделей.

В conformance checking существующие модели сравниваются с полученными из журнала событий данными. В [4] описаны проблемы, поднимаемые процессами проверки соответствий, вводятся основные метрики и необходимые формальные элементы. В книге также делается упор на измерения precision и fitness – приведена логика построения их самых распространённых численных метрик. В [5] вводятся методы измерения всех метрик – в частности, описывается подход выравниваний для вычисления fitness.

Декларативный подход нацелен на создание границ поведения модели, позволяя процессу протекать так, как не запрещено ограничениями. Такой подход может быть противопоставлен императивному подходу, в котором модель описывает все допустимые последовательности. В [6] описывается необходимость использования декларативных языков с целью создания гибких и контролируемых моделей. В этой статье также вводится синтаксис и семантика LTL, на которых строится семантика декларативного шаблона. Идея предоставления широкого спектра механизмов, позволяющих гибкое поведение модели, отражена в создании системы управления рабочими процессами Declare [7]. Другим способом применения декларативного подхода является внедрение общих знаний о процессе в модель. В [8] используются декларативные шаблоны для внедрения общих знаний в уже существующую модель. Приведены алгоритмы проверки соответствия модели шаблонам, а также описаны алгоритмы модификации модели.

В [9] приведена структура общего алгоритма проверки соответствия декларативной модели и трассы.

В [10] развита идея использования техники выравниваний для подсчета метрики fitness, описанная в [4]. Для нахождения оптимального выравнивания трассы используется алгоритм A\*, подробно описанный в [11].

## ЗАКЛЮЧЕНИЕ

В ходе работы над проектом были изучены основные ограничения декларативных моделей и существующие алгоритмы проверки соответствия декларативной модели и журнала. На основе этого были описаны, реализованы и протестированы методы и алгоритмы проверки соответствия декларативной модели и ациклического журнала.

Полученные в ходе проведения экспериментов результаты показывают хорошую применимость алгоритмов к синтезированным моделям и журналам.

Проведение дальнейших исследований, направленных на изучение задачи проверки соответствия декларативной модели и журнала событий, целесообразно в следующих направлениях:

1. Добавление новых шаблонов ограничений, которые подразумевают наличие цикла в трассе; добавление унарных шаблонов ограничений.
2. Проверка непротиворечивости декларативной модели: по заданным ограничениям требуется проверить, существует ли непустая трасса, удовлетворяющая ограничениям.
3. Адаптирование библиотеки под настоящие модели и журналы событий и проведение испытаний алгоритмов на них.

## 5. Список литературы

- [1] Wil van der Aalst. Process Mining: Data Science in Action. — 2-е изд. — Berlin: Springer, 2016. — 467 с.
- [2] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, Allar Soo. Automated Discovery of Process Models from Event Logs: Review and Benchmark // ACM Transactions on Software Engineering and Methodology. — 2019. — Т. 28, № 3. — С. 1–41.
- [3] Sander J.J. Leemans, Dirk Fahland, Wil M.P. van der Aalst. Discovering Block-Structured Process Models // BPM 2013: Business Process Management Workshops. — Berlin: Springer, 2013. — С. 311–322.
- [4] Wil M.P. van der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski. Conformance Checking and Diagnosis in Process Mining // Business Process Management. — 2009. — Т. 5701. — С. 455–466.
- [5] Wil van der Aalst, Arya Adriansyah, Boudeijn van Dongen. Replaying History on Process Models for Conformance Checking and Performance Analysis // WIREs Data Mining and Knowledge Discovery. — 2012. — Т. 2, № 2. — С. 182–192.
- [6] Claudio Di Ciccio, Marco Montali. Declarative Process Specifications: Reasoning, Discovery, Monitoring // ACM Transactions on Management Information Systems. — 2015. — Т. 5, № 4. — С. 1–30.
- [7] W. M. P. van der Aalst, M. Pesic, H. Schonenberg. Declarative Workflows: Balancing between Flexibility and Support // Computer Science - Research and Development. — 2007. — Т. 23, № 2. — С. 99–113.
- [8] P.M. Dixit, J.C.A.M. Buijs, Wil M.P. van der Aalst, B.F.A. Hompes, J. Buurman. Using Domain Knowledge to Enhance Process Discovery // International Journal of Data Science and Analytics. — 2018. — Т. 5, № 4. — С. 273–290.
- [9] A. Burattin, F. M. Maggi, A. Sperduti. Conformance Checking Based on Multi-Perspective Declarative Process Models // Information Systems. — 2018. — Т. 74. — С. 21–37.
- [10] Massimiliano de Leoni, Fabrizio Maria Maggi, Wil M.P. van der Aalst. Aligning Event Logs and Declarative Process Models for Conformance Checking // BPM 2012. — Berlin: Springer, 2012. — С. 82–97.
- [11] Rina Dechter, Judea Pearl. Generalized Best-First Search Strategies and the Optimality of A\* // Journal of the ACM. — 1985. — Т. 32, № 3. — С. 505–536.
- [12] Wolfgang Reisig. Petri Nets: Modeling Techniques, Analysis Methods, Case Studies. — Berlin: Springer, 2013. — 300 с.

## КАЛЕНДАРНЫЙ ПЛАН РАБОТ

Таблица 1 – календарный план выполнения работ

Этап проекта	Описание работ	Ожидаемые результаты	Сроки выполнения
1	Анализ предметной области. Уточнение цели исследования, постановка задачи	Формулировка целей, задач	15.11.2024
2	Анализ литературы, выдвижение гипотезы	Формулировка гипотезы	20.01.2025
3	Разработка алгоритма	Создание математической модели или алгоритма	20.02.2025
4	Тестирование алгоритма, доказательство его корректности	Тестирование и улучшение алгоритма	08.03.2025
5	Подготовка отчета	Оформление результатов и выводов	10.04.2025