УДК 519.179.2 + 519.713.8 + 512.579
ББК 22.12. + 22.19

О СОГЛАСОВАНИИ ПОВЕДЕНИЯ ВЗАИМОДЕЙСТВУЮЩИХ ОБЪЕКТОВ

Бабичев А. В.[1]
(*ФГБУН Институт проблем управления им. В.А. Трапезникова РАН, Москва*)

Статья посвящается задаче согласования поведения множества взаимодействующих объектов. Рассматривается процедура согласования, которая на каждом шаге настраивает один объект и переводит его в состояние, согласованное с состоянием его соседей. При этом может оказаться, что некоторые ранее настроенные объекты перестанут быть согласованными. Исследуются условия, при которых процесс стабилизируется и всё множество взаимодействующих объектов может быть согласовано. Описываются свойства, которыми должны обладать объекты для того, чтобы любое их соединение в сеть было настраиваемым, а также приводится алгоритм настройки для любой такой сети.

Ключевые слова: функциональная сеть, решение системы уравнений, универсальная алгебра, метод итераций.

1. Введение

Постановка. На интуитивном уровне задачу, решаемую в настоящей работе, можно поставить следующим образом.

Имеется система взаимодействующих объектов. Эта система функционирует правильно только в том случае, когда состояние каждого объекта (параметры поведения объекта) «согласовано» с состояниями соседних с ним объектов.

[1] *Андрей Владимирович Бабичев, кандидат технических наук, (babichev@ipu.ru).*

То, что состояние объекта «согласовано» с состояниями соседей, пока никак не определяется. Предполагается только, что имеется процедура, которая настраивает произвольный объект. Эта процедура переводит заданный объект в состояние, согласованное с состояниями соседей.

Требуется настроить всю систему, при этом шагом настройки является настройка одного объекта (перевод его в состояние согласованное с соседями). Понятно, что при настройке одного элемента мы можем потерять настройки других, поэтому в общем случае нет гарантии, что процесс настройки всей системы может быть успешно завершен. Таким образом, при рассмотрении процесса настройки сразу возникает две задачи: первая – описать класс настраиваемых сетей (сетей, которые могут быть настроены), вторая – описать процедуру настройки (последовательность, в которой следует настраивать элементы сети).

Источники постановки. Исходная постановка возникла в связи с проектом Максима Хомякова «Хаос» (начало 80-х годов [19]). Проект «Хаос» был посвящен созданию базы данных, и в этом проекте был заложен аппарат, хотя и более продвинутый концептуально, но сходный с аппаратом триггеров в современных СУБД. Основной принцип использования триггеров можно описать следующим образом. Каждый триггер можно рассматривать как программу корректировки данных для того, чтобы обеспечить некоторые ограничения целостности. При внесении в базу каких-либо изменений могут нарушиться ограничения целостности – вызываются соответствующие триггеры, корректирующие данные. Применение триггеров может изменить другие ограничения целостности – вызовутся другие триггеры и так далее пока не будет достигнуто согласованное состояние — состояние, в котором все ограничения целостности выполнены. В подобной интерпретации (в терминах триггеров) рассматриваемая задача согласования выглядит следующим образом: как надо организовывать последовательность вызовов триггеров и какие ограничения на вид триггеров надо наложить, чтобы не зациклиться на стадии исполнения транзакции?

Работа была завершена при исследовании задач согласования управленческих решений (смотри работы [14, 15]), проблематику которых демонстрируют приводимый разделе 2 пример.

Используемый аппарат и используемые результаты.

В качестве формальной модели для систем взаимодействующих объектов используются диагональные системы уравнений (системы уравнений вида $(*)$): каждому объекту взаимно однозначно соответствует переменная, а значение каждой переменной равно текущему состоянию соответствующего объекта. Соответственно, настроенные состояния всей системы взаимодействующих объектов – это решения системы уравнений.

$$(*)\begin{cases} x_1 = t_1(x_1, \dots, x_n) \\ \dots \\ x_n = t_n(x_1 \dots, x_n) \end{cases}$$

Процесс настройки это процесс решения системы уравнений модифицированным методом итераций:

если в классическом методе итераций значения всех переменных одновременно заменяются значениями правых частей соответствующих им уравнений, то в модифицированном методе на каждом шаге вычисляется ровно одна переменная.

Используемый в работе модифицированный метод итераций берёт начало от методики упорядочивания переменных, используемой Бухбергером [2, 4, 8, 9] (1965 г.) при делении многочленов от многих переменных, эту же методику позже использовал Блэнд [16].

В работах [5, 6, 7, 17, 18] исследуется формализация процесса согласования, с одной стороны, более подробная, а с другой, более общая, нежели рассматриваемая в настоящей статье. Более того, тематика этих работ пересекается не только с тематикой статьи, но и с тематикой проекта «Хаос».

В связи с задачей обобщения публикуемых результатов, следует отметить, что классические теоремы о неподвижной точке к рассматриваемым задачам неприменимы (это проявляется в том, что применение классического метода итераций приводит к зацикливанию). Исключением является теорема Брауэра о непрерывном отображении шара в себя. Более того, в работах А.В. Малишевского («натуральные системы») [13], которые по тематике пересекаются как с темой настоящей работы, так и с вопросами обобщения теоремы Брауэра, продемонстрировано, что эта теорема при обосновании методов согласования играет ключевую роль.

Принцип изложения. По возможности все формальные определения и утверждения вынесены в приложение, в основном тексте оставлена лишь та часть формального аппарата, без которой нельзя сформулировать главный результат (теорема 1). Существует взаимно однозначное соответствие между диагональными системами уравнений и функциональными сетями (размеченными ориентированными графами с некоторыми дополнительными ограничениями). Поэтому в основном тексте упор делается на более наглядное представление с помощью функциональных сетей.

План изложения. В разделе 2 приводится игрушечный пример взаимодействующих объектов, на котором будут иллюстрироваться все вводимые в основном тексте определения.

В разделе 3 в общих терминах описывается стратегия настройки.

Соответствующие определения и формальное описание алгоритм настройки в терминах функциональных сетей приведены в разделе 4.

В разделе 5 формулируются условия, характеризующие область применимости предлагаемого метода.

В приложение вынесены: описание аппарата, используемого при обосновании метода настройки (приложение 1), и само обоснование (приложения 2, 3).

2. Задача о вечеринке

Чтобы описать проблематику на интуитивном уровне, рассмотрим следующий пример.

Пример 1. (Задача о вечеринке). Представим ситуацию, что организатор вечеринки собирается пригласить в гости троих знакомых x, y и z (участники вечеринки) в один из ближайших выходных дней (суббота либо воскресенье): и тот и другой день организатора полностью устраивает. Каждый из знакомых организатора (каждый участник вечеринки) наверняка может придти к нему либо в субботу, либо в воскресенье. Если все не могут собраться (например, x занят в субботу, а y в воскресенье), то встреча может состояться в сокращенном составе (в тот день, когда может z) , но лучше все же собраться вчетвером.

У каждого из участников есть наиболее удобный день, когда он может придти в гости, но он, может быть, согласится придти в другой день. Например, если y во что бы то ни стало хочет встретиться с x (только ради этого он и придет на встречу), то y согласится на любой день, который удобен для x. С другой стороны, если x не хочет встречаться с y (но желает встретиться с организатором), то x будет действовать по следующему правилу: x предложит собраться в тот день, который не выбрал y.

При этом организатору не известны мотивы, руководствуясь которыми участники выбирают тот или иной день встречи.

Организатор по очереди договаривается с каждым участником встречи. При каждом разговоре он, во-первых, сообщает собеседнику все известные ему пожелания, а во-вторых, выслушивает пожелание собеседника.

Поскольку мнение каждого участника вечеринки может меняться в зависимости от мнений остальных участников, то, возможно, придется по нескольку раз разговаривать с каждым. Эти переговоры надо провести таким образом, чтобы каждый, кто придет на встречу, остался бы доволен: если y хочет встретиться с x, а x не придет на встречу, то y будет считать вечер испорченным;

Каждый участник, не попавший на вечеринку, должен быть извещен о том, какой день встречи назначен (вдруг он передумает).

Необходимо сделать конечное число звонков и либо назначить день встречи, либо отменить встречу. Каким образом следует проводить переговоры (в каком порядке обзванивать участников)?●

Чтобы продемонстрировать нетривиальность рассмотренной постановки, опишем, какие проблемы могут возникнуть при проведении переговоров.

Пример 2. На рис. 1 представлены иллюстрации (пока что это неформализованные картинки) возможных взаимоотношений участников вечеринки (следует помнить, что организатор ничего об этих взаимоотношениях не знает):

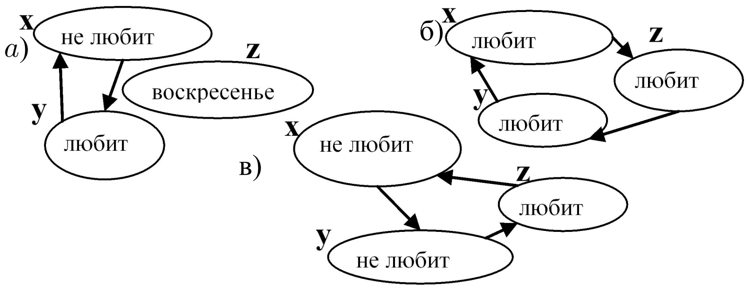


Рис. 1. Варианты взаимоотношений участников

a) мнение z не зависит от мнений x и y и это мнение не меняется в процессе переговоров (он/она по субботам никуда не ездит). Участник x ни в коем случае не хочет быть в одной компании с y (ей удобнее всего день, не удобный для y), y придет в любой день, в который придет x , а без x не придет;

б) участник x придет на встречу только вместе с z, z только с y, а y только вместе с x;

в) участник y не хочет встречаться с z, x не придет, если будет y, а z придет, только если будет x.

Предположим, что

- при проведении переговоров организатор многократно обзванивает участников по циклу z, y, x, начиная с z (как на рис. 2.и)), надеясь, что предпочтения участников рано или поздно перестанут меняться;

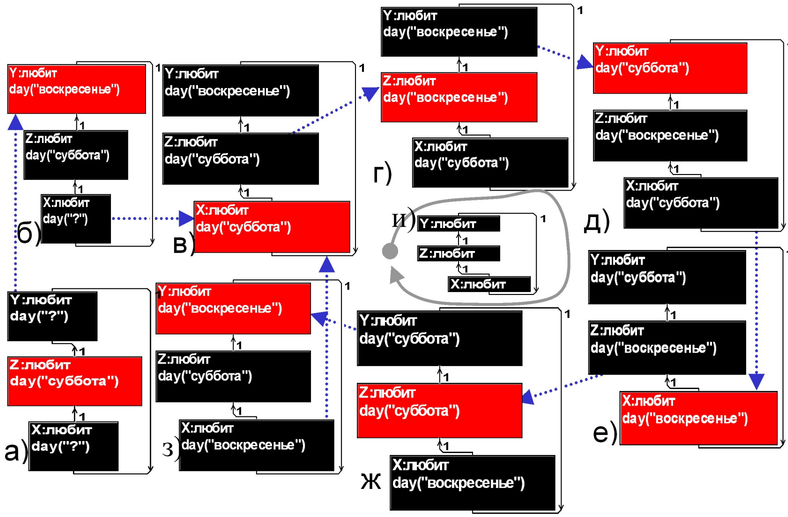


Рис. 2. Пример неправильной стратегии настройки

- до начала переговоров участнику z удобна суббота, а остальным – воскресенье (рис. 2.з));
- взаимоотношения участников (которые не известны организатору) соответствуют диаграмме, представленной на рис. 1.б): все друг друга любят и поэтому, если правильно провести переговоры, все соберутся и будут довольны

Но, как видно из рис. 2, подобная стратегия переговоров обречена на неудачу: при подобной стратегии переговоров процесс настройки не закончится никогда.●

3. Стратегия Киссинджера

Далее описывается техника настройки множества взаимодействующих объектов.

Эту технику, по-видимому, первым строго описал Б. Бухбергер (1965) при описании алгоритма деления многочленов со многими переменными [2, 4, 8, 9]. После Б. Бухбергера этот же алгоритм использовал Р. Блэнд (1977) для поиска базисных решений в задачах линейного программирования [16]. В дальнейшем эта техника будет проходить под названием стратегия Киссинджера (Г. Киссинджер использовал ее в 1973 г. при проведении трехсторонних переговоров после войны Судного дня).

На содержательном уровне стратегию Бухбергера – Киссинджера – Блэнда можно рассматривать как следующую стратегию согласования. Пусть все объекты перенумерованы. Тогда на каждом шаге должен настраиваться тот ненастроенный объект (ровно один), который имеет наименьший номер.

Определение 1. (*Стратегия Киссинджера*). *Пусть* $<<$ – *некоторый линейный порядок на конечном множестве взаимодействующих объектов* V. *Главным объектом из множества* V *будем называть минимальный по порядку* $<<$ *объект, который не является настроенным* (*главный объект существует тогда и только тогда, когда существуют ненастроенные объекты*). *Настройкой в соответствии со* $<<$ – *стратегией Киссинджера называется последовательность настроек главных объектов.*

Таким образом, при настройке по стратегии Киссинджера на каждом шаге настраивается тот ненастроенный объект, который имеет наименьший номер при нумерации по порядку $<<$.

Пример 3. Рассмотрим, как стратегия Киссинджера выглядит в применении к задаче о вечеринке.

Прежде всего следует отметить, что в задаче о вечеринке есть процедура, настраивающая произвольный объект – это звонок организатора участнику вечеринки. В то же время нет другого способа определить, в настроенном состоянии находится объект или нет, кроме как настроив его: лишь после настройки (после разговора) организатор вечеринки точно знает, в каком состоянии (в настроенном или нет) находился объект до настройки.

Таким образом, идентифицировать главный объект (и заодно настроить его) можно только обзвонив всех участников по порядку $<<$.

Итак, в применении к задаче о вечеринке стратегия Киссинджера выглядит следующим образом:

Назовем звонок информационным, если он каким-нибудь образом меняет представление организатора о предпочтениях участников. Например, самый первый звонок к кому-нибудь всегда является информационным (до звонка организатор ничего не знает о желаниях собеседника), повторный звонок является информационным только в том случае, если пожелание собеседника меняется по сравнению с пожеланиями на предыдущем звонке. Другими словами повторный звонок к объекту o является информационным, если до звонка объект o находился в ненастроенном состоянии.

Алгоритм настройки выглядит следующим образом.

1) Зафиксировать некоторый порядок $<<$ на элементах множества $\{x, y, z\}$ например $z{<<}y{<<}x$ (как в примере 2 и на рис. 2.и)).

2) Обзванивать участников в соответствии с порядком $<<$, при этом если очередной звонок является информационным, то прервать цикл и начать процесс переговоров заново (начиная с z).

3) Процесс настройки заканчивается, когда

a) либо все последние разговоры со всеми участниками оказались неинформационными, – в этом случае можно назначить день встречи (день который выбрало большинство участников),

b) либо после очередного звонка получены ровно те пожелания, которые были на предыдущих шагах (конфликт интересов, который проявляется как зацикливание) – следует отменить встречу, в противном случае среди участников будут обиженные на организатора.

На рис. 3 приведены примеры использования описанного алгоритма при упорядочивании $z{<<}y{<<}x$ и при различных взаимоотношениях между участниками вечеринки, перечисленных на рис. 1.

Следует подчеркнуть, что рассмотренная постановка – это лишь примитивный пример: каждый объект в этой задаче связан только с одним соседом, в общем же случае связи между объек-

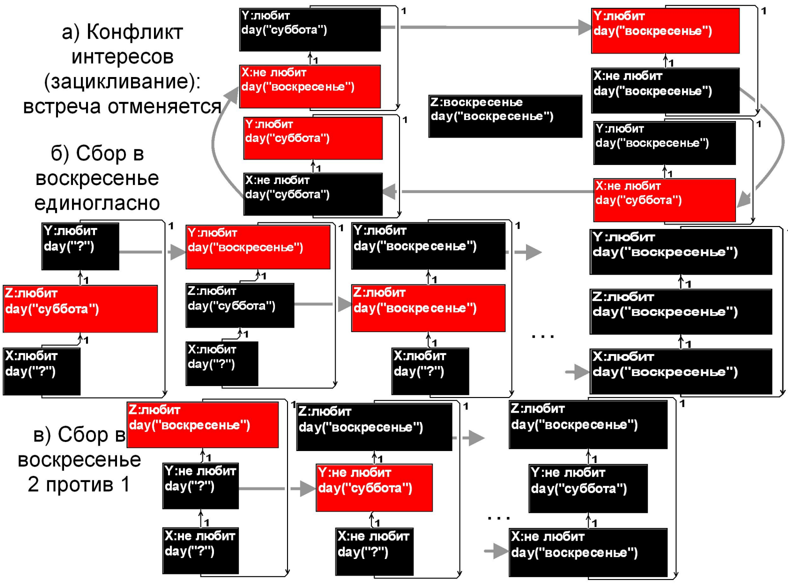


Рис. 3. Примеры использования стратегии Киссинджера

тами могут быть гораздо сложнее. Например, в состав участников вечеринки можно ввести «большевика» (и/или «меньшевика»), этот участник будет подсчитывать сторонников субботы и воскресенья и примыкать к большинству (к меньшинству). Замечательно то, что подобные изменения никак не влияют на работоспособность стратегии Киссинджера. •

На рис. 4 представлена иллюстрация работы программы осуществляющей настройку множества взаимодействующих объектов в соответствии со стратегией Киссинджера: в блоках осуществляется настройка $i-$го по порядку $<<$ объекта и сравнение состояния, полученного после настройки объекта, со старым состоянием объекта (с состоянием, в котором объект находился до шага настройки).

Настоящая работа посвящена исследованию множества ограничений, при которых, действуя по стратегии Киссинджера, гарантированно можно найти согласованное решение.

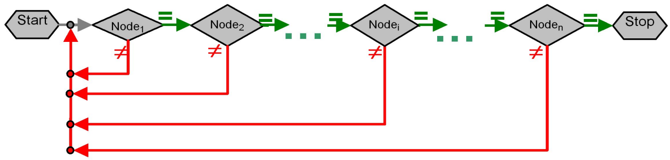


Рис. 4. Настройка по стратегии Киссинджера

4. Функциональные сети

Если для описания алгоритма настройки не требуется никакого аппарата, то для описания области применимости этого алгоритма необходимо уточнить, что понимается под терминами «сеть взаимодействующих объектов», «настроенный объект» и «настройка объекта». Уточнению перечисленных терминов и посвящен настоящий раздел.

В настоящей работе термин «сеть взаимодействующих объектов» отождествляется с термином «функциональная сеть» (формальное определение приводится в приложении: определение /33/[2]). Единственное, что требуется из этого определения для понимания дальнейшего, это следующее:

- функциональная сеть состоит из функциональных блоков, соединенных между собой (если N -– функциональная сеть, то $N.nodes$ – множество блоков присутствующих в ней);
- каждый блок имеет один выход (сигнал, порождаемый функциональным блоком) и какое-то множество (может быть, пустое) пронумерованных входов.

Кроме того, каждому блоку $x{\in}N.nodes$ соответствует название операции, которую он должен выполнять (это название операции обозначается $N.functor(x)$).

[2] *В приложении приведен список используемых в работе понятий. Поскольку нумерация определений сквозная, то в основном тексте иногда будут встречаться ссылки «вперед»: ссылки на определения из этого списка (первое определение в приложении имеет номер 8).*

• при соединении блоков между собой каждый вход каждого используемого блока должен быть подсоединен к выходу какого-нибудь блока (при этом не ограничено число входов, подсоединенных к заданному выходу). При этом блок, к выходу которого подсоединен вход с номером i блока x, обозначается $x[i]$ (определение /34/);

• в соответствии с определением /33/ функциональной сети вместо термина «функциональный блок сети» в дальнейшем будет использоваться термин «вершина сети»

• с понятием «функциональная сеть» тесно связано понятие «диагональная система уравнений»:

Каждой вершине $x \in N.nodes$ функциональной сети N соответствует уравнение вида $x = f(x[1], \ldots, x[n])$, где $f = N.functor(x)$ и n — число входов у вершины x. Соответственно, самой сети соответствует система уравнений (характеристическая система уравнений для функциональной сети N обозначается $system(N)$, определение /35)/, которая однозначно описывает исходную функциональную сеть.

Пример 4. На рис. 1 приведены примеры функциональных сетей. Стрелки направлены от входа каждой вершины к точке подсоединения этого входа (таким образом, стрелки ориентированы против направления распространения сигналов). Имена операций принимают значения из множества {*"любит"*,*"не любит"*,*"воскресенье"*}. Например, для сети N, представленной на рис. 1a), $N.nodes = \{x, y, z\}$,
при этом $N.functor(x)$=*"не любит"*, $N.functor(y)$=*"любит"*, $N.functor(z)$=*"воскресенье"*, $x[1] = y$ и $y[1] = x$.

Для функциональных сетей, представленных на рис. 1, характеристические системы уравнений имеют вид:

$a)\left\{\begin{array}{l} x = \text{“не любит”}(y), \\ y = \text{“любит”}(x), \\ z = \text{“воскресенье”}; \end{array}\right.$ б) $\left\{\begin{array}{l} x = \text{“любит”}(z), \\ z = \text{“любит”}(y), \\ y = \text{“любит”}(x); \end{array}\right.$ в) $\left\{\begin{array}{l} x = \text{“не любит”}(y). \\ y = \text{“не любит”}(z). \\ z = \text{“любит”}(x). \end{array}\right.$ •

Теперь, когда определено понятие «сеть взаимодействующих объектов», можно перейти к определению понятий «настройка объекта», «настроенный объект» и «настроенная сеть».

Для того чтобы определить эти понятия, воспользуемся классическим понятием универсальной алгебры (определение /25/). На содержательном уровне универсальная алгебра (в дальнейшем, поскольку других алгебр в работе не будет, будем пользоваться коротким термином алгебра) – это всего лишь множество не соединенных друг с другом функциональных блоков разного типа: тип блока (или марка изделия, записанная на корпусе) однозначно определяет количество входов у блока и правила его работы. Для того чтобы описать алгебру (или, говоря проще, чтобы описать имеющееся сырьё для производства функциональных сетей), необходимо задать:

- возможный диапазон входных и выходных сигналов имеющихся блоков (это множество всех возможных входных и выходных сигналов называется множеством элементов алгебры);
- множество имеющихся типов блоков (это множество называется множеством функциональных символов алгебры и именно из этого множества выбираются имена операций приписанных вершинам сети $N.functor$);
- правила функционирования блоков каждого типа (эти правила, которые каждому типу блока сопоставляют реализуемую блоком функцию, называются интерпретацией множества функциональных символов. При этом блок, не имеющий входов, является источником постоянного сигнала, а значение выхода у блока с входами функционально зависит от значений сигналов поступающих на его входы).

Таким образом, любая алгебра A задается тройкой (Q, F, μ), где $Q-$ множество элементов алгебры (диапазон сигналов); $F-$ множество функциональных символов (типы блоков) и μ - интерпретация (правила функционирования блоков). При этом если Q – конечное множество, то A называется конечной алгеброй.

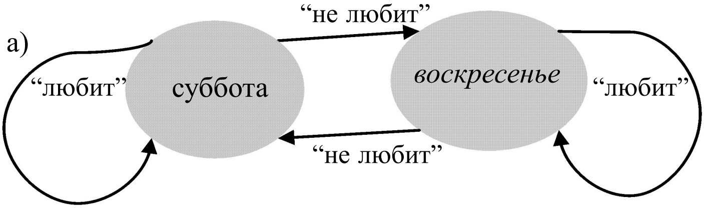
Для выражения того факта, что сеть построена из блоков задаваемых алгеброй A, используется термин сеть [представленная] над алгеброй A (определение /39/).

<u>*Пример* 5.</u> Рассмотрим функциональные сети, представлен-

ные на рис. 1, и опишем алгебру (назовем ее A_{party}), над которой представлены эти схемы.

Эти схемы состоят из следующих типов блоков: *“любит”* (с одним входом), *“не любит”* (с одним входом) и *“воскресенье”* (без входов). В соответствии с условием задачи о вечеринке (примеры 1, 2) множество входных и выходных сигналов блоков принадлежит множеству $\{$*суббота*, *воскресенье*$\}$.

а)



б)

Func / Arg	**“любит”**	**“не любит”**
суббота	*суббота*	*воскресенье*
воскресенье	*воскресенье*	*суббота*

Рис. 5. Алгебра A_{party}

Итак, $A_{party} = (\{$*суббота*, *воскресенье*$\}, \{$*“любит”*, *“не любит”*, *“воскресенье”*$\}, \mu)$, где $\mu($*“воскресенье”*$) =$ *воскресенье* (константа), а интерпретация функциональных символов *“любит”* и *“не любит”* задана на рис. 5, с помощью диаграммы (рис. 5а) и таблицы (рис. 5б) переходов конечного автомата.

Содержательно: если y *“любит”* x , то y назначает тот же день встречи, что и x, а, если x *“не любит”* y, то x назначает день, отличный от дня, назначенного y.●

Определение 2. *Пусть N – функциональная сеть (/33/) над алгеброй (/39/) $A = (Q, F, \mu)$ (/25/) и V – множество вершин сети N ($V = N.nodes$ /33/). Тогда множеством состояний сети N над A (обозначается $States(A, N)$) называется множество всех отображений из V в Q.*

Содержательно функциональная сеть с множеством вершин

V находится в состоянии $p\in States(A,N)$ тогда и только тогда, когда для любой вершины $x\in V$ значение сигнала на выходе x равно $p(x)$.

Следующее определение уточняет, что такое «шаг настройки объекта x» (это переход к новому состоянию, в котором выходной сигнал блока x равен значению функции реализуемой блоком при текущих значениях входных сигналов)

Определение 3. *(в /40/ приведено более строгое определение.) Пусть N функциональная сеть над алгеброй A (/39/). Для каждой вершины $x\in N.nodes$ функциональной сети N определим отображение $\langle{}^{node}_{step}\rangle_x : States(A,N) \to States(A,N)$ (/2/) следующим образом. Пусть вершина x имеет n входов и пусть f – функция, реализуемая этой вершиной (f – это функция соответствующая функциональному символу $N.functor(x)$ в алгебре A). Тогда для любого состояния $p \in States(A,N)$ $\langle{}^{node}_{step}\rangle_x(p)$ – это такое состояние $q \in States(A,N)$, что для любой вершины $y\in V$*

- *$q(y) = p(y)$, если $x \neq y$ и*
- *$q(x) = f(p(x[1]),\dots,p(x[n]))$ (в соответствии с определением /34/ $x[i]$ – это вершина, порождающая сигнал для i-го входа x).*

Замечание . Фактически в определении /3/ описано то, как меняются значения переменных на одном шаге метода итераций (метод итераций применяется для поиска решения характеристической системы уравнений сети), в том случае, когда шаг итерации применяется не ко всем вершинам, а ровно к одной.

Определение 4. *Пусть N – функциональная сеть (/33/) над алгеброй A (/39/), V – множество вершин схемы N ($V = N.nodes$ (/33/)) и $p \in States(A,N)$ (/2/). Говорят, что в состоянии p вершина $x\in V$ настроена, если $\langle{}^{node}_{step}\rangle_x(p) = p$. Состояние p называется настроенным (или согласованным), если в нем все вершины множества V настроены.*

Замечание . Равенство $\left\langle {node \atop step} \right\rangle_x(p) = p$ означает, что в состоянии p уравнение вершины $x = f(x[1], \ldots, x[n])$ является равенством в алгебре A: что $p(x) = f(p(x[1]), \ldots, p(x[n]))$ (здесь f – функция, реализуемая вершиной x).

Таким образом, в процессе настройки функциональной сети N ищется решение характеристической для N системы уравнений (утв. 16.следствие 1),

Пример 6. Любому согласованному в терминах задачи о вечеринке (примеры 1, 2, 3) состоянию какой-нибудь из сетей, представленных на рис. 1, соответствуют такие значения выходных сигналов у вершин x, y, z, которые являются решением соответствующей характеристической системы уравнений в алгебре A_{party} (пример 5). Наоборот: любому решению характеристической системы уравнений (определение /31/) соответствует настроенное состояние сети.●

Определение 5. *(В /42/ приведено более общее определение). Пусть N – функциональная сеть над алгеброй A (/39/) и V множество ее вершин. Последовательность $\pi : \{1, 2, \ldots\} \rightarrow States(A, N)$ (/2/) (конечная либо бесконечная) состояний схемы называется настроечной последовательностью, если для каждого не последнего члена последовательности с номером i существует такая вершина $x \in V$, что $\pi(i+1) = \left\langle {node \atop step} \right\rangle_x(\pi(i))$ (/3/).*

Настроечная последовательность называется избыточной, если в ней присутствует более одного согласованного состояния.

Замечания. 1. Каждый элемент настроечной последовательности описывает значения переменных на соответствующем шаге итерационного процесса решения характеристической системы уравнений модифицированным методом итераций.

2. Понятие избыточной последовательности вводится как аналог сходящейся последовательности (любую конечную избыточную настроечную последовательность можно продолжить до бесконечной сходящейся последовательности).

Следующее определение является переформулировкой определения стратегии Киссинджера в терминах состояний функциональных схем.

Определение 6. *(Более корректно это определение сформулировано в /46/). Пусть N – функциональная сеть над алгеброй A (/39/) и $<<$ линейный порядок на множестве V вершин схемы N. Для любого ненастроенного состояния $q \in States(A, N)$ (/2/) главной ненастроенной вершиной (обозначается $\langle{chief \atop node}\rangle(<<, q)$) будем называть наименьшую по порядку $<<$ ненастроенную вершину (/4/) из множества V.*

Пусть $\pi : \{1, 2, \ldots\} \to States(A, N)$ - настроечная последовательность (/5/) состояний схемы N (/2/). Последовательность π называется $<<$ последовательностью Киссинджера для схемы N, если для любого не первого члена этой последовательности с номером i выполняется, либо
$\pi(i) = \langle{node \atop step}\rangle_{\langle{chief \atop node}\rangle(<<, \pi(i-1))}(\pi(i-1))$ *(/3/),*
либо $\pi(i) = \pi(i-1)$ и при этом $\pi(i-1)$ – настроенное состояние (/4/).

Замечание . Таким образом, стратегия Киссинджера соответствует модифицированному методу итераций: на каждом шаге итерации вычисляется новое значение ровно одной переменной (той, которая является наименьшей из всех не удовлетворяющих соответствующему равенству переменных).

5. Комбинаторные алгебры

В этом разделе обсуждается следующий вопрос: «В каких случаях гарантирована сходимость последовательности Киссинджера»?

О том, что метод итераций, основанный на стратегии Киссинджера, не слабее обычного метода итераций, позволяет судить хотя бы тот факт, что всякий раз, когда гарантирована сходимость метода итераций (например, все отображения сжимающие либо все отображения ограничены и монотонны) гарантирована также сходимость и модифицированного метода.

В связи с дальнейшим изложением следует подчеркнуть, что все доказанные результаты касаются лишь очень частного случая: случая, когда сеть рассматривается над конечной алгеброй (в этом случае сходимость эквивалентна тому, что последовательность стабилизируется через конечное число шагов). Для рассмотрения общего случая требуется существенно более сложный аппарат (в частности, требуется понятие точности настройки).

Приводимое далее определение «регулярно устойчивых алгебр» вводится только для сокращения терминологии (для краткого обозначения тех алгебр, на которых стратегия Киссинджера работает). Этот термин нужен только для формулировки теоремы 1 и нигде больше он использоваться не будет. Уже в определении 6, чтобы очертить ту область, где есть строгие результаты, термин «сходящаяся последовательность» был заменен на термин «избыточная последовательность».

Определение 7. *(пояснения по поводу использования ссылок вперёд приведены в сноске [2] на странице 16) Конечная алгебра A (/25/) называется регулярно устойчивой, если для любой конечной (с конечным множеством вершин) схемы N над A (/39/) и любого линейного порядка $<<$ на ее вершинах любая бесконечная $<<$ последовательность Киссинджера (/46/) является избыточной (/42/).*

Конечная алгебра A (/25/) называется устойчивой, если для любой конечной (с конечным множеством вершин) схемы N над A (/39/) существует такой линейный порядок $<<$ на ее вершинах, что любая бесконечная $<<$ последовательность Киссинджера (/46/) является избыточной (/42/).

Целью настоящего раздела являются формулировка теоремы 1, в которой утверждается, что класс регулярно устойчивых алгебр совпадает с классом комбинаторных алгебр.

Использование термина «комбинаторная алгебра[3]» вызвано

[3] *Рассматриваемые комбинаторные алгебры не имеют отношения к*

тем, что в теории конечных автоматов (как уже говорилось, понятие автомата совпадает с понятием унарной алгебры) существует понятие комбинаторный автомат, которое в точности подходит под введенное определение (на самом деле определение комбинаторных алгебр базируется на классическом понятии комбинаторного автомата).

На содержательном уровне комбинаторная алгебра – это алгебра, в которой классический метод итерации работает для любого уравнения от одной переменной. Например, уравнение от одной переменной может иметь вид $x = f(x, g(x, 1))$.

В конечном случае (а именно такой случай и рассматривается в работе) это означает, что при решении уравнения $x = t(x)$ (t – сколь угодно сложное выражение с одной переменной, например $f(x, g(x, 1))$) над конечной комбинаторной алгеброй точное решение отыскивается за конечное число итераций. В бесконечном же случае для работоспособности стратегии Киссинджера потребуются дополнительные ограничения. Одним из таких ограничений является условие равномерной сходимости (хватит и равномерной фундаментальности) последовательности отображений $t(x), t(t(x)), t(t(t(x))), \ldots$.

Из теории автоматов ([1, 12]) известно следующее необходимое и достаточное условие комбинаторности.

Утверждение 1. *Конечный автомат* (Q, F, μ) *(конечная унарная алгебра* (Q, F, μ) *(/25/)) является комбинаторным тогда и только тогда, когда выполняется следующее условие. Для любой входной последовательности* α *(для алгебраической функции* f_α*, задаваемой строкой* α*), для любого состояния* $q \in Q$ *(для любого элемента алгебры* $q \in Q$*) и любого натурального числа* n *из* $(f_\alpha)^n(q) = q$ *следует* $f_\alpha(q) = q$.

Пример 7. Автомат, представленный на рис. 5, не является комбинаторным автоматом (алгебра A_{party} из примера 5 не является комбинаторной алгеброй), так как

тем комбинаторным алгебрам, которые изучаются в λ *- исчислении.*

“не любит”^{2}(*воскресенье*) = *“не любит”*(*“не любит”*(*воскресенье*)) = *воскресенье* и, в то же время, *“не любит”*(*воскресенье*) =*суббота*.●

Пример 8. Далее приводятся примеры достаточных условий комбинаторности (рассматриваются лишь конечные алгебры)

a) если на элементах алгебры можно задать линейный порядок так, чтобы все алгебраические операции задавали монотонные функции, то алгебра комбинаторна;

b) если на элементах алгебры можно задать метрику, при которой все алгебраические операции задают сжимающее отображение, то алгебра комбинаторна;

c) объединение не пересекающихся по элементам комбинаторных алгебр -– комбинаторная алгебра;

d) прямое произведение комбинаторных алгебр — комбинаторная алгебра;

e) гомоморфный образ комбинаторной алгебры – комбинаторная алгебра. ●

Теорема 1. *Множество регулярно устойчивых алгебр (/7/) совпадает с множеством комбинаторных алгебр (/45/).*
Доказательство в приложении (утверждения 25.следствие 2 и 26).

Заключение

В работе были определены ограничения на поведение объектов, обеспечивающие возможность настройки любой функциональной сети, построенной из них (поведение объектов должны задаваться операциями комбинаторной алгебры), и был приведен алгоритм настройки, который гарантированно приводит к успеху при соблюдении введенных ограничений.

Главная тема дальнейших исследований в намеченном направлении: доказать гипотезу о том, что время настройки сети, представленной над фиксированной конечной комбинаторной алгеброй, пропорционально n^2, где n – число настраиваемых объектов. Еще одно направление – это обобщение приведенного результата на случай бесконечных алгебр.

Следует отметить, что алгоритм настройки един для всех комбинаторных алгебр и, следовательно, является хорошим претендентом на эвристический метод настройки схем над произвольными алгебрами (не обязательно комбинаторными и не обязательно конечными): случай, когда не известны ни правила поведения объектов, ни структура их взаимодействий.

Литература

1. АРБИБ М. А., ГИНЗБУРГ С., ЗЕЙГЕР П., и другие *Алгебраическая теория автоматов, языков и полугрупп* . – М.: Статистика, 1975. – 335 с. – [Электронный ресурс]. – URL: *http://bookre.org/reader?file=437779* (дата обращения: 10.07.2013).
2. АРЖАНЦЕВ И.В. *Базисы Грёбнера и системы алгебраических уравнений*. – М.:МЦНМО, 2003. – 68 с. – [Электронный ресурс]. – URL: *http://www.mccme.ru/free-books/dubna/arjantsev.pdf* (дата обращения: 10.07.2013).
3. АРТАМОНОВ В.А., САЛИЙ В.Н. , СКОРНЯКОВ Л.А., ШЕВРИН Л.Н., ШУЛЬГЕЙФЕР Е.Г. *Общая алгебра*. – Т.2. СМБ. – М.:«Наука» Физматлит, 1991. – 479. – [Электронный ресурс]. – URL: *http://bookre.org/reader?file=441272* (дата обращения: 10.07.2013).
4. БУХБЕРГЕР Б., КОЛЛИНЗ ДЖ., ЛООС Р. *Компьютерная алгебра: символьные и алгебраические вычисления.* – М.: Мир, 1986. – 392 с. – [Электронный ресурс]. – URL: *http://bookre.org/reader?file=437353* (дата обращения: 10.07.2013).
5. ВАЛИЕВ М.К., ДЕХТЯРЬ М.И., ДИКОВСКИЙ А.Я. *Системы агентов, управляемых логическими программами: сложность верификации* // Программирование. – 2009. – №5. – С. 37–56. – [Электронный ресурс]. – URL: http://pagesperso.lina.univ-

nantes.fr/~dikovsky-a-1/publies/MASJProgr09.pdf (дата обращения: 30.10.2013).

6. ДЕХТЯРЬ М.И., ДИКОВСКИЙ А.Я. *Анализ поведения дискретных динамических систем средствами логичекого программирования* // Программирование. – 1996. – №3. – С. 3–16. – [Электронный ресурс]. – URL: http://homepages.tversu.ru/~p000103/ZHPROG.pdf (дата обращения: 30.10.2013).
7. ДЕХТЯРЬ М.И. *Обновления баз данных при динамических ограничениях целостности* // В сб. «Системная информатика» – 2002. – №8. – С. 72–142. – [Электронный ресурс]. – URL: http://homepages.tversu.ru/~p000103/Nov-rus.pdf (дата обращения: 30.10.2013).
8. ДЭВЕНПОРТ ДЖ., СИРЭ И., ТУРНЬЕ Э. *Компьютерная алгебра*.– М.: Мир, 1991. – С. 126–139 – [Электронный ресурс]. – URL: *http://bookfi.org/book/758936* (дата обращения: 10.07.2013).
9. КОКС Д., ЛИТТЛ ДЖ., О'ШИ Д. *Идеалы, многообразия и алгоритмы*. – М.:Мир, 2000. – С. 11–150. – [Электронный ресурс]. – URL: *http://bookre.org/reader?file=440091* (дата обращения: 16.07.2013).
10. КОН П. *Универсальная алгебра*. – М.: Мир, 1968. – – [Электронный ресурс]. – 352 с. – URL: *http://bookre.org/reader?file=567818* (дата обращения: 10.07.2013).
11. КУРОШ А.Г. *Лекции по общей алгебре*. – 2-е изд. М.: Физматлит, 1973. – [Электронный ресурс]. – 399 с. – URL: *http://bookre.org/reader?file=441263* (дата обращения: 10.07.2013).
12. ЛАЛЛЕМАН Ж. *Полугруппы и комбинаторные приложения*.– М.: Мир, 1985. – С. 174–228. – [Электронный ресурс]. – URL: *http://bookre.org/reader?file=440008* (дата обращения:

10.07.2013).
13. МАЛИШЕВСКИЙ А.В. *Качественные модели в теории сложных систем*. – М.:«Наука» Физматлит, 1998. – С. 48–281. – [Электронный ресурс]. – URL: *http://bookre.org/reader?file=561940* (дата обращения: 30.10.2013).
14. НОВИКОВ Д.А. *Управление проектами: организационные механизмы.* – М.: ПМСОФТ, 2007. – [Электронный ресурс]. – 140 с. – URL: *http://www.aup.ru/books/m185/* (дата обращения: 30.10.2013).
15. ТРАХТЕНГЕРЦ Э.А. *Компьютерная поддержка переговоров при согласовании управленческих решений.* – М.: Синтег, 2003. – 273 с.
16. BLAND R.G. *New finite pivot rules for simplex method* // Math. Oper. Res. – 1977. – Vol. 2. – P. 103–107.
17. DEKHTYAR M., DIKOVSKY A., DUDAKOV S. *On Complexity of Updates Through Integrity Constraints* // Proc. of the First Int. Conf. on Computational Logic (CL 2000), LNAI, Vol. 1861 2000. – P. 867–881. – [Электронный ресурс]. – URL: http://pagesperso.lina.univ-nantes.fr/~dikovsky-a-1/publies/cmplog00.pdf (дата обращения: 30.10.2013).
18. DEKHTYAR M., DIKOVSKY A., VALIEV M. *Temporal Verification of Probabilistic Multi-Agent Systems* // In: Trakhtenbrot/Festschrift, [A. Aviron et al. (Eds.)] LNCS 4800. – Springer, 2008. – P. 256–265. – [Электронный ресурс]. – URL: http://library.keldysh.ru/prep_vw.asp?pid=2882 (дата обращения: 30.10.2013).
19. KHOMYAKOV M., BIDER I. *Achieving Workflow Flexibility through Taming the Chaos* // OOIS 2000 - 6th international conference on object oriented information systems. – Springer, 2000. – P. 85–92. – [Электронный ресурс]. – URL:

http://www.ibissoft.se/publications/tamingchaos.pdf
(дата обращения: 26.05.2013).

Приложение[4]

1. Используемые термины и обозначения

1.1. ЧИСЛА (НАТУРАЛЬНЫЕ)

Операции и отношения: $+$, -, $<$, $>$, $\leqslant$, $\geqslant$, max, min.

$\{1,2,\ldots\}$- ($\{0,1,\ldots\} \overset{def}{=} \{1,2,\ldots\}\cup\{0\}$) множество натуральных чисел.

При $n\in\{0,1,\ldots\}$ $\{1,...,n\}$ – начальный отрезок натурального ряда длины n (при $n=0$ $\{1,...,n\}=\emptyset$).

$max(s)$ ($min(s)$) — максимальный (минимальный) элемент не пустого множества чисел s (значения $min(s)$ и $max(s)$ не для всех множеств чисел определены, но заведомо определены для непустых конечных множеств, кроме того $min(s)$определено для любого не пустого множества $s\subseteq\{0,1,\ldots\}$).

1.2. МНОЖЕСТВА

Операции и отношения: $\cap$, $\cup$, $\setminus$, $\times$, $\in$, $\notin$, $\subseteq$ (пересечение, объединение, вычитание, прямое произведение, принадлежность/не принадлежность элемента множеству, включение множества).

$\emptyset$ – пустое множество.

Если s – множество и $n\in\{1,2,\ldots\}$, то $s^n \overset{def}{=} s\times\ldots\times s$ (прямая
степень множества: $\{a,b\}^1=\{\{a\},\{b\}\}\neq\{a,b\}$ и $\{a\}^2\times\{a\}=\{((a,a),a)\}\neq\{a\}\times\{a\}^2=\{(a,(a,a))\}\neq\{a\}^3=\{(a,a,a)\}$).

$|s|$ – мощность множества s.

2^s – множество подмножеств множества s.

[4] *Приложение приводится в авторской редакции*

1.3. ОТНОШЕНИЯ И ОТОБРАЖЕНИЯ

Множеством отношений (бинарных отношений) (иногда бинарные отношения называют также ориентированными графами) над Q называется множество всех подмножеств $Q{\times}Q$ (множество $2^{Q\times Q}$) . Тот факт, что $(x, y){\in}s$, принято записывать следующим образом: $x\ s\ y$.

Для любых отношений $r, s{\subseteq}Q{\times}Q$ (в приведенных далее обозначениях используются квадратные скобки, для того чтобы отличать прямое произведение множеств/отношений и степень/рефлексивно- транзитивное замыкание отношения от прямой степени/итерации множества)

- $r_{\bullet}s \overset{def}{=} \{(a, b) : (a, c) \in r$ и $(c, b) \in s\}$ (произведение отношений);
- $s^{[0]} \overset{def}{=} \{(x, x) : x \in s\}$ (отношение равенства) обозначается $=_s$ (либо $=$ при известном s). При этом $\neq_s \overset{def}{=} s^2 \backslash {=_s}$;
- для любого $i{\in}\{1, 2, \ldots\}$ $s^{[i]} \overset{def}{=} s^{[i-1]}{}_{\bullet}s$ (степень отношения);
- $s^{[-1]} \overset{def}{=} \{(a, b) : (b, a){\in}s\}$ (обозначение s^{-1} тоже допустимо);
- $s^{[*]} \overset{def}{=} \cup\{s^{[i]}{:}i{\in}\{0, 1, \ldots\}\}$ – рефлексивно-транзитивное замыкание:
- $dom(r) \overset{def}{=} \{x : (x, y){\in}r\}$;
- $vals(r) \overset{def}{=} \{y : (x, y){\in}r\}$.

1.3.1. Множество эквивалентностей на множестве Q

(всех тех отношений $r{\subseteq}Q^2$, которые замкнуты относительно рефлексивно-транзитивного замыкания ($r{=}r^{[*]}$) и обращения ($r{=}r^{-1}$)) обозначается $Equivalences(Q)$.

1.3.2. Множество линейных порядков на множестве Q

(всех тех отношений $r{\subseteq}Q^2$, которые удовлетворяют равенствам: $r{=}r^{[*]}$, $r{\cap}r^{-1}{=}r^{[0]}$, $r{\cup}r^{-1}{=}Q^2$), обозначается $\langle{}^{Line}_{Orders}\rangle(Q)$

Если Q – конечное непустое множество и $<<{\in}\langle{}^{Line}_{Orders}\rangle(Q)$ — линейный порядок на Q, то наибольший по порядку $<<$ элемент множества Q обозначается $max_{<<}(s)$.

1.3.3. Множество отображений из Q в Q'

(таких отношений $f{\subseteq}(Q{\cup}Q')^2$, что $dom(f) = Q, vals(f){\subseteq}Q'$ и для любого $x{\in}Q$ $|\{y : (x, y) \in f\}| = 1$) обозначается $Q{\to}Q'$.

Тот факт, что $f{\in}Q{\to}Q'$, принято записывать следующим образом $f{:}Q{\to}Q'$ и для любого $x{\in}Q$ тот единственный элемент $y{\in}Q'$ для которого $(x, y){\in}f$ обозначается $f(x)$.

Отображение $f{:}Q{\to}Q'$ называется обратимым, если $f^{-1}{\in}$ $vals(f){\to}Q$.

Отображение $f{:}Q{\to}Q'$ называется отображением на Q', если $vals(f) = Q'$.

Обратимое отображение $f{:}Q{\to}Q'$ на Q' называется взаимно однозначным отображением.

Пусть $Q{\subseteq}\{0, 1, \ldots\}$. Тогда отображение $f{:}Q{\to}\{0, 1, \ldots\}$ называется монотонным в том и только в том случае, когда для любых $n, m{\in}\{0, 1, \ldots\}$ из $n{\leqslant}m$ следует $f(n){\leqslant}f(m)$.

Если $f{\in}Q{\to}Q'$ и s – множество (не требуется $s{\subseteq}Q$), то отображение $f{\cap}(s \times Q') \in (s \cap Q) \to Q'$ принято обозначать через $f|_s$.

Сужение операции произведения отношений на отображения (отношение $\bullet|_{(s\to Q)^2}$) называется операцией композиции отображений (обозначается $\bullet$).

Обозначения. Несмотря на то, что отображение является множеством, мы будем использовать для обозначения отображения $f^{[n]}$ то же обозначение, которое используется для прямой степени множества f^n. Конфликт при выборе способа интерпретации обозначения s^n должен разрешаться по следующему правилу: если множество s объявлено как отображение (если указано, что $s{\in}Q{\to}Q'$ для некоторых множеств Q и Q'), то запись s^n следует интерпретировать как $s^{[n]}$, в противном случае как прямая степень множества s.

Утверждение 2. *Пусть Q – множество и $r, r', r''{\subseteq}Q{\times}Q$. Тогда $(r_\bullet r')_\bullet r''{=}r_\bullet(r'_\bullet r'')$* (/1.3/). *При этом если $r, r'{\in}Q{\to}Q$* (/1.3.3/), *то и $r_\bullet r'{\in}Q{\to}Q$.* ***Без доказательства***

1.4. СТРОКИ И ПОСЛЕДОВАТЕЛЬНОСТИ

Отображения (/1.3.3/) из начального отрезка натурального ряда $\{1, ..., n\}$, где $n \in \{0, 1, \ldots\}$, во множество V (конечные последовательности) называются строками над алфавитом V. При этом отображения из $\{1, 2, \ldots\} \to V$ называются (бесконечными) последовательностями над V.

Множество $\{1, ..., n\} \to V$ (множество всех строк длины $n \in \{0, 1, \ldots\}$ над алфавитом V) обозначается V^n (так же как прямая степень множества V).

Множество всех строк над алфавитом V обозначается $V*$: $V* \overset{def}{=} \cup \{V^n : n \in \{0, 1, \ldots\}\}$.

Если $\alpha, \beta \in V*$ – строки над алфавитом V, то

- $length(\alpha)$ – длина строки α ($length(\alpha) \overset{def}{=} |dom(\alpha)|$);
- строка длины 0 называется пустой строкой и обозначается ε (хотя это не существенно для дальнейшего, но на самом деле $\emptyset = \varepsilon$);
- если $\alpha \in V* \setminus \{\varepsilon\}$ не пустая строка, то для любого $i \in \{1, ..., length(\alpha)\}$ $\alpha(i)$ – i–й символ строки α (поскольку $\alpha \in \{1, ..., length(\alpha)\} \to V$) и $last(\alpha) \overset{def}{=} \alpha(length(\alpha))$ – последний символ строки α.
- результат конкатенации α и β (результат приписывания β после α) будет в дальнейшем обозначаться $\alpha^\wedge\beta$, либо $\alpha\beta$.
- в дальнейшем, когда это не будет вызывать неоднозначности, элементы алфавита V и строки длины 1 будут обозначаться одинаковым образом (иногда символ алфавита $f \in V$ будет обозначать такую строку α, что $length(\alpha) = 1$ и $\alpha(1) = f$). Но в тех случаях, когда элемент $a \in V$ требуется отличить от строки a длины 1, для обозначения строки используется обозначение $\{a\}$ ($\{a\} \in V^1$ – это строка длины 1, первым символом которой является a). Таким образом $\{a\}(1) = a$ и $length(\{a\}) = 1$.
- любая строка $\alpha \in V*$ является (по определению /1.4/) отображением, а значит (по определению /1.3.3/), и множеством. Поэтому использование стандартной нотации приводит к тому, что разные объекты: $\alpha^\wedge\alpha ...^\wedge\alpha$ (n раз) (строка из $V*$) и $\alpha \times \alpha \times ... \times \alpha$ (n раз), обозначаются одинаковым образом α^n. Поэтому для записи $\alpha^\wedge\alpha ...^\wedge\alpha$ (n раз) будет использоваться обозначение $\alpha^{\wedge n}$.

Определение 8. *Пусть V – множество. Определим отображения $prefixes{:}V^*{\rightarrow}2^{(V^*)}$ и $suffixes{:}V^*{\rightarrow}2^{(V^*)}$ (/1.4/) следующим образом: для любой строки $\alpha{\in}V^*$ $prefixes(\alpha) \stackrel{def}{=}$ $\{\alpha' : \alpha'{}^\wedge\beta'{=}\alpha\}$ и $suffixes(\alpha) \stackrel{def}{=} \{\beta' : \alpha'{}^\wedge\beta'{=}\alpha\}$.*

Определение 9. *Пусть V – множество. Определим отображения $Prefix{:}V^*{\times}\{0,1,\ldots\}{\rightarrow}V^*$ и $Concat{:}(V^*)^*{\rightarrow}V^*$ следующим образом.*

Для любых $\pi{\in}V^*$ и $i{\in}\{1,2,\ldots\}$

$$Prefix(\pi,i) \stackrel{def}{=} \begin{cases} \pi, & \text{если } i \notin dom(\pi) \cup \{0\} \\ \pi', & \text{если } i \in dom(\pi), length(\pi') = i \\ & \text{и } \pi' \in prefixes(\pi) \end{cases} \quad \text{(/1.3.3,1.4,8/)}.$$

Для любых $\pi \in (V^*)^*$

$$Concat(\pi) \stackrel{def}{=} \begin{cases} \varepsilon, & \text{если } \pi = \varepsilon \\ Concat(\pi')^\wedge\alpha, & \text{если } \pi = \pi'{}^\wedge\{\alpha\} \\ & \text{и } \alpha \in V^* \end{cases} \quad \text{(/1.4/)} .$$

Определение 10. *Пусть V – множество и $r{\subseteq}V{\times}V$ – отношение (/1.3/). Продолжением отношения r на строки называется отношение $r' \subseteq V^*{\times}V^*$ такое, что для любых $\alpha,\beta{\in}V^*$(/1.4/) $\alpha\ r'\ \beta$ тогда и только тогда, когда выполняются следующие условия:*

- *$length(\alpha) = length(\beta)$ (/1.4/)*
- *для любого $i \in \{1,...,length(\alpha)\}$ выполняется отношение $\alpha(i)\ r\ \beta(i)$*

В дальнейшем продолжение отношения r на строки будет обозначаться $[*]^r$(в частности, для любого отношения $r{\subseteq}V{\times}V$ $\varepsilon\ [*]^r\varepsilon$).

Утверждение 3. *Пусть $r \subseteq V{\times}V$ – отношение (/1.3/). Тогда*

- *если $r \in V{\rightarrow}V$ (/1.3.3/), то $[*]^r{\in}\ V^*{\rightarrow}V^*$ (/10, 1.3.3/);*
- *если r – отношение эквивалентности (/1.3.1/), то $[*]^r$– отношение эквивалентности (/10,1.3.1/).* ***Без доказательства***

Определение 11. *Пусть S – множество и $<<\in\langle{}^{Line}_{Orders}\rangle(S)$ – линейный порядок на множестве S (/1.3.2/). Для любого конечного подмножества $s \subseteq S$ и любого $x \in s$ номером x в s в соответствии с порядком $<<$ (обозначается $num_{<<,s}(x)$) называется число $num_{<<,s}(x) \stackrel{def}{=} |\{y : y{\in}s\ \&\ y << x\}|$.*

Утверждение 4. *Пусть S – множество, $<<\in\langle{}^{Line}_{Orders}\rangle(S)$ – линейный порядок на множестве S (/1.3.2/) и $s \subseteq S$ конечное подмножество. Тогда отображение $num_{<<,s}{:}s{\to}\{1,...,|s|\}$ (/11/) является обратимым отображением на $\{1,...,|s|\}$ (отношение $(num_{s,<<})^{-1}{\subseteq}\ s{\times}\{1,2,\ldots\}$ является отображением из $1,...,|s|\}$ на s).* ***Без доказательства***

Замечание . Таким образом, $(num_{s,<<})^{-1}$ это такая строка длины $|s|$ над алфавитом s, в которой все элементы расположены по возрастанию (для любого $i{\in}\{1,...,|s|\}$ выполняется равенство $(num_{<<,s})^{-1}(i) = max_{<<}(\{x : x{\in}s\&num_{<<,s}(x){\leqslant}i\})$).

Определение 12. *Пусть Q – конечное множество, $<<\in\langle{}^{Line}_{Orders}\rangle(Q)$ - линейный порядок на Q (/1.3.2/), $s{\subseteq}Q$ и $\varphi{:}s{\to}Q'$ некоторое отображение (/1.3.3/). Введем следующие обозначения:*

Отображение $(num_{s,<<})^{-1}\ {:}\{1,...,|s|\}{\to}Q$ (/11,1.3/) будем обозначать $string(s,<<)$ ($string(s,<<)$ это такая строка $\alpha{\in}Q^{|s|}$, что для любых $i,j{\in}\{1,...,|s|\}$ $i{\leqslant}j$ тогда и только тогда, когда $\alpha(i) << \alpha(j)$).

Отображение $[*]^{\varphi}(string(s,<<))\ \in Q'^{|s|}$ (/11,10/) будем обозначать $string(s,<<,\varphi)$ ($string(s,<<,\varphi)$ это такая строка $\beta{:}\{1,...,|s|\}{\to}Q'$, что для любых $i{\in}\{1,...,|s|\}$ $\beta(i){=}\varphi(string(s,<<)(i))$: строка $string(s,<<,\varphi)$ получается из строки $\lambda{=}\ string(s,<<)$ заменой каждого символа $\lambda(i)$ на $\varphi(\lambda(i))$).

Определение 13. *Пусть V множество, $\alpha{\in}V^*$ -последовательность и $s{\subseteq}\{1,2,\ldots\}$. Тогда s - подпоследовательностью последовательности α (обозначается $subseq(s,\alpha)$- это последовательность, задаваемая множеством позиций $s{\cap}dom(\alpha)$ в строке α) называется* *такая последовательность* $\beta \in V^*$*, что* $|dom(\beta)| = |dom(\alpha) \cap s|$ *и для любого* $n \in dom(\beta)$ *выполняется равенство*

$$\beta(n) = \alpha(|\{i : i \in dom(\alpha) \cap s) \& i \leqslant n\}|).$$

Утверждение 5. *Пусть* V *множество,* $\alpha \in V^*$ *(/1.4/) - последовательность и* $s \subseteq \{1, 2, \ldots\}$*. Тогда* $subseq(s, \alpha) = string(s \cap dom(\alpha), \leqslant, \alpha|_s)$ *(/12,13/).* ***Без доказательства***

Определение 14. *Пусть* V *множество,* $V' \subseteq V$ *и* $\varphi : V' \to V^*$ *некоторое отображение. Строковым гомоморфизмом, порожденным отображением* φ*, называется отображение* $\varphi^{Strings(V)} : V^* \to V^*$ *такое, что для любой строки* $\alpha \in V^*$ *и для строки* $\pi \in (V^*)^*$ *такой, что* $dom(\pi) = dom(\alpha)$ *и для всех* $i \in dom(\pi)$ $\pi(i) = \begin{cases} \varphi(\alpha(i)), & \text{если } \alpha(i) \in V' \\ \{\alpha(i)\}, & \text{если } \alpha(i) \notin V' \end{cases}$, $\varphi^{Strings(V)}(\alpha) = Concat(\pi)$ *(/9/).*

1.5. ТЕРМЫ

Определение 15. *Множеством функциональных символов называется множество* F *с определенным на нем отображением* $arn : F \to \{0, 1, \ldots\}$ *(/1.3.3/), которое сопоставляет каждому элементу* $f \in F$ *неотрицательное целое число – арность.*

Замечание . В новосибирской школе вместо термина «множество функциональных символов» используется термин «сигнатура»: сигнатура это пара (F, arn), где F – множество (функциональных символов) и $arn \in F \to \{0, 1, \ldots\}$.

Определение 16. *Пусть* F *–множество функциональных символов (/15/) и* V *– множество функциональных символов арности 0, не пересекающееся с* F *(множество переменных).*

Множеством $Terms(F, V)$ *термов над* F *и переменными* V *называется минимальное множество строк над алфавитом* $F \cup V$*, удовлетворяющее следующим условиям:*

- $V^1 \subseteq Terms(F,V)$ *(/1.4/);*
- *Если* $f \in F$ *и* $\pi \in Terms(F,V)^{arn(f)}$*, то*
  $\{f\}^\wedge Concat(\pi) \in Terms(F,V)$ *(/9,1.4/).*

Термы из множества $Terms(F,\emptyset)$ *(выражения без переменных) называются основными над множеством функциональных символов* F *(или основными).*

Замечание . Следует обратить внимание на то, что $Terms(F,V) = Terms(F \cup V, \emptyset)$ и для любых множеств V', V'' $Terms(F, V' \cup V'') = Terms(F \cup V', V'' \setminus V')$.
В дальнейшем терм $ft_1 \dots t_n = \{f\}^\wedge t_1 \dots {}^\wedge t_n$ будет для наглядности записываться в виде $f(t_1, \dots, t_n)$. То, что такое соглашение корректно провозглашается в следующем утверждении.

Утверждение 6. *Пусть* F *– множество функциональных символов (/15/) и* V *– множество функциональных символов арности 0, не пересекающееся с* F *(множество переменных). Тогда для любого* $t \in Terms(F,V)$ $t \neq \varepsilon$ *и при этом существует и единственна такая последовательность* $\pi \in (Terms(F,V) \setminus \{t\})^*$*, что* $t = \{t(1)\}^\wedge Concat(\pi)$ *(/9/).* ***Без доказательства***

Определение 17. *Определим отображения*

- $topFunctor: Terms(F,V) \to F \cup V$*,*
- $Sons: Terms(F,V) \to Terms(F,V)^*$
- $\langle {}^{sub}_{terms} \rangle : Terms(F,V) \to 2^{Terms(F,V)}$ *и*
- *для любого множества* $W \subseteq V$ $vars_W : Terms(F,V) \to 2^V$

следующим образом.

Для любых $t \in Terms(F,V) \cup \{\varepsilon\}$ *и* $W \subseteq V$

- $topFunctor(t) \stackrel{def}{=} t(1)$ *(поскольку* $t-$ *не пустая строка, то значение* $t(1)$ *определено).*
- $Sons(t)$ *такая (согласно утв.6 единственная) последовательность* $\pi \in Terms(F,V)^*$*, что* $t = \{t(1)\}^\wedge Concat(\pi)$
- $\langle {}^{sub}_{terms} \rangle (t) \stackrel{def}{=}$
  $\{\tau : \tau \in Terms(F,V) \& \alpha^\wedge \tau^\wedge \beta = t \& \alpha, \beta \in (F \cup V)^*\}$

- $vars_W(t) \stackrel{def}{=} \langle {}^{sub}_{terms} \rangle (t) \cap W$.

В дальнейшем, когда множество переменных V известно из контекста, вместо записи $vars_V(t)$ будет использоваться сокращение $vars(t)$.

Утверждение 7. *Для любых $t \in Terms(F,V)$ и $\tau \in \langle {}^{sub}_{terms} \rangle (t)$ справедливо включение $Sons(\tau) \in (\langle {}^{sub}_{terms} \rangle (t) \setminus \{t\})^*$.*
При этом $length(Sons(\tau)) = arn(\tau(1))$(это утверждение понадобится в определении /26/). ***Без доказательства***

1.6. ПОДСТАНОВКИ

Определение 18. *Подстановкой (переменных) называется такое отображение (/1.3.3/) φ:$Terms(F,V) \to Terms(F,V)$ (/16/) , что для некоторого отображения f:$V \to Terms(F,V)$ (/16/), справедливо равенство $\varphi = f^{Strings(F \cup V)}|_{Terms(F,V)}$ (/14/).*

Множество всех подстановок переменных из V принадлежащих множеству $Terms(F,V) \to Terms(F,V)$ в дальнейшем обозначается $substitutions(F,V)$.

Если $\varphi \in substitutions(F,V)$ - подстановка, то отображение f:$V \to Terms(F,V)$ такое, что для любого $x \in V$ $f(x) = \varphi(\{x\})$ будет обозначаться $\varphi|_V$.

Подстановка $\varphi \in substitutions(F,V)$ называется основной, если для любой переменной $x \in V$ $\varphi(\{x\}) \in Terms(F,\emptyset)$ – основной терм. Множество основных подстановок из $substitutions(F,V)$ будет обозначаться $valuations(F,V)$

Ядром подстановки $\varphi \in substitutions(F,V)$ называется множество $kernel(\varphi) \stackrel{def}{=} \{x : x \in V \& \varphi(x) \neq x\}$ (в частности, ядро тождественной подстановки – пустое множество).

Обозначения. В дальнейшем для любых подстановок $\varphi, \varphi' \in substitutions(F,V)$ (/18/) их композиция $\varphi_\bullet\varphi'$ (/1.3,1.3.3/) (в соответствии с последующим утверждением 10 $\varphi_\bullet\varphi' \in substitutions(F,V)$) будет обозначаться $\varphi'(\varphi)$, либо $\varphi'\varphi$ (в соответствии с утверждением 2 расстановка скобок не существенна: $\varphi\varphi'\varphi'' \stackrel{def}{=} (\varphi(\varphi'))(\varphi'') = \varphi(\varphi'(\varphi''))$).

Кроме того, любое множество S пар вида $x{\to}t$, где $x{\in}V$, $t{\in}Terms(F,V)$, такое что $\{x{\to}t, x{\to}t'\}{\subseteq}S$ влечет $t{=}t'$, может быть использовано для обозначения такой подстановки $\varphi_S{\in}substitutions(F,V)$, что

$kernel(\varphi){\subseteq}\{x : x{\to}t{\in}S\}$ и

для любой пары $x{\to}t{\in}S$ $\varphi(x){=}t$.

Например, $\{x{\to}t\}$ – обозначение для такой подстановки $\varphi{\in}substitutions(F,V)$, что $\varphi(x){=}t$ и для любого $y{\in}V\backslash\{x\}$ выполняется равенство $\varphi(y){=}y$.

Утверждение 8. *Пусть $\varphi{:}V{\to}Terms(F,V)$ –отображение из множества переменных во множество термов. Тогда существует и единственна подстановка $\varphi'{\in}substitutions(F,V)$ такая, что $\varphi'|_V{=}\varphi$.* ***Без доказательства***

Следствие 1. *Пусть $\varphi,\varphi'{\in}substitutions(F,V)$ и $t{\in}Terms(F,V)$. Тогда $\varphi(t){=}\varphi'(t)$ тогда и только тогда, когда для любой переменной $x{\in}vars_V(t)$ $\varphi(x){=}\varphi'(x)$.* ***Без доказательства***

Определение 19. *Пусть $\varphi : V \to Terms(F,V)$ –отображение из множества переменных во множество термов. Подстановка $\varphi'{\in}substitutions(F,V)$ такая, что $\varphi'|_V{=}\varphi$ будет называться $\varphi-$ подстановкой. В дальнейшем $\varphi-$ подстановка $\varphi'{\in}substitutions(F,V)$ будет обозначаться $\varphi^{Terms(F)}$, либо, если F известно из контекста, φ^{Terms}.*

Утверждение 9. *Пусть $\varphi{\in}substitutions(F,V)$ и $\psi{\in}valuations(F,V)$. Тогда $\psi_\bullet\varphi{\in}valuations(F,V)$ и $\varphi_\bullet\psi{=}\psi$.* ***Без доказательства***

Утверждение 10. *Подстановки (/18/) замкнуты относительно операции композиции (/1.3.3/) (Если $\varphi,\varphi' \in substitutions(F,V)$ (/18/), то $\varphi_\bullet\varphi' \in substitutions(F,V)$ (/1.3, 1.3.3/)).* ***Без доказательства***

1.7. СИСТЕМЫ УРАВНЕНИЙ

В последующих определениях вводится понятие системы уравнений.

Все определения почти стандартны, за тем исключением, что все уравнения имеют специальный вид: «переменная»= «выражение». Кроме того, накладывается следующее ограничение: у различных уравнений системы должны быть различные левые части.

Определение 20. *Уравнениями (или равенствами) над множеством переменных V и множеством функциональных символов F (/15/) будем называть элементы множества $equations(F,V) \stackrel{def}{=} V{\times}(Terms(F,V)\backslash V^1)$ (/16/). В дальнейшем элементы $(x,t){\in}equations(F,V)$ будут обозначаться $x{=}t$, где $x{\in}V$ – переменная, а $t{\in}Terms(F,V)\backslash V^1$ (/16/) ($equations(F,V){=}\{x{=}t : x{\in}V \& t{\in}Terms(F,V)\backslash V^1\}$).*

Определение 21. *Множество $s{\subseteq}equations(F,V)$ (/20/) будем называть системой уравнений (диагональной системой уравнений) над переменными из V и функциональными символами из F (/15/), если из $x{=}t{\in}s$ и $x{=}t'{\in}s$ следует $t{=}t'$. Множество всех диагональных систем уравнений над переменными из V и функциональными символами из F будем обозначать через $Systems(F,V)$ ($\emptyset{\in}Systems(F,V)$).*

Определение 22. *На множестве систем уравнений определим отображение $\left\langle {bound \atop Vars} \right\rangle{:}Systems(F,V){\to}2^V$ (/21/) следующим образом: для любой системы $s{\in}Systems(F,V)$ $\left\langle {bound \atop Vars} \right\rangle(s) \stackrel{def}{=} \{x : x{=}\tau{\in}s\}$.*

Определение 23. *Пусть F –множество функциональных символов (/15/) и V- множество переменных (множество функциональных символов арности 0) не пересекающееся с F. Определим отображение $\left\langle {System \atop ToSubs} \right\rangle{:}Systems(F,V){\to}substitutions(F,V)$*

следующим образом.

Для всех $s{\in}Systems(F,V)$ *(/21/)* $\left\langle{System \atop ToSubs}\right\rangle(s)$ *это такая подстановка* $\varphi{\in}substitutions(F,V)$ *(/18/), что* $kernel(\varphi) \overset{def}{=} \{y : y{=}t{\in}s\&y{\neq}t\}$ *(/18/) и для любого* $x{\in}kernel(\varphi)$ $x{=}\varphi(x){\in}s$.

Определение 24. *Пусть* $s{\in}Systems(F,V)$ *(/21/).* *Множеством внутренних подстановок системы уравнений* s*, называется наименьшее множество подстановок* $\left\langle{system \atop Subs}\right\rangle(s){\subseteq}substitutions(F,V)$*, удовлетворяющее следующим условиям:*

- *Для любой системы уравнений* $s'{\subseteq}s$
  $$\left\langle{System \atop ToSubs}\right\rangle(s') \in \left\langle{system \atop Subs}\right\rangle(s);$$
- *Если* $\varphi,\varphi'{\in}\left\langle{system \atop Subs}\right\rangle(s)$*, то* $\varphi(\varphi'){\in}\left\langle{system \atop Subs}\right\rangle(s)$.

1.8. ИНТЕРПРЕТАЦИЯ И ПОДОБИЕ ТЕРМОВ И ПОДСТАНОВОК

Определение 25. *Универсальной алгеброй [10, 11] называется тройка* $A{=}\ (Q,F,\mu)$*, где*

Q *– не пустое множество элементов алгебры,*

F *–множество функциональных символов (/15/), такое, что для любого* $q{\in}Q{\cap}F$ $arn(q) = 0$

$\mu{:}F{\to}{\cup}\{Q^i{\to}Q\ :\ i{\in}\{0,1,\ldots\}\}$ *– интерпретация функциональных символов – отображение, которое каждому* $f{\in}F$ *ставит в соответствие отображение* $\varphi(f){:}Q^{arn(f)} \ \to\ Q$ *(/1.3.3/), определенное над последовательностями длины* $arn(f)$ *(/1.4/) такое, что для любого* $q{\in}Q{\cap}F$ $\mu(q)(\varepsilon) = q$.

Содержательно, ограничение $\mu(q)(\varepsilon) = q$ *для* $q{\in}Q{\cap}F$ *означает, что элементы алгебры в качестве функциональных символов, используются только для обозначения самих себя.*

В частности, если для $c{\in}F$ $arn(c) = 0$*, то областью определения* $\varphi(c)$ *является множество, состоящее из единственного элемента* ε *– последовательность длины 0.*

В дальнейшем мы будем пользоваться термином «алгебра» вместо термина «универсальная алгебра».

Если Q и F– конечные множества, то алгебра называется конечной.

Определение 26. *Пусть $A=(Q,F,\mu)$ – алгебра (/25/) и $t\in Terms(F,\emptyset)$. Будем говорить, что отображение $\varphi:\langle^{sub}_{terms}\rangle(t)\to Q$ является A - интерпретацией терма t, если для любого $\tau\in\langle^{sub}_{terms}\rangle(t)$ выполняется равенство $\varphi(\tau)=\mu(\tau(1))([*]^{\varphi}(Sons(\tau)))$ (если $\varphi(\tau)=\mu(f)(\varphi(\tau_1),\ldots,\varphi(\tau_n))$, где $\tau=f(\tau_1,\ldots,\tau_n)$). В соответствии с утверждением 7 определение корректно.*

Утверждение 11. *Пусть $A=(Q,F,\mu)$ – алгебра (/25/). Тогда для любого $t\in Terms(F,\emptyset)$ существует и единственна A- интерпретация.* ***Без доказательства***

Определение 27. *Пусть $A=(Q,F,\mu)$ – алгебра (/25/). Определим отображение $Interpretation_A:Terms(F,\emptyset)\to Q$ (/16/) следующим образом:*

для любого $t\in Terms(F,\emptyset)$ (/16/), $Interpretation_A(t)$ равно $\varphi(t)$, где φ A- интерпретация терма t (согласно утверждению 11 определение корректно)

Определение 28. *Алгебра $A=(Q,F,\mu)$ (/25/) называется выразительной, если $\{Interpretation_A(t) : t\in Terms(F,\emptyset)\}=Q$*
Выразительным замыканием A (обозначается $extension(A)$) называется алгебра $(Q,F\cup Q,\mu')$, такая, что $\mu'(f)=\mu(f)$ для всех $f\in F$.
Содержательно, выразительное замыкание алгебры A получается из A добавлением всех ее элементов в качестве констант (функциональных символов нулевой арности).

Определение 29. *Пусть $A=(Q,F,\mu)$ – алгебра (/25/), и $B=extension(A)$ – её выразительное замыкание (/28/). В дальнейшем, для любого терма $t\in Terms(F\cup Q,\emptyset)$ элемент алгебры $Interpretation_B(t)\in Q$ будет обозначаться $value_A(t)$.*

Кроме того, если V – множество (переменных) не пересекающееся с $F{\cup}Q$, и $\varphi \in valuations(F \cup Q, V)$, то $\langle{}^{subs}_{Value}\rangle_A(\varphi)$это такое отображение $f{:}V{\to}Q$, что $f(x){=}value_A(\varphi(x))$ для всех $x{\in}V$.

Определение 30. *Пусть $A{=}(Q,F,\mu)$ – алгебра (/25/). Будем говорить, что термы $t1$, $t2{\in}Terms(F{\cup}Q,V{\setminus}Q)$ (/16/) подобны в алгебре A (обозначается $t1{\equiv}_{A,terms}t2$ либо $t1{\equiv}_A t2$), если для любой основной подстановки $\varphi{\in}valuations(Q,V{\setminus}Q)$ справедливо равенство $value_A(\varphi(t1)){=}value_{A'}(\varphi(t2))$ (/27/). Будем говорить, что подстановки $\varphi,\varphi'{\in}substitutions(F{\cup}Q,V{\setminus}Q)$ (/18/) подобны в алгебре A (обозначается $\varphi{\equiv}_{A,subs}\varphi'$, либо $\varphi{\equiv}_A\varphi'$), если для любой переменной $x{\in}V{\setminus}Q$ термы $\varphi(x)$ и $\varphi'(x)$ подобны ($\varphi(x){\equiv}_A\varphi'(x)$).*

Утверждение 12. *Пусть $A{=}(Q,F,\mu)$ – алгебра (/25/) и V множество (переменных) не пересекающееся с $F{\cup}Q$. Тогда*

- *${\equiv}_{A,subs}{\in}Equivalences(substitutions(F{\cup}Q,V))$ (/1.3.1/),*
- *${\equiv}_{A,terms}{\in}Equivalences(Terms(F{\cup}Q,V))$ и*
- *для любых $\varphi,\psi,\varphi',\psi'{\in}substitutions(F{\cup}Q,V)$ (/18/) и любых $t,t'{\in}Terms(F{\cup}Q,V)$ (/16/) таких, что $\varphi{\equiv}_{A,subs}\varphi'$, $\psi{\equiv}_{A,subs}\psi'$ и $t{\equiv}_{A,terms}t'$ выполняются отношения $\varphi(t){\equiv}_{A,terms}\varphi'(t')$ и $\varphi\psi{\equiv}_{A,subs}\varphi'\psi'$ (/30/). Без доказательства*

1.9. РЕШЕНИЯ СИСТЕМ УРАВНЕНИЙ

Далее с помощью введенного определения интерпретации термов определяется понятие решения и параметрического решения системы уравнений.

Определение 31. *Пусть $A{=}(Q,F,\mu)$ – алгебра (/25/), V - множество (переменных), не пересекающееся с $F{\cup}Q$, и $s \in Systems(F,V)$ – система уравнений (/21/). Отображение $\varphi{:}V{\to}Q$ называется решением s в A, если для любого уравнения $x{=}t{\in}s$ (/20/) справедливо равенство $\varphi(x){=}value_A(\varphi^{Terms}(t))$ (/18,27/) (равенство $\varphi(x){=}value_A(\varphi^{Terms}(t))$ эквивалентно тождеству $\varphi(x) \equiv {}_A\varphi^{Terms}(t)$).*

Множество всех решений системы уравнений s в A обозначается $solutions(s, A)$

Определение 32. *Пусть $A{=}(Q, F, \mu)$ – алгебра (/25/), V - множество (переменных), не пересекающееся с $F{\cup}Q$, и $s \in Systems(F, V)$ – система уравнений (/21/). Отображение $\varphi{:}V{\to}Terms(F{\cup}Q, V)$ (/16/) называется параметрическим решением s в A, если для любого отображения $\psi{:}V{\to}Q$ отображение $\langle{}^{subs}_{Value}\rangle_A(\psi^{Terms}(\varphi^{Terms}))$ (/29,19/) является решением s в A (/31/) (любое решение является параметрическим решением, но не наоборот). Множество всех параметрических решений системы $s{\in}Systems(F, V)$ над A будет обозначаться $\langle{}^{Parametric}_{Solutions}\rangle_{A,\,V}(s)$.*

Параметрическое решение $\varphi{\in}\langle{}^{Parametric}_{Solutions}\rangle_{A,V}(s)$ системы s называется полным, если для любого решения $\psi{\in}solutions(s, A)$ системы s существует отображение $\lambda{:}V{\to}Q$ такое, что выполняется $\langle{}^{subs}_{Value}\rangle_A(\lambda^{Terms}(\varphi^{Terms})){=}\psi(x)$ (/29,19/).

Будем говорить, что параметрическое решение $\varphi{\in}\langle{}^{Parametric}_{Solutions}\rangle_{A,V}(s)$ системы $s{\in}Systems(F, V)$, является свободным решением s, если $\varphi^{Terms} \in \langle{}^{system}_{Subs}\rangle(s)$ (/19,24/). Множество всех свободных решений системы $s{\in}Systems(F, V)$ будем обозначать $\langle{}^{Free}_{Solutions}\rangle_A(s)$.

Утверждение 13. *Пусть $A{=}(Q, F, \mu)$ –алгебра (/25/), V -множество (переменных), не пересекающееся с $F{\cup}Q$, и $\varphi{\in}\langle{}^{Parametric}_{Solutions}\rangle_{A,V}(s)$-параметрическое решение (/32/) системы уравнений $s{\in}Systems(F, V)$ (/21/). Тогда для любой подстановки $\psi{\in}substitutions(F{\cup}Q, V)$ справедливо включение $\psi(\varphi^{Terms(F\cup Q,V)}){\in}\langle{}^{Parametric}_{Solutions}\rangle_{A,V}(s)$.* ***Без доказательства***

Следствие 1. *Пусть $A{=}(Q, F, \mu)$ –алгебра (/25/), V -множество (переменных), не пересекающееся с $F{\cup}Q$, и φ- параметрическое решение (/32/) системы уравнений (/21/) $s{\in}Systems(F, V)$ ($\varphi{\in}\langle{}^{Parametric}_{Solutions}\rangle_{A,V}(s)$). Тогда для любой основной подстановки $\psi{\in}valuations(F{\cup}Q, V)$ справедливо включение $\langle{}^{subs}_{Value}\rangle_A(\psi\varphi^{Terms(F\cup Q,V)}){\in}solutions(s, A)$*

(значение $\left\langle {}^{subs}_{Value} \right\rangle_A(\psi\varphi^{Terms\ (F\cup Q,V)})$ определено в силу утв. 9: $\psi\varphi^{Terms(F\cup Q,V)}\in valuations(F\cup Q,V)$). ***Без доказательства***

Следующее утверждение фиксирует, тот факт, что при замене любой переменной в правой части уравнения системы на правую часть уравнения, описывающего заменяемую переменную, решения не теряются.

Утверждение 14. *(Все свободные решения системы уравнений являются полными решениями). Пусть $A=(Q,F,\mu)$ –алгебра (/25/) и $s\in Systems(F,V)$ – система уравнений (/21/). Тогда любое свободное решение $\varphi:V \to Terms(F,V)$ ($\in\left\langle {}^{Free}_{Solutions} \right\rangle_A(s)$ /32/) системы s над A является полным решением s над A (/32/).* ***Без доказательства***

2. Функциональные сети

2.1. СИНТАКСИС

Определение 33. *Функциональной сетью называется четверка $G = (V,R,F,functor)$, где*

- *V– множество вершин;*
- *F– множество функциональных символов (/15/), такое, что $V\cap F=\emptyset$;*
- *$functor:V \to F\cup V$ –такое отображение, что для любой вершины $x\in V$ из $functor(x)\in V$ следует $functor(x)=x$;*
- *$R\subseteq V\times\{1,2,\ldots\}\times V$ – множество дуг, помеченных натуральными числами, такое, что для любой вершины $x\in V$ выполняются следующие условия:*
  - *Если $functor(x)\in F$, то $\{(i,y):(x,i,y)\in R\}\in (\{1,...,arn(functor(x))\} \to V)$ (/15,1.3.3,1.4/);*
  - *Если $functor(x)=x$, то $\{(i,y):(x,i,y)\in R\}=\emptyset$.*

Таким образом, если из вершины $x\in V$ выходит n дуг, то вершине соответствует функциональный блок f с n входами ($functor(x)=f$, $f\in F$ и $arn(f)=n$), у которого все входы пронумерованы. Если же вершина x является висячей (нет выходящих из x дуг), то вершине соответствует функциональный блок, не *имеющий входов (генератор постоянного сигнала) либо входной сигнал (в случае, когда $functor(x)=x$).*

Функциональная сеть (в дальнейшем сеть) (V,R,F,$functor$) называется конечной, если V – конечное множество. Везде в дальнейшем под термином сеть будет подразумеваться конечная сеть.

Пусть W множество, в дальнейшем множество всех таких конечных сетей $G = (V, R, F', functor)$, что $V \subseteq W$ и $F' \subseteq F$, будем обозначать $netwares(W, F)$.

Если $G = (V, R, F', functor) \in netwares(W, F)$, то $G.nodes$ является обозначением для V, $G.arcs$ является обозначением для R и $G.functor$ обозначением для $functor$.

Определение 34. *Пусть $G = (V, R, F, functor)$ – функциональная сеть (/33/), $x \in V$, $functor(x) \neq x$ и $arn(functor(x)) > 0$. Тогда для $i \in \{1, ..., arn(functor(x))\}$ $x[i]$ –это такая однозначно определенная вершина, что $(x, i, x[i]) \in R$. В случае, когда $functor(x) = x$, либо когда $functor(x) \neq x$ и $i > arn(functor(x))$, значение $x[i]$ полагается равным $\emptyset$.*

Таким образом, если x – функциональный блок в сети, то $x[i]$ – это тот блок, к которому в сети подсоединен i - ый вход блока x, либо $\emptyset$, в случае если у x нет входа с номером i.

В следующем определении уточняется, что такое характеристическая система уравнений для функциональной схемы.

Определение 35. *Характеристической системой уравнений для сети $N = (V, R, F, functor) \in netwares(W, F)$ (/33/) называется система уравнений $system(N) \in Systems(F, V)$ (/21/) такая, что $system(N) \overset{def}{=} \{x = \{f\}^\wedge \pi_x :$*

$$x \in V \& f = functor(x) \& f \neq x \& \pi_x = \{(i, y) : (x, i, y) \in R\}\}$$

Пояснение. В силу определения /33/
$\pi_x = \{(i, y) : (x, i, y) \in R\} \in V^{arn(f)}$ (/1.4,15/), а значит определение корректно. Здесь существенным образом используется определение /1.3.3/ понятия отображения. На содержательном уровне
$system(N) = \{x = f(x[1], \ldots, x[arn(f)]) : x \in V \& f = functor(x) \& f \neq x\}$

2.2. ПЛАН ВЫЧИСЛЕНИЯ И ПОДСТАНОВКА ВЫЧИСЛЕНИЯ

Определение 36. *Пусть* $N = (V, R, F, functor)$ *- функциональная сеть (/33/). Произвольную последовательность* $\pi \in (\{1, 2, \ldots\} \to 2^V \setminus \{\emptyset\}) \cup (2^V \setminus \{\emptyset\})^*$ *(конечную при* $\pi \in (2^V \setminus \{\emptyset\})^*$*, либо бесконечную при* $\pi \in \{1, 2, \ldots\} \to 2^V \setminus \{\emptyset\}$*) будем называть планом настройки сети* N*. Множество всех конечных планов настройки* N *(множество всех строк над* $2^V \setminus \{\emptyset\}$*) будем обозначать* $\left\langle {}^{Computer}_{Plans} \right\rangle(N)$*, а множество всех планов настройки* N *будем обозначать* $\left\langle {}^{AllComputer}_{Plans} \right\rangle(N)$*.*

Определение 37. *Пусть* $N = (V, R, F, functor)$ *- функциональная сеть (/33/) и* $s \subseteq V$*. Итерационной* s *- подстановкой (обозначается* $iteration_{N,s}$*) называется такая подстановка* $\varphi \in substitutions(F, V)$ *(/18/), что*

$kernel(\varphi) = \{x : x \in s \& functor(x) \neq x\}$ *(/18/) и*

для любого $x \in kernel(\varphi)$ $x = \varphi(x) \in system(N)$ *(/35/).*

Определение 38. *Пусть* $N = (V, R, F, functor)$ *– функциональная сеть (/33/). Определим отображение* $\left\langle {}^{Computer}_{Sub} \right\rangle_N : \left\langle {}^{Computer}_{Plans} \right\rangle(N) \to substitutions(F, V)$ *(/36,18/) во множество подстановок следующим образом. Для любой строки* $\alpha \in \left\langle {}^{Computer}_{Plans} \right\rangle(N)$ *(/36/)*

- *если* $\alpha = \varepsilon$ *(/1.4/), то* $\left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha)$ *– тождественная подстановка (*$kernel(\left\langle {}^{Computer}_{Sub} \right\rangle_N(\varepsilon)) = \emptyset$ *(/18/)).*
- *если* $\alpha \neq \varepsilon$ *и* $\alpha = \alpha(1)^\wedge\alpha'$ *(/1.4/), то* $\left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha) = iteration_{N,\alpha(1)} \left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha')$ *(/18|*$_{Обозначения}$*,37/: Таким образом* $\left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha) = iteration_{N,\alpha(1)}(\left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha')) = \left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha') \bullet iteration_{N,\alpha(1)}$*).*

Множеством подстановок, порождаемых сетью N*, называется множество* $\left\langle {}^{NetWork}_{Subs} \right\rangle(N) \stackrel{def}{=} \{\left\langle {}^{Computer}_{Sub} \right\rangle_N(\alpha) : \alpha \in \left\langle {}^{Computer}_{Plans} \right\rangle(N)\}$*.*

Утверждение 15. *Пусть* $N = (V, R, F, functor)$ *– функциональная сеть (/33/). Тогда* $\left\langle {}^{NetWork}_{Subs} \right\rangle(N) = \left\langle {}^{system}_{Subs} \right\rangle(system(N))$*.*

Без доказательства

2.3. СЕМАНТИКА

Определение 39. *Говорят, что функциональная сеть $N = (V, R, F, functor)$ (/33/) задана над алгеброй $A = (Q, F', \mu)$ (/25/), если $F \subseteq F'$ и $V \cap (F \cup Q) = \emptyset$.*

Определение 40. *Пусть $N = (V, R, F, functor)$ – функциональная сеть (/33/) над алгеброй $A = (Q, F, \mu)$ (/25,39/). Для любого множества $s \subseteq V$ на множестве $States(A, N)$ (в соответствии с /2/ $States(A, N) = N.nodes \to Q$ /33/) определим отображение $step_{A,N,s} : States(A, N) \to States(A, N)$ следующим образом: для любого состояния $g : V \to Q$ $step_{A,N,s}(g)$ это такое отображение $g' : V \to Q$, что*

для любого $x \in (V \setminus s) \cup \{y : y \in V \& functor(y) = y\}$ $g'(x) = g(x)$,

а для любого $x \in s \setminus \{y : y \in V \& functor(y) = y\}$ и для последовательности $\pi \in Q^{arn(functor(x))}$ такой, что $\pi(i) = g(x[i])$, для всех $i \in dom(\pi)$, выполняется $g'(x) = \mu(functor(x))(\pi)$.

Замечание . В частности, для любых $g \in States(A, N)$ и $x \in N.nodes$ $\left\langle {node \atop step} \right\rangle_x(g) = step_{A,N,\{x\}}(g)$ (/3,40/).

Определение 41. *Пусть $N = (V, R, F, functor)$ – функциональная сеть (/33/) над алгеброй $A = (Q, F, \mu)$ (/25,39/). Определим отображение $\left\langle {effective \atop Plans} \right\rangle_{A,N} : States(A, N) \to 2^{\left\langle {Computer \atop Plans} \right\rangle(N)}$ следующим образом.*

Для любого $f \in States(A, N)$ и $\pi \in \left\langle {Computer \atop Plans} \right\rangle(N)$ $\pi \in \left\langle {effective \atop Plans} \right\rangle_{A,N}(f)$ тогда и только тогда, когда состояние $\left\langle {Computer \atop Result} \right\rangle_{A,N}(f, \pi)$ является настроенным (/4/: Планы настройки из множества $\left\langle {effective \atop Plans} \right\rangle_{A,N}(f)$ будут планами полной настройки для состояния f).

Настроечная последовательность π называется полной, если в ней есть согласованное состояние (если существует $i \in dom(\pi)$, такое, что $\pi(i)$- согласованное состояние). Настроечная последовательность называется избыточной, если в ней присутствует более одного согласованного состояния.

Утверждение 16. *Пусть* $N = (V, R, F, functor)$ *– функциональная сеть* *(/33/)* *над алгеброй* $A = (Q, F, \mu)$ *(/25,39/)* *и* $f \in States(A, N)$. *Тогда вершина* $x \in V$ *настроена в состоянии* f *в том и только в том случае, когда выполняется ровно одно из следующих условий:*

- *Либо* $functor(x) = x$
- *Либо* $functor(x) = g \in F$ *и отображение* $f{:}V \to Q$ *является решением системы (из одного уравнения)* $\{x = \{g\}^\wedge \pi\}$, *где* $\pi \in V^{arn(g)}$ *и для любого* $i \in dom(\pi)$ $\pi(i) = x[i]$. ***Без доказательства***

Следствие 1. *Пусть* N *- функциональная сеть* *(/33/)* *над алгеброй* A *(/40/)*. *Тогда состояние* $f \in States(A, N)$ *(/2/)* *является настроенным* *(/4/)* *в том и только в том случае, когда* $f \in solutions(system(N), A)$. ***Без доказательства***

Определение 42. *Пусть* N *функциональная сеть* *(/33/)* *над алгеброй* $A = (Q, F, \mu)$ *(/40/)*. *Определим отображения*

$\left\langle {}^{Computer}_{Protocol} \right\rangle_{A,N} : States(A, N) \times \left\langle {}^{AllComputer}_{Plans} \right\rangle(N) \to States(A, N)^*$ *и*

$\left\langle {}^{Computer}_{Result} \right\rangle_{A,N} : States(A, N) \times \left\langle {}^{Computer}_{Plans} \right\rangle(N) \to States(A, N)$

следующим образом:

- *Для любых* $f \in States(A, N)$ *и* $\tau \in \left\langle {}^{AllComputer}_{Plans} \right\rangle(N)$ $\left\langle {}^{Computer}_{Protocol} \right\rangle_{A,N}(f, \tau)$ *это такая последовательность* π *элементов из* $States(A, N)$, *которая является последовательностью минимальной длины, из всех последовательностей, удовлетворяющих следующим условиям:*
  - $1 \in dom(\pi)$ *и* $\pi(1) = f$;
  - *если* $i \in dom(\tau) \cap dom(\pi)$, *то* $i + 1 \in dom(\pi)$ *и при этом* $\pi(i + 1) = step_{A,N,\tau(i)}(\pi(i))$ *(/40/)*.
- *Для любого* $\tau \in \left\langle {}^{Computer}_{Plans} \right\rangle(N)$ $\left\langle {}^{Computer}_{Result} \right\rangle_{A,N}(f, \tau) \overset{def}{=} last(\left\langle {}^{Computer}_{Protocol} \right\rangle(f, \tau))$ *(/1.4/)*.

Последовательность состояний $\pi \in (\{1, 2, \ldots\} \to States(A, N)) \cup (States(A, N)^*)$ *(/40/)* *(конечная, при* $\pi \in States(A, N)^*$, *либо бесконечная)* *называется настроечной последовательностью, если существует план настройки*

$\tau \in \langle^{AllComputer}_{Plans}\rangle(N)$ *такой, что* $\pi = \langle^{Computer}_{Protocol}\rangle(\pi(1), \tau)$.

Определение 43. *Пусть* N *сеть (/33/) над алгеброй* A *(/40/). Множеством надежных планов настройки сети* N *в алгебре* A *называется такое множество* $\langle^{SaFe}_{Plans}\rangle_A(N) \subseteq \langle^{Computer}_{Plans}\rangle(N)$, *что для любых* $\pi \in \langle^{SaFe}_{Plans}\rangle_A(N)$ *и* $f \in States(A, N)$ $\pi \in \langle^{effective}_{Plans}\rangle_{A,N}(f)$.

Утверждение 17. *Пусть* $G = (V, R, F, functor)$ *– функциональная сеть (/33/),* $A = (Q, F, \mu)$ *– алгебра (/25/) и* $s \subseteq V$. *Тогда для любого состояния* $f \in States(A, G)$ *выполняется тождество* $(step_{A,G,s}(f))^{Terms(F\cup Q)} \equiv {}_A f^{Terms(F\cup Q)}(iteration_{N,s})$ *(/19/) (подстановка* $iteration_{N,s}$ *определена в /37/).*

Доказательство. Пусть $x{\in}V$ и $\varphi{=}f^{Terms(F\cup Q)}$(/19/: $\varphi \in substitutions(F \cup Q, V)$). Достаточно доказать, что $step_{A,G,s}(f)(x){=}value_A(\varphi(iteration_{N,s})(x))$ (/29,40,19,37/). Возможны следующие варианты: $x{\notin}s$ и $x{\in}s$.

Пусть $x \notin s$, либо $functor(x) = x$. Тогда $step_{A,G,s}(f)(x) = f(x)$ (/40/) (по определению /40/ $step_{A,G,s}$). В то же время, поскольку $x \notin kernel(\varphi)$, то $\varphi(iteration_{N,s})(x) = \varphi(iteration_{N,s}(x)) = \varphi(x)$ (/37/),
а значит, $\varphi(iteration_{N,s})(x) = f^{Terms(F\cup Q)}(x) = f(x)$.

Пусть $x{\in}\ s$, $functor(x){=}g$ и $g{\neq}x$.

Возможны следующие варианты

1) $arn(g) = 0$,

2) $arn(g) = n > 0$

В первом случае $\varphi(iteration_s(x)) = \varphi(g) = g$ (/37/), а значит $value_A(\varphi(iteration_{G,s}))(x) = \mu(g)(\varepsilon)$ (/29,37/). В то же время $step_{A,G,s}(f)(x) = \mu(g)(\varepsilon)$ (/40/), по определению $step_{A,G,s}$ (/40/).

Во втором случае $iteration_{G,s}(x){=}g(x[1],\ldots, x[n])$ (/37,34/), а значит
$value_A(\varphi(iteration_{N,s}))(x) =$ (/29,37/) $value_A(\varphi(iteration_{G,s}(x))) = Interpretation_A(g(\varphi(x[1]),\ldots, \varphi(x[n]))) =$ (/27,37,34/) $\mu(g)(Interpretation_A(\varphi(x[1])), \ldots, Interpretation_A(\varphi(x[n]))) = step_{A,G,s}(f)(x)$ (/27, 34, 40/). **Конец доказательства**

Следствие 1. *Пусть* $N = (V, R, F, functor)$ *– функциональная сеть (/33/) над алгеброй* $A = (Q, F, \mu)$ *(/25,40/) и* $s \subseteq V$. *Тогда для любого состояния* $f \in States(A, N)$ $step_{A,N,s}(f) = \langle {subs \atop Value} \rangle_A (f^{Terms(F \cup Q)}(iteration_{N,s}))$ *(/29,19/).* ***Без доказательства***

Теорема 2. *Пусть* $N = (V, R, F, functor)$ *– функциональная сеть (/33/) над алгеброй* $A = (Q, F, \mu)$ *(/25,40/) и* $\pi \in \langle {Computer \atop Plans} \rangle (N)$. *Тогда для любого состояния* $f \in States(A, N)$ *(/40/)* $(\langle {Computer \atop Result} \rangle_{A,N}(f, \pi))^{Terms(F \cup Q)} \equiv {}_A f^{Terms(F \cup Q)}(\langle {Computer \atop Sub} \rangle_N(\pi))$ *(/19, 42, 30, 38/)*.

Доказательство. Проведем индукцию по длине π.
Базис: $\pi = \varepsilon$ (/1.4/). В этом случае утверждение верно, поскольку $\langle {Computer \atop Result} \rangle_{A,N}(f, \pi) = f$ (/42/), а $\langle {Computer \atop Sub} \rangle_N(\pi)$ (/38/) – тождественная подстановка.
Шаг индукции. Пусть $\pi = \pi(1)^\wedge \pi'$ (/1.4/) и пусть $\alpha = \langle {Computer \atop Protocol} \rangle_{A,N}(f, \pi)$ (/42/). Поскольку $\pi \neq \varepsilon$, то $length(\alpha) > 1$ (/1.4/).

По предположению индукции $(\langle {Computer \atop Result} \rangle_{A,N}(\alpha(2), \pi'))^{Terms(F \cup Q)} \equiv_A \alpha(2)^{Terms(F \cup Q)}(\langle {Computer \atop Sub} \rangle_N(\pi'))$ (/19,42, 30, 38/).

При этом, поскольку $\langle {Computer \atop Result} \rangle_{A,N}(\alpha(2), \pi') = \langle {Computer \atop Result} \rangle_{A,N}(\alpha(1), \pi) = \langle {Computer \atop Result} \rangle_{A,N}(f, \pi)$ (/42/), то $(\langle {Computer \atop Result} \rangle_{A,N}(f, \pi))^{Terms(F \cup Q)} \equiv_A (\alpha(2)^{Terms(F \cup Q)})(\langle {Computer \atop Sub} \rangle_N(\pi'))$ (/19,42, 30, 38/).

Таким образом, достаточно доказать соотношение $\alpha(2)^{Terms(F \cup Q)}(\langle {Computer \atop Sub} \rangle_N(\pi')) \equiv_A (f^{Terms(F \cup Q)})(\langle {Computer \atop Sub} \rangle_N(\pi))$ (/19, 38, 30/).

Действительно, в силу утверждения 17 $\alpha(2)^{Terms(F \cup Q)} \equiv_A f^{Terms(F \cup Q)}(iteration_{N,\pi(1)})$ (/19, 30, 37/).

Таким образом, $\alpha(2)^{Terms(F \cup Q)}(\langle {Computer \atop Sub} \rangle_N(\pi')) \equiv_A f^{Terms(F \cup Q)}(iteration_{N,\pi(1)})\langle {Computer \atop Sub} \rangle_N(\pi')$.

Но по определению /38/ $\langle {Computer \atop Sub} \rangle_N(iteration_{N,\pi(1)})\langle {Computer \atop Sub} \rangle_N(\pi') = \langle {Computer \atop Sub} \rangle_N(\pi)$ (/37,38/).
Конец доказательства

Следствие 1. *Пусть* $G = (V, R, F, functor)$ *– функциональная сеть (/33/),* $A = (Q, F, \mu)$ *– алгебра (/25/),* $f \in States(A, G)$ *- состояние сети (/40/) и* $\pi \in \left\langle{}^{Computer}_{Plans}\right\rangle(N)$ *план вычислений. Тогда* $\left\langle{}^{Computer}_{Result}\right\rangle_{A,N}(f, \pi) = \left\langle{}^{subs}_{Value}\right\rangle_A(f^{Terms(F \cup Q)}\left\langle{}^{Computer}_{Sub}\right\rangle_N(\pi))$. ***Без доказательства*** *(переформулировка теоремы 2).*

Следствие 2. *Пусть* $N = (V, R, F, functor)$ *– сеть (/33/),* $A = (Q, F, \mu)$ *– алгебра (/25/) и* $\pi \in \left\langle{}^{Computer}_{Plans}\right\rangle(N)$ *(/36/). Тогда следующие утверждения равносильны друг другу.*

1) $\pi \in \left\langle{}^{SaFe}_{Plans}\right\rangle_A(N)$ *(/43/)*

2) *Отображение* $\left\langle{}^{Computer}_{Sub}\right\rangle_N(\pi)|_V$ *(/38,18/) является свободным (а значит полным) решением (/32/)* $system(N)$ *(/35/) над* A

3) *Для любой подстановки* $\varphi \in \left\langle{}^{NetWork}_{Subs}\right\rangle(N)$ *(/38/)* $\left\langle{}^{Computer}_{Sub}\right\rangle_N(\pi)\varphi \equiv_A \left\langle{}^{Computer}_{Sub}\right\rangle_N(\pi)$. ***Без доказательства***

3. Обоснование стратегии Киссинджера

В следующем определении вводится обозначение $t^{x,n}$для терма (/16/) $\{x \to t\}^n(x)$ (в соответствии с /1.3.3|$_{\text{Обозначения}}$/ $\varphi^n = \varphi^{[n]}$ является обозначением для подстановки $\underbrace{\varphi\varphi\ldots\varphi}_{n \text{ раз}}$).

Определение 44. *В дальнейшем запись* $t^{x,n}$*, где* $x \in V$*– переменная,* $t \in Terms(F, V)$ *– терм (/16/), а* $n \in \{0, 1, \ldots\}$*, будет использоваться для обозначения терма, представляемого записью* $\{x \to t\}^n(x)$ *(/18, 1.3.3|*$_{\text{Обозначения}}$*,18|*$_{\text{Обозначения}}$*/:* $\{x \to t\}$ *– подстановка, переводящая* x *в* t *и оставляющая остальные переменные на месте).*

Примеры.

1) Если $x \notin vars(t)$ (/17/ если терм t не содержит переменной x), то при всех $n > 0$ $t^{x,n} = t$ (/44/).

2) Если $t{=}x$ (если терм является переменной), то при всех $n \geqslant 0$ $t^{x,n}{=}x^{x,n}{=}x$ (/44/).

Определение 45. *Пусть* $A = (Q, F, \mu)$ *конечная алгебра* (/25/). *Будем говорить,* A *– комбинаторная алгебра, если для любого терма* $t \in Terms(F, \{x\})$*, где* $x \notin F \cup Q$ *и любого* $q \in Q$ *существует такое* $n \in \{1, 2, \ldots\}$*, что* $\{x \to q\}\ t^{x,n} \equiv_A \{x \to q\}\ t^{x,n+1}$.

В следующем определении (фактически оно повторяет определение 6 стратегии Киссинджера) вводятся термины и обозначения, используемые при доказательстве.

Определение 46. *Пусть* $N = (V, R, F, functor)$ *конечная функциональная сеть* (/33/) *над алгеброй* A (/40/) *и* $<< \in \langle{}^{Line}_{Orders}\rangle(V)$ *линейный порядок* (/1.3.2/) *на множестве вершин* V.

Для любого не настроенного состояния $q \in States(A, N)$ (/4,40/) *главной не настроенной вершиной (обозначается* $\langle{}^{chief}_{node}\rangle_{N,\,A}(<<, q)$*) будем называть наименьшую по порядку* $<<$ *не настроенную вершину из множества* V (/4/). *В тех случаях, когда* N *и* A *ясны из контекста будем пользоваться обозначением* $\langle{}^{chief}_{node}\rangle(<<, f)$ *вместо* $\langle{}^{chief}_{node}\rangle_{N,A}(<<, f)$.

Пусть $f \in States(A, N)$. *План вычислений* $\pi \in \langle{}^{AllComputer}_{Plans}\rangle(N)$ *называется* $<<, f$*-планом Киссинджера, если* π *является такой последовательностью минимальной длины из множества* $States(A, N)^* \cup (\{1, 2, \ldots\} \to States(A, N))$*, что выполняются следующие условия*

- *если* f *не настроенное состояние, то* $1 \in dom(\pi)$ *и* $\pi(1) = \{\langle{}^{chief}_{node}\rangle(<<, f)\}$ (/1.3.3,1.4/);
- *если* $i \in dom(\pi)$ *и* $g = \langle{}^{Computer}_{Result}\rangle_{A,N}(f, \pi|_{\{1,\ldots,i\}})$ *не настроенное состояние* (/4/), *то* $i+1 \in dom(\pi)$ (/1.3.3/) *и* $\pi(i+1) = \{\langle{}^{chief}_{node}\rangle(<<, g)\}$ (/40,46/)

Если последовательность $\pi \in \langle{}^{AllComputer}_{Plans}\rangle(N)$ *является* $<<, f$*-планом Киссинджера, то последовательность состояний* $\langle{}^{Computer}_{Protocol}\rangle_{A,N}(f, \pi)$ (/42/) *называется* $(f, <<)$*- настройкой сети* N.

3.1. РЕГУЛЯРНАЯ УСТОЙЧИВОСТЬ КОМБИНАТОРНЫХ АЛГЕБР

Ближайшая цель – доказать утверждение 25.следствие 2 о том, что любая конечная комбинаторная алгебра (/45/) регулярно устойчива (доказательство утверждения 26, из которого следует обратное утверждение, существенно проще и не требует аппарата, излагаемого ниже).

На рис 6 представлена схема доказательства утверждения 25.следствие 2.

(в этой схеме не указаны ссылки на используемые утверждения и определения из других разделов, в частности основным инструментом доказательства является теорема 2 и ее следствия).

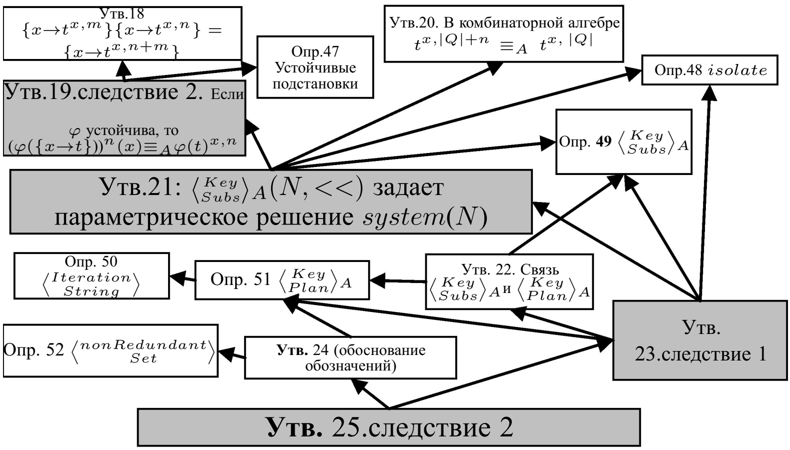


Рис. 6. Схема доказательства

Доказательство 25.следствие 2 проводится в два этапа:

На первом этапе (который заканчивается утверждением 21)

1. В /49/ на вершинах сети (вершины являются переменными соответствующей сети N системы уравнений $system(N)$) определяется специальная подстановка $\langle{Key \atop Subs}\rangle_A(N,<<)$ (в /49/ эта подстановка называется ключом к решению), которая практически не зависит от дуг исходной сети N.

2. В утв. 21 доказывается, что подстановка $\langle{Key \atop Subs}\rangle_A(N,<<)$ задаёт параметрическое решение системы уравнений $system(N)$ (хотя для рассматриваемой задачи полнота решения не нужна получаемое решения оказывается даже полным).

Единственным трудным промежуточным утверждением в приводимом далее доказательстве утверждения 21, является интуитивно очевидное утверждение 19.следствие 2. Все предшествующие 19.следствие 2 утверждения служат как промежуточный материал для доказательства 19.следствие 2.

На втором этапе (заканчивается утверждением 25.следствие 2)

Доказывается, что при настройках сетей над конечными комбинаторными алгебрами, настройка по стратегии Киссинджера приводит ровно к тому же результату, что и настройка в соответствии с тем планом настройки, который соответствует подстановке $\left\langle {Key \atop Subs} \right\rangle_A (N, <<)$ (/49/).

Этот план, обозначаемый $\left\langle {Key \atop Plan} \right\rangle_A (N, <<)$ явным образом описан в определении /51/ (и устроен так, что $\left\langle {Key \atop Subs} \right\rangle_A (N, <<) = \left\langle {Computer \atop Sub} \right\rangle_N (\left\langle {Key \atop Plan} \right\rangle_A (N, <<))$).

Поэтому, поскольку $\left\langle {Key \atop Subs} \right\rangle_A (N, <<)$ задает параметрическое решение план $\left\langle {Key \atop Plan} \right\rangle_A (N, <<)$ всегда приводит к согласованному состоянию (об этом лишний раз напоминается в тривиальном утверждении 22.следствие 1).

Идея доказательства сводится к следующему:
Если из плана настройки $\left\langle {Key \atop Plan} \right\rangle_A (N, <<)$ (/51/) сети находящейся в начальном состоянии f выбросить все «бесполезные шаги» (шаги, не изменяющие текущих состояний), то получится $<<, f$-план Киссинджера (/46/).

«Бесполезные шаги» определены в /52/: это те номера шагов, которые не входят во множество $\left\langle {nonRedundant \atop Set} \right\rangle_{A,N,<<} (f)$ (где f – начальное состояние сети N над алгеброй A)

Таким образом, $<<, f$-план Киссинджера (/46/) является подпоследовательностью плана $\left\langle {Key \atop Plan} \right\rangle_A (N, <<)$ (/51/), а протокол настройки Киссинджера (f,<<)- настройка (/46/)) является подпоследовательностью последовательности $\left\langle {Computer \atop Protocol} \right\rangle (f, \left\langle {Key \atop Plan} \right\rangle_A (N, <<))$.

Доказательство основано на утверждении 23.следствие 1 (все остальные утверждения безыдейны и очевидны, а их доказательства не следует рассматривать иначе как проверку корректности используемых определений)

Первый этап При обосновании будут проводиться некоторые формальные выкладки с композициями подстановок вида $\{x{\to}t\}$, $\{x{\to}t^{x,n}\}$, $\{x{\to}t\}^n$ и т.д. Основной принцип доказательства равенств подобных композиций состоит в следующем. Если φ и φ' две подстановки такие, что $kernel(\varphi){\subseteq}\{x\}$ и $kernel(\varphi'){\subseteq}\{x\}$, то $\varphi = \varphi'$ ($\varphi{\equiv_A}\varphi'$) тогда и только тогда, когда $\varphi(x) = \varphi'(x)$ ($\varphi(x) \equiv_A\varphi'(x)$). Более общая форма этого приема лежит в определении подстановок и сформулирована в утверждении 8 (определении /30/): две подстановки φ, φ' равны (эквивалентны) тогда и только тогда, когда для любой переменной x равны (эквивалентны) значения $\varphi(x)$ и $\varphi'(x)$.

Утверждение 18. *Пусть* $t{\in}Terms(F,V)$ *и* $x{\in}V$. *Тогда для любых* $n, m{\in}\{0, 1, \ldots\}$ $\{x{\to}t^{x,n}\}{=}\{x{\to}t\}^n$ *(/18|*$_{\text{Обозначения}}$*,44/) и* $\{x{\to}t^{x,m}\}\{x{\to}t^{x,n}\}{=}\{x{\to}t^{x,n+m}\}$.

Доказательство. Докажем равенство $\{x{\to}t^{x,n}\}{=}\{x{\to}t\}^n$ (/18|$_{\text{Обозначения}}$,44/): поскольку x единственная переменная которая может измениться в этих подстановках, то (утв. /8/) достаточно доказать, что $\{x{\to}t^{x,n}\}(x){=}\{x{\to}t\}^n(x)$.

$\{x{\to}t^{x,n}\}(x){=}t^{x,n}$ (поскольку $\{x{\to}\tau\}$ (/18|$_{\text{Обозначения}}$/) обозначение такой подстановки φ, что $\varphi(x){=}\tau$)

$\{x{\to}t\}^n(x){=}t^{x,n}$ (по определению /44/ $t^{x,n}$). **Конец доказательства**.

Определение 47. *Пусть* $A = (Q, F, \mu)$ *– алгебра (/25/) и* $\varphi \in substitutions(F \cup Q, V)$ *(/18/) – подстановка. Будем говорить, что подстановка* $\varphi \in substitutions(F, V)$ *(/18/) устойчива над* A*, если справедливо тождество* $\varphi \equiv {}_A\varphi^2$ *(/30/, если* $\varphi \equiv {}_A\varphi(\varphi)$*).*

Устойчивые подстановки (/47/) будут играть ключевую роль при доказательстве утверждения 25.следствие 2.

Утверждение 19. *Пусть* $A = (Q, F, \mu)$ *– алгебра (/25/) и* $\varphi \in substitutions(F{\cup}Q, V)$ *(/18/) – устойчивая в* A *подстановка (/47/). Тогда для любых* $x \in V \setminus kernel(\varphi)$ *(/18/) и* $t \in Terms(F, V)$ *(/16/)* $\varphi\{x \to t\}\varphi \equiv {}_A\{x \to \varphi(t)\}\varphi$ *(/18,18|*$_{\text{Обозначения}}$*,30/).*

Доказательство. Поскольку φ - устойчивая подстановка, то доказываемое тождество эквивалентно тождеству
$\varphi\{x\to t\}\varphi\equiv_A\{x\to\varphi(t)\}\varphi\varphi$

В силу определения /30/ отношения $\equiv_A$, достаточно доказать, что для любой переменной $y\in V$
$(\varphi\{x\to t\})\varphi(y)\equiv_A(\{x\to\varphi(t)\}\varphi)\varphi(y)$ (/30/). Пусть $\tau=\varphi(y)$

Докажем равенство $(\varphi\{x\to t\})(\tau)=\{x\to\varphi(t)\}\varphi(\tau)$.
Согласно утверждению 8.следствие 1, для доказательства равенства достаточно для каждой переменной $z\in vars(\tau)$ доказать равенство $(\varphi\{x\to t\})(z)=\{x\to\varphi(t)\}\varphi(z)$.

Пусть $z\in vars(\tau)$. Возможны варианты 1) $x\neq z$ и 2)$x=z$.
1.Если $z\neq x$, то $(\varphi\{x\to t\})(z)=\varphi(z)$ и $\{x\to\varphi(t)\}\varphi(z)=\varphi(z)$.
2.При $x=z$ $(\varphi\{x\to t\})(z)=(\varphi\{x\to t\})(x)=\varphi(t)$ и
$\{x\to\varphi(t)\}\varphi(z)=\{x\to\varphi(t)\}\varphi(x)=\{x\to\varphi(t)\}(x)=\varphi(t)$. **Конец доказательства**

Следствие 1. *Пусть $A=(Q,F,\mu)$ – алгебра (/25/) и $\varphi\in substitutions(F\cup Q,V)$ (/18/) – устойчивая в A подстановка (/47/). Тогда для любых $x\in V\setminus kernel(\varphi)$ (/18/), $t\in Terms(F,V)$ (/16/) и $n\in\{0,1,\ldots\}$ выполняются тождества $\{x\to\varphi(t)^{x,n}\}(\varphi)\equiv_A(\varphi(\{x\to t\}))^n\varphi\equiv_A\{x\to\varphi(t)^{x,n}\}\varphi$ (/18,44,18|Обозначения,1.3.3|Обозначения/).*

Доказательство.
Проведем индукцию по n :
Базис тривиален: $(\varphi(\{x\to t\}))^0\varphi=\varphi$ и
$\{x\to\varphi(t)^{x,0}\}\varphi=\{x\to x\}\varphi=\varphi$.
Шаг индукции. Пусть $n>0$, тогда $(\varphi(\{x\to t\}))^n\varphi=$
$(\varphi(\{x\to t\}))^{n-1}\varphi\{x\to t\}\varphi\equiv_A$/*индуктивное предположение*/
$\{x\to\varphi(t)^{x,n-1}\}\varphi\{x\to t\}\varphi\equiv_A$/*утв. 19*/
$\{x\to\varphi(t)^{x,n-1}\}\{x\to\varphi(t)\}\varphi=$ /* утв. 18*/$\{x\to\varphi(t)^{x,n}\}\varphi$. **Конец доказательства**

Следствие 2. *Пусть $A=(Q,F,\mu)$ – алгебра (/25/) и $\varphi\in substitutions(F\cup Q,V)$ (/18/) – устойчивая в A подстановка (/47/). Тогда для любых $x\in V\setminus kernel(\varphi)$ (/18/), $t\in Terms(F,V)$ (/16/) и $n\in\{0,1,\ldots\}$ выполняется тождество $(\varphi(\{x\to t\}))^n(x)\equiv_A\varphi(t)^{x,n}$ (/44,18|Обозначения,1.3.3|Обозначения/).*

Доказательство.
$(\varphi(\{x{\to}t\}))^n(x){=}$/*$\varphi(x) = x$*/$(\varphi(\{x{\to}t\}))^n\varphi(x){\equiv_A}$/*утв.19.следствие 1*/
$\{x{\to}\varphi(t)^{x,n}\}\varphi(x){=}$/*$\varphi(x) = x$*/
$\{x{\to}\varphi(t)^{x,n}\}(x){=}\varphi(t)^{x,n}$. **Конец доказательства**

Утверждение 20. *Пусть $A = (Q, F, \mu)$ – конечная комбинаторная алгебра (/45/). Тогда для любых $t{\in}Terms(F, V)$ (/16/), $x{\in}V$ и $n{\in}\{0, 1, \ldots\}$ $t^{x,|Q|+n} \equiv_A t^{x,|Q|}$ (/44,30/).* ***Без доказательства***

Определение 48. *Пусть F – множество функциональных символов (/15/) и W – множество не пересекающееся с F. Определим отображение*
$isolate{:}netwares(W, F){\times}2^W \to netwares(W, F)$ (/33/) следующим образом. Для любой сети
$N = (V, R, F, functor) \in netwares(W, F)$ (/33/) и любого множества $s{\subseteq}W$ $isolate(N, s) = (V, R', F, functor')$ это такая сеть, что выполняются следующие условия:

- $R' = R{\cap}((W{\setminus}s){\times}\{1, 2, \ldots\}{\times}(W{\setminus}s))$
- *для любого $x{\in}V$*

$$functor'(x) = \begin{cases} x, & \text{если } x{\in}s \\ functor(x), & \text{если } x{\notin}s \end{cases}.$$

В дальнейшем вместо записи $isolate(N, s)$ будем использовать запись $N.isolate(s)$. При этом будем говорить, что сеть $N.isolate(s)$ получается из сети N изолированием вершин множества s.

Определение 49. *Пусть $A = (Q, F, \mu)$ – конечная алгебра (/25/), $N = (V{,}R{,}F{,}functor)$- конечная сеть (/33/) и $<<$ - линейный порядок на V. Ключом к решению для тройки $(A, N, <<)$ назовем - подстановку $\langle{}^{Key}_{Subs}\rangle_A(N, <<) \in substitutions(F, V)$, которая определяется индукцией по мощности $\langle{}^{bound}_{Vars}\rangle(system(N))$ следующим образом:*

- *Если $|\langle{}^{bound}_{Vars}\rangle(system(N))| = 0$ (если $system(N){=}\emptyset$), то*

$\langle{}^{Key}_{Subs}\rangle_A(N, <<) \overset{def}{=} (=_V)^{Terms}$ *- тождественная подстанов-*

ка (kernel($\langle{}^{Key}_{Subs}\rangle_A(N,<<)$)=$\emptyset$)

- *Если* $|\langle{}^{bound}_{Vars}\rangle(system(N))| > 0$, *то*

$\langle{}^{Key}_{Subs}\rangle_A(N,<<) \stackrel{def}{=} (\varphi(iteration_{N,\{x\}}))^{|Q|}\varphi$, *где* $x{=}max_{<<}(\langle{}^{bound}_{Vars}\rangle(system(N)))$ *и* $\varphi{=}\langle{}^{Key}_{Subs}\rangle_A(N.isolate(\{x\}),<<)$.

Утверждение 21. *Пусть $A = (Q,F,\mu)$ – комбинаторная алгебра (/45/), $N = (V,R,F,functor)$- конечная сеть (/33/) и $<<$ - линейный порядок на V.*
Тогда $\langle{}^{Key}_{Subs}\rangle_A(N,<<)|_V$ свободное решение /32/ $system(N)$ над A.

Доказательство. Проведем индукцией по $|\langle{}^{bound}_{Vars}\rangle(system(N))|$.
Базис: (случай $\langle{}^{bound}_{Vars}\rangle(system(N)){=}\emptyset$) тривиален (все элементы $States(A,N)$ являются решениями и тождественная подстановка принадлежит $\langle{}^{NetWork}_{Subs}\rangle(N){=}\ \langle{}^{system}_{Subs}\rangle(system(N))$ (/24/ и утв. 15)).
Индуктивный шаг. Пусть $\langle{}^{bound}_{Vars}\rangle(system(N)){\neq}\emptyset$. Положим $x \stackrel{def}{=} max_{<<}(\langle{}^{bound}_{Vars}\rangle(system(N)))$, $N' \stackrel{def}{=} N.isolate(\{x\})$, $\theta_x \stackrel{def}{=} iteration_{N,\{x\}}$, $\tau \stackrel{def}{=} \theta_x(x)$ и $\varphi \stackrel{def}{=} \langle{}^{Key}_{Subs}\rangle_A(N',<<)$.
Тогда $system(N'){=}system(N)\backslash\{x{=}\tau\}$ и $\{x{=}\tau\}{\in}system(N)$.

Поскольку $\varphi{\in}\langle{}^{NetWork}_{Subs}\rangle(N)$ (по индуктивному предположению) и $iteration_{N,\{x\}}{\in}\langle{}^{NetWork}_{Subs}\rangle(N)$ (/38,24/), то $\langle{}^{Key}_{Subs}\rangle_A(N,<<){\in}\langle{}^{system}_{Subs}\rangle(system(N))$ (/24, 38/ и утв.15). Следовательно, для доказательства утверждения достаточно убедиться в том, что $\langle{}^{Key}_{Subs}\rangle_A(N,<<)|_V = ((\varphi\ \theta_x)^{|Q|}\varphi)|_V$ является параметрическим решением (/32/) $system(N)$ в A. Для этого достаточно доказать, что для любого уравнения $y{=}t{\in}system(N)$ $(\varphi\ \theta_x)^{|Q|}\varphi(y){\equiv}_A(\varphi\ \theta_x)^{|Q|}\varphi(t)$.
Возможны следующие варианты $y{=}x$ и $y{\neq}x$.
Пусть $y{=}x$. Тогда $t{=}\tau$. Необходимо доказать тождество $t_{leftpart} = (\varphi\ \theta_x)^{|Q|}\varphi(x){\equiv}_A(\varphi\ \theta_x)^{|Q|}\varphi(\tau) = t_{rightpart}$.

Поскольку (по индуктивному предположению) $\varphi|_V$ – свободное решение $system(N')$, то (теорема 2.следствие 2 пункты 2 и 3) $\varphi\varphi{\equiv}_A\varphi$, следовательно φ - устойчивая подстановка.

Учитывая устойчивость φ, преобразуем терм $t_{rightpart}$ =

$(\varphi\ \theta_x)^{|Q|}\varphi(\tau)$ следующим образом:
$t_{rightpart}$=$(\varphi\ \theta_x)^{|Q|}\varphi(\tau)$ = /*$\theta_x = \{x{\to}\tau\}$*/
$(\varphi\ \{x{\to}\tau\})^{|Q|}\varphi(\{x{\to}\tau\}(x))$=$((\varphi\ \{x{\to}\tau\})^{|Q|}\varphi\{x{\to}\tau\})(x)$=
$(\varphi\ \{x{\to}\tau\})^{|Q|+1}(x)\equiv_A$/*утв.19.следствие 2*/ $\varphi(t)^{x,|Q|+1}$.
Аналогично
$t_{leftpart} = (\varphi\ \theta_x)^{|Q|}\varphi(x)$=//$\varphi(x) = x$
$(\varphi\ \theta_x)^{|Q|}(x)\equiv_A$ /*утв.19.следствие 2*/ $\varphi(t)^{x,|Q|}$.
Тождество $\varphi(t)^{x,|Q|}\equiv_A\varphi(t)^{x,|Q|+1}$ справедливо в силу утверждения 20.
Пусть $y{\neq}x$. Поскольку $\varphi|_V$ параметрическое решение (по предположению индукции), то $\psi\varphi(y)\equiv_A\psi\varphi(t)$ для любой подстановки $\psi{\in}substitutions(F,V)$ в том числе и для $\psi = (\varphi\ \theta_x)^{|Q|}$. **Конец доказательства**

Второй этап

Определение 50. *Определим отображение* $\langle{}^{Iteration}_{String}\rangle$:$\{1,2,\ldots\}{\times}\{0,1,\ldots\}{\to}\{1,2,\ldots\}^*$ *следующим образом для любых* $n{\in}\{1,2,\ldots\}$ *и* $m{\in}\{0,1,\ldots\}$ $\langle{}^{Iteration}_{String}\rangle(n,m)$=

$$\begin{cases} \varepsilon, & \text{если } m = 0 \\ ((\langle{}^{Iteration}_{String}\rangle(n,m-1)^\wedge\{m\})^{\wedge n})^\wedge\langle{}^{Iteration}_{String}\rangle(n,m-1), & \text{если } m > 0 \end{cases}$$

Если V множество и $f:\{1,...,n\}\to V$ *отображение, то строку* $[*]^f(\langle{}^{Iteration}_{String}\rangle(n,m))$ *(/10/) будем обозначать* $\langle{}^{Iteration}_{Protocol}\rangle_f(n,m)$.

Пример 9. $\langle{}^{Iteration}_{String}\rangle$(2,0)=$\varepsilon$
$\langle{}^{Iteration}_{String}\rangle(2,1) = ((\varepsilon^\wedge\{1\})^{\wedge 2})^\wedge\varepsilon$= $\{1\}^\wedge\{1\}$ (=“11”)
$\langle{}^{Iteration}_{String}\rangle(2,2)$ =$((\langle{}^{Iteration}_{String}\rangle(2,1)^\wedge\{2\})^{\wedge 2})^\wedge\langle{}^{Iteration}_{String}\rangle(2,1)$ =
$((\{1\}^\wedge\{1\}^\wedge\{2\})^{\wedge 2})^\wedge(\{1\}^\wedge\{1\})$ =
$\{1\}^\wedge\{1\}^\wedge\{2\}^\wedge\{1\}^\wedge\{1\}^\wedge\{2\}^\wedge\{1\}^\wedge\{1\}$ (=“11211211”).●

Определение 51. *Пусть* $N = (V,R,F,functor)$*- конечная сеть (/33/),* $A = (Q,F,\mu)$ *– конечная алгебра (/25/),* $<<$ *- линейный порядок на* V*,* $s = \langle{}^{bound}_{Vars}\rangle(system(N))$ *и* f:$\{1,...,|s|\}{\to}V^1$ *отображение такое, что для любого* $i\in dom(f)$ *выполняется равенство* $f(i) = \{{num_{s,<<}}^{-1}(i)\}$.

Ключевым планом для тройки $(A, N, <<)$ назовем последовательность $\langle{}^{Key}_{Plan}\rangle_A(N,<<)\in\langle{}^{Computer}_{Plans}\rangle(N)$, равную последовательности $\langle{}^{Iteration}_{Protocol}\rangle_f(|Q|,|s|)$.

Утверждение 22. *Пусть $N = (V, R, F, functor)$- конечная сеть (/33/), $A = (Q, F, \mu)$ – конечная алгебра (/25/) и $<<$ - линейный порядок на V. Тогда*
$\langle{}^{Key}_{Subs}\rangle_A(N,<<) = \langle{}^{Computer}_{Sub}\rangle_N(\langle{}^{Key}_{Plan}\rangle_A(N,<<))$. ***Без доказательства***

Следствие 1. *Пусть $A = (Q, F, \mu)$ – комбинаторная алгебра (/45/), $N = (V, R, F, functor)$- конечная сеть (/33/) и $<<$ - линейный порядок на V. Тогда для любого состояния $f\in States(A, N)$ состояние $\langle{}^{Computer}_{Result}\rangle_{A,N}(f, \langle{}^{Key}_{Plan}\rangle_A(N,<<))$ является настроенным.*

Доказательство. Пусть $\pi=\langle{}^{Key}_{Plan}\rangle_A(N,<<)$.

Поскольку (утв.22) $\langle{}^{Key}_{Subs}\rangle_A(N,<<)=\langle{}^{Computer}_{Sub}\rangle_N(\pi)$,

то (Теорема 2.следствие 1)
$\langle{}^{Computer}_{Result}\rangle_{A,N}(f,\pi) = \langle{}^{subs}_{Value}\rangle_A(f^{Terms(F\cup Q)}\langle{}^{Key}_{Subs}\rangle_A(N,<<))$.

Поскольку (в силу утв.21) $\langle{}^{Key}_{Subs}\rangle_A(N,<<)|_V$ - параметрическое решение $system(N)$, то (в силу утв.13.следствие 1)
$\langle{}^{subs}_{Value}\rangle_A(f^{Terms(F\cup Q)}\langle{}^{Key}_{Subs}\rangle_A(N,<<))$- решение $system(N)$,
а значит $\langle{}^{Computer}_{Result}\rangle_{A,N}(f,\pi)$– согласованное состояние. **Конец доказательства**

Утверждение 23. *Пусть $n\in\{1, 2, \ldots\}$, $m\in\{0, 1, \ldots\}$ и $\pi=\langle{}^{Iteration}_{String}\rangle(n, m)$. Тогда для любого $i\in dom(\pi)$*
$\langle{}^{Iteration}_{String}\rangle(n, \pi(i) - 1) \in suffixes(Prefix(i - 1, \pi))$.

Доказательство. Для любых $k \in \{1, ..., m\}$ и $\alpha \in \{1, ..., m\}^*$ обозначим множество
$\{i : i \in dom(\alpha)\&\alpha(i) = k\}$ через $\langle{}_{Occurrences}^{Var}\rangle(k, \alpha)$,
множество $\{\alpha(i) : i \in dom(\alpha)\&k \leqslant \alpha(i)\}$ через $\langle{}^{notLess}_{Vals}\rangle(k, \alpha)$,
а строку $Prefix(min(\langle{}_{Occurrences}^{Var}\rangle(k, \alpha)), \alpha)$ через $\langle{}^{first}_{Prefix}\rangle(k, \alpha)$.

По определению /50/ $\langle{}^{Iteration}_{String}\rangle(n, m)$
$\langle{}^{first}_{Prefix}\rangle(k, \pi)=\langle{}^{Iteration}_{String}\rangle(n, k - 1)^\wedge\{k\}$.

Следовательно для доказательства утверждения достаточно доказать, что для любого $i \in \langle {}^{Var}_{Occurrences} \rangle (k, \pi)$ выполняется $\langle {}^{first}_{Prefix} \rangle (k, \pi) \in suffixes(Prefix(i, \pi))$.

Докажем, что для любого $i \in \langle {}^{Var}_{Occurrences} \rangle (k, \pi)$ $\langle {}^{first}_{Prefix} \rangle (k, \pi) \in suffixes(Prefix(i, \pi))$.

Доказательство проведем индукцией мощности множества $\langle {}^{notLess}_{Vals} \rangle (k, Prefix(i, \pi))$.

Базис. $|\langle {}^{notLess}_{Vals} \rangle (k, Prefix(i, \pi))| = 1$. В этом случае $\langle {}^{notLess}_{Vals} \rangle (k, Prefix(i, \pi)) = \{k\}$. Следовательно, (по определению /50/ $\langle {}^{Iteration}_{String} \rangle (n, m)$), для некоторого $n' \in \{1, ..., n\}$ $Prefix(i, \pi) = \langle {}^{first}_{Prefix} \rangle (k, \pi)^{\wedge n'}$. Таким образом, $\langle {}^{first}_{Prefix} \rangle (k, \pi) \in suffixes(Prefix(i, \pi))$.

Шаг индукции. Пусть $|\langle {}^{notLess}_{Vals} \rangle (k, Prefix(i, \pi))| > 1$, $k' = max(\langle {}^{notLess}_{Vals} \rangle (k, Prefix(i, \pi)))$ и $\alpha = \langle {}^{first}_{Prefix} \rangle (k', \pi)$. Тогда $\langle {}^{Var}_{Occurrences} \rangle (k, \alpha) \neq \emptyset$ (по определению /50/ $\langle {}^{Iteration}_{String} \rangle (n, m)$) и для любого $i' \in \langle {}^{Var}_{Occurrences} \rangle (k, \alpha)$ $\langle {}^{first}_{Prefix} \rangle (k) \in suffixes(Prefix(i', \pi))$ (по индуктивному предположению).

Следовательно, для любых $n' \in \{1, 2, \ldots\}$ и $i'' \in \langle {}^{Var}_{Occurrences} \rangle (k, \alpha^{\wedge n'})$ $\langle {}^{first}_{Prefix} \rangle (k) \in suffixes(Prefix(i'', \alpha^{n}))$.

В то же время (по определению /50/ $\langle {}^{Iteration}_{String} \rangle (n, m)$) существует $n'' \in \{1, ..., n+1\}$ такое, что $Prefix(i, \pi) = Prefix(i, \alpha^{\wedge n''})$. **Конец доказательства**

Следствие 1. *Пусть* $A = (Q, F, \mu)$ *– конечная комбинаторная алгебра (/45/),* $N = (V, R, F, functor)$*– конечная сеть (/33/),* $<<$ *- линейный порядок на* V*,* $\pi = \langle {}^{Key}_{Plan} \rangle_A (N, <<)$*,* $i \in dom(\pi)$ *и* $\pi(i) = \{x\}$*. Тогда для любого состояния* $f \in States(A, N)$ *все вершины множества* $V \backslash \{y : x << y\}$ *являются настроенными в состоянии* $\langle {}^{Computer}_{Protocol} \rangle (f, \pi)(i)$*.*

Доказательство. Пусть $N' = N.isolate\{(y : x << y\})$. В силу утверждения 23 $Prefix(i-1, \pi) = \pi'^{\wedge}\pi''$, где $\pi'' = \langle {}^{Key}_{Plan} \rangle_A (N', <<)$, а $\pi' \in \langle {}^{Computer}_{Plans} \rangle (N)$.

Таким образом, для подстановок $\varphi = \langle {}^{Computer}_{Sub} \rangle_N (Prefix(i-1, \pi)) \in \langle {}^{NetWork}_{Subs} \rangle (N)$, $\varphi' \in \langle {}^{Computer}_{Sub} \rangle_N (\pi') \in \langle {}^{NetWork}_{Subs} \rangle (N)$ и

$\varphi'' = \langle{}^{Computer}_{Sub}\rangle_N(\pi'') \in \langle{}^{NetWork}_{Subs}\rangle(N)$ выполняется соотношение (опр /38/ $\langle{}^{Computer}_{Sub}\rangle_N$) $\varphi = \varphi'\varphi''$.

В силу утверждения 22 $\varphi'' = \langle{}^{Key}_{Subs}\rangle_A(N', <<)$, а, следовательно (утв.21), $\varphi''|_V$– параметрическое решение (/32/) $system(N')$ над A. В таком случае (утв. 13.следствие 1) $\langle{}^{subs}_{Value}\rangle_A(f^{Terms(F\cup Q,V)}\varphi'\varphi'')$- решение $system(N')$ над A.

Поскольку (Теорема 2.следствие 1) $\langle{}^{subs}_{Value}\rangle_A(f^{Terms(F\cup Q,V)}\varphi'\varphi'') = \langle{}^{Computer}_{Result}\rangle_{A,N}(f, Prefix(i-1,\pi)) = \pi(i)$, то (утв 16.следствие 1) состояние $\pi(i)$ сети N' настроено, а значит (опр. /4/) настроены все вершины из множества $N.isolate(\{y : x << y\})$. **Конец доказательства**

Определение 52. *Пусть $A = (Q, F, \mu)$ – конечная алгебра (/25/), $N = (V, R, F, functor)$- конечная сеть (/33/), $<<$ - линейный порядок на V и $\pi = \langle{}^{Key}_{Plan}\rangle_A(N, <<)$.*

Определим отображения

$\langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}: States(A, N) \to 2^{\{1,2,\ldots\}}$ и

$\langle{}^{nonRedundant}_{Plan}\rangle_{A,N,<<}: States(A, N) \to \{1, 2, \ldots\}^*$ следующим образом:

Для любого $f \in States(A, N)$

- $\langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f) \overset{def}{=} \{i : i \in dom(\pi) \& \langle{}^{Computer}_{Protocol}\rangle(f, \pi)(i) \neq \langle{}^{Computer}_{Protocol}\rangle(f, \pi)(i+1)\}$
- $\langle{}^{nonRedundant}_{Plan}\rangle_{A,N,<<}(f)$ *это строка которая является монотонным и взаимно-однозначным отображением из множества $\{1, ..., |\langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f)|\}$ на $\langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f)$.*

Утверждение 24. *Пусть $A = (Q, F, \mu)$ – конечная алгебра (/25/), $N = (V, R, F, functor)$ - конечная сеть (/33/), $<<$ - линейный порядок на V, $\tau = \langle{}^{Key}_{Plan}\rangle_A(N, <<)$, $f \in States(A, N)$, $s = \langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f)$ и $\pi = subseq(s, \tau)$ (π -подпоследовательность, протокола τ, порожденная множеством $s \subseteq \tau$). Тогда для любого $i \in \langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f)$ выполняются следующие равенства:*

1) $\pi(num_{s,\leqslant}(i)) = \tau(i)$,
2) $\langle{}^{Computer}_{Protocol}\rangle(f, \pi)(num_{s,\leqslant}(i)) = \langle{}^{Computer}_{Protocol}\rangle(f, \tau)(i)$

Доказательство.

1. Поскольку $num_{s,\leqslant}:s\rightarrow\{1,...,|s|\}$ взаимно-однозначное отображение (утв. 4), то равенство $\pi(num_{s,\leqslant}(i))= \tau(i)$ для всех $i\in s$ равносильно равенству $\pi(j) = \tau({num_{s,\leqslant}}^{-1}(j))$ для всех $j\in dom(\pi)$.

Равенство $\pi(j) = \tau({num_{s,\leqslant}}^{-1}(j))$ следует из утверждения 5: $\pi = subseq(s,\tau) =$ /*утв. 5*/ $string(s\cap dom(\tau),\leqslant,\tau) =$ /*$s \subseteq dom(\tau)$*/$=$ $string(s,\leqslant,\tau)$ $=$/*опр./12/*/ $=$ ${num_{s,\leqslant}}^{-1}\bullet\tau$, где $\bullet$ - операция композиции отображений. Таким образом $\pi(j) = ({num_{s,\leqslant}}^{-1}\bullet\tau)(j) = \tau({num_{s,\leqslant}}^{-1}\ (i))$. (В частности $\pi(1) = \tau(min(s))$).

2.Равенство $\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(num_{s,\leqslant}(i)+1)= \left\langle{Computer \atop Protocol}\right\rangle(f,\tau)(i+1)$ является следствием (по определению /42/ $\left\langle{Computer \atop Protocol}\right\rangle$) равенства $\pi(num_{s,\leqslant}(i)) = \tau(i)$ (это равенство доказано в пункте 1. утв 24). Действительно, пусть $g = \left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(num_{s,\leqslant}(i))= \left\langle{Computer \atop Protocol}\right\rangle(f,\tau)(i)$ и $m = \pi(num_{s,\leqslant}(i))= \tau(i)$. Тогда (по определению /42/) $step_{A,N,m}(g) = \left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(num_{s,\leqslant}(i)+1)= \left\langle{Computer \atop Protocol}\right\rangle(f,\tau)(i+1)$.

Вместе с тем (по определению /52/ $\left\langle{nonRedundant \atop Set}\right\rangle$ и по определению /42/ $\left\langle{Computer \atop Protocol}\right\rangle$) $\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(num_{s,\leqslant}(1))=$ /*опр/52/*/$f=$ /*опр/42/*/ $\left\langle{Computer \atop Protocol}\right\rangle(f,\tau)(1)$. **Конец доказательства**

Утверждение 25. *Пусть $A=(Q,F,\mu)$ – комбинаторная алгебра (/45/), $N = (V,R,F,functor)-$ конечная сеть (/33/), $<<$ - линейный порядок на V и $\pi =\left\langle{Key \atop Plan}\right\rangle_A(N,<<)$. Тогда для любого состояния $f\in States(A,N)$ и любого $i\in\left\langle{nonRedundant \atop Set}\right\rangle_{A,N,<<}(f)$ $\pi(i) = \{\left\langle{chief \atop node}\right\rangle_{N,A}(<<,\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i))\}$.*

Доказательство. Пусть $\pi(i)=\{x\}$. В соответствии с утверждением 23.следствие 1 все вершины множества $V\backslash\{y : x << y\}$ являются настроенными в состоянии $\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i)$. Поскольку $\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i)\neq\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i+1)$ (по определению /52/ $\left\langle{nonRedundant \atop Set}\right\rangle_{A,N,<<}(f)$), то вершина x не настроена в состоянии $\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i)$. Таким образом, x –минимальная по порядку $<<$ вершина не настроенная в состоянии $\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i)$. В таком случае (определение /46/) $x =\left\langle{chief \atop node}\right\rangle_{N,A}(<<,\left\langle{Computer \atop Protocol}\right\rangle(f,\pi)(i))$.

Конец доказательства

Следствие 1. *Пусть* $A = (Q, F, \mu)$ *– конечная комбинаторная алгебра* (/45/), $N = (V, R, F, functor)$*- конечная сеть и* $<<$ *- линейный порядок на* V*. Тогда для любого состояния* $f \in States(A, N)$ *подпоследовательность* $subseq(\langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f), \langle{}^{Key}_{Plan}\rangle_A(N,<<)) \in \langle{}^{Computer}_{Plans}\rangle(N)$ *последовательности* $\langle{}^{Key}_{Plan}\rangle_A(N,<<) \in \langle{}^{Computer}_{Plans}\rangle(N)$ *является* $<<, f$*-планом Киссинджера* (/46/).

Доказательство. Пусть $s = \langle{}^{nonRedundant}_{Set}\rangle_{A,N,<<}(f)$, $\tau = \langle{}^{Key}_{Plan}\rangle_A(N,<<)$ и $\pi = subseq(s, \tau)$.

Возможны следующие варианты $s = \emptyset$ и $s \neq \emptyset$.

Если $s = \emptyset$, то (опр /13/ $subseq(\emptyset, \tau)$) $\pi = \varepsilon$ и (опр /52/ $\langle{}^{nonRedundant}_{Set}\rangle$) $f = \langle{}^{Computer}_{Result}\rangle_{A,N}(f, \pi)$ следовательно (утв 22.следствие 1) f настроенное состояние. Следовательно (в соответствии с определением /46/) $\pi = \varepsilon$ является $<<, f$ -планом Киссинджера (/46/).

Пусть $s \neq \emptyset$. В силу определения /46/ достаточно убедиться в том, что для любого $i \in dom(\pi)$ $\pi(i) = \{\langle{}^{chief}_{node}\rangle_{N,A}(<<, \langle{}^{Computer}_{Protocol}\rangle(f, \pi)(i))\}$.

По определению /13/ $subseq(s, \tau)$ последнее равносильно, тому, что для любого $j \in s$ $\pi(num_{s,\leqslant}(j)) = \{\langle{}^{chief}_{node}\rangle_{N,A}(<<, \langle{}^{Computer}_{Protocol}\rangle(f, \pi)(num_{s,\leqslant}(j)))\}$.
Согласно утв. 24 $\langle{}^{Computer}_{Protocol}\rangle(f, \pi)(num_{s,\leqslant}(j)) = \langle{}^{Computer}_{Protocol}\rangle(f, \tau)(j)$ и $\pi(num_{s,\leqslant}(j)) = \tau(j)$, а согласно утв.25 $\tau(i) = \{\langle{}^{chief}_{node}\rangle_{N,A}(<<, \langle{}^{Computer}_{Protocol}\rangle(f, \tau)(i))\}$. **Конец доказательства**

Следствие 2. *(Достаточность условия комбинаторности в теореме 1) Любая конечная комбинаторная алгебра* (/45/) *является регулярно устойчивой* (/7/). ***Без доказательства***

3.2. УСТОЙЧИВЫЕ И РЕГУЛЯРНО УСТОЙЧИВЫЕ АЛГЕБРЫ

Следующий пример демонстрирует тот факт, что существуют конечные устойчивые алгебры (/7/), которые не являются комбинаторными (/45/).

<u>*Пример* 10.</u> Пусть $F = \{\emptyset, v, a, f\}$ – множество функциональных символов (/15/), для которого $arn(\emptyset) = arn(v) = 0$, $arn(a) = 1$ и $arn(f) = 2$. Пусть $\tau = \{x \to v\}((f(x, a(x)))^{x,2})= f(f(v, a(v)), a(f(v, a(v)))) \in Terms(F, \emptyset)$ (/18,44,16/).

Рассмотрим множество $Q = (\langle {}^{sub}_{terms} \rangle(\tau) \backslash \{\tau\}) \cup \{\emptyset\}$ (/17/: $Q = \{v, a(v), f(v, a(v)), a(f(v, a(v))), \emptyset\}$) и алгебру $A_\tau = (Q, F, \mu)$ (/25/), в которой μ определено следующим образом:

- если $c \in \{\emptyset, v\}$, то $\mu(c)(\varepsilon) = c$,
- для любого $t \in Q$ $\mu(a)(t) = \begin{cases} a(t), & \text{если } a(t) \in Q \\ \emptyset, & \text{если } a(t) \notin Q \end{cases}$;
- для любых $t, t' \in Q$

$$\mu(f)(t, t') = \begin{cases} f(t, t'), & \text{если } f(t, t') \in Q \\ v, & \text{если } f(t, t') = \tau \\ \emptyset, & \text{если } f(t, t') \notin Q \cup \{\tau\} \end{cases}$$

На рис 7 представлено строение этой алгебры в виде таблицы операций (7a) и в виде функциональной сети (7б) (вершина соответствующая элементу алгебры $\emptyset$ на рис. 7б не изображена).

Эта алгебра не является комбинаторной (/45/), поскольку для терма $t = f(x, a(x)) \in Terms(F, \{x\})$ (/16/) выполняется равенство $value_{A_\tau}(\{x \to v\} t^{x,2}) = v$ (/27, 18,44/) и в то же время $value_{A_\tau}(\{x \to v\} t) = f(v, a(v)) \neq v$ (/27, 18/).

Вместе с тем, для любой функциональной сети (/33/) $G = (W, R, \{\emptyset, v, a, f\}, functor)$, для любого состояния $g \in States(A_\tau, G)$ (/40/) и любого линейного порядка $<< \in \langle {}^{Line}_{Orders} \rangle(W)$, такого, что из $functor(x) = f$ и $functor(y) = a$ следует $x << y$ (g,$<<$)- настройка (/46/) является полной (/41/).

Без доказательства ●

Оставшаяся часть раздела посвящена доказательству того, что любая некомбинаторная алгебра (/45/) не является регулярно устойчивой (/7/).

<u>**Определение 53.**</u> *Определим отображение* $EqNet : equations(F, \{x\}) \to netwares(Terms(F, \emptyset), F)$ *(/20,33/) следующим образом. Для любого уравнения* $x = t \in equations(F, \{x\})$

а) $\mu(f)$:

$x_1 \backslash x_2$	v	$a(v)$	$f(v,a(v))$	$a(f(v,a(v)))$	$\emptyset$
v	$\emptyset$	$f(v,a(v))$	$\emptyset$	$\emptyset$	$\emptyset$
$a(v)$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$f(v,a(v))$	$\emptyset$	$\emptyset$	$\emptyset$	v	$\emptyset$
$a(f(v,a(v)))$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

$\mu(a)$:

x	$\mu(a)(x)$
v	$a(v)$
$a(v)$	$\emptyset$
$f(v,a(v))$	$a(f(v,a(v)))$
$a(f(v,a(v)))$	$\emptyset$
$\emptyset$	$\emptyset$

б)

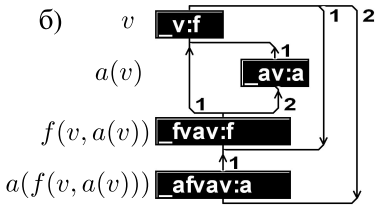


Рис. 7. Устойчивая некомбинаторная алгебра.

(/20/) $EqNet(x{=}t){=}\ (\langle^{sub}_{terms}\rangle(t)\backslash\{x\}, R, F, topFunctor)$ (/17/) такая сеть (/33/), что $(\tau, i, \tau'){\in}R$ тогда и только тогда, когда выполняется одно из следующих условий:

- *либо $Sons(\tau)(i) = x$ и $\tau'{=}t$ (/17/);*
- *либо $Sons(\tau)(i){=}\tau'$ (/17/).*

Утверждение 26. *(Необходимость условия комбинаторности в теореме 1.) Пусть $A = (Q, F, \mu)$ – конечная некомбинаторная алгебра (/45/). Тогда существует такое уравнение $u{\in}equations(F, \{x\})$ (/20/), что для некоторого линейного порядка $<<$ (/1.3.1/) на множестве вершин схемы $EqNet(u)$ (/53/), существует такое состояние $f{\in}States(A, EqNet(u))$ (/40, 53/), что (f,$<<$)- настройка (/46/) схемы $EqNet(u)$ не является полной (/41/).*

Доказательство. Поскольку A – некомбинаторная алгебра (/45/), существуют терм $t \in Terms(F, \{x\})$ (/16/), $q{\in}Q$ и целое

$k > 1$ такие, что $value_A(\{x \to q\}t^{x,k})=q$ и $value_A(\{x \to q\}t)\neq q$ (/27, 18, 44/).

Рассмотрим схему $N = EqNet(x = t)$ (/53/) и состояние $f\in States(A, N)$ (/40/) такое, что для любой вершины $\tau \in \langle{}^{sub}_{terms}\rangle(t)\backslash\{t\}$ (/17/ в соответствии с определением /53/ вершины $EqNet$ являются подтермами t) $f(\tau)=value_A(\{x \to q\}(\tau))$ (/27,18/). Это состояние не согласовано (/4/), и единственной не настроенной вершиной (/4/) является вершина x (поскольку $value_A(\{x\to q\}\ (t))\neq q$ (/27, 18/)).

Рассмотрим такой линейный порядок (/1.3.2/) $<<$ на множестве $\langle{}^{sub}_{terms}\rangle(t)\backslash\{t\}$ (/17/) ($<<\in\langle{}^{Line}_{Orders}\rangle(\langle{}^{sub}_{terms}\rangle(t) \setminus \{t\})$) , в котором вершина x является наибольшей (/1.3.2/). Тогда $(f,<<)$- настройка схемы N (/46/) – бесконечная последовательность (не является полной (/41/)).

Конец доказательства

ABOUT THE TUNING OF THE INTERACTING OBJECTS BEHAVIOR

Andrey Babichev, Institute of Control Sciences of RAS, Moscow, Cand.Sc. (*babichev@ipu.ru*).

Abstract: We investigate the problem of interacting subjects' coordination and consider a procedure, which iteratively tunes the state of one subject to reconcile it with states of its neighbors. Every step results in coordination of one subject, but other subjects may become uncoordinated even if they were tuned before. We derive the conditions when this process stabilizes. We also discuss subjects' characteristics which guarantee tunability of any network of such subjects, and construct a corresponding tuning algorithm.

Keywords: functional flow block diagram, system of equations solution, universal algebra, iterative method..

Статья представлена к публикации членом редакционной коллегии Б. Т. Поляком

.

Поступила в редакцию 22.02.2013.
Опубликована 30.11.2013.