

Логика четырех состояний.

1.Символы:

Логика четырех состояний оперирует четырьмя символами:

1. Т – тезис (ВЕРА, ПРАВДА, ОСЬ X) (соответствует состоянию логическая «ПРАВДА» в формальной логике).
2. А – антитезис (ОТРИЦАНИЕ, ЛОЖЬ, ОСЬ Y) (соответствует состоянию логическая «ЛОЖЬ» в формальной логике).
3. S – синтез (ПОНИМАНИЕ, НЕОПРЕДЕЛЕННОСТЬ, ОСЬ Z) (соответствует такому выражению формальной логики: «ПРАВДА ИЛИ ЛОЖЬ = ПРАВДА И ЛОЖЬ», где «ИЛИ» – операция слабой дизъюнкции. «И» – операция конъюнкции, знак «=» – означает что правая и левая часть выражения тождественно равны. В логике четырех состояний данное выражение является отдельным символом S. При переходе от логики четырех состояний к формальной логике мы будем пользоваться левой частью этого тождества. Т.о. значение этого выражения в формальной логике – «ПРАВДА ИЛИ ЛОЖЬ = ПРАВДА».
4. N – нуль (РЕАЛИЗАЦИЯ, ВНЕ_ПОНИМАНИЯ, ОСЬ ВРЕМЕНИ) (соответствует такому выражению формальной логики: «НЕ (ПРАВДА ИЛИ ЛОЖЬ) = НЕ (ПРАВДА И ЛОЖЬ)», где «НЕ» – операция отрицания. В логике четырех состояний данное выражение является отдельным символом N. При переходе от логики четырех состояний к формальной логике мы также будем пользоваться только левой частью этого тождества. Т.о. значение этого выражения в формальной логике – «НЕ (ПРАВДА ИЛИ ЛОЖЬ) = ЛОЖЬ».

2.Тезисы:

А) Любой объект (логическое выражение) может находиться **только** в одном из 4-ех состояний по отношению к любому другому объекту:

1. данное выражение истинно (Т) (пример: белый цвет)
2. данное выражение ложно (А) (пример: черный цвет)
3. данное выражение неопределенно (S) (пример: серый цвет),
4. данное выражение бессмысленно, т.е. вне понимания (N) (пример: соленый цвет)

Указанные состояния являются дополняющими друг к другу и образуют полную группу состояний.

Б) Символы имеют прямой и обратный порядок следования. Что позволяет их использовать в качестве цифр :

1. Прямой Порядок (синтезирующее направление), (индукция): $N \rightarrow T \rightarrow A \rightarrow S$
пример: мы собираемся купить вещь:
0) Копим деньги и подыскиваем товар в магазине. Это состояние РЕАЛИЗАЦИИ (N)
1) Покупаем вещь, при этом радуемся покупке и тем новым возможностям, которые она предоставляет в наше распоряжение. Это состояние ВЕРЫ (Т).
2) Начинаем все больше замечать, что купленная вещь не отвечает нашим

ожиданиям от нее, т.к. в процессе эксплуатации возникают проблемы.
Это состояние ОТРИЦАНИЯ (А).

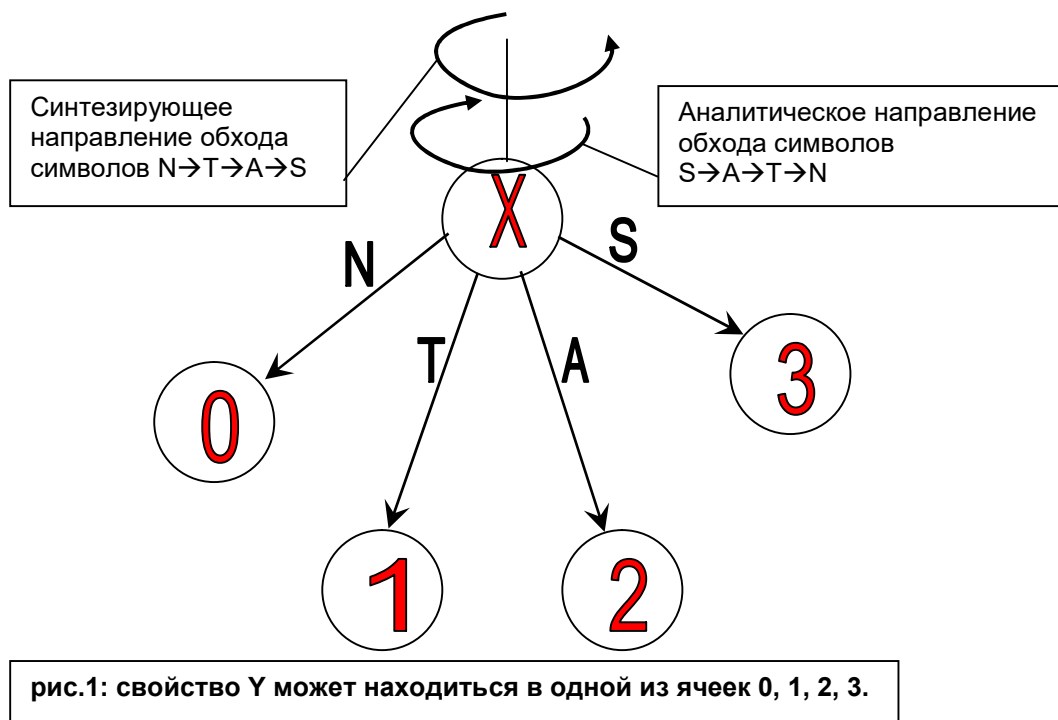
3) Наконец мы приходим к ПОНИМАНИЮ(S) того, какой бы мы хотели видеть вещь. На основе этого ПОНИМАНИЯ переходим к п. 0.

2. Обратный Порядок (аналитическое направление), (дедукция): $S \rightarrow A \rightarrow T \rightarrow N$

Символы соответствуют таким цифрам $N=0$, $T=1$, $A=2$, $S=3$. (Т.о. арифметика логики четырех состояний целиком соответствует арифметике с числами по основанию 4)

3. Сдвоенная бипирамида.

Возьмем некоторый произвольный объект X . И некоторое произвольное свойство (любой другой объект) Y . В соответствии с тезисом А) высказывание «свойство Y присуще объекту X » может находиться в одном из четырех состояний. Отобразим это графически (см. рис 1)



Таким образом, свойство Y попадает в одну из ячеек 0,1,2 либо 3. Обобщая наш пример, можно сказать, что абсолютно все свойства Y_0, Y_1, \dots, Y_n по отношению к объекту X можно распределить по ячейкам 0,1,2,3.

Рассуждая далее – мы можем сказать, что ячейки 0,1,2 и 3 содержат 4 объекта, которые определяют собой объект X в терминах логики четырех состояний.

Раз это так, то по аналогии с объектом X - каждый из объектов 0,1,2 и 3 имеет свои четыре группы свойств N, T, A, S . Причем для каждого из объектов 0,1,2,3 определена уже одна группа свойств через объект X (рис. 2).

В дальнейшем, любые рассматриваемые объекты и их возможные (*неспаренные*) и существующие (*спаренные*) логические связи мы будем называть *логическим построением*. Изображение на рис.2 является логическим построением.

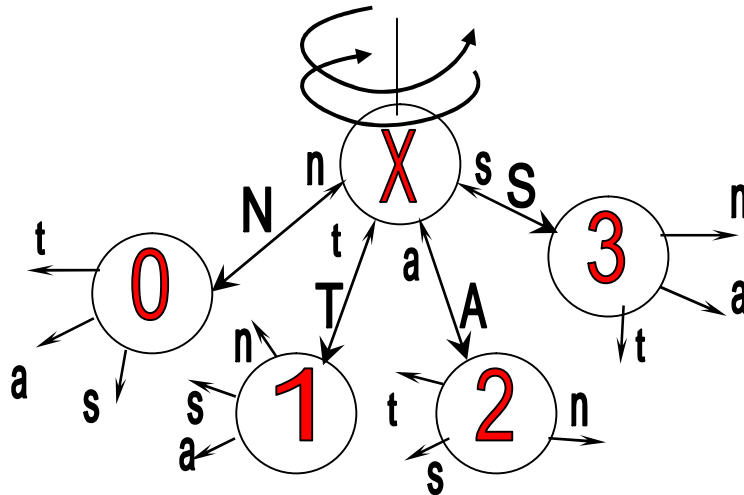


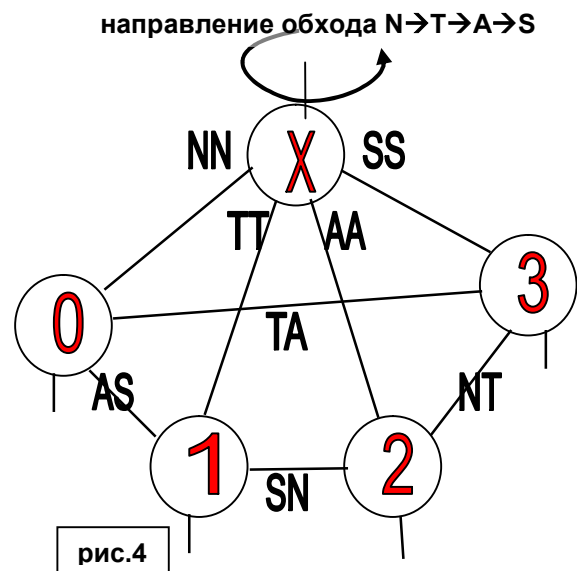
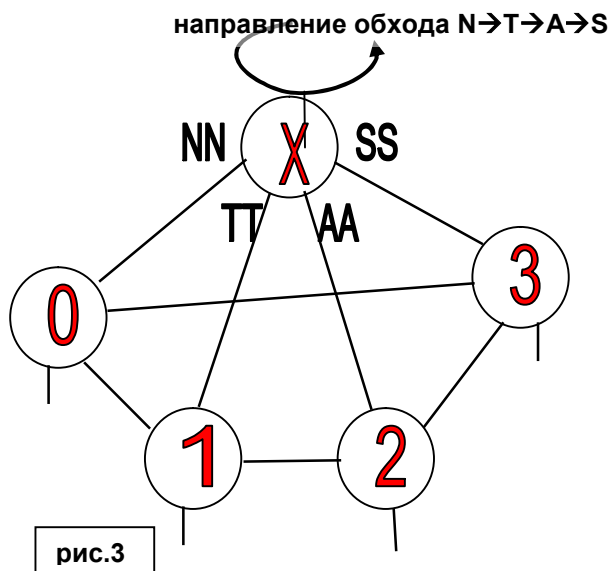
рис.2 Логическое построение

В нашем логическом построении объект X – станем называть – *эталонным объектом*, а объекты 0,1,2,3 – *вспомогательными объектами*. Можно предположить, что группы свойств N,T,A,S вспомогательных объектов связаны не только с эталонным объектом X, но и между собой.

Предлагается следующий алгоритм построения связей между вспомогательными объектами :

Алгоритм бипирамида:

1. Соединим вспомогательные объекты между собой, как показано на рис.3.
2. Зададим порядок обхода символов (например, прямой порядок $N \rightarrow T \rightarrow A \rightarrow S$)
3. Определим значение связи 0—1, 1—2, 2—3 и 3—0 следующим образом:
 - 3.1 для каждой связи вспомогательных объектов в качестве возможных остаются два символа. Укажем их в соответствии с выбранным порядком.
 - 3.2 Например связь 0—1 можно описать символами S или A и только ими (символ N уже использован на построение связи 0—X для объекта 0, аналогично символ T уже использован на построение связи 1—X для объекта 1). Т.к. мы задали прямой порядок $N \rightarrow T \rightarrow A \rightarrow S$, то для связи 0—1 будем записывать последовательность AS (по заданному направлению обхода символов $N \rightarrow T \rightarrow A \rightarrow S$ сначала идет символ A, затем идет символ S)
4. В результате определения остальных связей (1-2, 2-3 и 3-0) получим рис.4



5. Опять рассмотрим объекты 0,1,2 и 3 и установим последовательности символов (*комбинации*) для оставшихся неопределенными связей 0—... , 1—... , 2—... и 3—... :
- 5.1. Для объекта 0, по первым символам всех уже установленных связей: N, T, A – использованы. Свободен S.
- 5.2. Для объекта 0, по вторым символам: N, A, S – использованы. Свободен T.
- 5.3. В итоге Связь 0—... определена, как ST
6. Поступим аналогичным образом для связей: 1—... , 2—... и 3—... .
7. Объединим связи 0—... , 1—... , 2—... и 3—... в объект Z.
8. Объект Z полностью определен в терминах логики четырех состояний и называется *результатирующим объектом*. (Рис. 5а)

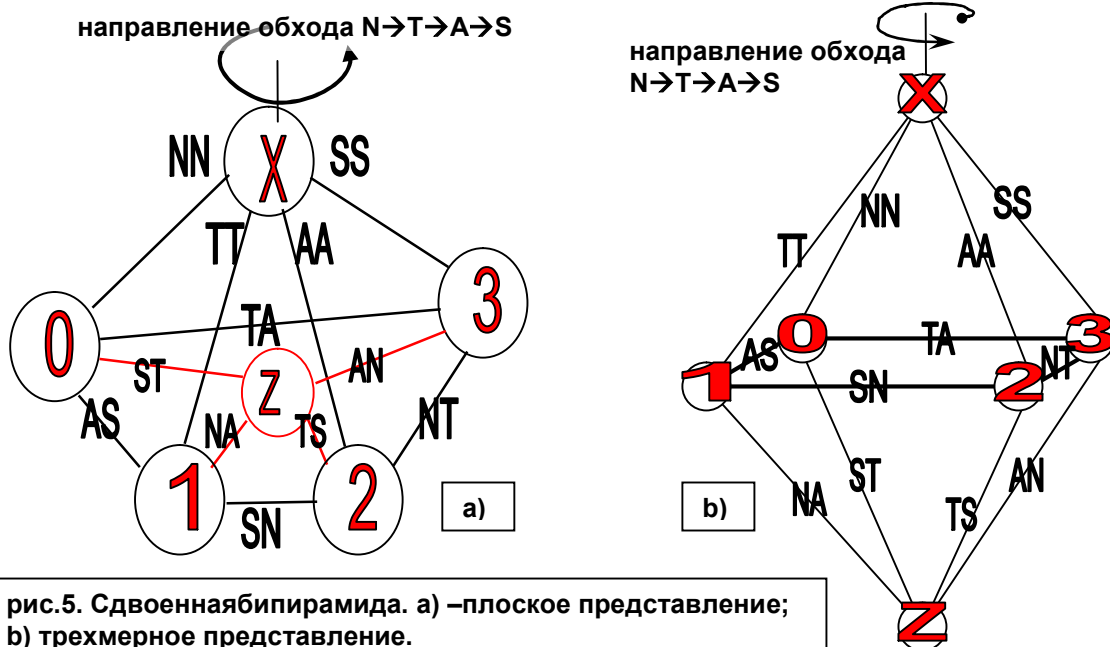


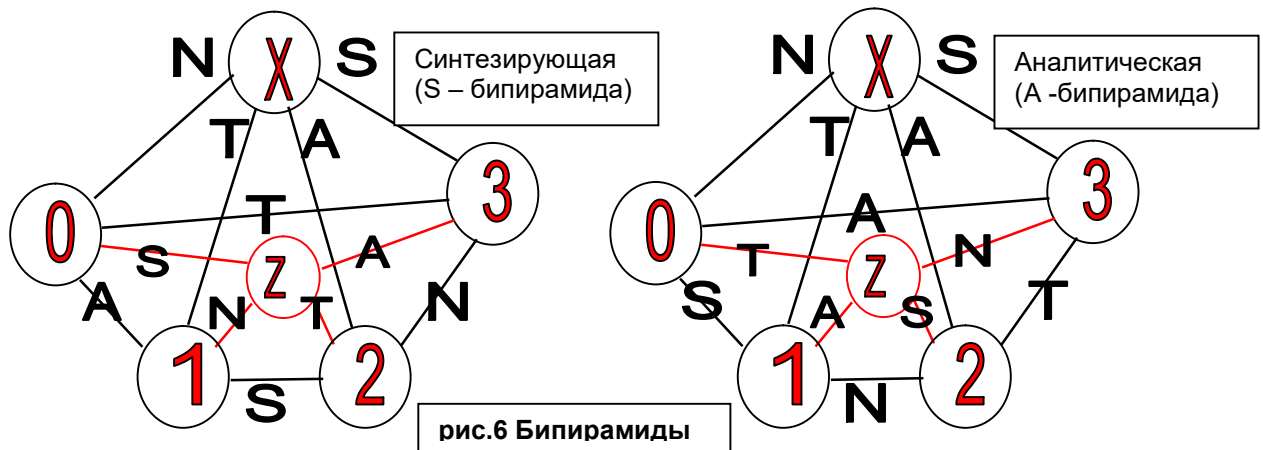
рис.5. Сдвоенная бипирамида. а) –плоское представление; б) трехмерное представление.

Анализ результатов и определения:

1. Полученное логическое построение является бипирамидой (рис.5б), поскольку представляет собой две пирамиды, имеющие общее основание (объект 0, объект 1, объект 2, объект 3) и две вершины (объект X и объект Z). Бипирамида, распадается еще на 2 бипирамиды. Связи одной, определяются по первому символу, а связи второй – по второму символу (рис.6). Поэтому будем называть наше построение на рис 5. *сдвоенной логической бипирамидой*. Процесс, при котором определяется, какая из бипирамид соответствует сдвоенной бипирамиде, будем называть *локализацией бипирамиды*. *Локализацией* в общем смысле будем называть процесс выбора логического символа из некоторого набора допустимых символов (*комбинации символов*) для соотнесения его со связью. Сдвоенную бипирамиду будем также называть *нелокализованной логической бипирамидой*, а каждое из

построений изображенных на рис.6 – локализованной логической бипирамидой.

- Замечаем, что для построения локализованной бипирамиды, необходимо задать 5 связей (4 связи эталонного объекта и одну из связей вспомогательных объектов). Если мы зададим только 4 связи, то получим нелокализованную бипирамиду. Заметим также, что всего в бипирамиде 12 связей.
- Сдвоенная бипирамида на рис.5 построена исходя из синтезирующего направления обхода символов $N \rightarrow T \rightarrow A \rightarrow S$ (прямой порядок). При построении нелокализованной бипирамиды, исходя из аналитического направления обхода $S \rightarrow A \rightarrow T \rightarrow N$ (обратный порядок), все символы в двухсимвольных комбинациях ее связей поменяются местами. Воспользовавшись этим свойством, будем называть бипирамиду, получающуюся при локализации сдвоенной бипирамиды по **первым** символам двухсимвольных комбинаций ее связей:
- А) *синтезирующей бипирамидой (S - бипирамидой)*, при условии, что сдвоенная бипирамида построена исходя из синтезирующего направления обхода символов $N \rightarrow T \rightarrow A \rightarrow S$ (левая часть рис.6).
- В) *аналитической бипирамидой (A - бипирамидой)*, при условии, что сдвоенная бипирамида построена исходя из аналитического направления обхода символов $S \rightarrow A \rightarrow T \rightarrow N$ (правая часть рис.6)
- из 2 и 3 следует, что если в логической бипирамиде определен любой один объект (то есть, определены все связи этого объекта) и одна из всех связей остальных объектов – то мы можем определить, является ли это построение аналитическим или синтезирующим по отношению к эталонному объекту.



4. Карта чисел

Числа в терминах логики четырех состояний.

В соответствии с Тезисом Б. Символы логики четырех состояний имеют порядок следования. Порядковые номера (начиная с 0) символов при синтезирующем направлении обхода символов ($N \rightarrow T \rightarrow A \rightarrow S$), сопоставленные с самими символами станем называть *цифровым представлением символов логики четырех состояний* или *цифрами логики четырех состояний*.

Как уже замечалось ранее, арифметика логики четырех состояний базируется на арифметике по основанию 4. Т.е. имеем всего 4 цифры: N(0), T(1), A(2), S(3).

Привычная для нас арифметика базируется на арифметике по основанию 10 (всего 10 цифр: 0,1,2,3,4,5,6,7,8,9), а арифметика, принятая, например, для вычислительной техники – базируется на арифметике по основанию 2 (всего 2 цифры: 0 и 1).

Число, записанное арабскими цифрами (0,1,2,3) станем называть, как это принято в математике, *числом по основанию 4* или *четверичным числом*, а число, записанное символами полной логики (N, T, A, S) – *симметрией*.

Запись четверичного числа (числа по основанию 4), как и симметрии, - тривиальна и однозначно переводится в число с другим основанием:

Например, симметрия SAN переводится в четверичное число, а затем в десятичное следующим образом:

$$SAN = 320_4 = (3 \cdot 4^2 + 2 \cdot 4^1 + 0 \cdot 4^0)_{10} = (48 + 8 + 0)_{10} = 56_{10} \text{ или}$$

$$SAN = 56_{10} = (5 \cdot 10^1 + 6 \cdot 10^0)_{10}$$

(нижний индекс определяет основание системы счисления)

Существует важное отличие между арифметикой чисел по основанию 4 (или любому другому основанию) и арифметикой симметрий. В арифметике симметрий мы оперируем не абстрактными числами, выражающими только количество или порядок следования некоторых элементов, а логическими понятиями (симметриями), выражающими сложную, иерархическую симметрию какой-либо одной группы элементов, относительно другой группы элементов. Поэтому в арифметике симметрий реализуется более строгое отношение к ведущим нулям и разрядности симметрий, чем в арифметике чисел (в частности, по основанию 4). Например, одно и то же число

$23_4 = 023_4 = 0023_4$ имеет разный логический смысл, выражающийся как неравенство симметрий $AS \langle \rangle NAS \langle \rangle NNAS$. Анализ логического смысла симметрий (как абсолютного, так и по отношению друг к другу) является важной задачей теории логики четырех состояний.

Графическое представление четверичных чисел/симметрий.

Графически симметрия представляется с помощью фрактала. Шаблоном этого фрактала служит построение на рис.7. Шаблон состоит из 4-ех квадрантов, соответствующих символам логики четырех состояний.

Стрелками указано направление обхода символов.

Шаблон фрактала – это один разряд симметрии.

Старшие разряды симметрии находятся в БОльших квадрантах фрактала, младшие разряды, соответственно, находятся в меньших квадрантах (рис. 8). Таким образом, каждое число/симметрия выражается положением квадранта своего самого младшего разряда на всем поле значений фрактала. Геометрически, симметрия выглядит как точка (наименьший квадрант) на плоскости (поле значений фрактала). Набор

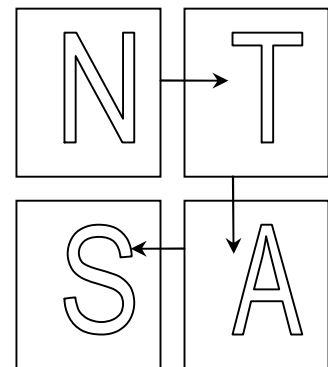


рис.7 Шаблон фрактала

цифровых (десятичных) представлений симметрий, записанный в квадрантах фрактала показан на рис. 9.

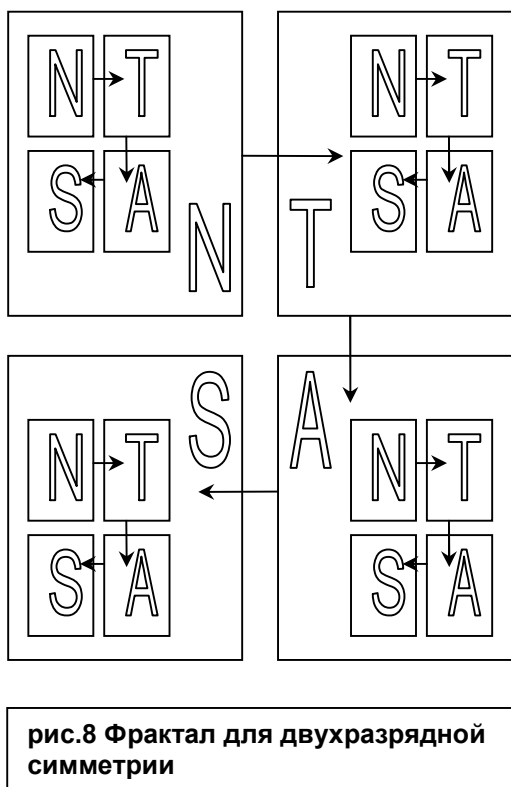


рис.8 Фрактал для двухразрядной симметрии

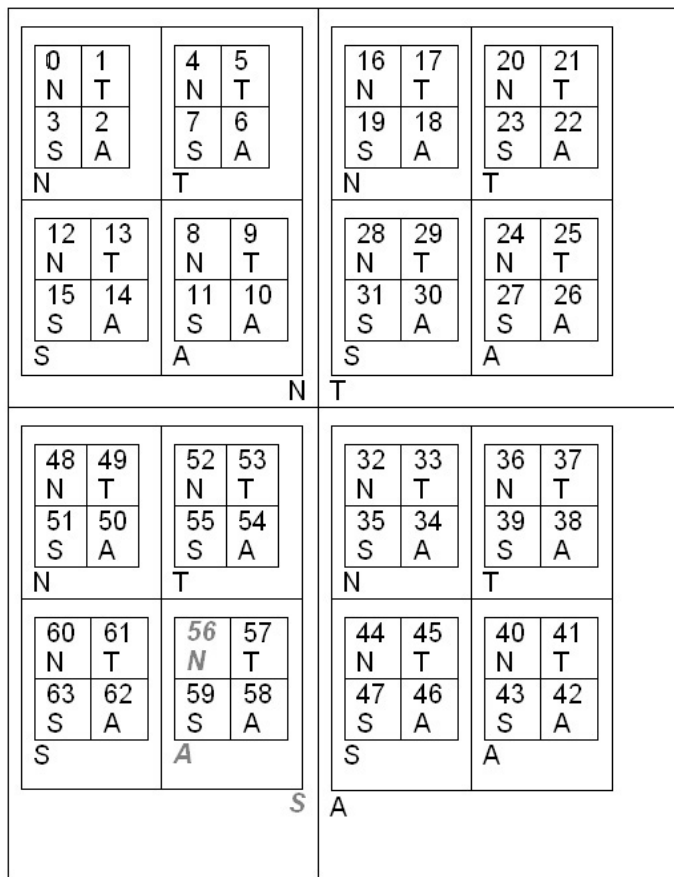


рис.9 Фрактал для трехразрядных чисел / симметрий. Число $56_{10} = SAN$ выделено цветом и курсивом.

Такое построение будем называть *картой чиселразрядности X*, где X - степень вложенности фрактала или количество разрядов в симметрии. Например, на рис.9 указана карта чисел разрядности 3.

Анализ карты чисел и определения:

1. Карта чисел является удобной и наглядной формой записи и анализа симметрий, но не единственно возможной.
2. Направление обхода символов в шаблоне фрактала – *синтезирующее*.
3. Расстояние на карте между двумя симметриями связано с общностью в их логике.
4. Симметрии с одинаковыми символами во всех разрядах сосредоточены в углах карты чисел. Такие симметрии будем называть *когерентными*.
5. Анализ симметрий следует проводить, в направлении от старшего разряда к младшему. Старшие разряды задают общие логические аспекты, младшие разряды – детализуют эти аспекты.

В приложении приведен исходный код компьютерной программы на языке Си для расчета карт чисел заданной разрядности:

5.Связь между симметрией и логической бипирамидой.

Вернемся к алгоритму бипирамиды. При расчете по этому алгоритму мы получаем сдвоенную бипирамиду (рис.5).

На шаге 7 алгоритма мы вводим результирующий объект Z , как объединение неспаренных связей вспомогательных объектов. Уберем результирующий объект сдвоенной бипирамиды. В этом случае сдвоенная бипирамида превращается в сдвоенный (нелокализованный) объект (Z_0), который, по аналогии с сдвоенной бипирамидой, распадается на синтезирующий (Z_{0s}) и аналитический (Z_{0a}) объект. Это проиллюстрировано на рис.10

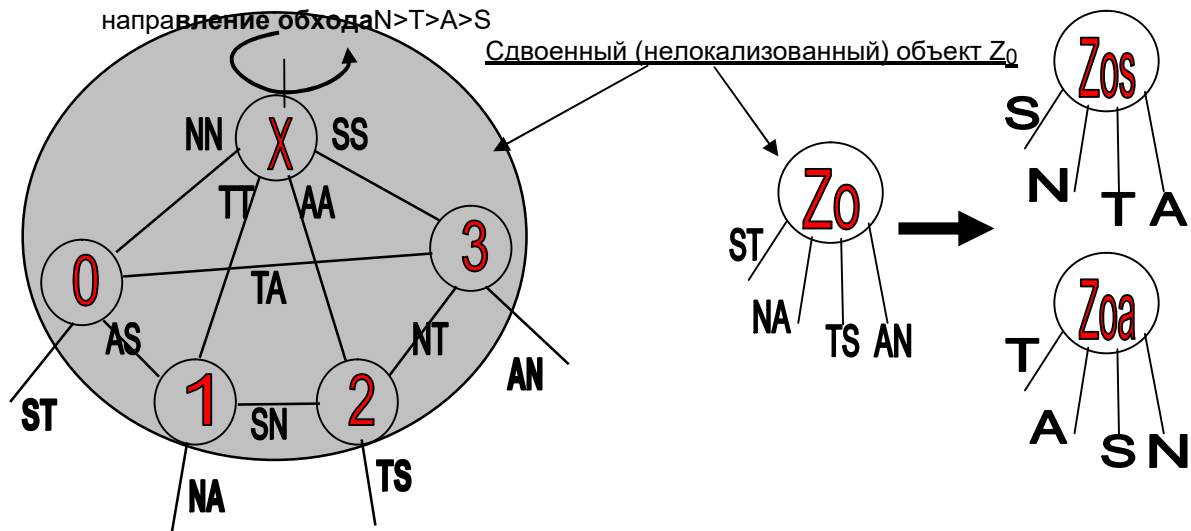


рис.10 Сдвоенный объект Z_0 , распадается на два объекта синтезирующий (Z_{0s}) и аналитический (Z_{0a})

При локализации, сдвоенный объект превращается в обычный объект. Его можно использовать, как эталонный для построения еще одной сдвоенной бипирамиды. Причем направление обхода (синтезирующее или аналитическое) новой сдвоенной бипирамиды, уже будет определено на основании локализации сдвоенного объекта предыдущей сдвоенной бипирамиды. Таким образом, можно создавать цепочки из локализованных бипирамид, а при размыкании связей других объектов (вершин бипирамид) можно создавать разнообразные пространственные логические сети. Кроме того, можно создавать цепочки из нелокализованных бипирамид. Пример такой логической цепочки из двух нелокализованных бипирамид изображен на рис.11.

Вспомогательные объекты любой бипирамиды в цепочке и их связи будем называть *звеньями цепочки*. Нумерацию звеньев, будем проводить от верхней бипирамиды к нижней. Таким образом, объекты 0,1,2,3 – образуют звено 1, а объекты 00, 11,22, 33 – звено 2. Эталонный объект – является вырожденным случаем и представляет собой звено 0. Результирующий объект – звено 3.

Отсюда следует, что любая отдельно взятая бипирамида представляет собой цепочку из трех звеньев (звено 0, звено 1, звено 2), два из которых – звено 0 (эталонный объект) и звено 2 (результирующий объект) - вырождены.

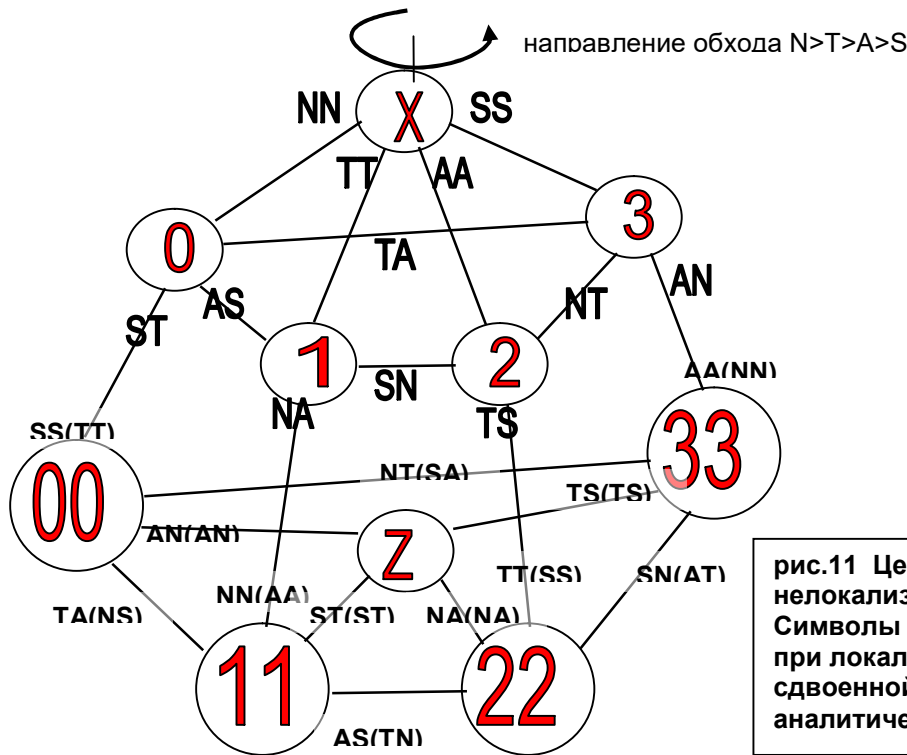


рис.11 Цепочка из двух нелокализованных бипирамид. Символы в скобках используются при локализации верхней сдвоенной бипирамиды, как аналитической.

пояснения:

- не следует путать обозначения вспомогательных объектов (0,1,2,3,00,11,22,33) с числовым представлением символов логики четырех состояний N(0), T(1), A(2), S(3) или симметриями.
- Двухсимвольные комбинации в скобках записаны исходя из аналитического направления. Таким образом, они записаны в направлении, обратном двухсимвольным комбинациям вне скобок.

Анализ результатов и определения

Заметим, что:

1. Как видно из построения на рис.11, значения связей результирующего объекта Z не зависят от локализации верхней сдвоенной бипирамиды. Отсюда следует, что при увеличении длины цепочки, связи любого звена будут выражены максимум четырехсимвольной комбинацией. При этом выбор символа для каждой связи будет определяться только локализациями текущей и предыдущей нелокализованной бипирамиды.
2. связи 00-11,11-22,22-33 и 33-00 вспомогательных объектов нижней нелокализованной бипирамиды могут принимать всю группу значений (N,T,A,S). Для определения этих связей необходима локализация обоих нелокализованных бипирамид.
3. Путь от эталонного объекта к результирующему объекту, можно выразить через симметрию (при условии локализации всех нелокализованных бипирамид цепочки). При этом старший разряд симметрии соответствует одной из связей эталонного объекта, а младший разряд – одной из связей результирующего объекта.

Под (логическим) путем мы подразумеваем расстояние между объектами, выраженное минимальным набором связей. Заметим также, что путь от эталонного объекта к результирующему объекту цепочки всегда проходит через вертикальные ребра бипирамид.

Логической траекторией или логическим выводом – будем называть набор связей, соединяющий два объекта произвольным образом. Заметим, что если какой-либо участок траектории, связывающей эталонный объект с результирующим, проходит через связь между вспомогательными объектами одной бипирамиды (например, через связь 00-11), то такая траектория не может быть путем.

Связь между двумя вспомогательными объектами одной бипирамиды будем называть *паразитной*. Паразитные связи лежат на горизонтальных ребрах бипирамид.

Логический путь – является самой короткой, а значит и самой эффективной траекторией, связывающей два логических выражения: предпосылку (эталонный объект) и вывод (результирующий объект). Следовательно, для построения эффективных логических выводов необходимо, по возможности, исключать паразитные связи между вспомогательными объектами.

При построении кратчайшего логического вывода желательно обходить:

1. связи между понятиями, которые указывают на то, что предпосылка находится вне пределов рассмотрения (N) и понятиями, которые подтверждают достоверность предпосылки (T). Другими словами связь (0-1), вспомогательных объектов на рис 11.
2. связи между понятиями, которые подтверждают достоверность предпосылки (T) и понятиями, которые устанавливают ложность этой предпосылки (A). Другими словами связь (1-2), вспомогательных объектов на рис 11.
3. связи между понятиями, которые устанавливают ложность предпосылки (A) и понятиями, которые указывают на неопределенность этой предпосылки (S). Другими словами связь (2-3), вспомогательных объектов на рис 11.
4. связи между понятиями, которые указывают на неопределенность предпосылки (S) и понятиями, которые указывают на то, что предпосылка находится вне пределов рассмотрения (N) или на то, как эта предпосылка реализована (N). Другими словами связь (3-0), вспомогательных объектов на рис 11.

Иначе логический вывод увеличится, и не будет представлять собой логический путь.

Всегда существуют четыре логических пути, связывающих эталонный и результирующий объекты в цепочке.

Заметим также, что для построения длинных (сложных и обстоятельных) логических выводов необходимо пользоваться паразитными связями.

6. Поведение цепочки нелокализованных бипирамид при увеличении количества звеньев.

Рассмотрим цепочку, изображенную на рис.12. В ней 8 звеньев. Напомним, что звенья 0 и 7 – являются вырожденными и представляют собой соответственно: эталонный и результирующий объекты. Заметим, что звено 6 полностью повторяет звено 2. Таким образом, при дальнейшем увеличении числа звеньев

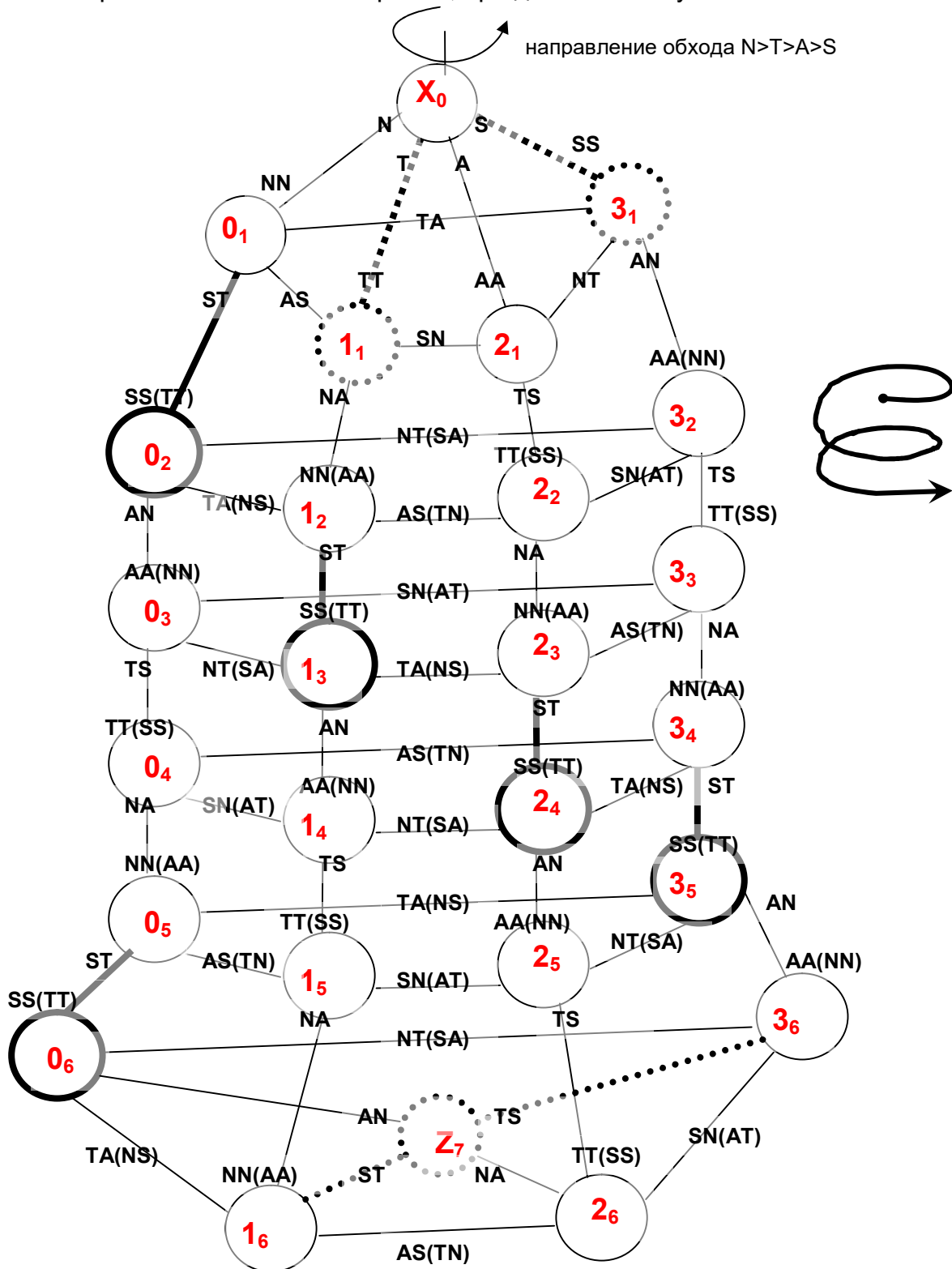


рис.12 Цепочка из 8 звеньев (шесть нелокализованных бипирамид). Нижний индекс в обозначениях объектов указывает на звено, к которому относится объект.

мы получим многократное повторение фрагмента из звеньев со второго, по пятое. Заметим также, что *звено 1 (первое звено)* никогда не повторяется, какую бы длину не имела цепочка. То есть первое звено является *уникальным звеном*.

Также следует отметить, что при локализации цепочки бипирамид первое звено перестает быть уникальным и многократно повторяется. Цепочки локализованных бипирамид будут рассмотрены в отдельной главе. Первое звено и каждое из вырожденных звеньев цепочки нелокализованных бипирамид станем называть *уникальными звеньями*. То есть звеньями, которые не повторяются при увеличении длины цепочки.

Явление многократного повторения некоторого фрагмента логического построения назовем *свойством избыточности*.

Логическое построение, объекты которого взаимосвязаны между собой и не имеют неспаренных связей, будем называть *логически замкнутым построением*.

Логическое построение, в котором хотя бы один объект имеет, хотя бы одну неспаренную связь называется *фрагментом*.

Фрагмент из четырех последовательных и неуникальных звеньев станем называть *полным фрагментом*. Заметим, что каждое последующее звено в полном фрагменте, повернуто на один шаг (90^0) в соответствии с направлением обхода, заданном для эталонного объекта. Это проиллюстрировано на рис.12, на примере объектов, обведенных толстой линией. Пунктирной, толстой линией указано вырожденное продолжение этой тенденции для объектов в уникальных звеньях. Таким образом, все звенья цепочки повернуты относительно друг – друга и образуют *спираль (спираль сдвига звеньев)*, направленную от эталонного объекта к результирующему объекту. Направление витков спирали соответствуют заданному в цепи синтезирующему направлению обхода.

Фрагмент, насчитывающий менее четырех последовательных и неуникальных звеньев, станем называть *неполным фрагментом*. Количество звеньев в неполном фрагменте будем называть *остатком*. Остаток может быть равен единице, двойке либо тройке. При отсутствии неполного фрагмента в построении – остаток равен нулю.

Фрагмент, насчитывающий более четырех последовательных и неуникальных звеньев цепи, назовем *избыточным фрагментом*. Заметим, что избыточный фрагмент может содержать в себе несколько полных фрагментов, либо несколько полных фрагментов и один неполный. То есть избыточный фрагмент может быть *кратным* либо *некратным* полному фрагменту. В общем случае *кратностью* мы будем называть количество полных фрагментов в избыточном, полном или неполном фрагменте цепи.

В избыточном фрагменте кратность всегда больше либо равна 1.

В полном фрагменте кратность всегда равна 1.

В неполном фрагменте кратность всегда равна 0.

Набор уникальных звеньев ([объект X], [звено 1], [объект Z]) будем называть *уникальным фрагментом*, а набор вырожденных звеньев ([объект X], [объект Z]) будем называть *вырожденным фрагментом*. Таким образом, уникальный фрагмент состоит из вырожденного фрагмента и уникального звена 1. Все остальные фрагменты: полный фрагмент, неполный фрагмент и избыточный фрагмент - будем называть *неуникальными фрагментами*.

Таким образом, любой неуникальный фрагмент можно охарактеризовать, как фрагмент с кратностью K и остатком R . Где K – количество полных фрагментов, а R – количество звеньев неполного фрагмента. Например, неуникальный (избыточный) фрагмент цепи, изображенной на рис.12 является фрагментом с кратностью 1 и остатком 1.

Цепь бипирамид, состоящую только из одного полного и одного уникального фрагмента будем называть *полной и неизбыточной*. Такая цепь имеет 7 звеньев (с X_0 по Z_6) или, другими словами, 5 бипирамид.

Цепь, состоящую из уникального фрагмента и неполного фрагмента, будем называть *неполной*. Такая цепь имеет 4, 5 или 6 звеньев или от 2 до 4 бипирамид, включительно.

Цепь, состоящую из уникального фрагмента и избыточного фрагмента, будем называть *избыточной*. Она имеет более 7 звеньев или более 5 бипирамид. Такая цепь представлена на рис. 12.

Заметим, что цепь, состоящая только из уникального фрагмента, соответствует бипирамиде. Поэтому бипирамиду можно называть также *уникальной цепью*.

Только в уникальной цепи (бипирамиде) все звенья уникального фрагмента идут последовательно одно за другим и связаны между собой. Будем называть это свойство – *свойством последовательности уникального фрагмента* или *свойством уникальности*. Все цепи, кроме бипирамиды лишены этого свойства. В таких цепях последнее звено уникального фрагмента [объект Z] отделено от остальной своей части с помощью неуникального фрагмента.

Заметим также, что увеличение количества звеньев в цепи может происходить только за счет увеличения количества звеньев в неуникальном фрагменте. Таким образом, только неуникальный фрагмент обладает *свойством роста*. Свойство роста и свойство уникальности являются взаимоисключающими свойствами. То есть, при увеличении количества звеньев в уникальной цепи (бипирамиде), она теряет свое свойство уникальности и приобретает свойство роста. По мере роста неуникального фрагмента он проходит три последовательные стадии: неполный фрагмент → полный фрагмент → избыточный фрагмент. Заметим, что при переходе неуникального звена от полного фрагмента к избыточному фрагменту в построении появляется свойство избыточности.

Цепь, состоящую только из неуникального фрагмента, будем называть *неуникальной цепью*. В такой цепи связи последнего звена неуникального фрагмента соединены со связями первого звена этого же фрагмента. Геометрически, неуникальная цепь представляет собой тор. Неуникальная цепь, состоящая из одного полного фрагмента, изображена на рис.13. При этом нумерацию звеньев в неуникальной цепи будем проводить, начиная с номера 2. Этим мы подчеркиваем отсутствие уникального фрагмента в неуникальной цепи.

В частности, отсутствие звена 0 (эталонного объекта) и звена 1.

Отсутствие уникального фрагмента в цепи и, следовательно, отсутствие бипирамид в цепи нелокализованных бипирамид будем называть *свойством нового качества цепи*.

Неуникальные цепи являются важной логической структурой. Более подробно неуникальные цепи и свойства цепей будут рассмотрены в отдельной главе.

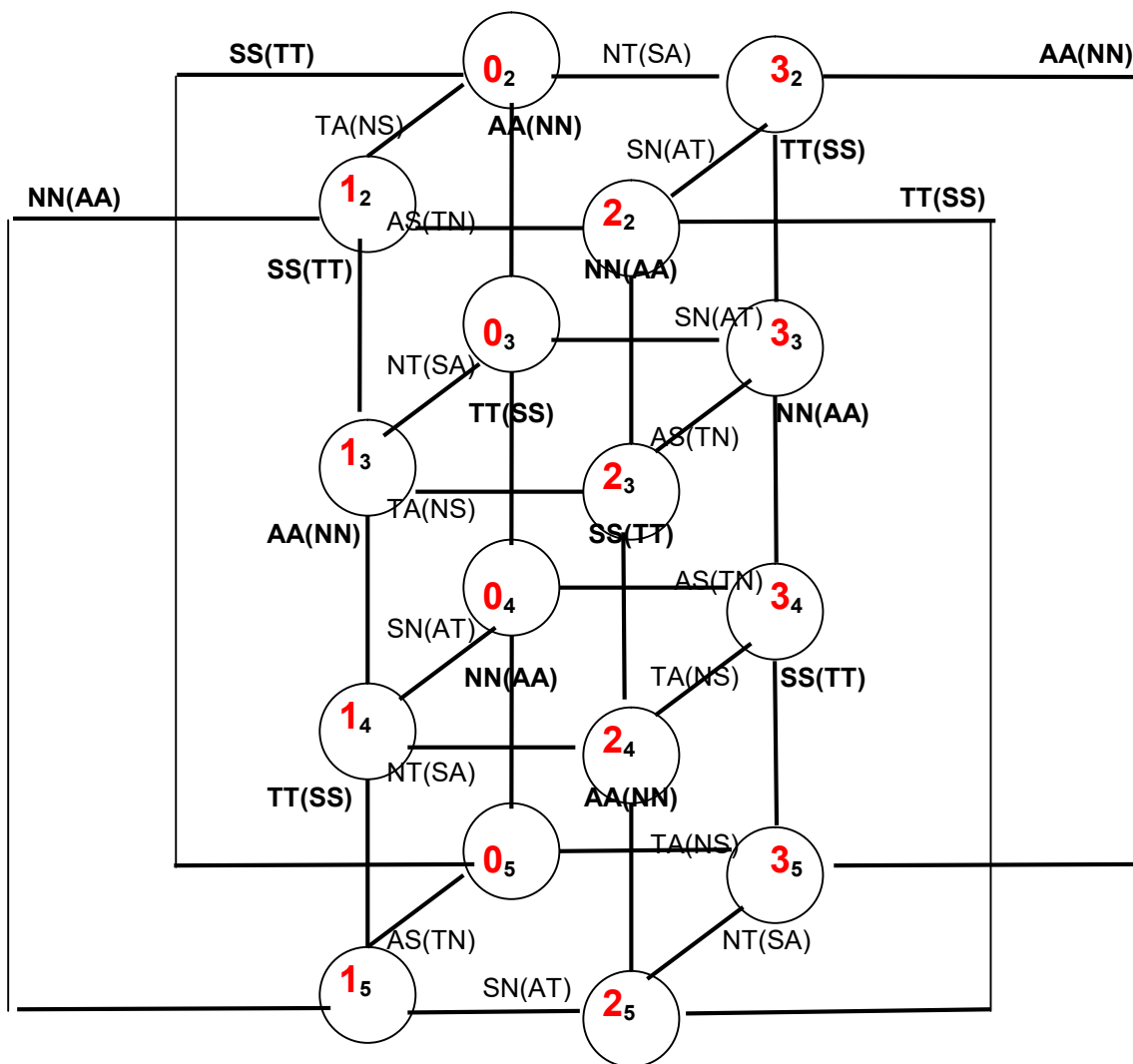


рис.13 Неуникальная цепь из четырех последовательных и неуникальных звеньев или одного полного фрагмента.

7.Абстрагирование и внедрение. Абстрактный объект.

В логической бипирамиде используется 4 связи для описания каждого объекта в терминах остальных объектов. В свою очередь, каждая из связей между объектами в пирамиде описана одним из символов логики четырех состояний: N, T, A, S (напомним, что бипирамида является полностью локализованным построением). Внутреннее устройство объекта в бипирамиде никак не выражено и его поведение мы можем определить только по значению связей N, T, A, S объекта с другими объектами. Поэтому мы говорим, что объект определен или реализован остальными объектами. Такой способ реализации объекта будем называть *внешней реализацией объекта*. Другими словами, *внешней реализацией объекта в терминах логического построения* будем называть фрагмент логического построения, остающийся при отделении от него этого объекта.

Любой объект, отделенный от логического построения, или вообще не связанный с каким-либо логическим построением будем называть *абстрактным* объектом.

О внутреннем устройстве абстрактного объекта можно сказать, что оно реализовано или только реализуется (утверждение 1) и, что оно лежит вне нашего понимания

(утверждение 2). Утверждения 1 и 2 соответствуют символу N в логике четырех состояний. *Внутренней реализацией* любого объекта будем называть реализацию (связь N) этого объекта при отделении его от логического построения. Другими словами *внутренняя реализация объекта* – это реализация (связь N) соответствующего ему абстрактного объекта.

Мы будем говорить, что при отделении (*абстрагировании*) объекта от логического построения связь N этого объекта замыкается на внутреннюю реализацию последнего для поддержания логического существования (смысла) объекта. Таким образом, абстрактные объекты имеют всего три связи и изображаются в соответствии с рис. 14а или в соответствии с рис. 14b, если мы хотим задать обозначение абстрактному объекту. На рис. 14b замкнутая связь N просто не отображена. Кроме того, абстрактный объект может быть изображен вместе с обозначением и замкнутой связью N, как это показано на рис. 14с.

При отображении наших логических построений мы будем пользоваться как абстрактными, так и отдельно - стоящими объектами. *Отдельно - стоящие объекты* - это обычные объекты, связанные с логическим построением, но, для наглядности, изображенные отдельно от построения (рис. 14d). Их не стоит путать с абстрактными объектами. Абстрактные объекты имеют только 3 связи. Часто в виде отдельно-стоящего объекта изображается внешняя реализация абстрактного объекта (т.е. фрагмент, остающийся при отделении(абстрагировании) объекта удобно отобразить в виде некоторого отдельно-стоящего объекта).

Процесс прикрепления абстрактного объекта к логическому построению, обратный процессу абстрагирования, будем называть *внедрением абстрактного объекта*.

При внедрении связь N абстрактного объекта размыкается, и он становится обычным объектом.

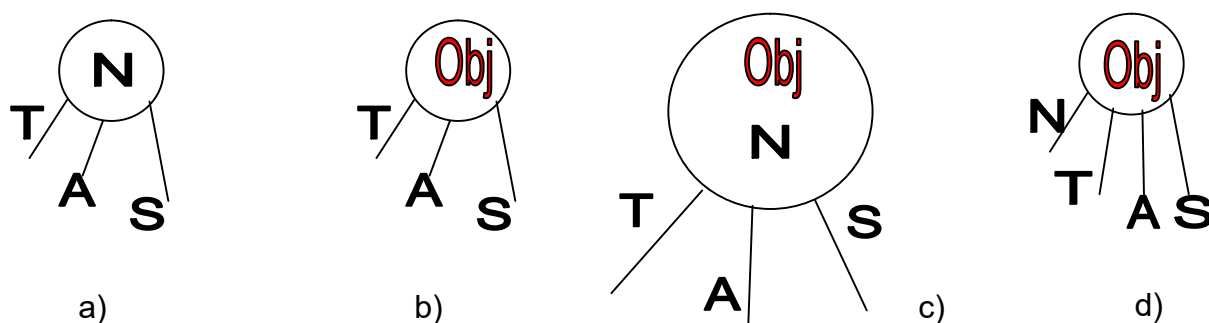


Рис.14 а), b), c) -изображения абстрактного объекта. d) – отдельно- стоящий объект.

Также как обычные объекты, абстрагированию и внедрению могут быть подвергнуты *сдвоенные объекты*. В качестве примера, вспомним процесс выделения сдвоенного объекта из сдвоенной бипирамиды (рис.10, глава 5). Согласно введенным нами определениям, сдвоенному объекту Z_0 (левая часть рис.15)

будет соответствовать абстрактный нелокализованный объект Z (правая часть рисунка 15). Абстрактный нелокализованный объект Z - это тот самый результирующий объект, который мы удалили из сдвоенной бипирамиды. А отдельно-стоящий сдвоенный объект Z_0 – является его внешней реализацией.

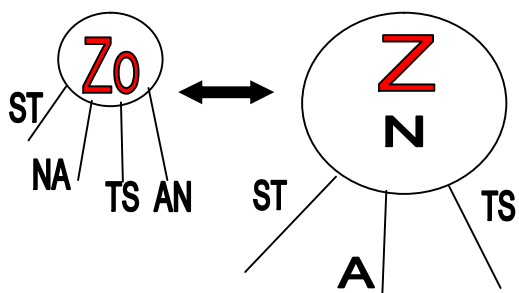


рис.15 Отдельно- стоящий и абстрактный нелокализованные объекты

Следует заметить, что при абстрагировании результирующего объекта станет локализованной одна из его связей (связь А). Это вытекает из определения процесса абстрагирования. А именно:

1. при отделении результирующего объекта от сдвоенной бипирамиды связи этого объекта теряют свое соответствие с оставшимися фрагментами.
2. для замыкания связи N на внутреннюю реализацию результирующего объекта, необходимо локализовать эту связь.
3. Связь N может быть локализована из связей NA и AN и только из них. При этом, вне зависимости от способа локализации, связь N замкнется и останется связь А.

При внедрении абстрактного результирующего объекта в цепочку локализованных бипирамид – связь N должна быть локализована, в соответствии с требованиями внешней реализации результирующего объекта. Процессы абстрагирования и внедрения имеют важное значение при изменении длины цепочки бипирамид и при арифметических операциях с симметриями.

8. Абстрактные логические системы. Тетраэдр.

Абстрактные объекты имеют три свободных связи и могут объединяться с другими абстрактными объектами. Логическое построение, все объекты которого используют только три связи, исключая связь N, для объединения между собой будем называть *абстрактным логическим построением*. Все объекты такого логического построения считаются абстрактными.

Логический тетраэдр.

Создадим абстрактное логическое построение. Для этого возьмем некоторый произвольный объект 0, в качестве эталонного объекта. Будем считать его абстрактным. В соответствии с тезисом А, все остальные объекты (свойства) по отношению к объекту 0 могут находиться в одном из четырех состояний (N_0, T_0, A_0, S_0). Нижний индекс символов, указывает на их принадлежность к объекту 0. Сгруппируем эти свойства в объекты 0,1,2,3 (рис.16). Заметим, что в соответствии с определением абстрактного объекта, все свойства лежащие вне понимания, по отношению к объекту 0 (N_0), являются его внутренней реализацией (N_0).

Теперь рассмотрим вспомогательные, абстрактные объекты 1, 2 и 3. Каждая из групп свойств T_0, A_0 и S_0 объекта 0 – является внутренней реализацией, соответствующего абстрактного объекта 1, 2 и 3. Другими словами $N_1 = T_0, N_2 = A_0, N_3 = S_0$ (рис.17).

Объединим связи с одинаковыми символами, каждого из объектов 0,1,2,3 между собой (рис.17):

Алгоритм тетраэдр:

1. объединим все связи эталонного объекта 0 с соответствующими им связями вспомогательных объектов 1,2,3. А именно (рис.18):
 - а) T_0 с T_1 ;
 - б) A_0 с A_2 ;
 - в) S_0 с S_3 ;
2. Для каждого вспомогательного объекта остаются две неспаренные связи. Они объединяются однозначным образом, поскольку, между любыми двумя вспомогательными объектами есть только один общий символ. А именно (рис.18):
 - а) S_1 с S_2 ;

b) $T_2 \subset T_3$;
c) $A_3 \subset A_1$;

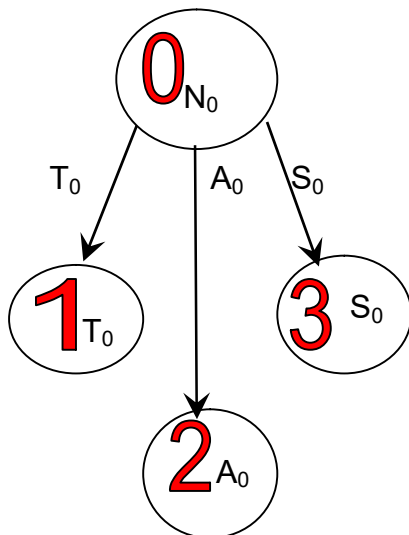


рис.16

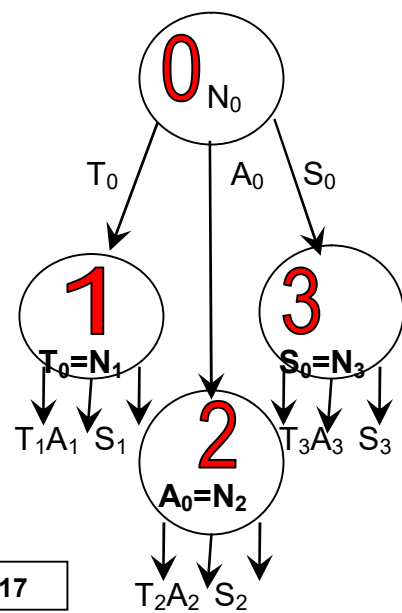


рис.17

Анализ результатов:

1. Тетраэдр является логически замкнутым построением, каждый объект которого связан со всеми остальными объектами построения. Такие построения будем называть *полностью связанными логическими построениями*.

2. В алгоритме тетраэдр нельзя задать направление обхода символов. Оно всегда синтезирующее ($N \rightarrow T \rightarrow A \rightarrow S$). Это важное свойство следует из абстрактности эталонного объекта.

Действительно, при построении тетраэдра, чтобы разграничить свойства эталонного объекта по группам нам необходимо иметь этот эталонный объект. В абстрактных объектах используется связь N для поддержания существования объектов. Таким образом, символ N – должен быть первым при выборе направления обхода. Единственное направление, отвечающее такому ограничению – синтезирующее. (Тем не менее

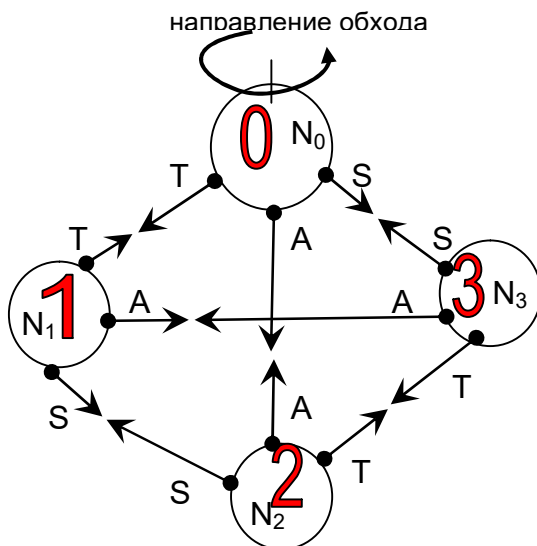


рис.18

возможно исследовать алгоритм тетраэдр и его свойства с заданным аналитическим направлением обхода.)

3. Алгоритм тетраэдр, является вырожденным вариантом алгоритма бипирамида.
4. Внутренняя реализация абстрактных объектов зависит от способа их рассмотрения. Это проиллюстрировано на примере объединения группы свойств 1 абстрактного объекта 0 в абстрактный объект 1
 - 4.1 Если мы рассматриваем свойства из группы 1, по отношению к объекту 0 - то все они характеризуются символом T, исходя из наших действий по сортировке этих свойств (рис.16).
 - 4.2 Если мы рассматриваем каждое свойство в группе 1, по отношению к объекту 1 – то они характеризуются уже символом N, поскольку логический объект 1 и, соответственно, его отношения с остальными объектами реализованы, на базе этих свойств (рис.17).

Объект, в отношении которого рассматривается внутренняя реализация абстрактного объекта, будем называть *реализующим объектом*.

Абстрактный объект, внутреннюю структуру которого определяет реализующий объект, будем называть *реализуемым объектом*.

Для случая 4.1 реализующим является объект 0, а реализуемым является объект 1.

Для случая 4.2 реализующий и реализуемый объекты совпадают и являются объектом 1.

Выбор реализующего объекта произволен. Необходимым условием для выбора объекта, в качестве реализующего, является наличие связи с реализуемым объектом.

Если реализуемый объект не имеет связей с другими объектами – то в качестве реализующего объекта выступает он сам, используя свою связь N. Таким образом, у каждого абстрактного объекта всегда имеется не менее одного и не более четырех реализующих объектов.

У каждого абстрактного объекта в логически замкнутом построении (не только абстрактном) всегда есть ровно четыре реализующих объекта, включая его самого. Иначе данное логическое построение имеет неспаренные связи и является фрагментом.

В общем случае мы будем рассматривать внутреннюю реализацию абстрактного объекта по отношению к нему самому (т.е. как символ N, записанный внутри объекта).

В некоторых случаях, чтобы подчеркнуть выбор реализующего объекта мы будем указывать внутреннюю реализацию, как символ логики четырех состояний с индексом, в виде обозначения соответствующего реализующего объекта.

При этом символ логики четырех состояний будет соответствовать значению связи реализующего объекта с реализуемым объектом.

5. Связи между объектами логического построения не зависят от способа рассмотрения. Например, на рис. 18, логическая связь между объектами 0 и 1 не зависит от того, к какому из объектов, 0 или 1, мы относим эту связь. Поскольку в любом случае она будет выражена символом T. Данное свойство связи относится не только к абстрактным, но и ко всем логическим построениям.

6. Независимость от способа рассмотрения будем называть *условием материальности*. И наоборот зависимость от способа рассмотрения будем называть *условием идеалистичности* или *условием абстрактности*.

7. При внедрении всех абстрактных объектов, тетраэдр превращается из абстрактного логически замкнутого и полностью связанного построения в фрагмент неабстрактного логического построения (рис. 19).

Процесс внедрения всех объектов абстрактного построения будем называть *внедрением абстрактного построения*.

Все разомкнутые связи любого внедренного абстрактного построения или фрагмента неабстрактного построения будем называть *точками крепления* или *валентными связями*.

Количество таких точек будем называть *валентностью* фрагмента.

Существует 4 типа валентных связей в соответствии с количеством логических символов.

При внедрении абстрактного логического построения образуются валентные связи типа N. При этом, например, валентность внедренного тетраэдра будет равна 4.

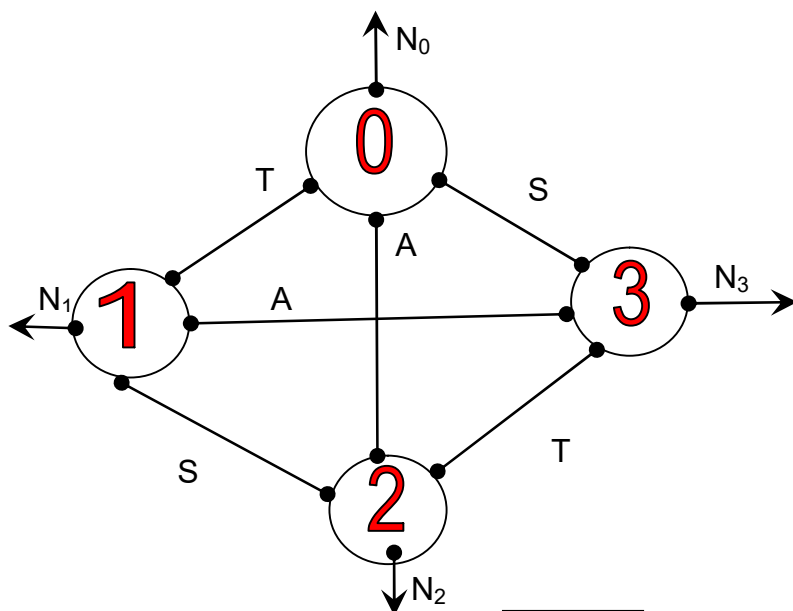


рис.19

ВЫВОДЫ:

1. Считаем, что полученные результаты можно рассматривать как предпосылки для выражения законов естественных наук через алгоритмы, символы и понятия теории логики четырех состояний.

2. Мы пришли к более глубокому пониманию внутренней и внешней реализации объекта. Внешняя реализация внедренного объекта – является такой вырожденной формой внутренней реализации, при которой объект никак сам себя не определяет. Его полностью

определяет логическое построение, в которое он внедрен.

3. Введены понятия идеалистичности внутренней реализации логических объектов и материальности связей между ними.
Считаем, что физическая материя является сложным фрактальным логическим построением. В этом построении каждый объект является отдельным логическим построением. Другими словами, каждая система может быть рассмотрена в качестве объекта следующей системы. Материальные свойства такой системы будут проявляться, по мере увеличения вложенности фрактала, как свойства связей между объектами на каждом уровне фрактала (в каждой логической системе).
Таким образом, физическая материя является логической системой, с большим количеством связей, возможно стремящимся к бесконечности и связи между логическими объектами и сами объекты, определяют физические законы.

9.Цепи тетраэдров. Треугольные призмы.

Аналогично бипирамиде, тетраэдры можно организовать в виде цепочки. Начнем изменять тетраэдр в соответствии с предлагаемым алгоритмом.

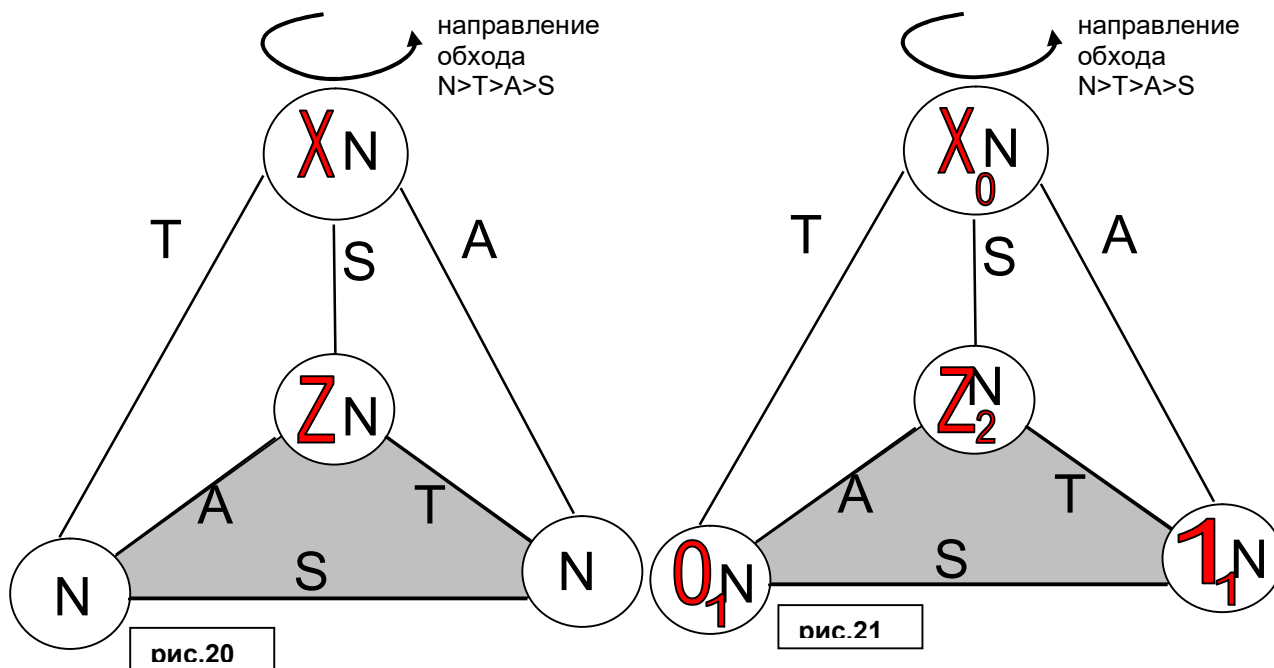
Алгоритм треугольная призма.

1. В тетраэдре выберем произвольным образом два различных объекта и назовем их эталонным (X) и результирующим (Z) объектами. Например, как на рис.20.

Замечания к выбору объектов X и Z:

- 1.1 Возможность произвольного выбора эталонного объекта тетраэдра следует из его полной логической связанности, а также, из однозначности связей в тетраэдре. Данная возможность отсутствует в нелокализованной бипирамиде, где место эталонного объекта однозначно определено его внешней реализацией. В бипирамиде (локализованной бипирамиде), также как и в тетраэдре, существует возможность произвольного выбора эталонного объекта.
- 1.2 В бипирамиде, место результирующего объекта, однозначно определяется выбором эталонного объекта. В тетраэдре этого не происходит благодаря полной логической связанности объектов тетраэдра (каждый объект тетраэдра связан со всеми остальными объектами тетраэдра (см. алгоритм тетраэдр, анализ результатов, пункт 1)). Поэтому каждый из оставшихся трех

неэталонных объектов может считаться результирующим. Таким образом, в качестве результирующего объекта тетраэдра мы вправе выбрать любой объект кроме эталонного объекта.



1.3 После выбора эталонного объекта в тетраэдре, необходимо выбрать результирующий объект.

Наиболее логичным (не обязательным) выбором из трех оставшихся неэталонных объектов, в качестве результирующего является объект, имеющий S – связь с эталонным объектом.

Поясним наш выбор:

а) В тетраэдре всегда задано синтезирующее направление обхода $N \rightarrow T \rightarrow A \rightarrow S$. Следовательно, группа свойств, образующая объект, имеющий S – связь с эталонным объектом будет рассматриваться последней.

б) Результирующий объект является последним звеном цепочки и при установлении символов для каждой связи (*расчете связей*) цепочки также рассматривается последним.

1.4 Возможность выбора эталонного объекта, результирующего объекта и направления обхода в логическом построении будем называть *степенями свободы* логического построения.

2. Оставшиеся два объекта будем называть вспомогательными объектами. Объекту, имеющему T-связь с эталонным объектом, дадим обозначение 0, а объекту, имеющему A-связь с эталонным объектом дадим обозначение 1. Данные обозначения отражают очередность символов при синтезирующем направлении обхода $N \rightarrow T \rightarrow A \rightarrow S$. Таким образом, мы подчеркиваем, что объект 0 (символ T) рассматривается относительно эталонного объекта раньше, чем объект 1 (символ A). Объекты 0 и 1, а также их связи будем называть *звеном треугольной призмы*. Эталонный и результирующий объекты будем называть – *вырожденными звеньями треугольной призмы*, по аналогии с цепочкой (нелокализованных) бипирамид. Можно сказать, что тетраэдр состоит из трех звеньев треугольной призмы, два из которых вырождены. Нумерацию звеньев будем проводить от эталонного объекта к результирующему объекту. Начинать нумерацию звеньев будем с нуля и записывать в виде индексов к обозначениям объектов, входящих в звено (рис.21)

3. *Треугольной призмой* или *цепочкой тетраэдров* будем называть модификацию тетраэдра, имеющую не менее четырех звеньев (рис.22).

Напомним, что сам тетраэдр имеет только три звена, два из которых вырождены. *Количество тетраэдров в треугольной призме* будем определять количеством невырожденных звеньев треугольной призмы. Например, треугольная призма, изображенная на рис 23 состоит из трех тетраэдров и пяти звеньев (звено 0 – эталонный объект, звено 1, звено 2, звено 3, звено 4 – результирующий объект).

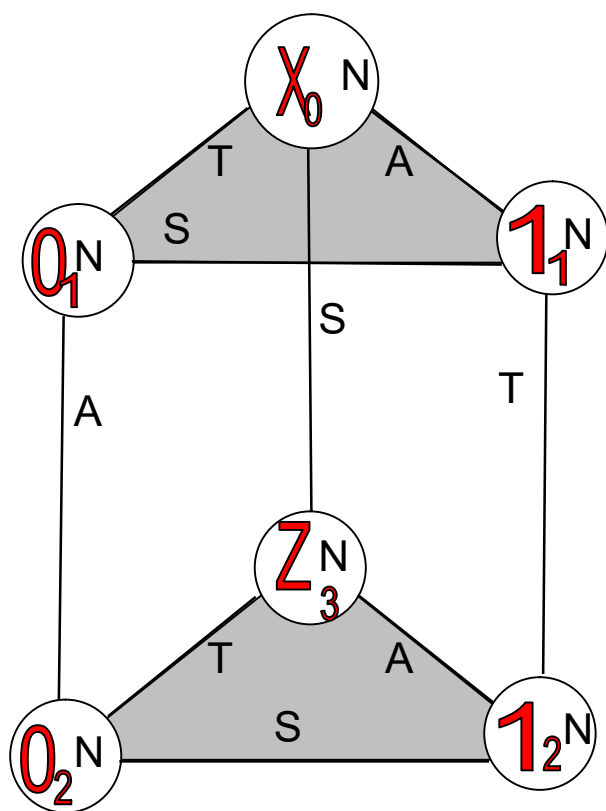


рис.22

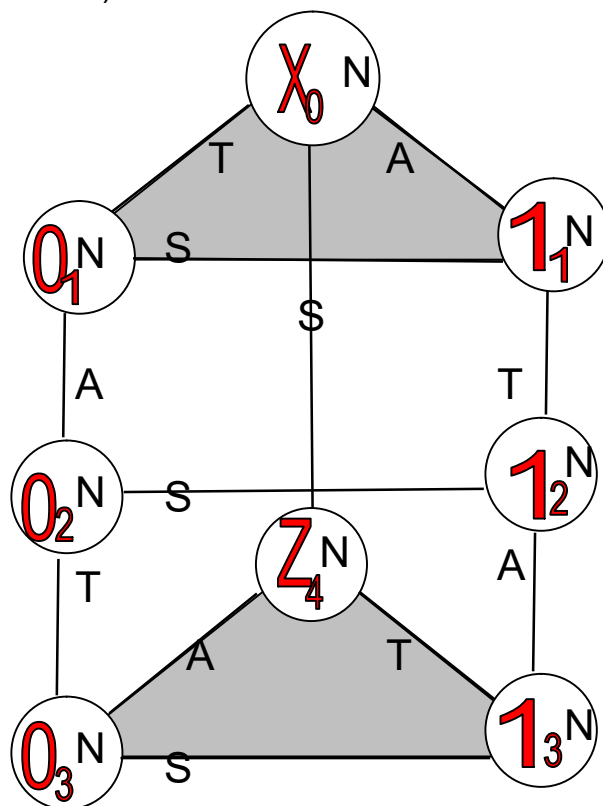
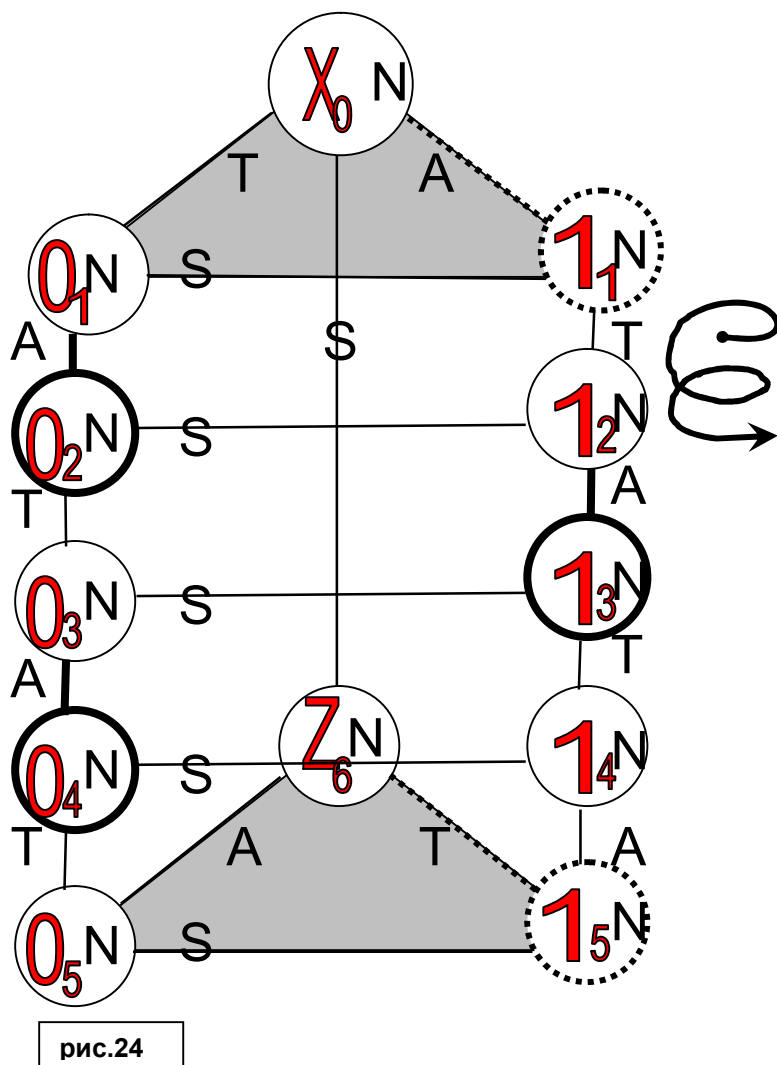


рис.23

4. Фрагмент, состоящий из звена 0 (эталонного объекта) и звена 1 будем называть *эталонной плоскостью*. Фрагмент, состоящий из последнего звена (результирующего объекта) и предстоящего ему звена будем называть *результирующей плоскостью*. Связь между вспомогательными объектами в каждом невырожденном звене будем называть *ректификационной связью*. Связь между эталонным объектом и результирующим объектом будем называть *абстрактным путем* или *связью между вырожденными звеньями*. Эту связь мы будем называть также, *связью между эталонной плоскостью и результирующей плоскостью*.
5. Определение символов связей (расчет связей) в треугольных призмах тривиален и проводится аналогично расчету связей в логическом тетраэдре. Также как и при расчете связей в тетраэдре каждая связь в треугольной призме определяется однозначно. Расчет связей для каждого звена призмы следует проводить от эталонного объекта (звено 0) к результирующему объекту (последнее звено). Связи между объектами в звене рассчитываются в соответствии с направлением обхода символов в треугольной призме.
6. Поскольку в качестве результирующего объекта был выбран вспомогательный объект, имеющий S – связь с эталонным объектом, то построенную треугольную призму будем называть *треугольной S – призмой*. Аналогично можно построить *треугольную T – призму* и *A – призму*.

10. Поведение треугольной призмы при увеличении количества звеньев.

Рассмотрим треугольную S – призму, изображенную на рис.24. В ней 7 звеньев. При этом звенья 0 и 1 – образуют эталонную плоскость, а звенья 5 и 6 – результирующую плоскость. Обе плоскости являются фрагментами треугольной призмы. Заметим, что звено 2 полностью повторяет звено 4. Следовательно, при дальнейшем увеличении количества звеньев мы получим многократное повторение звеньев со второго по третье включительно. То же самое можно сказать о поведении A – призмы и T – призмы. Они изображены на рис.25 и рис.26 соответственно. Таким образом, фрагмент треугольной призмы, состоящий из двух последовательных звеньев, при условии, что эти звенья не входят в эталонную или результирующую плоскость будем называть *плоскостью абстрактного вывода* или *плоскостью вывода*, поскольку эта плоскость параллельна абстрактному пути. (Напомним, что абстрактный путь - это связь между эталонным и результирующим объектами треугольной призмы).



Если объект 0 самого верхнего звена плоскости абстрактного вывода связан с предыдущим звеном символом A, то такую плоскость будем называть *плоскостью абстрактного A-вывода*. (Пример из рис.24. Плоскость из объектов $0_2 1_2 1_3 0_2$). Если объект 0 самого верхнего звена плоскости абстрактного вывода связан с предыдущим звеном символом T, то такую плоскость будем называть *плоскостью абстрактного T-вывода*. (Пример из рис.24. Плоскость из объектов $0_3 1_3 1_4 0_4$). Наконец, если объект 0 самого верхнего звена плоскости абстрактного вывода связан с предыдущим звеном символом S, то такую плоскость будем называть *плоскостью абстрактного S-вывода*. Заметим, что плоскости абстрактного S – вывода присутствует только в T – призме и A – призме. Плоскости абстрактного S – вывода отсутствуют в S - призме.

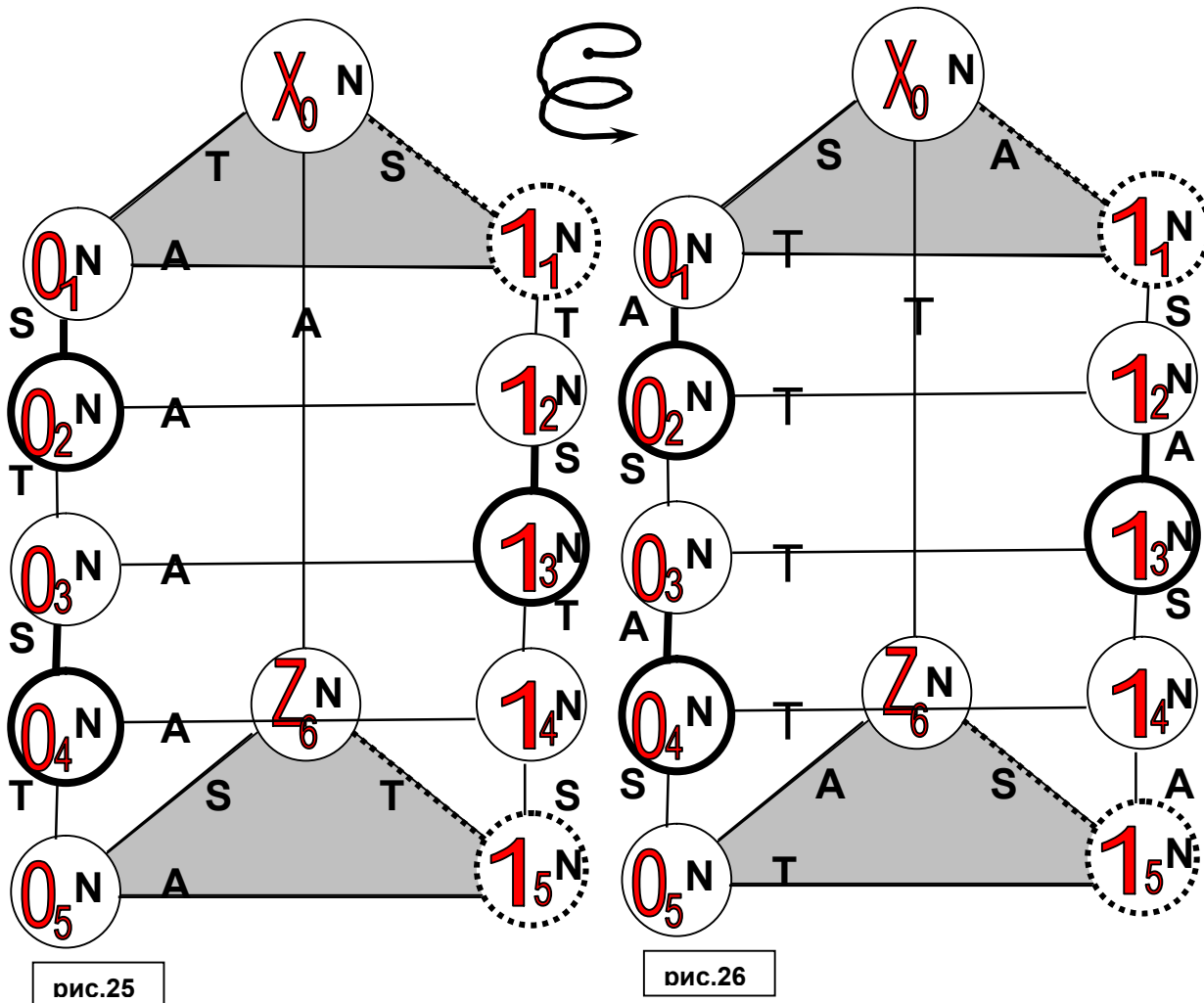
Также как и в цепочке нелокализованных бипирамид, каждое звено треугольной призмы повернуто на один шаг в соответствии с синтезирующим направлением обхода ($N \rightarrow T \rightarrow A \rightarrow S$). Это проиллюстрировано на рис. 24, 25 и 26 на примере объектов, обведенных толстой линией. Пунктирной, толстой линией указано продолжение этой тенденции для звеньев входящих в эталонную и результирующую

плоскости абстрактного вывода связан с предыдущим звеном символом S, то такую плоскость будем называть *плоскостью абстрактного S-вывода*. Заметим, что плоскости абстрактного S – вывода присутствует только в T – призме и A – призме. Плоскости абстрактного S – вывода отсутствуют в S - призме.

Также как и в цепочке нелокализованных бипирамид, каждое звено треугольной призмы повернуто на один шаг в соответствии с синтезирующим направлением обхода ($N \rightarrow T \rightarrow A \rightarrow S$). Это проиллюстрировано на рис. 24, 25 и 26 на примере объектов, обведенных толстой линией. Пунктирной, толстой линией указано продолжение этой тенденции для звеньев входящих в эталонную и результирующую

плоскости абстрактного вывода связан с предыдущим звеном символом S, то такую плоскость будем называть *плоскостью абстрактного S-вывода*. Заметим, что плоскости абстрактного S – вывода присутствует только в T – призме и A – призме. Плоскости абстрактного S – вывода отсутствуют в S - призме.

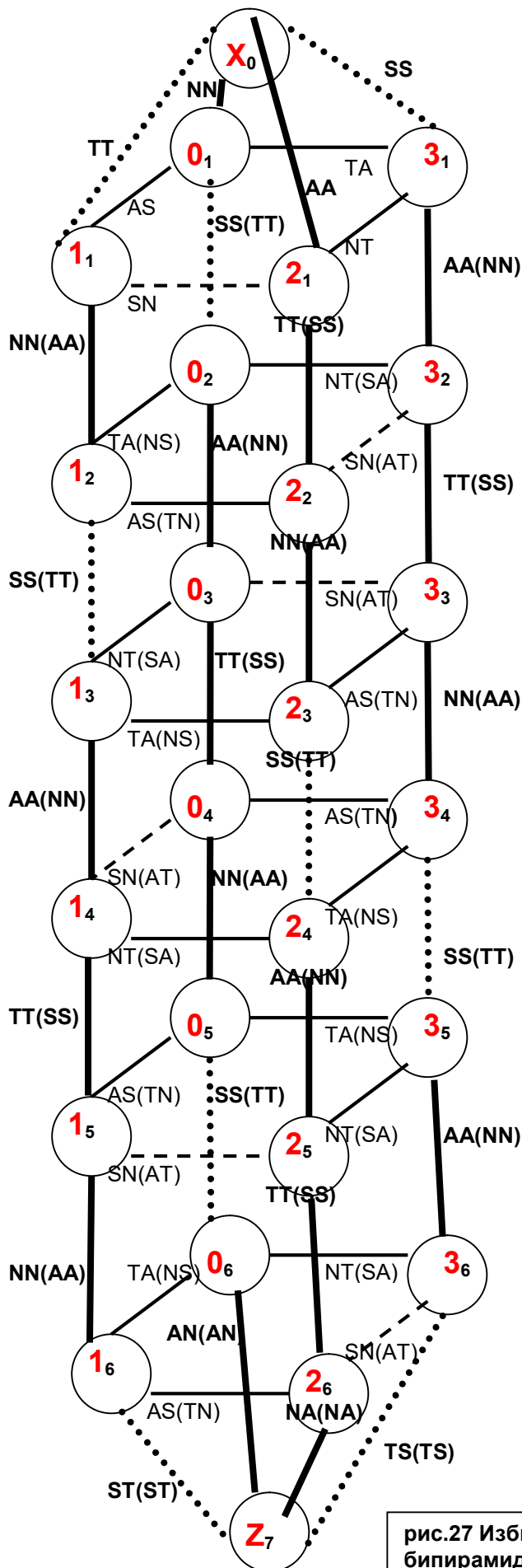
плоскости. Таким образом, все звенья цепочки повернуты относительно друг – друга и образуют спираль сдвига звеньев.



Предположение: Считаем, что дальнейшее исследование свойств треугольных призм приведет к объяснению современных понятий дискретности и непрерывности, с точки зрения логики четырех состояний, а также к определению «плоская проекция объемного построения», при условии введения в теорию понятия объема и пространства.

11 Локализованные цепочки бипирамид.

В главе 6 мы рассмотрели цепочки нелокализованных бипирамид на примере избыточной цепочки, состоящей из 8 звеньев (рис.12). Эта цепочка изображена на рис.27 в трехмерном представлении. Пунктирной линией указано направление сдвига звеньев относительно друг-друга (спираль) и вырожденное продолжение этой тенденции для уникальных звеньев. Каждая бипирамида, входящая в цепочку нелокализованных бипирамид может быть локализована в виде S – бипирамиды, либо A – бипирамиды. *Когерентной локализацией* - будем называть такую локализацию бипирамид цепочки, при которой все бипирамиды цепочки локализуются в соответствующие им синтезирующие бипирамиды (*когерентная S – локализация*) либо все бипирамиды цепочки локализуются в соответствующие им аналитические бипирамиды (*когерентная A – локализация*).



Вернемся к цепочке нелокализованных бипирамид, изображенной на рис.27. При когерентной локализации эта избыточная цепочка нелокализованных бипирамид распадается на две цепочки: на когерентную S – локализованную избыточную цепочку бипирамид (рис.28) и когерентную A – локализованную избыточную цепочку бипирамид (рис.29). Цепочки на рисунках 28 и 29 изображены в трехмерном представлении. *Некогерентной локализацией* будем называть такую локализацию всех бипирамид цепочки, при которой часть бипирамид цепочки локализируются в соответствующие им синтезирующие бипирамиды, а часть бипирамид цепочки локализируются в соответствующие им аналитические бипирамиды.

Локализованным звеном цепочки будем называть соответствующее звено соответствующей локализованной бипирамиды, входящей в цепочку.

Периодической локализацией будем называть такую некогерентную локализацию, при которой локализованные звенья цепочки подчиняются периодическому закону. То есть такую некогерентную локализацию цепочки, при которой каждый фрагмент из локализованных звеньев одного типа локализации (S – локализация или A – локализация), сменяется фрагментом из локализованных звеньев другого типа локализации (A – локализация и S – локализация, соответственно). При этом количество звеньев в любом S – локализованном фрагменте цепочки постоянно. Это количество звеньев будем называть *S – периодом цепочки*.

рис.27 Избыточная цепочка нелокализованных бипирамид из 8 звеньев.

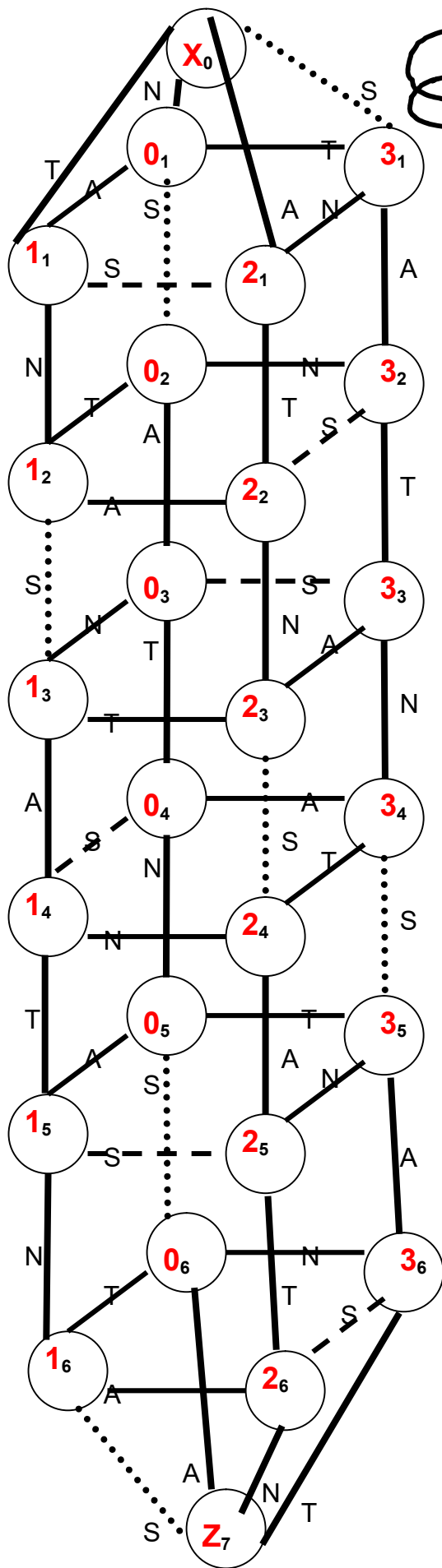


рис.28 Когерентная S – локализованная цепочка бипирамид

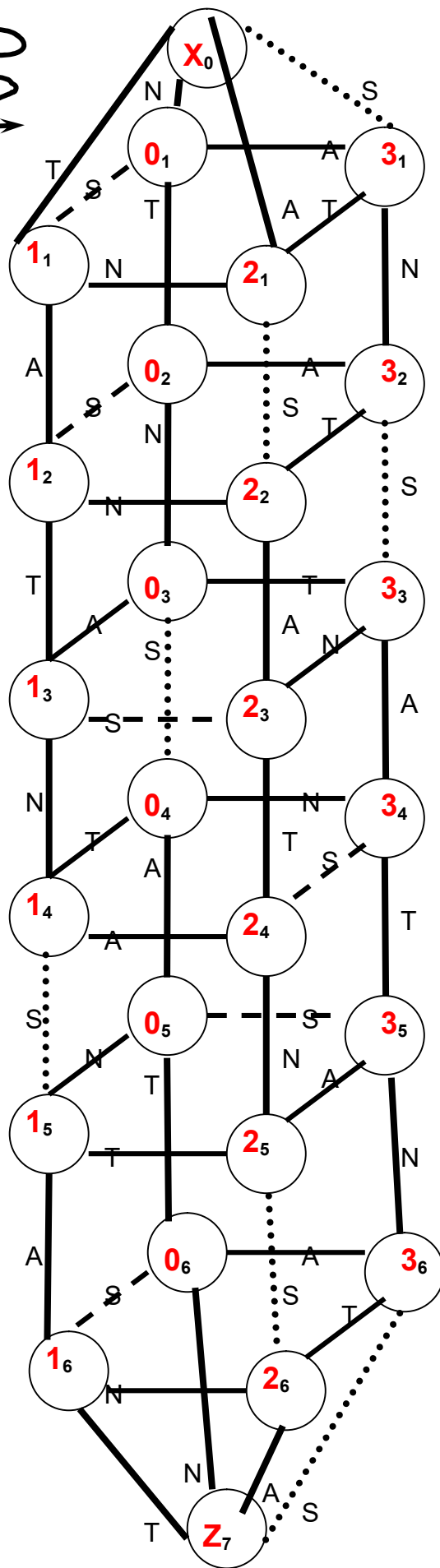


рис.29 Когерентная A – локализованная цепочка бипирамид

Соответственно и количество звеньев в любом A – локализованном фрагменте цепочки постоянно. Это количество звеньев будем называть A – *периодом цепочки*.

Периодическую локализацию цепочки, при которой S – и A – периоды равны между собой будем называть *эквипериодической локализацией*.

Периодическую или эквипериодическую локализацию цепочки, эталонный объект (звено 0) которой S – локализован, будем называть, соответственно, *периодической синусоидальной локализацией* и *эквипериодической синусоидальной локализацией*. Эквипериодическая синусоидальная локализация с периодом 1 изображена на рис 30. Данная локализация соответствует избыточной цепочке, изображенной на рис.27.

Периодическую локализацию цепочки, эталонный объект (звено 0) которой A – локализован, будем называть, соответственно, *периодической косинусоидальной локализацией* и *эквипериодической косинусоидальной локализацией*. Эквипериодическая косинусоидальная локализация с периодом 1 изображена на рис 31. Данная локализация также соответствует избыточной цепочке, изображенной на рис.27.

Заметим, что :

1. При эквипериодических локализациях с периодами равными единице мы получаем аналитическое направление витков спирали сдвига звеньев.
2. При когерентных S – и A – локализациях, а также в нелокализованной цепочке бипирмид мы получаем синтезирующее направление витков спирали сдвига звеньев.
3. В когерентной A – локализации и в эквипериодической косинусоидальной локализации с периодом 1 первое звено не теряет свое уникальное свойство, также кратность и остаток цепи не меняются. То есть неуникальный фрагмент не изменяется.
4. При когерентной S – локализации и при эквипериодической синусоидальной локализации с периодом 1 первое звено теряет свое уникальное свойство и многократно повторяется. Заметим, что при этом уникальный фрагмент цепи превращается в вырожденный фрагмент. Соответственно длина неуникального фрагмента цепи увеличивается на одно звено. В зависимости от длины цепочки меняется кратность и/или остаток ее неуникального фрагмента. Цепи, не имеющие уникального фрагмента, а имеющие только вырожденный фрагмент будем называть *вырожденными цепями*. При этом неуникальный фрагмент вырожденной цепи будем называть *невырожденным фрагментом*.

При периодических и эквипериодических локализациях с периодами более чем единица мы получим чередование S – когерентно и A – когерентно локализованных фрагментов в одной цепочке. Длины чередующихся фрагментов будут равны значениям соответствующих периодов. Под длиной фрагмента или цепи мы понимаем количество звеньев в этом фрагменте или цепи.

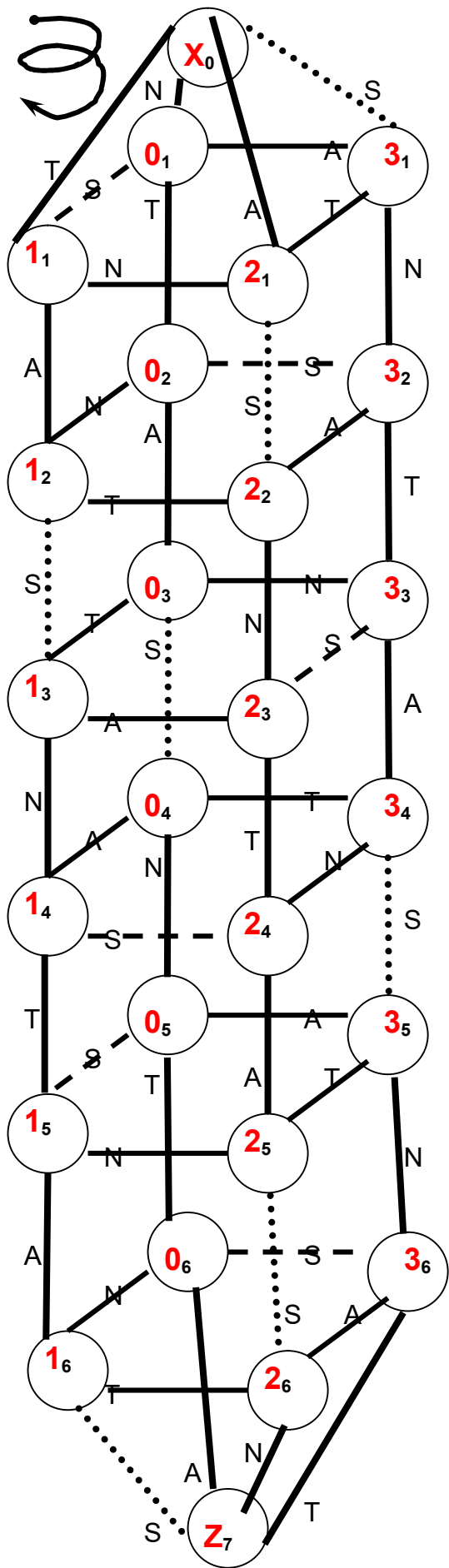


рис.30 Синусоидальная эквипериодическая цепочка с периодом 1

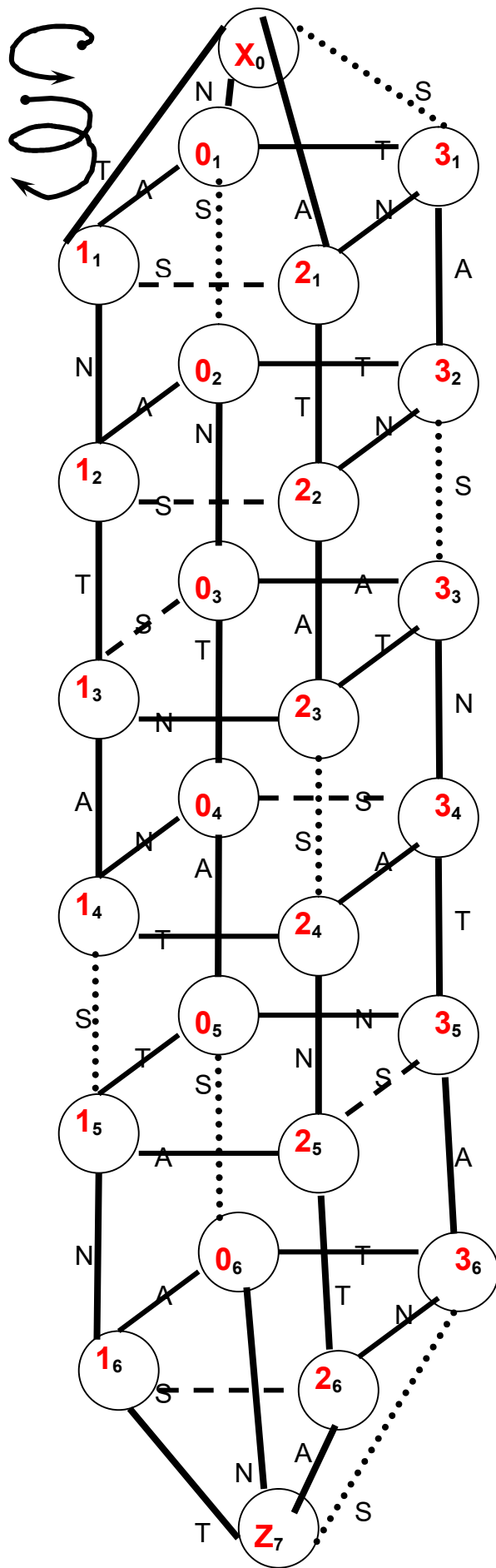


рис.31 Косинусоидальная эквипериодическая цепочка с периодом 1

12. Формальный объект. Формальные построения ромб(квадрат) и кольцо.

В формальной логике любое логическое выражение (предикат) может находиться в двух состояниях по отношению к любому другому выражению. Эти состояния в формальной логике называются, «ПРАВДА» и «ЛОЖЬ». По аналогии с объектом логики четырех состояний (далее четверичный объект) и абстрактным объектом изображение на рис 31 будем называть *отдельно – стоящим формальным объектом*. Формальный объект имеет две связи в соответствии с количеством его состояний. При этом состояние «ПРАВДА» будем указывать символом 1, а состояние «ЛОЖЬ» символом 0. В логике четырех состояний, символы 1 и 0, сопоставленные,

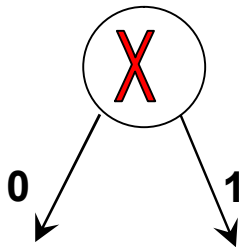


рис.31 Отдельно – стоящий формальный объект

соответственно, с состояниями «ПРАВДА» и «ЛОЖЬ» формальной логики будем называть *формальными символами*.

По аналогии с алгоритмами бипирамида и тетраэдр, объединяющими в логические построения, соответственно, четверичные и абстрактные объекты, для формального объекта предлагается свой алгоритм

При этом логические построения, состоящие из формальных объектов, будем называть *формальными построениями*.

Алгоритм ромб (квадрат).

1. Возьмем произвольное логическое выражение в качестве эталонного объекта (объект X на рис.32) формального построения

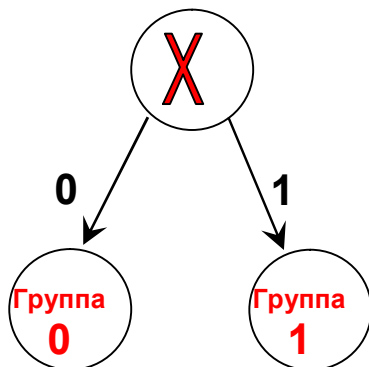


рис.32 Эталонный объект и его группы свойств

2. Любые другие логические выражения (считаем их свойствами объекта X) можно рассортировать на две группы (группа 0 и группа 1) в соответствии с их логическим состоянием («ЛОЖЬ», «ПРАВДА») по отношению к эталонному объекту X (рис 32).

3. Каждая из групп свойств (группа 0 и группа 1) на рис.32 – является самостоятельным объектом (объект 0 и объект 1 на рис 33, соответственно). Будем называть эти объекты вспомогательными объектами. Заметим, что одна из связей каждого вспомогательного объекта уже определена эталонным объектом

4. Объединим неспаренные связи вспомогательных объектов в результирующий объект Z (рис 34).

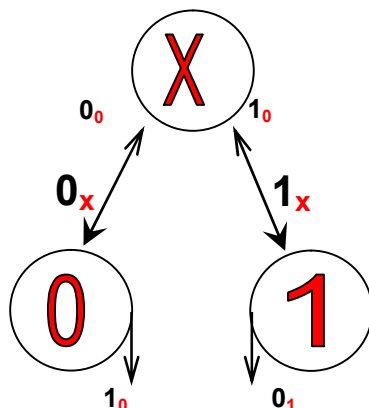


рис.33 Эталонный и вспомогательные объекты. Нижний индекс символа указывает на объект, к которому принадлежит связь

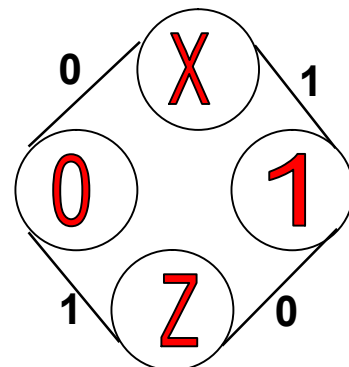
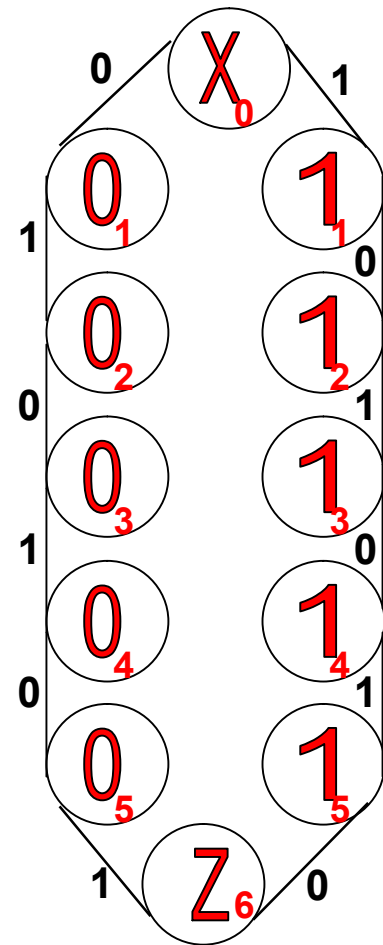


рис.34 Формальное построение ромб (квадрат)

Полученное логически-замкнутое построение будем называть *формальным построением ромб* или *формальным построением квадрат*.

Эталонный и результирующий объекты являются вырожденными звеньями ромба, а вспомогательные объекты образуют звено 1. При отделении результирующего объекта от ромба мы можем наращивать количество невырожденных звеньев и создавать цепочки из ромбов. Такие цепочки будем называть *кольцами* (рис.35). Заметим, что вспомогательные объекты ромба не имеют общих связей (паразитных связей). Таким образом вспомогательные объекты любого невырожденного звена не связаны между собой. Заметим также, что в кольце существуют только две логические траектории, связывающие эталонный объект и результирующий объект. Длины этих траекторий равны и эти траектории являются логическими путями. Заметим, что символы связей любого звена в кольце инвертированы, относительно соответствующих символов связей последующего звена и предыдущего звена.



Преобразование формального объекта в четверичный объект

Как введено в главе 1, существует соответствие между символами логики четырех состояний (N, T, A, S) и между формальными символами (0,1). Это соответствие может быть выражено как четыре равенства:

- (1)
- 1) $N = \overline{NE} (1 \text{ ИЛИ } 0) = \underline{0}$
 - 2) $T = \underline{1}$
 - 3) $A = \underline{\underline{0}}$
 - 4) $S = (1 \text{ ИЛИ } 0) = \underline{1}$

рис.35 Цепочка из 5 ромбов или 7 звеньев (кольцо из семи звеньев) Нижний индекс в обозначениях объектов указывает на звено, к которому относится объект.

Отметим, что символы формальной логики имеют порядок следования, при этом порядок $0 \rightarrow 1$ соответствует прямому порядку или синтезирующему направлению обхода символов логики четырех состояний (*прямой формальный порядок или синтезирующее направление обхода формальных символов*). Порядок $1 \rightarrow 0$ соответствует обратному порядку или аналитическому направлению обхода символов логики четырех состояний (*обратный формальный порядок или аналитическое направление обхода формальных символов*).

Заметим, что символы T и S соответствуют символу 1, а символы A и N соответствуют символу 0. Процесс выяснения соответствия между заданными символами (выражениями) логики четырех состояний и символами (выражениями) формальной логики будем называть *формализацией символов логики четырех состояний* (*формализацией выражений логики четырех состояний*). Соответственно, обратный процесс, при котором заданному символу (выражению) формальной логики ставится в соответствие символ (выражение) или комбинация

символов (группа выражений) логики четырех состояний будем называть *деформализацией символов (выражений) формальной логики*.

Рассмотрим отдельно- стоящий формальный объект. Такой объект изображен на рис.31. Деформализуем символы его связей.

Символ 1 деформализуется в виде двухсимвольной комбинации, которую можно записать как TS, если мы используем синтезирующее направление обхода ($N \rightarrow T \rightarrow A \rightarrow S$) либо ST, при выборе аналитического направления обхода ($S \rightarrow A \rightarrow T \rightarrow N$). Если мы не задаем направление обхода при деформализации, то такую *нелокализованную деформализацию* будем указывать как TS(ST), при этом комбинация в скобках используется при выборе аналитического направления обхода символов.

Аналогично, символ 0 можно деформализовать в виде комбинаций NA, AN либо NA(AN) в зависимости от выбранного направления обхода.

В общем случае:

1. деформализацию, при которой выбирается синтезирующее направление обхода символов логики четырех состояний будем называть *S – локализованной деформализацией* или *S – деформализацией*. При S – деформализации символов 1 и 0 мы получаем комбинации символов TS и NA, соответственно.
2. деформализацию, при которой выбирается аналитическое направление обхода символов логики четырех состояний будем называть *A – локализованной деформализацией* или *A – деформализацией*. При A – деформализации символов 1 и 0 мы получаем комбинации символов ST и AN, соответственно.
3. деформализацию, при которой не указывается направление обхода символов логики четырех состояний будем называть *нелокализованной деформализацией* или *SA – деформализацией*. При нелокализованной деформализации символов 1 и 0 мы получаем комбинации символов TS(ST) и NA(AN), соответственно.

Любой формальный объект, связи которого деформализованны, то есть определены через символы логики четырех состояний (N,T,A,S) или комбинации из этих символов будем называть *деформализованным объектом*. Отдельно – стоящие деформализованные объекты с разными типами деформализации изображены на рис.36.

Отметим отличие между процессами деформализации и локализации. При локализации мы выбираем символ из некоторого набора допустимых символов. При деформализации мы, в частности, заменяем символ формальной логики комбинацией символов логики четырех состояний. Таким образом, локализация является одной из стадий формализации, при которой определяется, какая именно комбинация символов логики четырех состояний ставится в соответствие формальному символу в соответствии с выбранным направлением обхода. Отметим также, что для однозначного определения символов связей деформализованных объектов, в общем случае (SA - деформализация) необходимы две *последовательные локализации*. Для S – деформализованных и A-деформализованных объектов одна из локализаций была проведена в процессе их деформализации, поэтому для однозначного определения их связей необходима только одна дополнительная локализация. При первой локализации следует исходить из направления обхода, реализованного в формальном объекте ($(0 \rightarrow 1) =$

$(N \rightarrow T \rightarrow A \rightarrow S) = S$ – локализация либо $(1 \rightarrow 0) = (S \rightarrow A \rightarrow T \rightarrow N) = A$ – локализация).

При второй локализации следует исходить из направления обхода, задаваемого для уже деформализованного объекта $((N \rightarrow T \rightarrow A \rightarrow S) = S$ – локализация либо $(S \rightarrow A \rightarrow T \rightarrow N) = A$ – локализация).

Таким образом, для однозначного определения символов связей деформализованного объекта необходимо знать направление обхода в формальном объекте и задать направление обхода в деформализованном объекте.

Деформализованный объект, связи которого однозначно определены, будем называть дважды локализованным деформализованным объектом или *2L-деформализованным объектом*.

Кроме того, S – деформализованный объект и A – деформализованный объект будем называть единожды локализованными деформализованными объектами или *1L-деформализованными объектами* (рис. 36b и рис.36c), а SA – деформализованный объект будем называть нуль – локализованными деформализованным объектом или *0L-деформализованным объектом* (рис 36a). При этом число, стоящее перед символом L будем называть *степенью локализации*.

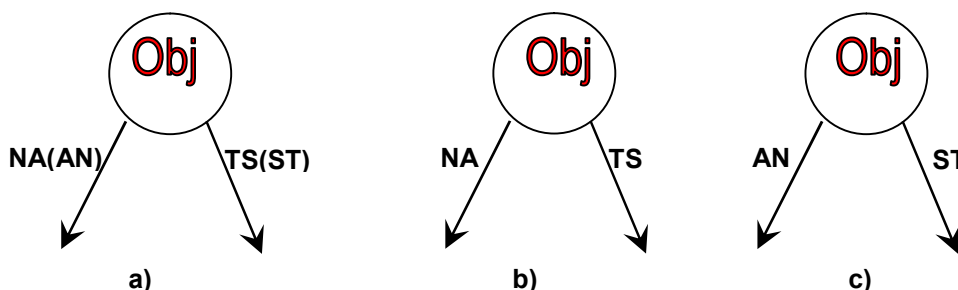


рис.36 Отдельно – стоящие деформализованные объекты:

- a) объект с нелокализованной деформализацией (0L- деформализованный объект)
- b) S – деформализованный объект (разновидность 1L – деформализованного объекта)
- c) A – деформализованный объект (разновидность 1L – д формализованного объекта)

Локализации, проводящиеся одна за другой, применительно к одному и тому же объекту или логическому построению будем называть *последовательными локализациями*. Например, двойная локализация представляет собой две последовательные локализации. В отличие от последовательных локализаций, *параллельными локализациями* будем называть локализации, проводящиеся по одному разу, применительно к каждому объекту в группе объектов или в логическом построении. Например, периодические и когерентные локализации цепочки бипирамид состоят, в частности, из набора параллельных локализаций всех звеньев (а значит и всех объектов), входящих в цепочку.

Всего возможно 4 типа двойных локализаций приводящих 0L – деформализованный объект в 2L – деформализованный объект, а именно:

1. Две последовательные S – локализации, то есть последовательная S – когерентная локализация. При этом символы связей формального объекта 0 и 1 переходят в символы N и T деформализованного объекта, соответственно.
2. Две последовательные A – локализации, то есть последовательная A – когерентная локализация, при этом символы связей формального объекта 0 и 1 также переходят в символы N и T деформализованного объекта, соответственно.
3. S – локализация и следующая за ней A – локализация, то есть последовательная эквипериодическая синусоидальная локализация с

периодом 1 или синусоидальная локализация.

При этом символы связей формального объекта 0 и 1 переходят в символы A и S деформализованного объекта, соответственно.

4. A – локализация и следующая за ней S – локализация, то есть последовательная эквипериодическая косинусоидальная локализация с периодом 1 или косинусоидальная локализация.
При этом символы связей формального объекта 0 и 1 также переходят в символы A и S деформализованного объекта, соответственно.

Таким образом, при однозначном определении связей формального объекта в символах логики четырех состояний, то есть при преобразовании формального объекта в 2L – деформализованный объект мы можем получить только два типа объектов (рис 37):

1. Объекты, у которых символы связей 0 и 1 деформализованы в символы N и T, соответственно. Такие объекты будем называть *когерентными* (рис 37a). В таких объектах направление обхода символов заданное для формального объекта совпадает с направлением обхода символов соответствующего деформализованного объекта (2L – деформализованного объекта).
2. Объекты, у которых символы 0 и 1 деформализованы в символы A и S, соответственно. Такие объекты будем называть *эквипериодическими* (рис 37b). В таких объектах направление обхода символов заданное для формального объекта противоположно направлению обхода символов соответствующего деформализованного объекта (2L – деформализованного объекта).

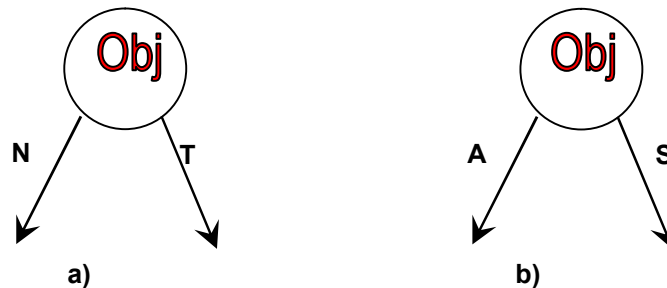


рис.37 Отдельно – стоящие 2L – деформализованные объекты:
а) когерентный объект б) эквипериодический объект

Заметим, что при преобразовании формального объекта в 2L-деформализованный объект символы 0 и 1 связей формального объекта ни при каких условиях не могут быть деформализованы в символы логики четырех состояний A(0) и T(1), соответственно либо в символы логики четырех состояний N(0) S(1), соответственно. Символы в скобках соответствуют формализациям символов логики четырех состояний. Группу символов A(0) и T(1) и группу символов N(0) и S(1) мы рассмотрим в главе формализация.

И наоборот, при преобразовании формального объекта в 2L-деформализованный объект символы 0 и 1 связей формального объекта могут быть деформализованы только в символы логики четырех состояний N(0) и T(1) (когерентный объект) либо в символы логики четырех состояний A(0) S(1) (эквипериодический объект), соответственно.

Таким образом процесс деформализации состоит максимум из трех стадий (0,1 и 2) и может порождать из формального объекта 5 деформализованных объектов с разной степенью локализации (от 0 до 2). А именно:

Алгоритм деформализация:

0. преобразование: формальный объект \rightarrow 0L – деформализованный объект.
При этом порождается один тип объектов:
SA – деформализованный объект (нелокализованный деформализованный объект на рис.36а).
1. преобразование: 0L – деформализованный объект \rightarrow 1L – деформализованный объект. При этом может порождаться два типа объектов:
S – деформализованный объект (рис.36b) и A – деформализованный объект (рис.36с).
2. преобразование: 1L – деформализованный объект \rightarrow 2L – деформализованный объект. При этом может порождаться два типа объектов:
когерентный объект (рис.37а) и эквипериодический объект (рис.37b).

Заметим, что алгоритм деформализация имеет структурное и количественное сходство с алгоритмом бипирамида. В алгоритме бипирамида из заданного эталонного объекта порождается 5 объектов, столько – же объектов порождается из заданного формального объекта при деформализации (количественное сходство). Кроме того, все объекты бипирамиды организованы в виде трех звеньев, как и объекты, порождаемые при деформализации организованы в виде трех стадий (структурное сходство).

Процесс объединения двух деформализованных объектов, имеющих одну степень локализации, но принадлежащих к разным типам деформализованных объектов будем называть *объединением деформализованных объектов в четверичный объект* или *объединением формальных объектов в четверичный объект*.

В общем случае, если не оговорено обратное, то оба объединяемых деформализованных объекта порождены из одного формального объекта.

таким образом, в четверичный объект объединяются:

1. когерентный и эквипериодический объекты – при этом образуется однозначно локализованный четверичный объект (рис.38).
2. s – деформализованный и a – деформализованный объекты – при этом образуется сдвоенный объект (нелокализованный объект) логики четырех состояний (рис.39). По значению связей этот объект соответствует результирующему объекту бипирамиды.
3. формальный объект и 0L – деформализованный объект – при этом образуется объект логики четырех состояний особого вида (рис.40). Две связи этого объекта выражены формальными символами и могут быть однозначно деформализованы, как на рис.41.

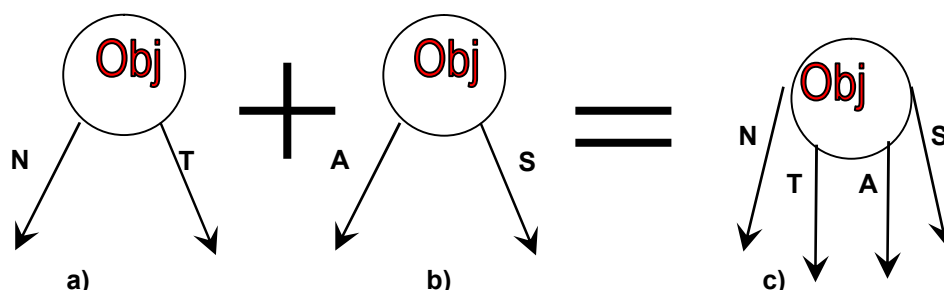


рис.38 Отдельно – стоящие 2L – деформализованные объекты когерентный объект а) и эквипериодический объект б) объединяются в четверичный объект с)

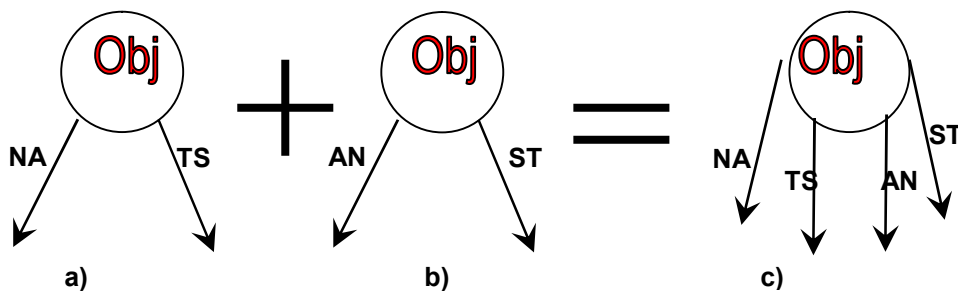


рис.39 Отдельно – стоящие 1L – деформализованные объекты s – деформализованный объект a) и a – деформализованный объект b) объединяются в четверичный объект c)

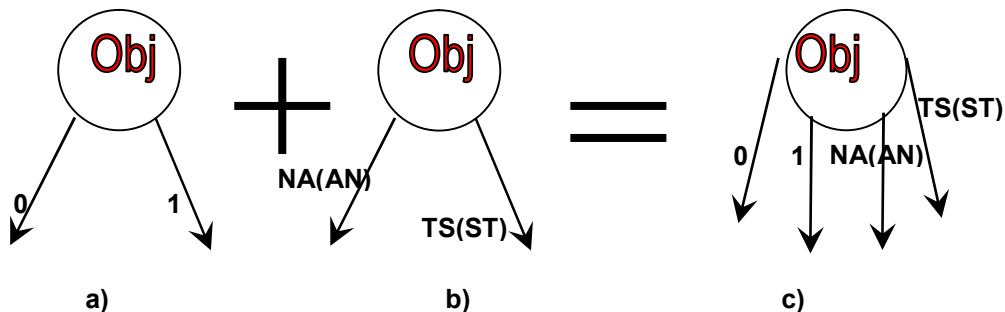


рис.40 Отдельно – стоящий формальный объект a) и 0L – деформализованный объект b) объединяются в четверичный объект особого вида c)

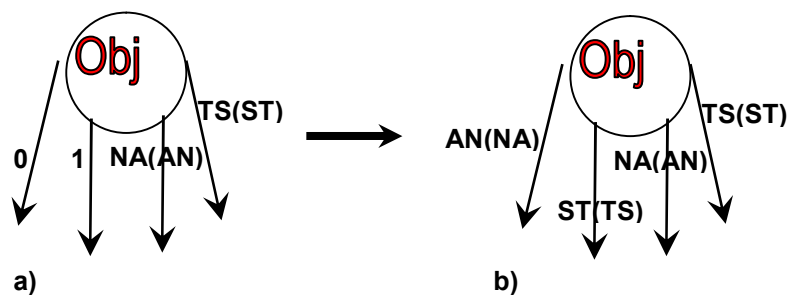


рис.41 деформализация связей четверичного объекта особого вида.

Процесс, обратный объединению будем называть *разделением четверичного объекта на формальные / деформализованные объекты*. Таким образом, четверичные объекты могут разделяться на деформализованные объекты тремя способами, соответствующими процессу объединения, рассмотренному выше. На рис.42 изображен процесс разделения четверичного объекта особого вида с полностью деформализованными связями.

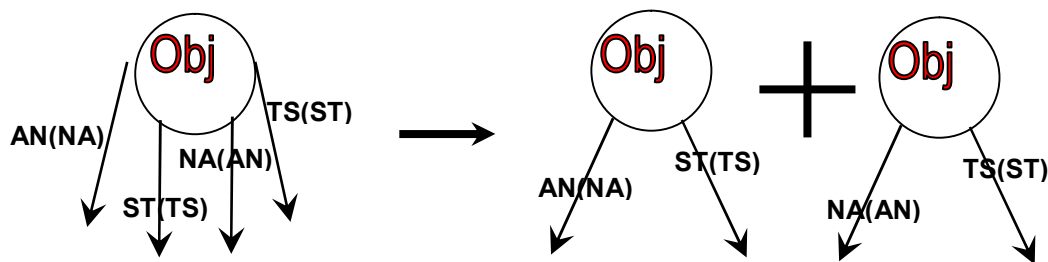


рис.42 разделение четверичного объекта особого вида на деформализованные объекты.

13. Абстрагирование формального объекта. Абстрактный формальный объект.

При отделении формального объекта от формального логического построения одна из его связей должна замкнуться на его внутреннюю реализацию для поддержания логического существования абстрактного формального объекта, по аналогии с абстрактным объектом логики четырех состояний. Вспомним, что при абстрагировании четверичного объекта на внутреннюю реализацию замыкается связь, описываемая символом N. В формальной логике этому символу соответствует формальный символ 0. Таким образом, при абстрагировании формального объекта на внутреннюю реализацию замыкается связь 0 (рис.43). Абстрактный формальный объект является логической константой ПРАВДА, имеющей только одну связь. Эта связь описывается символом 1.

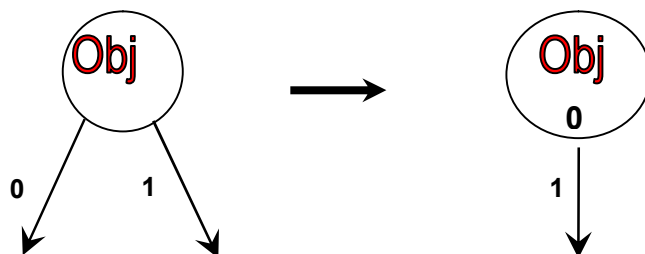


рис.43 абстрагирование формального объекта.

Поскольку формальный абстрактный объект имеет только одну связь, он может объединиться, выступая в качестве эталонного объекта, только с еще одним формальным абстрактным объектом (результатирующим объектом). Получаемое при этом логическое построение будем называть формальным абстрактным построением *отрезок* (рис 44). Отрезок не имеет свойства роста, и к нему не могут быть добавлены объекты. Построение отрезок является простейшим логическим построением. Оно состоит только из двух объектов: эталонного и результатирующего. В формальной логике, построение отрезок имеет смысл *равенства* одного предиката другому. В случае если наборы свойств объектов A1 и A2 совпадают, то построение отрезок имеет смысл *тождества*.

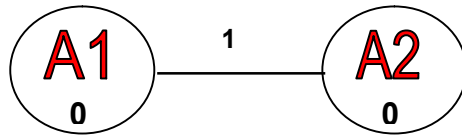


рис.44 абстрактное формальное построение.

Исходя из вышеизложенного, следует заметить, что возможно представить абстрагирование формального объекта, при котором на его внутреннюю реализацию замыкается связь 1. В этом случае объект является *константой ЛОЖЬ* формальной логики. Такой объект требует более широкого рассмотрения процессов абстрагирования, как для формальных объектов, так и для четверичных объектов. Данная задача выходит за рамки настоящей работы, поскольку требует создания теории симметричной данной (например, теории в которой синтезирующее направление обхода символов $N \rightarrow T \rightarrow A \rightarrow S$, считается аналитическим и в которой все логические построения созданы из заданного результирующего объекта, то есть в направлении от результирующего объекта к эталонному, то есть от следствия к причине). Заметим, что такая теория не согласуется с законами нашего мира. Тем не менее, мы рассмотрим формальные абстрактные логические построения с участием константы ЛОЖЬ. Таким образом, мы будем пользоваться константой ЛОЖЬ, считая ее невозможной (находящейся ВНЕ ПОНИМАНИЯ) с точки зрения теории логики четырех состояний. Константу ЛОЖЬ будем называть *анти-абстрактным объектом* формальной логики (рис.45), а логические построения с ее участием станем называть *анти-абстрактными*. При этом название анти-абстрактного объекта будем предварять символом минус «-».

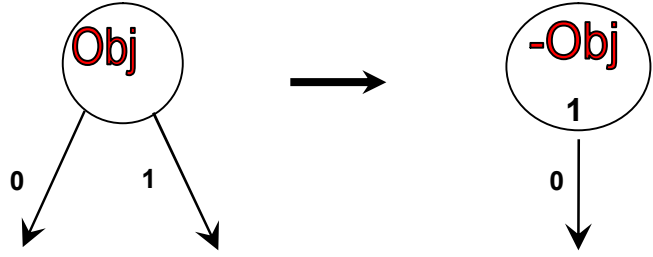


рис.45 анти-абстрагирование формального объекта.

По аналогии с алгоритмом отрезок анти-абстрактные объекты могут быть объединены в построение *анти-отрезок* (рис.46). Это построение обладает теми же свойствами, что и алгоритм отрезок.

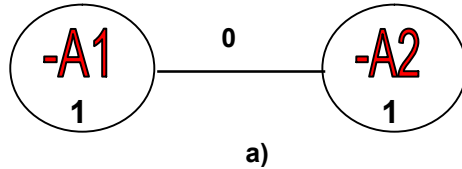


рис.46 анти-абстрактное формальное построение.

абстрактные и анти-абстрактные тождественные отрезки. Алгоритм символы логики четырех состояний.

1. Некоторую группу свойств будем считать объектом формальной логики.
2. Этому объекту дадим два обозначения X и Z и будем их рассматривать как два тождественных объекта формальной логики. При этом объект X будет выступать в роли эталонного объекта отрезка, а объект Z в роли результирующего объекта отрезка.
3. объектам X и Z будут соответствовать абстрактные объекты X_A и Z_A и анти-абстрактные объекты $-X_A$ и $-Z_A$, соответственно.
4. Теперь мы имеем четыре разных объекта (четыре константы). Два эталонных: X_A , $-X_A$ и два результирующих Z_A , $-Z_A$. Эталонные и результирующие объекты могут объединяться между собой в построение отрезков **всеми возможными способами**, поскольку представляют собой один и тот же объект (все константы тождественны между собой).
5. Таким образом, имеем, четыре вида тождественных отрезков, получаемых при объединении эталонного и результирующего объектов, разных видов абстрагирования (левая часть рис.47).

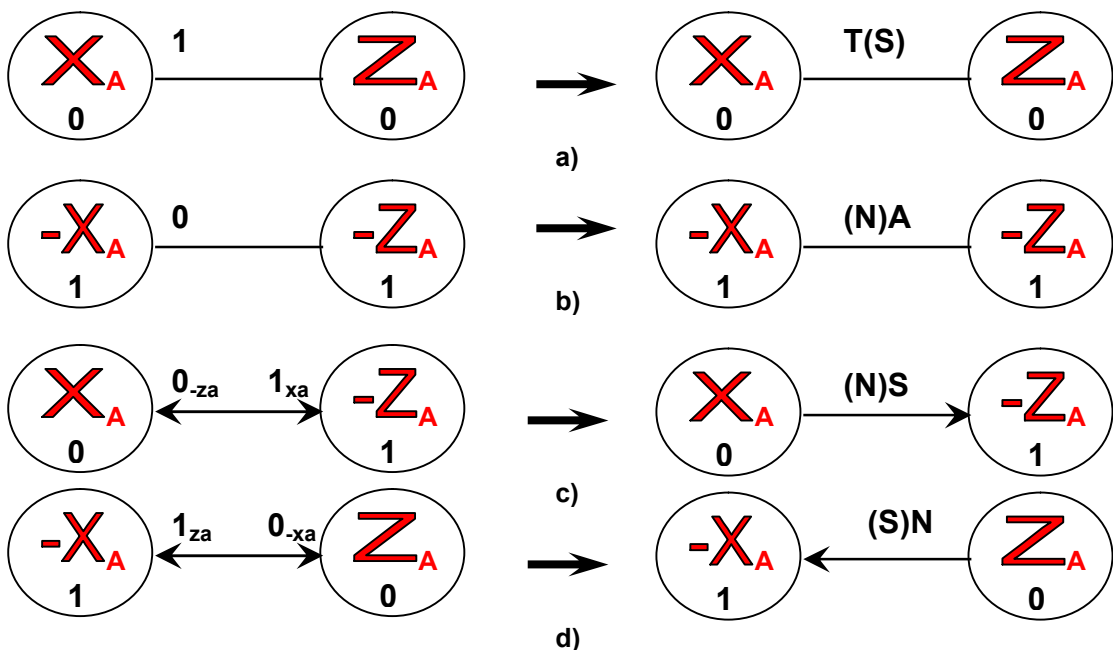


рис.47 абстрактный (a) и анти-абстрактные (b,c,d) отрезки . индексы символов связей в левой части рисунков c) и d) указывают на объекты, к которым принадлежат эти символы.

Выводы:

Отрезок а) и отрезок б) уже нами рассмотрены и являются абстрактным формальным построением и анти - абстрактным формальным построением, соответственно.

Интерес представляют связи между объектами в отрезках с) и д).

В зависимости от того, к какому объекту мы относим связь между ними, она меняет свое значение. Например, для отрезка с). Если мы считаем, что связь относится к эталонному объекту X_A , то связь выражается символом 1. Если мы считаем, что связь относится к результирующему объекту $-Z_A$, то связь выражается уже символом

0. Значение связи в отрезке d) также зависит от того, к какому объекту мы ее относим.

Таким образом, для связей в отрезках c) и d) не выполняются условия материальности, присущие всем связям в логических построениях. (гл.8.Абстрактные логические системы. Тетраэдр/выводы)

Связи в отрезках c) и d) – особые связи. В отличие от обычных связей, их значение зависит от способа рассмотрения. т.е для них выполняются условия идеалистичности.

Мы будем говорить, что эти связи имеют направления и являются *векторами*. Обычные связи не имеют направления и являются *скалярами*.

Первый метод определения направления вектора.

1. Направление вектора выражается через соответствующее направление обхода формальных символов (синтезирующее $0 \rightarrow 1$ или аналитическое $1 \rightarrow 0$). Для определения направления обхода необходимо определить комбинацию из двух формальных символов. Поэтому:
2. сначала мы считаем, что связь принадлежит результирующему объекту и, исходя из этого, определяем первый формальный символ связи.
3. Затем мы считаем, что связь принадлежит эталонному объекту и, исходя из этого, определяем второй формальный символ связи.
4. Таким образом, мы получаем комбинацию из двух формальных символов и, соответственно, направление обхода $0 \rightarrow 1$ или $1 \rightarrow 0$.

Например, вектор отрезка d) имеет аналитическое направление $1 \rightarrow 0$ (обратный порядок обхода формальных символов). Вектор отрезка c) имеет синтезирующее направление $0 \rightarrow 1$ (прямой порядок обхода формальных символов)

Второй метод определения направления вектора.

Направление вектора может быть определено, также, по формальным символам внутренней реализации эталонного и результирующего объектов. Вектор всегда направлен от символа 0 к символу 1. Если, при этом направление вектора совпадает с направлением от эталонного объекта к результирующему объекту, то этот вектор имеет синтезирующее направление, иначе аналитическое.

Предположения.

Так как связь между объектами, может быть рассмотрена, как самостоятельный логический объект и на основании вышеизложенного мы предполагаем, что:

1. условие материальности объекта (независимость свойств объекта от способа рассмотрения этого объекта) всегда является признаком того, что объект не имеет направления и является скаляром
2. условие идеалистичности объекта (зависимость свойств объекта от способа рассмотрения этого объекта) всегда является признаком того, что объект имеет направление и является вектором

считаем, что верно и обратное:

3. Если объект не имеет направления и описывается скалярной величиной – он материален (его свойства не зависят от способа рассмотрения).

4. Если объект имеет направление и описывается векторной величиной – он идеалистичен (его свойства зависят от способа рассмотрения).

Деформализация символов связей векторных и скалярных отрезков.

Деформализация связи каждого из отрезков изображена в правой части рис.46. При этом связь отрезка а) деформализуется в комбинацию T(S).

Локализацию этой комбинации следует проводить следующим образом: Символ T выбирается, если отрезок построен исходя из синтезирующего направления обхода символов (напомним, что такое направление задано для всех логических построений в нашей работе по умолчанию). В данном случае это означает также, что построение отрезка велось от эталонного объекта к результирующему объекту. Символ S выбирается, если построение велось в обратном направлении (от результирующего объекта к эталонному объекту). Это аналитическое направление обхода.

Другими словами, символ T выбирается при прикреплении результирующего объекта к заданному эталонному объекту (синтезирующее направление обхода символов). Символ S выбирается при прикреплении эталонного объекта к заданному результирующему объекту (аналитическое направление обхода символов).

Связи отрезков b),c) и d) деформализуются в комбинации (N)A, (N)S, (S)N. Поскольку эти отрезки являются анти-абстрактными построениями, то при заданном эталонном объекте у них реализуется второй символ в двухсимвольной комбинации связи. То есть при построении отрезка исходя из синтезирующего направления обхода символов, связь строится исходя из аналитического направления. Этот факт отражен в способе записи двухсимвольной комбинации связи каждого из отрезков. А именно: первый символ в двухсимвольной комбинации связи анти-абстрактного отрезка мы берем в скобки.

Вектора в отрезках c) и d) имеют неопределенность с точки зрения формальных символов либо находятся вне понимания с точки зрения формальных символов. Следовательно, для описания векторов в отрезках c) и d) принципиально допустимы два символа логики четырех состояний: N и S.

Поскольку в отрезке c) направление вектора совпадает с синтезирующим направлением обхода, то ему будет соответствовать комбинация (N)S.

В отрезке d) направление вектора противоположно синтезирующему направлению. То есть противоположно направлению от эталонного объекта к результирующему. Этому вектору будет соответствовать комбинация (S)N.

При конечной локализации всех отрезков мы получим две группы по четыре отрезка (рис.48). На рис.48 левая группа отрезков соответствует локализациям, при условии, что все четыре отрезка созданы исходя из синтезирующего направления обхода (направление от эталонного объекта к результирующему объекту).

Правая группа отрезков соответствует локализациям, при условии, что все четыре отрезка созданы исходя из аналитического направления обхода. То есть при прикреплении эталонного объекта к заданному результирующему объекту. Направление от результирующего объекта к эталонному объекту.

Анализ локализаций в правой группе отрезков выходит за пределы нашей работы, поскольку требует рассмотрения свойств логических построений, создаваемых в направлении от результирующего объекта к эталонному объекту.

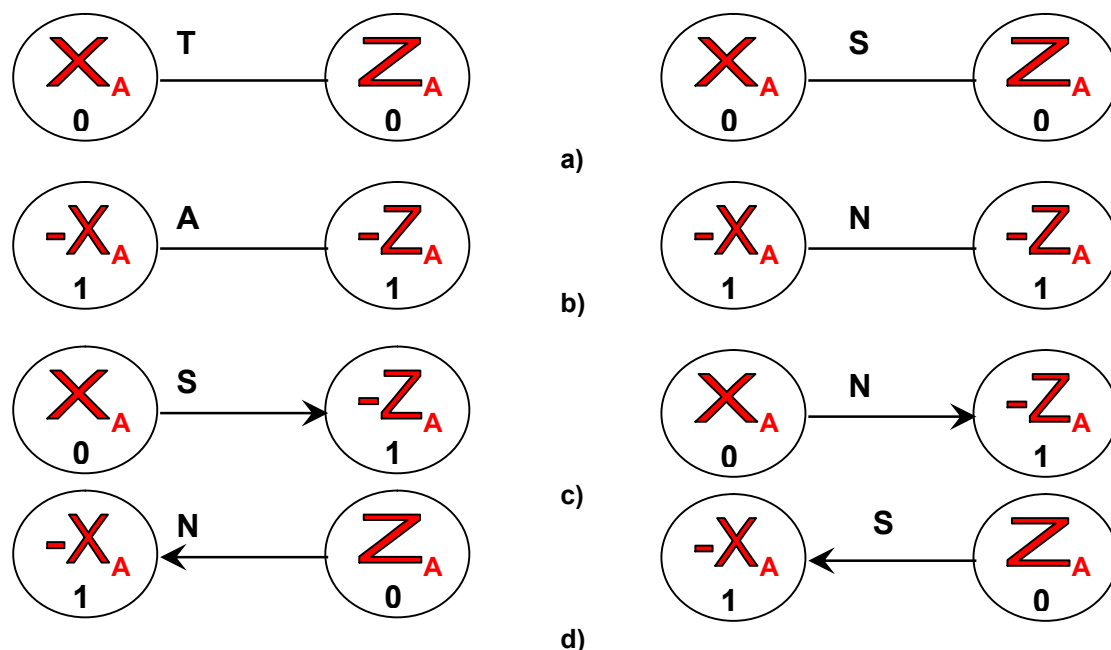


рис.48 конечные локализации отрезков.

Следует заметить, что в нашей работе все построения созданы в направлении от эталонного объекта к результирующему объекту. То есть в направлении от причины к следствию.

Локализации в левой группе отрезков являются представлением символов логики четырех состояний в виде (анти-) / абстрактных формальных логических построений.

Таким образом, при заданном синтезирующем направлении обхода и при условии тождественности эталонного и результирующего объектов в построении:

Символ Т – является абстрактным скалярным отрезком.

Символ А – является анти – абстрактным скалярным отрезком.

Символ S – является анти – абстрактным векторным отрезком синтезирующего направления.

Символ N – является анти – абстрактным векторным отрезком аналитического направления.

На примере логического построения отрезок показано, что синтезирующее направление обхода символов является также направлением от эталонного объекта к результирующему объекту (от причины к следствию). И наоборот, аналитическое направление обхода символов является также направлением от результирующего объекта к эталонному объекту (от следствия к причине).

Теоретико – практические применения логики четырех состояний.

1.Преобразования логических построений.

В предыдущих главах нами рассмотрены объекты и логические построения, которые можно разделить на четыре категории:

1. абстрактные формальные объекты (объекты с 1-ой связью). Логическое построение отрезок. При рассмотрении разных типов отрезков (2-ух векторов и 2-ух скаляров), получаем символы логики четырех состояний.
2. формальные объекты (объекты с 2-мя связями). Логическое построение квадрат. При увеличении количества звеньев получаем логическое построение кольцо.
3. абстрактные объекты логики четырех состояний (объекты с 3-мя связями). Логическое построение тетраэдр. При увеличении количества звеньев получаем логическое построение треугольная призма.
4. объекты логики четырех состояний (объекты с 4-мя связями). Логическое построение бипирамида. При увеличении количества звеньев получаем логическое построение цепочка бипирамид.

Также нами рассмотрены преобразования объектов из одних категорий в объекты других категорий (процессы абстрагирования и деформализации).

Интерес представляет процесс преобразования одних логических построений (алгоритмов) в другие логические построения(алгоритмы)

Преобразование треугольная призма из четырех звеньев (двух тетраэдров) – бипирамида.

Рассмотрим треугольную призму, состоящую из двух тетраэдров (рис. 49а). Она имеет сходство с бипирамидой по двум важным критериям: по количеству объектов и по отсутствию полной логической связанности.

Треугольная призма из двух тетраэдров, также как и бипирамида состоит из 6 объектов.

В отличие от тетраэдра, треугольная призма и бипирамида не является полностью связанными построениями (т.е. построениями в которых каждый объект имеет связь с остальными объектами).

Эти два момента являются основополагающими предпосылками, дающими возможность их взаимного преобразования.

Для преобразования в бипирамиду мы выбрали S – призму, поскольку, как показано в главе 9. «Цепи тетраэдров. Треугольные призмы», такая призма является наиболее рациональной с точки зрения логики четырех состояний. Преобразования других типов призм в бипирамиду является интересной задачей, которая позволит изучить и обобщить законы преобразования между абстрактными и неабстрактными логическими построениями.

Алгоритм преобразования треугольной призмы из четырех звеньев (двух тетраэдров) в бипирамиду.

1. Внедрим треугольную S-призму (рис.49а, 49b). В результате данная призма превратится в фрагмент неабстрактного логического построения с неспаренными связями N (рис. 49b). Всего таких связей 6.

2. Четное число разомкнутых связей N , принципиально, позволяет превратить фрагмент неабстрактного логического построения в замкнутое неабстрактное логическое построение. Другими словами, позволяет объединить абсолютно все разомкнутые связи полученного фрагмента.
3. Объединить разомкнутые связи можно только двумя способами (рис 49с, 49d). Штриховыми линиями указаны объединенные связи N .

Поясним наше утверждение

- 3.1 Все объекты фрагмента на рис. 49b имеют неспаренную связь N .
- 3.2 В соответствии с тезисом 1, каждый объект может находиться только в одном из четырех состояний по отношению к другому объекту. Следовательно, разомкнутые связи N можно взаимно объединить только у объектов не имеющих между собой спаренных связей. (заметим, что необходимое условие наличия таких объектов – отсутствие полной логической связанности в построении)
- 3.3 Нетрудно видеть, что такие объекты лежат на диагоналях трех плоскостей, $X_0 1_1 1_2 Z_3$, $0_1 1_1 1_2 0_2$, $Z_3 0_2 0_1 X_0$ призмы (обозначения плоскостей даны по названиям объектов, образующих плоскость).
- 3.4 Т.к. плоскости, представляют собой четырехугольники, то в каждой из них, имеется возможность провести только две диагонали. В соответствии с этим и определяется количество способов объединения всех разомкнутых связей N . Таких способа два. В первом случае объединяются связи N объектов: X_0 и 1_2 , 1_1 и 0_2 , 0_1 и Z_3 . Во втором случае объединяются связи N объектов: X_0 и 0_2 , 0_1 и 1_2 , 1_1 и Z_3 . Дополнительно заметим, что, так как каждая плоскость имеет общие вершины (объекты) с соседней плоскостью, то и каждая диагональ плоскости использует два объекта из соседних плоскостей. Поскольку каждый объект имеет только одну неспаренную связь N , то общие для соседних плоскостей вершины (объекты), уже имеющие замкнутую диагональную связь N не могут быть объединены с другими объектами.

1

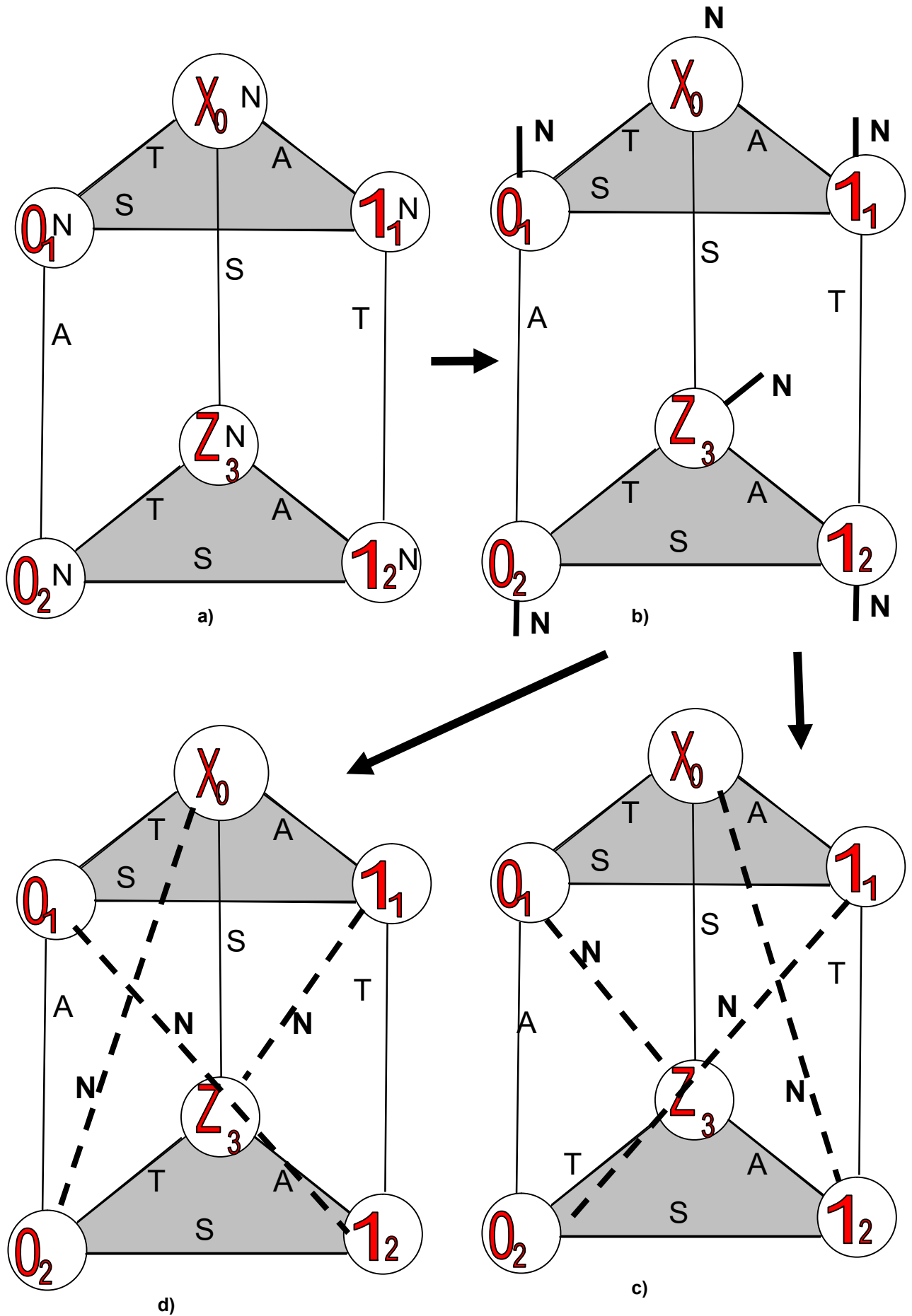


рис.49. а) треугольная S – призма б) фрагмент неабстрактного логического построения
 с), d) бипирамиды в форме призм с) X_1 – неабстрактная призма d) X_0 – неабстрактная призма

Таким образом, первое - же объединение разомкнутых связей N по диагонали одной из плоскостей, определит собой способ объединения всех остальных диагональных связей N .

- 3.5 Способ, при котором объединяются связи N эталонного объекта X_0 и вспомогательного объекта 1_2 , будем называть способом *диагонального $X1$ – объединения связей*. Соответственно второй способ, при котором объединяются связи N эталонного объекта X_0 и вспомогательного объекта 0_2 , будем называть способом *диагонального $X0$ – объединения связей*.
4. Замкнутое неабстрактное логическое построение, полученное из треугольной призмы с четырьмя звеньями (два тетраэдра) способом диагонального $X0$ – объединения связей N будем называть *неабстрактной $X0$ – призмой* (рис.49с). А аналогичное построение, полученное способом диагонального $X1$ – объединения – *неабстрактной $X1$ – призмой* (рис.49d).
5. Неабстрактные $X0$ – и $X1$ – призмы являются такими формами локализованных бипирамид у которых нет разграничения на эталонный, результирующий и вспомогательные объекты. Другими словами эти призмы являются бипирамидами у которых не определены звенья.

Общая методика определения звеньев в неабстрактных призмах

1. Выбрать новые эталонный и результирующий объекты в неабстрактной призме.

Пояснение:

В бипирамиде эталонный и результирующий объекты не имеют общих связей, в отличие от абстрактной треугольной призмы, где такая связь существует.

Поэтому в неабстрактной призме мы можем оставить эталонный объект и должны выбрать новый результирующий объект из вспомогательных объектов, не имеющих связей с эталонным. (*синтезирующий выбор*). Либо поступить иначе:

Оставить результирующий объект и выбрать новый эталонный объект, из вспомогательных объектов, не имеющих связей с результирующим объектом (*аналитический выбор*).

Третья возможность заключается в том, чтобы выбрать новые эталонный и результирующий объекты. (*смешанный выбор*). При этом смешанный выбор будет производиться из вспомогательных объектов призмы, не имеющих общих связей.

2. После синтезирующего, аналитического или смешанного выбора необходимо расположить объекты, связанные с эталонным объектом (это будут вспомогательные объекты) в соответствии с синтезирующим направлением обхода $N \rightarrow T \rightarrow A \rightarrow S$. Т.е. сначала изображается объект, имеющий связь N с эталонным объектом, затем изображается объект, имеющий связь T с эталонным объектом, и т.д. до объекта имеющего связь S с эталонным объектом.

3. Далее, необходимо обратить внимание на паразитные связи полученных вспомогательных объектов. При отсутствии какой-либо паразитной связи, между двумя последовательными вспомогательными объектами неабстрактная призма с данным выбором не может быть преобразована к бипирамиде. Следует вернуться к п.1 и сделать новый выбор.

4. Проверить наличие связи результирующего объекта со всеми вспомогательными объектами.

5. Проверить, имеет ли, каждый из 6 объектов полученной бипирамиды по четыре спаренных связи, соответствующих четырем символам N, T, A, S .

6. Определить тип локализации полученной бипирамиды (a – локализованная или s – локализованная бипирамида)

6. На основании методики изложенной в п.5 была получена s – локализованная бипирамида (рис.50a) из неабстрактной X0 – призмы при синтезирующем выборе объекта 1_2 (т.е. при выборе объекта 1_2 в качестве результирующего объекта бипирамиды, обозначено, как $Zb=1_2$ на рис.50a) Других бипирамид из неабстрактной X0 – призмы получить нельзя.
7. Также была получена а – локализованная бипирамида (рис.50b) из неабстрактной X1 – призмы при аналитическом выборе объекта 1_1 (т.е. при выборе объекта 1_1 в качестве эталонного объекта бипирамиды, обозначено, как $Xb=1_1$.на рис.50b). Других бипирамид из неабстрактной X1 – призмы получить нельзя.
8. При смешанном выборе (т.е. при выборе новых эталонного и результирующего объекта) в любой неабстрактной призме бипирамид получить нельзя.

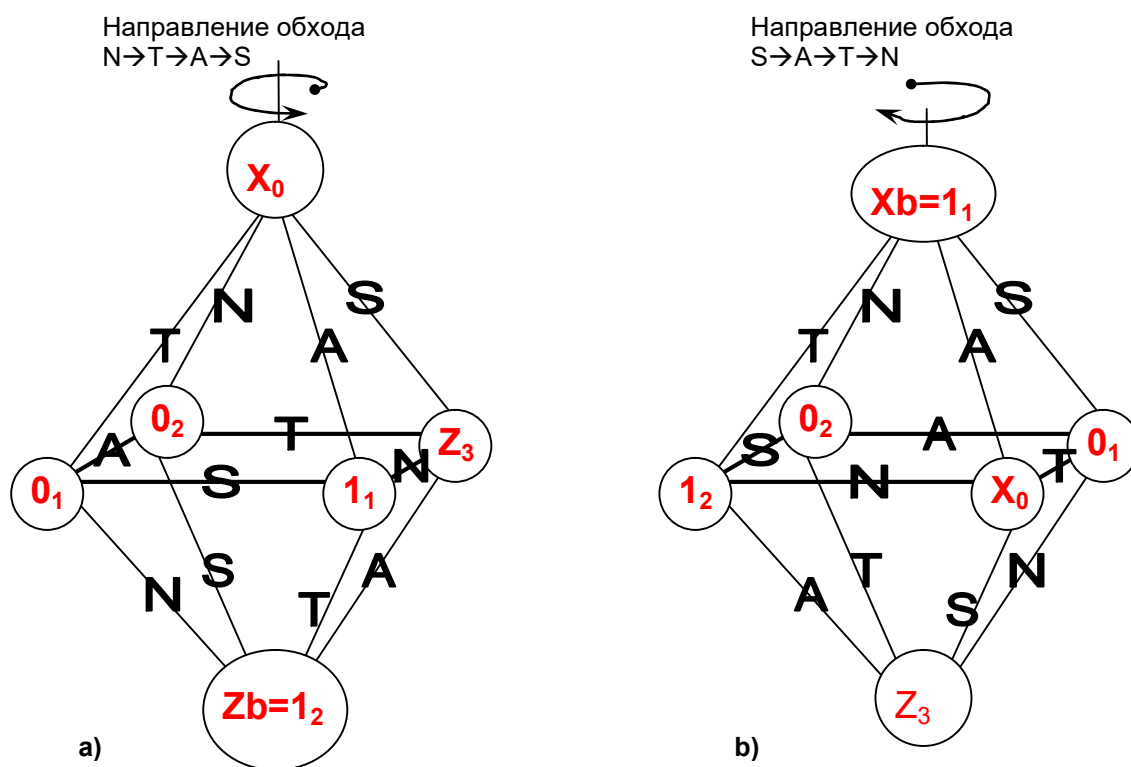


Рис.50
 а) s – локализованная бипирамида, образованная из X0 – неабстрактной призмы б) а – локализованная бипирамида, образованная из X1 – неабстрактной призмы

9. таким образом можно сделать вывод, что любая треугольная призма, состоящая из двух тетраэдров, может быть преобразована к локализованной бипирамиде только двумя способами:
 при диагональном X0 – объединении связей N – получаем s – локализованную бипирамиду
 при диагональном X1 – объединении связей N – получаем а – локализованную бипирамиду

Преобразование квадрат - тетраэдр.
Преобразование отрезки – квадрат.

Моделирование программной среды системы искусственного интеллекта на базе логики четырех состояний.

Предлагается следующая схема построения программной среды системы искусственного интеллекта (рис 1).

Данная система предназначена для преобразования текстовой информации.

В задачи системы входит:

1. преобразование некоторого входного потока символов алфавита естественного языка (входной текст) в новый выходной поток символов этого алфавита (выходной текст).
2. Наличие четкой семантики и (в меньшей степени) четкого синтаксиса в выходном тексте.
3. Близость семантики входного и выходного текста
- 4.

любой входной текст в выходной текст, имеющий несущий новые знания



—————▶ Поток данных в программной среде

.....▶ Поток управления в программной среде

15. Индуктивные последовательности слов.

Для выделения семантики каждого слова естественного языка предлагается простой итеративный алгоритм "индуктивные последовательности слов".

1. Ввод исходного слова (объект T)
2. Ввод антонима к T (A)
3. Ввод слова, объединяющего понятия T и A (S)
4. Ввод слова, определяющего реализацию понятия S (N)
5. Сохранение введенных слов (T, A и S) в массив M.
Запись S в T. Запись "пустой строки" в ячейки A и S.
6. переход к пункту 2

Практическое использование программы показывает, что для каждого заданного исходного слова получаются семантически-связанные последовательности, в среднем, из 6-8 слов.

массив из таких структур представляет собой семантическую сеть естественного языка, в которой смысл каждого слова определяется его отношениями с остальными словами.

При генерации достаточно большого числа последовательностей в массиве появятся пересечения слов. Таким образом структура семантической сети получает дополнительные разветвления, характеризующие собой неоднозначность понятий, используемых в естественном языке.

Каждая четверка слов, представляет собой неуникальное звено бипирамиды. Каждое слово в четверке - один из вспомогательных объектов.

16. Формализация. Иерархии уровней.

Рекурсивные уравнения анализа формальных логических символов.

Заметим, что для формализации символов логики четырех состояний N и S необходимо вычислить логическое выражение, в то время как, символы T и A формализованы по своему определению (см. равенства (1) в главе 12 и главу 1 «Символы»).

$$(1) \quad \begin{array}{l} 1) \mathbf{N} = \mathbf{NE} (1 \text{ ИЛИ } 0) = \underline{\mathbf{0}} \\ 2) \mathbf{T} = \underline{\mathbf{1}} \\ 3) \mathbf{A} = \underline{\mathbf{0}} \end{array}$$

$$4) S = (1 \text{ ИЛИ } 0) = \underline{1}$$

$$\underline{T=1; A=\text{not}(T)=0; S=f(T,A); N=\text{not}(f(T,A));}$$

$$\underline{f(T,A) = (T \text{ xor } A) \text{ xor } (T \text{ and } A) = T \text{ xor } A = T \text{ and } A}$$

$$\underline{S = (1 \text{ or } 0) = (1 \text{ xor } 0) \text{ xor } (1 \text{ and } 0)}$$

$$\underline{N = \text{not } (1 \text{ or } 0) = \text{not } ((1 \text{ xor } 0) \text{ xor } (1 \text{ and } 0))}$$

Группу символов T и A и группу символов N и S будем называть *уровнями*.

Группу символов T и A будем называть *простым уровнем*, поскольку при формализации символов этого уровня не требуется вычислять логических выражений.

Группу символов N и S будем называть *сложным уровнем*, поскольку при формализации каждого из символов этого уровня необходимо вычислить логическое выражение.

Символы логики четырех состояний в каждом уровне противоположны и образуют полную группу состояний («ПРАВДА», «ЛОЖЬ») с точки зрения формальной логики. Так как каждый символ уровня может быть определен на основании вычисления логической функции «НЕ» от другого символа уровня, то символы в уровнях будем также называть *микроуровнями*. При этом вычисляемый символ, является *сложным микроуровнем*, а символ, на основании которого, вычисляется сложный микроуровень, является *простым микроуровнем*.

Заметим, что определение каждого микроуровня, как простого или сложного, зависит от выбора направления обхода символов логики четырех состояний.

При выборе синтезирующего направления $N(0) \rightarrow T(1) \rightarrow A(0) \rightarrow S(1)$

T (1) – является **простым** микроуровнем **простого** уровня.

A (0) – является **сложным** микроуровнем **простого** уровня.

N (0) – является **простым** микроуровнем **сложного** уровня.

S (1) – является **сложным** микроуровнем **сложного** уровня.

Символы в скобках соответствуют формализациям символов логики четырех состояний.

Заметим что при переходе от простого уровня к сложному уровню формальные символы, соответствующих простых и соответствующих сложных микроуровней инвертируются. Таким образом, сложный уровень всегда инвертирован относительно простого уровня, также как сложный микроуровень всегда инвертирован относительно простого микроуровня.

При выборе аналитического направления $S(1) \rightarrow A(0) \rightarrow T(1) \rightarrow N(0)$ сложный и простой микроуровни поменяются местами в каждом уровне. Сами уровни не претерпевают изменений, поскольку обход символов логики четырех состояний при обоих направлениях (синтезирующем и аналитическом) начинается со сложного уровня.

Таким образом:

T (1) – является **сложным** микроуровнем **простого** уровня.

A (0) – является **простым** микроуровнем **простого** уровня.

N (0) – является **сложным** микроуровнем **сложного** уровня.

S (1)– является **простым** микроуровнем **сложного** уровня.

Рекурсивные уравнения анализа формальных логических символов.

$$m = \text{const}; m (\sim (0,1))$$

$$T_0 = m$$

$$A_0 = 1 - m$$

$$S_0 = T_0 * A_0$$

$$N_0 = 1 - (T_0 * A_0)$$

$$T_1 = S_0$$

$$A_1 = N_0$$

$$S_1 = T_1 * A_1$$

$$N_1 = 1 - (T_1 * A_1)$$

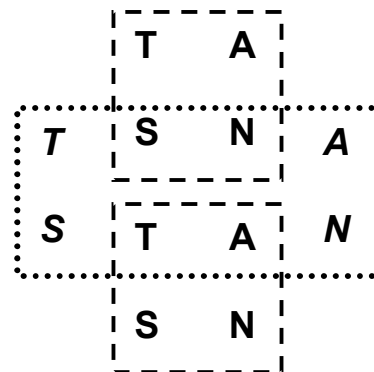
...

$$T_n = S_{n-1}$$

$$A_n = N_{n-1}$$

$$S_n = T_n * A_n$$

$$N_n = 1 - (T_n * A_n)$$



, а для) и символами формальной логики необходимо вычислить логическое выражение. Символы Т и А точно соотв

могут быть определены через состояния формальной логики (формальные символы). То есть через точное и ыми
олная логика оперирует четырьмя символами:

Т – тезис (ВЕРА, ПРАВДА, ОСЬ X)

соответствует значению логическая ПРАВДА формальной логики.

А – антитезис (ОТРИЦАНИЕ, ЛОЖЬ, ОСЬ Y)

соответствует значению логическая ЛОЖЬ формальной логики.



S – синтез (ПОНИМАНИЕ, НЕОПРЕДЕЛЕННОСТЬ, ОСЬ Z)
соответствует значению логическая ПРАВДА формальной логики, по отношению к правильности записи некоторого равенства. В полной логике символом S мы будем описывать неопределенные отношения между логическими выражениями.

N – нуль (РЕАЛИЗАЦИЯ, ВНЕ_ПОНИМАНИЯ, ОСЬ ВРЕМЕНИ)
соответствует значению логическая ЛОЖЬ в формальной логике, по отношению к правильности записи некоторого равенства. При этом мы считаем, что такое равенство записано не полностью. В полной логике символом N мы будем описывать все лежащие вне понимания и абсурдные отношения между логическими выражениями. Также об этих отношениях мы будем говорить, что они находятся в состоянии реализации.

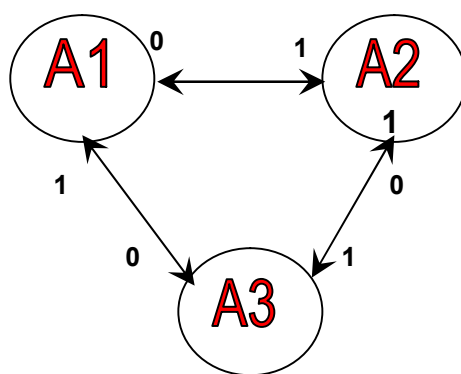


рис.43 абстрактное формальное построение.

11.Объединение полной логики в абстрактные логические построения.

Данная глава заканчивает краткое рассмотрение основ полной логики и систематизирует основные логические понятия и построения с применением теории полной логики.

СИМВОЛЫ

Символы могут быть представлены в виде отдельно стоящего объекта, как изображено на рис.26

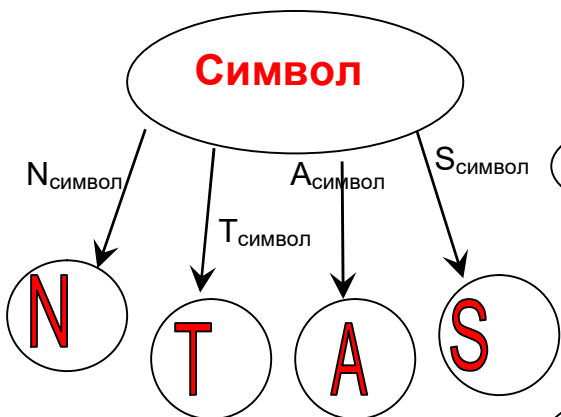


рис.26 объект символ

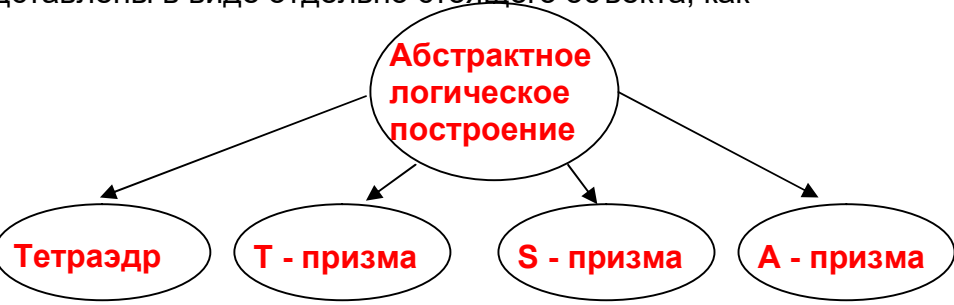


Рис.28

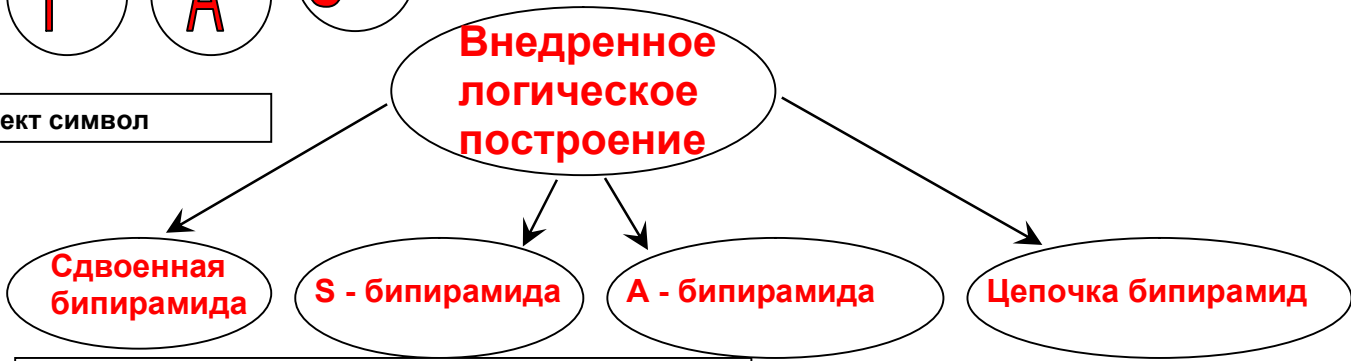


рис.27 объект внедренное логическое построение

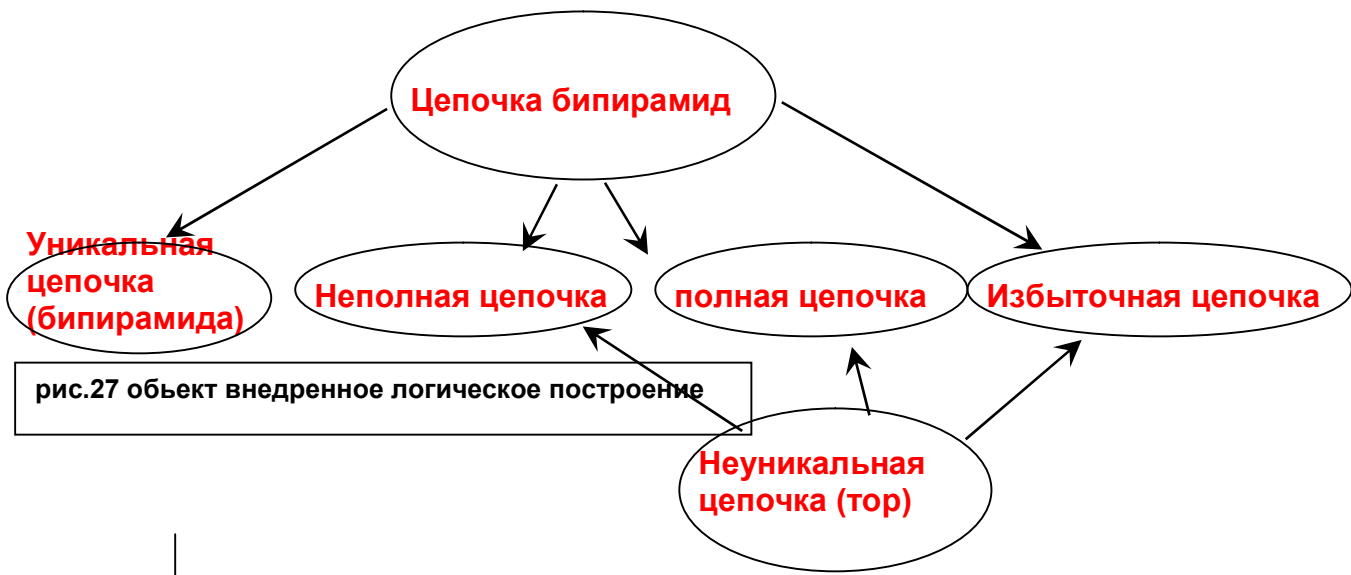


рис.27 объект внедренное логическое построение

А именно:
Символы NT, A, S
категории имеют Мы разработали несколько логических имеем

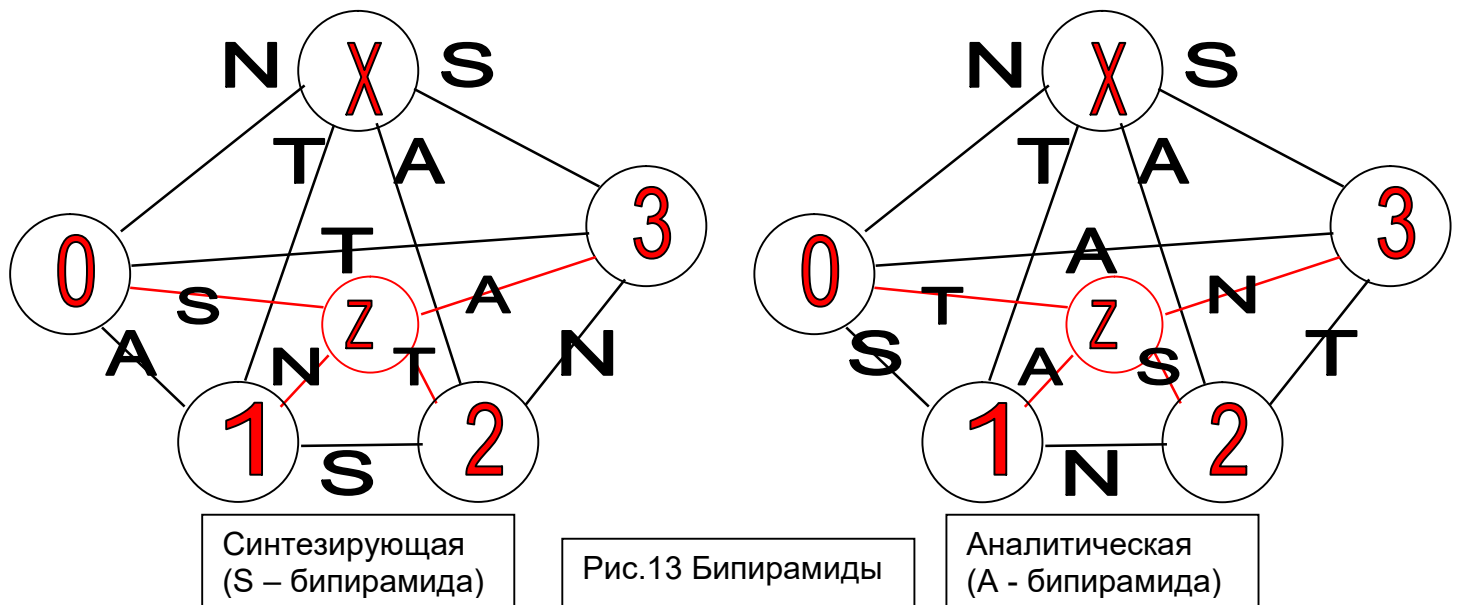


T₀ A₀ S₀

S

11. Преобразование S-бипирамиды в A-бипирамиду.

S-бипирамида и A-бипирамида изображены на рис.13. Вспомогательные объекты
Для преобразования S-бипирамиды в A-бипирамиду необходимо развернуть
Рассмотрим условия при которых S-
бипирамида превращается в A-бипирамиду. Нелокализованная бипирамида представляет собой
уникальную цепь из трех уникальных звеньев. Также как и в любой цепи бипирамид
звенья уникальной цепи сдвинуты друг относительно друга на один шаг.



Абстрактное логическое построение считается *логически-замкнутым*, если все объекты однозначно определены в символах полной логики и взаимосвязаны между собой. При этом используется только три связи. Связь N – не используется.

5. Пирамида – является логическим построением, все объекты которого определены в символах полной логики (т.е. имеют 4 связи N,T,A,S) и взаимосвязаны между собой. Будем называть такие построения *логически-замкнутыми системами*, поскольку каждый объект однозначно определен в терминах полной логики остальными объектами.

Заметим, что наиболее эффективная логическая траектория (логический путь) от эталонного объекта к результирующему объекту на полной и эффективной цепочке выражается шестью разрядами симметрии. На основании этого будем называть шестиразрядную карту чисел – *эффективной и полной картой чисел*.

Заметим, что путь на когерентно – локализованной цепочке никогда не будет выражен когерентной симметрией. Разрядность этой симметрии и соответствующей ей карте чисел будет равна количеству связей между звеньями цепочки.

/* этот кусок будет исключен и вынесен в отдельную главу

Количество операций по размыканию связи внутренней реализации (внедрению) объектов абстрактного построения будем называть *энергией внедрения*.

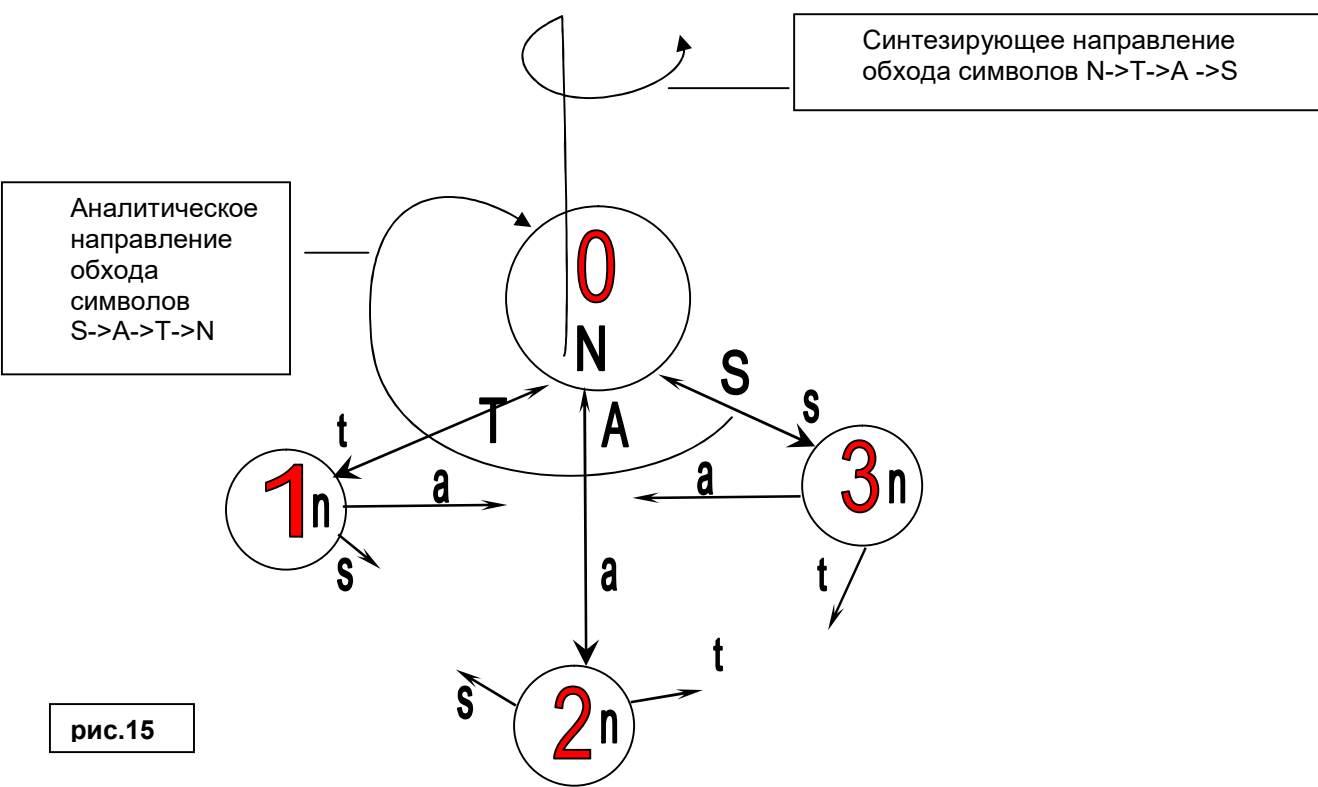
Существует 4 типа энергии в соответствии с 4 возможными способами рассмотрения реализуемого абстрактного объекта. Например, любой абстрактный объект, вне логического построения обладает N - энергией в размере 1. Так как ему необходима 1 операция внедрения. И при этом в качестве реализующего объекта может находиться только он сам, поэтому он обладает только N - энергией.

Разные типы энергии взаимосвязаны между собой через порядковые номера, соответствующих логических символов при синтезирующем и аналитическом направлениях обхода.

Разные типы энергии взаимосвязаны между собой через логическую структуру абстрактных объектов, реализующих друг – друга.

Массой объекта будем называть количество операций по разрыву логических связей, необходимое для абстрагирования объекта от логического построения.

***/ этот кусок будет исключен и вынесен в отдельную главу**



8.Пр
имен

рис.15

ение полной логики для построения компьютерных систем искусственного интеллекта.

Индуктивные последовательности.

Основной задачей при построении системы искусственного интеллекта является формализация естественного языка и построение базы знаний на основе этой формализации.

Предлагается следующая схема работы системы искусственного интеллекта:

- Ввод индуктивных последовательностей слов/понятий
- построение первичных ассоциативных связей (подсознание системы) на основе индуктивных последовательностей.
- Выделение часто используемых ассоциативных связей (вторичные ассоциативные связи или сознание системы)
- Выделение ассоциативных связей, необходимых для решения задачи, т.е. для построения логического вывода. Создание цепочки пирамид.
- Построение эффективной логической траектории на пирамиде.
- Самоанализ системы на основе истории построенных логических траекторий. Коррекция вторичных ассоциативных связей.

1. Построение базы знаний из слов естественного языка. Формализация слов естественного языка в терминах

, выдвигаем гипотезу, что для систем искусственного интеллекта максимально эффективными будут операции с симметриями на 7-разрядных картах чисел. Для доказательства этой гипотезы необходимо доказать, что полная и эффективная цепочка би - пирамид – является наиболее полной и эффективной логической структурой, связывающей эталонный и результирующий объекты. Мы показали полноту и эффективность связей этой структуры. Для полного доказательства необходимо определиться со смыслом вспомогательных объектов, каждого звена этой и аналогичных структур. И доказать соотнести

Вспомним, что вспомогательные объекты описывают эталонный объект в терминах полной логики (рис.1). Другими словами они являются *реализацией(N) полной логики*, эталонного объекта. Сам эталонный объект характеризуется *пониманием (S)полной логики* его внутреннего устройства.

Следовательно, важнейшее значение для эффективного логического вывода имеет однозначность распределения свойств исходной посылки по категориям полной логики (N,T,A,S).

Вернемся к карте чисел. Напомним, что карта чисел это фрактал, образованный из шаблона, изображенного на рис.7. Любая симметрия на карте чисел соответствует цепочке из локализованных би - пирамид. Шаблон фрактала – это четыре вспомогательных объекта пирамиды. Соответствие между квадрантами шаблона и каждым, из четырех вспомогательных объектов пирамиды устанавливается по связи вспомогательного объекта с эталонным. При этом каждая пирамида этой цепочки будет соответствовать одному разряду симметрии.

определенный в При анализе, а освободившиеся связи – будем считать

1. би-пирамида

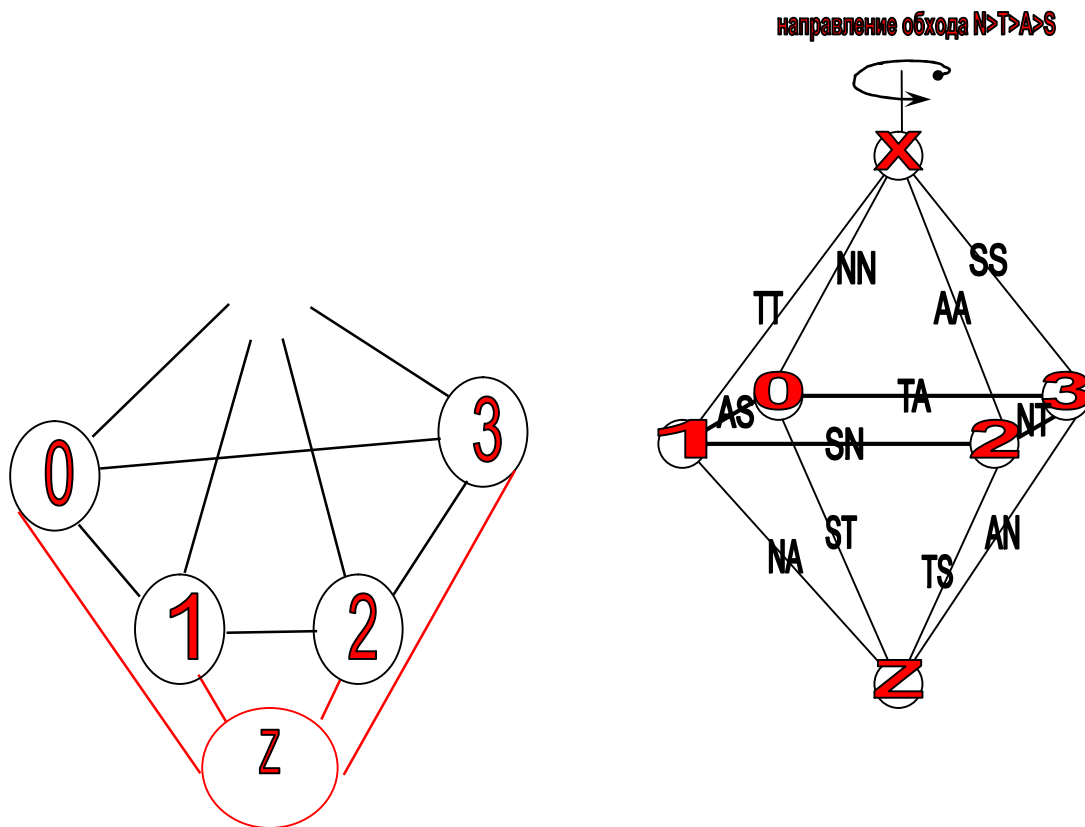


рис.5. Сдвоенная бипирамида

Гипотетический результирующий объект Z

примеру для тезиса Б, который иллюстрирует прямой порядок обхода символов: мы собираемся купить вещь:

- 0) Копим деньги и подыскиваем товар в магазине. Это состояние РЕАЛИЗАЦИИ (N)
- 1) Покупаем вещь и радуемся покупке. Это состояние ВЕРЫ (Т).
- 2) Начинаем все больше замечать, что купленная вещь не отвечает нашим ожиданиям от нее, т.к. в процессе эксплуатации возникают проблемы. Это состояние ОТРИЦАНИЯ (А).
- 3) Наконец мы приходим к ПОНИМАНИЮ (S) того, какой бы мы хотели видеть вещь. На основе этого ПОНИМАНИЯ переходим к п. 0.

Остановимся на пункте 3, а именно, на постулируемом нами переходе от состояния ПОНИМАНИЯ (S) к состоянию РЕАЛИЗАЦИИ (N). Этот переход аналогичен переносу в старший разряд для чисел. Например: $9+1=10$, при превышенувеличении значения некоторого счетчика.

$N(gjbcr)$

Сложение – вычитание симметрий (связь с пирамидой как перенос/заем)
 Анализ симметрий. Временисчисление в христианстве, как глобус чисел. Мах.
 Разрядность глобуса ≤ 7
 Идеальная замкнутая система (тетраэдр)

Гомогенный и гетерогенный тетраэдр
Двойственный тетраэдр. Внутренняя и внешняя структура тетраэдра.
N – как отображение внутреннего на внешнее. Отображение N.
Связи пирамиды - как объекты описанные полной логикой
Гомогенные и гетерогенные пирамиды

Тетраэдр:

Термины:

=====

внутренние элементы:

т - тезис.некоторое понятие

а - антитезис к т

с - синтез т и а.

пример:

т=студент

а=преподаватель

с=учебное заведение

tas4 - класс данных в который входят элементы т, а, с
также tas4 имеет методы для установки/чтения значений т, а, с
они имеют имена T(), A(), C().

Эти методы называются отображениями или внешними элементами

кроме того имеется внутренний элемент в (вариант),
который содержит код коммутации между т,а,с и T(), A(), C()

коды в:

=====

коды коммутации записываются в следующем порядке:

1. сначала берется шаблон "т_а_с", отражающий порядок следования внутренних элементов (тезис->антитезис->синтез)

2 в пустые места вписываются отображения соответствующие внутренним элементам

например :тT_aA_cC - задает такой порядок при котором

T()==т

A()==а

C()==с, прямой порядок

ограничение:

несколько внутренних элементов не могут отображаться в один и тот же внешний элемент

например: тT_aT_cC - запрещено т,а отображаются в T()

существует 6 комбинаций для в (базовые коды):

0.тT_aA_cC

1.тT_aC_cA

2.тA_aT_cC

3.тA_aC_cT

4.тC_aT_cA

5.тC_aA_cT

отображение тас4 в ТАС4(), отображение в
=====

для отображения всех внутренних элементов т,а,с,в .
введем понятие внешнего элемента В(), который
отражает состояние внутреннего элемента в.
остальные внешние элементы Т(), А(), С() - мы уже
имеем.

исследование В():
=====

итак мы добиваемся отображения всех внутренних
элементов тас4 во внешние. Что эквивалентно созданию
гипотетического зеркального объекта ТАС4.
допустим он у нас есть, тогда:

Т(), А(), С() - становятся внутренними элементами ТАС4
(но внешними для тас4)

по аналогии

т, а, с - становятся внешними элементами ТАС4
(но внутренними для тас4)

исходя из этого получаем алгоритм пересчета кода (в) в отображение В():

пусть у нас есть код тХ_аУ_сЗ, где ХУЗ одно
из всех возможных перестановок (ТАС, ТСА, АТС, АСТ, СТА, САТ):
итак:

- 1) имеем тХ_аУ_сЗ
- 2) меняем регистр букв ->Тх_Ау_Сз
- 3) меняем местами буквы в парах ->хТ_уА_зС
- 4) сортируем пары хТ, уА, зС таким образом,
чтобы порядок следования х, у, з совпадал с шаблоном "т _а _с".

например:

пусть в= тС_аТ_сА

определим В()

- 1) имеем тС_аТ_сА
- 2) меняем регистр буквТс_Ат_Са
- 3) меняем местами буквы в парах сТ_тА_аС
- 4) сортируем пары по шаблону: тА_аС_сТ
таким образом В(т)=тА_аС_сТ

применяя этот алгоритм (функция В()) ко всем базовым кодам получим,
что все базовые коды отображаются сами в себя кроме
базовых кодов 3 и 4, причем
В(3)=4 и В(4)=3

резюме:

объект тас4/ТАС4 двойственен и может быть рассмотрен
с одинаковым успехом, как набор тас4(т,а,с,в) или

как набор TAC4(T(), A(), C(), B()).

соображения:

tas4 всегда может распадаться на 2 различные tas4

tas4 с неполными отображениями.

=====

мы считаем что 0 можно заменить на (v) или V() исходя из определения 0.

если все прочие числа имеют вполне определенное значение, то 0 это число, которое обозначает отсутствие значения.

мы видим, что и (v) не имеет какого либо значения в терминах т,а,с (это просто комбинаторный элемент) и поэтому его удобно использовать в качестве термина 0.

ПОСТУЛАТ:

таким образом считаем, что 0 - это комбинаторный элемент, который определяет структуру, а не значение.

рассмотрим tas4 в которой один внутренний элемент не имеет отображения (далее мы будем говорить, что он отображен на B)

рассмотрим коды в этого объекта (внутренний неполный код N1 (ВНК1))

1) т->В 2) а->В 3) с->В

0.тВ_аТ_сА 0.тТ_аВ_сА 0.тТ_аА_сВ

1.тВ_аТ_сС 1.тТ_аВ_сС 1.тТ_аС_сВ

2.тВ_аА_сТ 2.тА_аВ_сТ 2.тА_аТ_сВ

3.тВ_аА_сС 3.тА_аВ_сС 3.тА_аС_сВ

4.тВ_аС_сТ 4.тС_аВ_сТ 4.тС_аТ_сВ

5.тВ_аС_сА 5.тС_аВ_сА 5.тС_аА_сВ

0) в->В

особый случай,

включает в себя базовые коды коммутации.

(см. выше)

заметим, что ВНК1 может быть однозначно преобразован к базовому коду. достаточно определить незадействованное отображение из набора T(),A(),C()

например:

тВ_аТ_сА ==>задействованы Т и А, значит свободно С=>

тС_аТ_сА (базовый код 4)

назовем функцию, работающую по этому алгоритму индукцией внутреннего неполного кода N1 (ИВНК1()).

теперь построим отображения V() для ВНК1,

исходя из алгоритма B():

в B()
тВ_аТ_сА =>тА_аС_вТ
.
.
.
.
тС_аА_сВ =>вС_аА_сТ

аналогично назовем вновь полученные коды -
- отображаемыми неполными кодами N1 (ОНК1).

заметим, что ОНК1 также может быть однозначно преобразован к базовому коду,
исходя из определения шаблона (т_а_с).

например:

тА_аС_вТ ==>тА_аС_сТ (базовый код 3)

назовем функцию, работающую по этому алгоритму индукцией
отображаемого неполного кода N1 (ИОНК1()).

планы на будущее:

- 1) ввести понятие комплекса из тас4
- 2) ввести понятие логически замкнутого комплекса из тас4
(все элементы (т,а,с,в,Т(),А(),С(),В()) связаны)
отсюда следует также понятие логической массы
- 3)на основе 2 ввести понятие логически замкнутой единичной тас4

4)

дедукцию вводить через ((в)B()) иртуализацию (т,а или с)
или (Т(), А(), или С())

виртуализация происходит не случайно а на
основе статистики или требования ?логической замкнутости? комплекса из тас4

====замечание=====

следует учесть, что

в тас4 с в=ВНК1 (1), 2), 3))

определить номер кода (в) через B() сложно,

поскольку на B() проецируется т, а или с.

??

номер кода (в) легко определить через

незадействованное отображение Т(), А() или С(),

которое и является отображением (в)

в случае логической замкнутости тас4

??

====

==замечание=====

ОНК1 имеет хотя бы одно отображение спроецированное на
внутренний элемент в. Таким образом тас4,
имеющее в=ОНК1 содержит в отображениях,
как (в) так и B()

????????????????????

неверно, в случае логической замкнутости тас4,
поскольку тогда на В() спроецировано не (в)
а (т,а или с)

????????????????????

=====

Математический аппарат, лежащий в основе всех современных интеллектуальных систем был создан в тридцатых годах двадцатого столетия в теоретических работах ряда выдающихся ученых математиков.

Булля, и Геттингенского университета. На основе логики Булля, разрабатываемой им, в качестве теории «не привязанной к чему либо определенному» базируется вся современная цифровая электроника.

Эта очень простая теория была развита и дополнена такими выдающимися учеными-теоретиками как Гильберт и Бернайс в их основополагающей работе «Основания математики». Выход «Оснований математики» ознаменовал собой прорыв в области

создана в Д.Фон Неймана и многих других первопроходцах.

В настоящее время основополагающее значение имеют интеллектуальные программные и аппаратные системы. То есть такие системы, уровень автоматизации которых приближен к уровню знаний и умений человека в широком диапазоне задач. Если в прикладном, то есть производственном, отношении уровень автоматически действующих систем гораздо выше возможностей человека, то в областях, связанных с интеллектуальной, в частности творческой деятельностью такой прогресс менее очевиден. Данная проблема, очевидно, сводится к проблеме узкой специализации автоматически действующих систем в четко определенном круге задач.

контого, что научить машину делать, что либо гораздо проще, чем научить ее думать. Прогресс достигнутый в области автоматизации рутинных, не творческих, во многих отношениях гораздо более. Отдельным и важным,

