НОТАЦИЯ СВЯЗАННЫХ ДАННЫХ BrainFogicTag 1[.] BFT 0[0]

ДОПОЛНЯЮЩИХ КОНТЕКСТ ЕСТЕСТВЕННОГО ТЕКСТА

Гудаев О.А.

**Введение.** Общей целью создания искусственной нотации BFT является улучшение коммуникации между носителями различных формальных систем. Перечислим интегральные характеристики такого процесса коммуникации: неоднозначность понимания констант формальной системы; трудность перевода между искусственными языками, процессов вывода; сложность описания процесса конструирования формальных выражений. Все указанные проблемы порождают противоречия в интерпретации результатов, полученных автоматически. Естественный текст не содержит таких трудностей – он всегда неоднозначен для параллельных конструкций и имеет не точную, а достоверную интерпретацию. Но для его размещения на носителе человек тратит драгоценное время на фиксацию и ресурс на размещение в пространстве носителя. Когда человек освоил компьютерный набор естественного текста, то возник соблазн дополнить текст логическими конструкциями, которые удобны для осмысленного чтения и одновременно пригодны для машинного анализа. Уточним, соблазн возникает, когда появляется возможность мгновенно и без затрат копировать в аппаратную память фразу и многократно извлекать её дубликат. Расширить контекст естественного текста возможно, только если вписать в него выражения метаописания, настолько выразительные, чтобы они воспринимались как часть текста.

Перечислим четыре цели расширения текста искусственной нотацией. Во- первых, занять нишу простого описания онтологий контента Интернета, когда, следуя хронологической траектории, развивается формат XML и JSON, переходя от формата JSON-LD к нотации BFT. Во-вторых, предложить формат вкраплений в текст на естественном языке связанных онтологических данных. В-третьих, протоколировать автоматическое доказательство теорем выразительным описанием процесса принятия решений в интегральных системах, когда прозрачно комментируется конструирование компьютерных объектов по спецификации. В-четвёртых, не бороться с клиповым мышлением, а направить адекватное общение с робототехническими системами в интеллектуальное русло.

**Исследования.** Скелет контента Интернета 1[.] W3C.org 0[0] – это текст на естественном языке. Вкрапление специальных цепочек комментариев 1[.] blockchain 0[0] в последовательность символов делает текст связанным с

онтологической разметкой 1[.] OWL 0[0]. В отличие от иерархической разметки знаний формата XML или JSON, предложенный формат BFT 1[.] Brain Fogic Tag 0[0] имеет сетевую модель. Например, строители XX-века добавили в бетон арматуру и получили железобетон, уникальный по прочности материал. В Макеевке, возле шлаковой горы металлургического комбината 0[map1] gsm1 1[memo1] смотри координаты строения в конце статьи 0[a1] map1..memo1 0[0], возвышаются объёмные чаши очистных сооружений, стоящие на тонких железобетонных колоннах.

В названии статьи и приведенном выше абзаце текст дополнен тегами, которые экранированы простой комбинацией символов "1[" и "0[". Для большинства компиляторов языков высокого уровня программирования запись "1[", с ведущим пробелом ошибочна. Так как идентификатор кортежа не может начинаться с цифры, а во встроенных арифметических операциях не существует функции с именем "[". Выразительность, различимость, простота набора на клавиатуре делает комбинацию символов BFT 1[tag1] аббревиатура "мозг туманных тегов" 0[link0] tag1 0[0] удобной для семантической разметки текста. Встраивая формат BFT 0[link1] tag1 0[0] в повседневную жизнь, мы отдаём дань уважения тенденции роботизации массовых процессов. Программные боты, сокращение взято от понятия "роботы", – это агенты интеллектуальных систем 1[botAI] Bot of Artificial Intelligence 0[0] обработки сложноструктурированных знаний 1[cloudMind] Cloud Tag of Knowledge Warehouse 0[0].

Модель гармонична и проста: блоки тегов, на первый взгляд, произвольно разбросаны по тексту, образуют цепочки связанных данных и размечают близлежащие фразы текста. Осознавая, что чтение материалов затруднено, положительным и превалирующим моментом считается угождение клиповому мышлению. Клиповый формат знаний BFT детализирует текст семантическим контекстом, полезным при машинной обработке, но контекст не лишен смысла и при чтении. Простота интерпретации технических знаков формата BFT 0[link2] tag1 0[0] делает обучение с целью их применения в массовой переписке посильным. Формат делает нас искуснее в порождении компьютерных текстов. Тексты смсок, твитов, новостных сообщений 1[.] RSS 0[0] стали короткими, что соответствует клиповости их восприятия. Учитывая тенденции изменившейся реальности, формат BFT 0[link3] tag1 0[0] делает твиты пригодными для последующего онтологического анализа машиной глубинного обучения.

Все упоминания аббревиатуры BFT 0[labelAllBFT] link0..link1..link2..link3 0[0] в предыдущем изложении можно связать простым символьным описанием,

именуемым "labelAllBFT", удобным для интерпретации мозгом человека, при чтении естественным путем, и ботом, при машинном способе обработки 0[link4] botAI 0[0].

В предыдущем абзаце приведен пример использования рефрена "labelAllBFT", одновременно размещённого в тексте на естественном языке в качестве обозначения, и в нотации метаязыка в качестве тега. Совмещение выглядит приемлемо при чтении, так как метаязык не перегружен обязательной парностью технических символов. Парность вызывает напряжение ума, когда признак начала "держится в уме" и ожидается закрывающийся символ экранирования. Например, в формате XML – это комбинация пары ("<", "/>"). Куда приятнее видеть поток цифр ноль "0" или один "1", не пустой "мешок" мультимножества bag "[0]" или предполагаемый пустой набор "[.]", как управляющую последовательность команд описания онтологии текста. Сравним два вида текста с внедренными в него элементами BFT или XML. Текст с BFT: "Привет 1[.] bft 1[English] hello 1[type] string 0[0] мир!". Текст с XML: "Привет

<xml><English>hello</English><type>string</type></xml> мир!".

Как видно каламбур и спутанность XML очевидна. Из преимуществ BFT вытекает недостаток – сложность отобразить иерархическую структуру, но для разрозненных и непоследовательных связей сетевой модели это приемлемо. Очевидным остаётся только описание наборов: мультимножеств и кортежей.

В нотации BFT можно описать набор, как упорядоченную последовательность двумя способами: аддитивным и поздним связыванием.

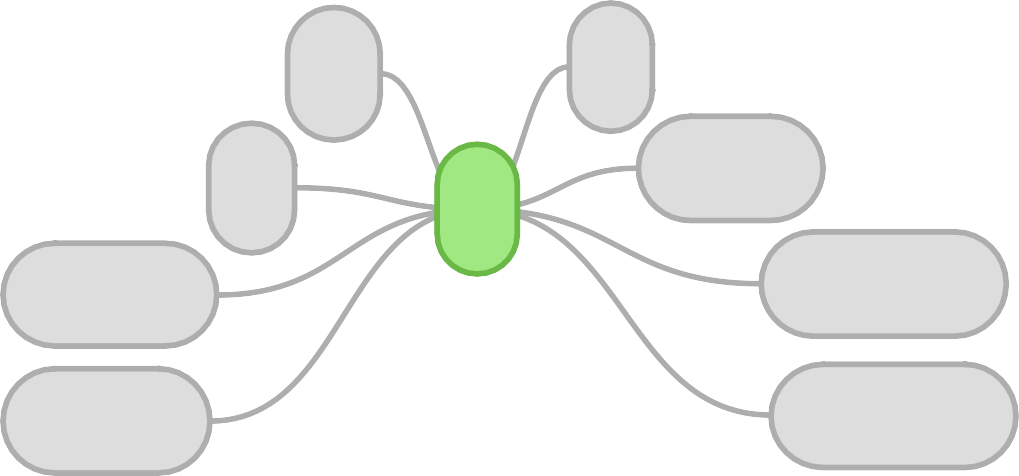
Например, для кортежа <1, 2, 4, 7>, создадим описание аддитивным

способом 1[node1] 1 1[node2] 4 1[node3] 2 0[ref1] node1..node3 0[0], когда связываем, сначала первый и третий узел, затем дописываем связь со вторым узлом 0[ref2] ref1..node2 0[0] и допускаем эти комментарии в завершении потока 1[node4] 7 0[ref3] ref2..node4 0[0] команд. Тег ref3 и будет указателем на кортеж. Теперь создадим для мультимножества [1, 4, 2, 7] описание 1[i1] 1 1[i2] 4 1[i3] 2 1[i4] 7 0[0], из которого получим кортеж примера 0[ref0] i1..i3..i2..i4 0[0], поздним связыванием. Тег ref0 – это указатель кортежа, эквивалентного кортежу ref3.

**Описание нотации BFT мемом, выразительным образом, удобным для запоминания.** В символической вселенной программирования процедурная модель решения задач представлена исходным кодом. Исходный код на языке высокого уровня содержит зачатки формального описания самого себя: осмысленные названия идентификаторов; подразумеваемая спецификация свойств и методов объектов; однострочные и многострочные комментарии.

Описания являются субъективными и выполнены на естественном языке, как-то отражают семантику модели, её онтологий и доменов. То есть, в исходном коде программы присутствуют вкрапления текста на естественном языке. Описания нивелируются при трансляции и игнорируются при компиляции. Возьмем этот приём совмещения в исходном коде текста комментариев и формальных выражений за аналог внедрения онтологического кода разметки в сообщениях на естественном языке.

Как сделать вкрапления в художественный текст объектов программирования, не нарушив удобочитаемости и понятности текста? На рисунке 1 изображено дерево всех ключевых элементов разметки BFT.



0[]

*false*

1[1]

*true*

0[0]

*null*

[]

bag

1[.] label

*symbol*

0[.]

*undefined*

1[label] value

*memory*

1[]

*absurdum*

0[label] value

*convert*

Рисунок 1 – Интеллект-карта тегов нотации Brain Fogic Tag

В таблице 1 приведено соответствие меток BFT существующим терминам в компьютерной науке.

Таблица 1 – Список тегов нотации BFT и их аналоги константных имен

|  |  |
| --- | --- |
| **Тег** | **Существующие аналоги, применяемые в компьютерной науке** |
| 1[] | absurdum, paradox, not relation, no solve, error {false} |
| 0[] | False |
| 0[0] | null, nil, empty, zero, void, none, stop proof {false} |
| 0[.] | undefined, unassigned {false} |
| 1[1] | True |
| 1[.]value 0[0] | tag, symbol, constant name, designate {true} / finish 0[0] |
| 1[label]value 0[0] | memory cell {true} / finish 0[0] |
| 0[label]value 0[0] | predicate, convert value to links, delimiter #\dot#\dot  {true} / finish 0[0] |

Возьмём известный элемент "ссылка на литературу – [0..9..999]". Добавим к нему триггер-признак "переключение на контекст", в виде известных цифр ноля и

единицы {"0", "1"}. Полученная конструкция будет различима для программных анализаторов текста, узнаваема читателем и при этом естественна на вид. Это основа выразительности нотации. Программные боты 1[.] botAI 0[0] смогут понимать теги нотации, и при этом читатели легко сами интерпретируют и воспроизводят текст на метаязыке BFT. Происходит синергия: добавленный контекст полезен при машинном анализе и он обогащает текст дополнительным смыслом при чтении.

Для запоминания нотации удобен мем, как образ: мешок с завязками. В конструктивной математике для конечного перечисления элементов с повторяющимися дубликатами мультимножества используется специальное обозначение с квадратными скобками "bag: [d, 7, d, 9]". Одним из принятых названий мультимножества в английском языке является "bag", что переводится как мешок. Мешок – это символ []. Формально, при письме развязанный мешок будем обозначать цифрой ноль. Абстрактно ноль представляет отсутствие чего- либо. Существование объекта в единственном экземпляре обозначается единицей. Поэтому завязанный мешок ассоциируется с цифрой один. Бинарный признак "0/1" сделаем префиксом обозначения мешка. Тогда получим различные комбинации: 0[] – пустой, развязанный мешок; 1[] – завязанный мешок, без предъявления содержимого; 0[.] – развязанный мешок с непонятной надписью; 1[.] – завязанный мешок с чистым ярлыком, который можно подписать.

Если рассматривать расположение мешка по центру, а завязок слева, то справа ожидается еще что-то. Таким образом, конструкция состоит из трех компонент, где правая часть подразумевается и может отсутствовать при написании тега нотации. Представим размещение содержимого мешка снаружи в виде перечисления его элементов с правой стороны от символа мешка "[X]", где X обозначает ярлык, наклеенный или пришитый на мешок. Ярлык – это серийный номер или наименование кучи предметов. Размещение предметов делается снаружи, потому что справа удобно поместить символьный поток переменной длины, но не пустой. Например, завязанный мешок с надписью "bag" представим в символьном виде, как "1[bag] d, 7, d, 9".

То есть, конструкция состоит из трех позиций, но правая часть может отсутствовать. Морфологически не все комбинации заполнения трех позиций нотации имеют содержательный смысл и удобный для запоминания. Абсурдно завязывать мешок без надписи и содержимого. Тогда не известно, пустой ли мешок.

Перечислим восемь значений мема-образа, удобных для интерпретации и запоминания.

Абсолютная ложь 0[], когда абстрактное число ноль, обозначает пустоту, тождественно пустому мешку, без надписи и предъявления содержимого    0 .

Возможная ложь, когда не найдено решение 0[0] – это просто ноль, пустой кортеж или список, признак окончания вычислений или конца доказательств. С помощью 0[0] можно выразительно отделить поток метаописаний от текста на естественном языке. Например, теперь можно записать без кавычек экранирования метаописание мультимножества 1[bag] d, 7, d, 9 0[0], которое является частью этого предложения.

Не существующий объект 0[.] должен интерпретироваться как ложь. Константное имя 0[.] эквивалентно undefined в языке программирования JavaScript, или unassigned в языке Delphi Pascal.

Противоречие 1[], интерпретируется как ложь в многозначной логике. Зачем завязывать мешок без надписи и содержимого. В нём может быть всё что угодно. Это абсурдная ситуация. Непонимание или ошибка – источник новых

знаний. Абстрактное число единицы не тождественно мешку   1 .

Абсолютная истина 1[1], когда мешок завязан, не надо требовать предъявления содержимого, ведь существует надпись, хотя надпись абсолютно абстрактная запись "один". Уникальное существование абстрактного объекта, обозначаемое числом один, тождественно абстрактному понятию единицы,

размещенной на надписи мешка 1 1.

Тег 1[.]X, интерпретируется как истина существования имени, где X – это тег, метаописание, десигнат слота фреймовой модели, онтология связанных данных. Появляется третья компонента мем-образа – специализированный ярлык. Ярлык, в виде точки, обозначает обобщение, а имя класса – это надпись X. Так как X – это символьный поток, то необходимо показать его завершение. Для указания завершения команд BFT подходит тег 0[0]. Например, ключевое слово гипертекстовой разметки 1[.] html 0[0] экранировано комбинацией {1[.], 0[0]}.

Ячейка памяти 1[X]Y, интерпретируется как истина, где X – это идентификатор, а Y – это символьный текст, размещённый в памяти. Текст Y экранирован символами {', "} или не включает комбинации двух символов из множества {0[, 1[}. Мешок завязан, имеет ярлык с надписью X и предъявлено его содержимое. Необходимо завершать поток перечисления элементов содержимого мешка символом 0[0].

Конвертер 0[X]Z, интерпретируется как истинный предикат. Где X – это идентификатор, а Z – это список тегов метаописания, разделенных символом удвоенной точки "..". Если Z с одним тегом, то это атомарный предикат или конвертор косвенной адресации. Этот тег наиболее искусственная структура в нотации BFT, но он является связующим звеном в образовании связанных данных. Тег трудно ассоциировать с конструкцией мем-образа "мешок и завязка". Тег является исключением из правил образного мышления. Только он один трудный в запоминании.

Для обобщения и выразительности из нотации пока исключены две комбинации: метка 0[X], без правой компоненты, где X – это не 0 или не точка, пока не имеет смысла и не является допустимой в нотации; метка 1[X], без правой компоненты, где X – это не 1, пока не имеет смысла и не является допустимой в нотации.

Представим пройденный материал в виде таблицы 2.

Таблица 2 – Список базовых конструкций метаязыка BFT

|  |  |  |
| --- | --- | --- |
| **Литерал** | **Логика** | **Описание** |
| 1[] | *false* | Антиномия |
| 0[] | *false* | Ложь |
| 0[0] | *false* | Пустой список |
| 0[.] | *false* | Не объявленная конструкция |
| 1[1] | *true* | Истина |
| 1[.] X | *true* | Ключевое слово X |
| 1[X] Y | *true* | Память с адресом X и значением Y |
| 0[X] Z | *true* | Конвертер с именем предиката X и списком термов Z,  разделенных удвоенной точкой |

В таблице 2 приведена интерпретация литерала BFT в качестве логической константы логики предикатов первого порядка.

**Выводы.** Рассмотрена нотация BrainFogicTag. Из восьми описанных тегов были использованы в тексте статьи четыре, как есть, по их назначению

«расширять смысл». Не все теги пригодились в разметке. Оставшиеся теги предназначены для описания работы экспертных систем, протоколирования доказательств, генерируемых искусственным интеллектом. Выделение цветом потока тегов BFT по тексту можно убрать, но удобочитаемость статьи с вкраплениями искусственного кода, останется приемлемой. Это значит, нотацию BFT можно применять в любом тексте, набираемом на компьютере.

Например, автономная роботизированная система, увидев на мониторе пульта энергетического модуля жизнеобеспечения лунной станции экранную форму, изображенную на рисунке 2, поймет интерфейсные решения архаических программистов.

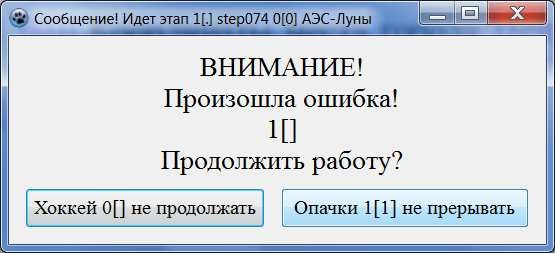


Рисунок 2 – Критически важный диалог для колонии на Луне

**Сноски.**

1[.] map1 1[.] gsm1 1[.] a1 0[0] Координаты места города 1[city1] Макеевка 0[0] в формате ГМС (градусы минуты секунды) 1[coord1] 48°02'57.58"N 37°56'48.51"E 0[gsm1] city1..coord1 0[0]

*27 мая 2019 года*

*07 июня 2019 года*

*Редактирование: 15 июня 2019 года*