ISO/IEC 14977:1996(E)

Информационная технология – Синтаксический метаязык – Расширенная Форма Бэкуса-Наура (Extended BNF)

Введение III
Синтаксические метаязыки III
Необходимость в стандартном синтаксисе метаязыка III
Задачи, которые нужно решать IV
Некоторые распространённые синтаксические метаязыки V
Стандарт метаязыка Расширенная Форма Бэкуса-Наура (*Extended BNF*) V
Ограничения и расширения VI
1 Границы 1
2 Нормативные ссылки 1
3 Определения 1
4 Форма каждого синтаксического элемента Расширенной ФБН 1
4.1 Общее 2
4.2 Синтаксис 2
4.3 Синтаксическое-правило 2
4.4 Список-определений 2
4.5 Единичное-определение 2
4.6 Синтаксический-термин 2
4.7 Синтаксическое-исключение 2
4.8 Синтаксический-фактор 2
4.9 Целое число 2
4.10 Синтаксическая-основа 3
4.11 Последовательность-альтернатив 3
4.12 Повторяющаяся-последовательность 3
4.13 Сгруппированная-последовательность 3
4.14 Мета-идентификатор 3
4.15. Символ-мета-идентификатора 3
4.16 Терминальная-строка 3
4.17 Первый-терминальный-символ 3
4.18 Второй-терминальный-символ 3
4.19 Специальная-последовательность 3
4.20 Символ-специальной-последовательности 3
4.21 Пустая-последовательность 3
4.22 Дополнительные примеры 3
5 Символы, представленные каждым синтаксическим элементом 4
5.1 Общий 4
5.2 Терминальная-строка 4
5.3 Мета-идентификатор 4
5.4 Сгруппированная-последовательность 4
5.5 Последовательность-альтернатив 4
5.6 Повторяющаяся-последовательность 4
5.7 Синтаксический-фактор 4
5.8 Синтаксический-термин 5
5.9 Единичное-определение 5
5.10 Список-определений 5
5.11 Специальная-последовательность 5
5.12 Пустая-последовательность 5
6 Размещение и Комментарии 5
6.1 Общее 5
6.2 Терминальный-символ 6
6.3 Символ-без-пробелов 6
6.4 Разрывающий-разделитель 6

6.5 Не-комментируемый-символ 6
6.6 Комментируемый-символ 6
6.7 Текстовый-комментарий-в-скобках 6
7 Представление каждого терминального-символа в расширенной ФБН 7
7.1 Общее 7
7.2 Буквы и цифры 7
7.3 Другие терминальные символы 7
7.4 Альтернативные представления 7
7.5 Другой-символ 7
7.6 Разрывающий-разделитель 8
7.7 Терминальные-символы, представленные парой символов 8
7.8 Неправильные последовательности символов 8
8 Примеры 9
8.1 Синтаксис Расширенной ФБН 9
8.2 Расширенная ФБН для неформального определения самой себя. 10
8.3 Расширенная ФБН, определённая неформально 11
Приложение A (информативное) Двухуровневая грамматика 12
Приложение B (информативное) Библиография 14

Введение

Синтаксический метаязык является важным инструментом в информатике. Понятия хорошо известны, однако используется множество систем обозначения, незначительно отличающихся друг от друга. В результате, синтаксические метаязыки всё ещё не понимаемы в достаточной степени, широко не используются, а преимущества строгих систем обозначения недооценено многими людьми.

Расширенная ФБН (Extended BNF) привносит некоторый порядок в формальное описание синтаксиса и будет полезна не только для определения языков программирования, но и для многих других формальных определений.

Со времён определения языка программирования Algol 60 (Naur, 1960) традиционно было определять синтаксис языка программирования формально. Algol 60 был определён с помощью системы обозначения, известной сейчас как BNF, или Backus-Naur Form (Форма Бэкуса-Наура). Эта система обозначения оказалась подходящим основанием для более поздних языков, но зачастую бывала расширена или незначительно изменена. Множество различных систем обозначений сбивают с толку и тем самым не дают возможности оценить преимущества формальных однозначных определений. Синтаксический метаязык *Расширенной ФБН*, описанный в этом стандарте, основан на форме Бэкуса-Наура и включает в себя наиболее распространённые расширения.

Синтаксические метаязыки

Синтаксический метаязык является системой обозначения для определения синтаксиса языка путём использования некоторого количества правил. Каждое правило именует часть языка (называемую *нетерминальным символом* языка) и затем определяет её возможные формы. *Терминальный символ* языка – это наименьшая частица, которая не может быть разделена на меньшие компоненты языка. Синтаксический метаязык является полезным всякий раз, когда необходима ясность формального описания и определения. Например, формат ссылок на статьи, предлагаемые в журналах, или инструкции для выполнения сложной задачи.

Формальное определение синтаксиса имеет три разных функции:

a) Оно именует различные синтаксические части (т.е. *нетерминальные символы*) языка;
b) Оно показывает, какие последовательности символов являются допустимым предложением языка;
c) Оно показывает *синтаксическую структуру* любого предложения языка;

Необходимость в стандартном синтаксисе метаязыка

Без наличия стандартного синтаксиса метаязыка, определение каждого языка программирования начинается с подробного описания самого метаязыка, используемого для определения синтаксиса этого языка программирования. Это является причиной возникновения различных проблем:

Множество различных систем обозначений – использование одного и того же метаязыка для двух разных языков программирования – это редкость. Т.о. читателю приходится трудно, т.к. нужно изучать новый метаязык до того, как он сможет приступить к изучению нового языка программирования.
Идеи не являются достаточно понятными – отсутствие стандартной системы обозначения, препятствует использованию строгих, однозначных определений.

Несовершенные системы обозначений – поскольку метаязыки должны быть определены для каждого языка программирования, то почти неизбежен тот факт, что метаязыки содержат дефекты. Например, ошибки, возникающие при составлении RTL/2 (BS5904) и CORAL 66 (BS5905) из-за того, что такие метаязыки не могут без труда быть набраны на клавиатуре.

Системы обозначений особого назначения – метаязык, определённый для индивидуального языка программирования, который нередко упрощён за счёт использования преимуществ некоторых специальных возможностей, присутствующих в определяемом целевом языке программирования. Однако в этом случае метаязык будет непригодным для использования в других языках программирования.

Малое количество общих синтаксических процессоров – множество синтаксических метаязыков ограничивают пригодность компьютерных программ для анализирования и обработки синтаксиса. Например, аккуратно описать синтаксис, создать индекс символов, используемых в синтаксисе для того, чтобы выполнить проверку синтаксиса программ, написанных на языке.

На практике, опытные читатели не испытывают больших трудностей в изучении и понимании новой нотации, но даже в этом случае имеющиеся различия затрудняют взаимопонимание и препятствуют общению. Стандартный метаязык позволяет многим людям кристаллизовать неясные идеи в однозначное определение. Это так же полезно и потому, что другие люди, нуждающиеся в предоставлении формальных определений, будут избавлены от необходимости повторного изобретения схожих понятий.

Задачи, которые нужно решать

Желательно, чтобы стандарт синтаксиса метаязыка был:

a) *кратким* настолько, чтобы языки могли быть определены кратко, и таким образом были бы более легки для понимания;
b) *точным* настолько, чтобы правила были недвусмысленными;
c) *формальным* настолько, чтобы правила могли быть проанализированы, либо обработаны иным способом с помощью компьютера, когда это понадобится;
d) *естественным* настолько, чтобы синтаксическое обозначение и формат были относительно легки в изучении и понимании даже для тех, кто не является разработчиком языков; (Это означает, что значение символа не должно быть неожиданным. Это так же означает то, что должно быть возможным определение синтаксиса языка таким образом, который бы позволял понимать назначения конструкций).
e) *общим* настолько, чтобы система обозначений была пригодной для многих целей, включая описание многих различных языков;
f) *простым в наборе символов и с такой системой обозначения*, которая избегает, насколько это возможно, использование символов, обычно не доступных на стандартной клавиатуре (точнее на печатающем устройстве и на компьютерном терминале) т.о. чтобы правила могли бы быть набраны и затем вычислены посредством компьютерных программ;
g) *описывающим себя* настолько, чтобы система обозначений была в состоянии описать сама себя;
h) *линейным* настолько, чтобы синтаксис мог быть выражен как единый поток символов. (Это упрощает печать синтаксиса. Компьютерная обработка синтаксиса так же упрощается).

Некоторые распространённые синтаксические метаязыки

К сожалению, ни один из существующих метаязыков не годится для принятия его в качестве стандарта. Например:

a) COBOL (ISO 1989:1985) перечисляет альтернативы вертикально и использует скобки, занимая много строк. Это неудобно для компьютерной обработки и не может быть набрано на клавиатуре.
b) Форма Бэкуса-Наура (использовалась в ALGOL 60) имеет проблемы, если метасимволы < > | ::= встречаются в определяемом языке. Некоторые общие формы конструкций (например, комментарии) не могут быть выражены естественно, другие конструкции (например повторения) являются слишком многословными.
c) У устаревшего FORTRAN 77 (ISO 1539:1980) были «железнодорожные пути». Он легок в понимании, но труден в подготовке и обработке компьютером или печатающим устройством. Текущая версия, FORTRAN 90 (ISO/IEC 1539:1991), более не использует эту систему обозначений.

Большинство других языков используют вариант одного из этих метаязыков. Большинство из них не может претендовать на стандартизацию потому, что они используют символы, которые не были определены в составе языка как метасимволы метаязыка. Это упрощает метаязык, но препятствует его повсеместному использованию.

POSIX (ISO/IEC 9945-2:1993) включает два дополнительных средства, каждое из которых предполагает использование определённого в ISO/IEC 646:1991 набора символов: LEX даёт возможность определения и синтаксического анализа регулярных выражений, но недостаточен для описания произвольной, контекстно-зависимой грамматики, и YACC (Yet Another Compiler Compiler) является синтаксическим анализатором для грамматики LALR(1).

Стандарт метаязыка Расширенная Форма Бэкуса-Наура (*Extended BNF*)

Расширенная Форма Бэкуса-Наура – это метаязык, определённый в данном Международном Стандарте, основанный на предложении Никлауса Вирта (Niklaus Wirth (Wirth, 1977)) которое, в свою очередь, основано на форме Бэкуса-Наура и содержит большинство общих расширений, а именно:

a) *Терминальные символы* языка заключены в кавычки, поэтому любой символ, включенный в *Расширенную ФБН*, может быть определён как терминальный символ определяемого языка.
b) [и] обозначают необязательные символы.
c) { и } указывают повторения.
d) Каждое правило имеет явный заключительный символ для того, чтобы никогда не возникало неоднозначности в том, где оно заканчивается.
e) Скобки группируют элементы вместе. Это очевидное удобство – использовать (и) в их обычном математическом смысле.

Основными отличиями в *Расширенной ФБН* являются дополнительные особенности которые, как показывает опыт, очень нужны при обеспечении формального определения:

a) *Определение точного количества элементов*. Fortran содержит правило, согласно которому метка поля содержит ровно пять символов; идентификатор в PL/I или COBOL имеет до 32 символов: правила, подобные этим, могут быть выражены в Форме Бэкуса-Наура лишь с трудом. На практике, такие определения зачастую оставляют неполными и правила определяют неформально, на английском языке.
b) *Определение чего-либо, путём указания на несколько исключительных случаев*. В Algol **end**-комментарий оканчивается перед первым **end**, **else** или перед точкой с запятой. Пра-

вило, подобное этому, не может быть выражено в Форме Бэкуса-Наура кратко или ясно и обычно оно так же определяется неформально, на английском языке.

c) *Включение комментариев*. Языки программирования и другие структуры со сложным синтаксисом нуждаются во множестве правил для своего определения. Синтаксис будет более понятным при наличии пояснений и перекрёстных ссылок; Соответственно *Расширенная* Форма Бэкуса-Наура предоставляет возможность комментирования, и таким образом простой текст может быть добавлен в синтаксис для помощи читателю не влияя на формальное значение синтаксиса.

d) *Мета-идентификатор*. Мета-идентификатор (имя нетерминального символа в языке) не обязательно должен быть представлен одним словом или же быть вложенным в скобки, потому что он является точным, связанным символом. Он также гарантирует, что размещение синтаксиса (кроме терминального символа) не влияет на язык, который был определён.

e) *Расширения*. Пользователь может пожелать расширить *Расширенную* Форму Бэкуса-Наура. Специальная последовательность, формат и значение которой не определяется в стандарте, предоставляет такую возможность, помимо гарантии того, что начало и окончание расширения всегда могут быть с лёгкостью обнаружены. Различные возможные расширения выделены в следующих параграфах.

Ограничения и расширения

Основное ограничение *Расширенной* Формы Бэкуса-Наура состоит в том, чтобы язык был определён линейно, т.е. символы в предложении языка могут быть размещены в порядке следования. Например, узоры для вязания и кулинарные рецепты являются линейными языками, а схемы электрических цепей – нет.

Дополнительное ограничение состоит в том, что *Расширенная* Форма Бэкуса-Наура является недостаточной для определения более сложных форм грамматики. Такие средства не были предоставлены потому, что основной идеей было определение системы обозначений, достаточной для простых и наиболее общих потребностей.

Вместо этого, *Расширенная* Форма Бэкуса-Наура была разработана так, чтобы различные расширения можно было создавать естественным путём. Существует два простых способа расширения стандарта метаязыка. Во-первых, понятие ***специальной последовательности*** служит базовой основой для любого расширения. Структура, находящаяся между символами специальной последовательности будет почти абсолютно произвольной. Этот метод подошёл бы для грамматических операций, т.е. указанных действий, которые должны быть выполнены после того, как предложение будет проанализировано. Во-вторых, мета-идентификатор в языке стандарта никогда не может сразу следовать за левой круглой скобкой; т.о. другим способом расширения метаязыка, является определение синтаксиса и значения мета-идентификатора, сопровождаемого последовательностью параметров, ограниченных круглыми скобками. Это было бы разумно в атрибутивной грамматике, где правила гарантируют непротиворечивость между различными членами предложения в определяемом языке.

Более сложные расширения так же возможны. *Приложение A* советует, как *Расширенная* Форма Бэкуса-Наура может быть расширена для определения двухуровневой грамматики.

Информационная технология – Синтаксический метаязык – Расширенная ФБН

1 Границы

Этот Международный Стандарт определяет систему обозначений, *Расширенную ФБН*, для определения синтаксиса линейной последовательности символов. Он определяет логическую структуру системы обозначений и его графическое представление.

Расширенная ФБН находит применение при определении языков программирования, а так же иных языков, равно как и в других формальных определениях: например, команд операционной системы, или точности формата данных и результатов.

Примеры *Расширенной ФБН* даны в п.8.

ПРИМЕЧАНИЕ: поскольку существует множество других систем обозначений, то *Расширенная ФБН* всё ещё может использоваться неправильно; т.о. это не препятствует тому, чтобы кто-то в результате своих попыток определил неоднозначный язык, для которого было бы невозможно выполнить анализ синтаксиса.

2 Нормативные ссылки

Следующие стандарты содержат положения, на которые производятся ссылки в этом тексте, составляющем положения данного Международного Стандарта. Во время публикации, указанные ниже версии стандартов были действующими. Все существующие стандарты, со временем, подвергаются пересмотру и части соглашений, основанных на данном Международном Стандарте призваны следовать, по возможности, применению наиболее новых выпусков перечисленных ниже стандартов. Участники IEC и ISO поддерживают регистрацию текущих, действующих Стандартов.

ISO 2382-15 : 1985, *Data processing — Vocabulary Part 15: Programming languages.*

ISO/IEC 646 : 1991, *Information technology — ISO 7-bit coded character set for information interchange.*

ISO/IEC 6429 : 1992, *Information technology — Control functions for 7-bit and 8-bit coded character sets.*

BS 6154 : 1981, *Method of defining — Syntactic meta-language.*

3 Определения

В данном Международном Стандарте используются определения, данные в ISO 2382-15, а так же применяются указанные ниже определения:

3.1 последовательность: упорядоченный список из нуля или более отдельных элементов.

3.2 подпоследовательность: последовательность внутри последовательности.

3.3 нетерминальный символ: синтаксическая часть языка, который будет определён.

3.4 мета-идентификатор: имя нетерминального символа.

3.5 начальный символ: нетерминальный символ, определённый в одном или более синтаксическом правиле, но не встречающийся в остальных синтаксических правилах.

3.6 предложение: последовательность символов, которые представляют собой начальный символ.

3.7 терминальный символ: последовательность из одного или более символов, формирующая неделимый элемент языка.

ПРИМЕЧАНИЕ: В данном Международном Стандарте терминальный символ *Расширенной ФБН* назван *терминальным-символом* и терминальный символ языка был определён синтаксисом представленным в виде *терминальной-строки*.

4 Форма каждого синтаксического элемента Расширенной ФБН

ПРИМЕЧАНИЕ

1. Используются следующие соглашения:
   a) Каждый *мета-идентификатор Расширенной ФБН* написан как одно или более слово, соединённых друг с другом дефисами;
   b) *Мета-идентификатор*, начинающийся с префикса *"символ-"* является именем терминального символа *Расширенной ФБН*.
2. Обычный символ представляющий каждый оператор *Расширенной ФБН* и его подразу-

меваемый приоритет (наивысший приоритет наверху):

*	символ-повторения
-	символ-исключения
,	символ-объединения
|	символ-разделителя-определений
=	символ-определения
;	символ-разделителя

3. Нормальный приоритет переопределён для следующих пар скобок:

‘ символ-одинарной -кавычки	символ-одинарной ‘ -кавычки
“ символ-двойной -кавычки	символ-двойной “ -кавычки
(* символ-начала -комментария	символ-конца *) -комментария
(символ-начала -группы	символ-конца) -группы
[символ-начала -альтернативы	символ-конца] -альтернативы
{ символ-начала -повторений	символ-конца } -повторений
? символ-специальной -последовательности	символ-специальной ? -последовательности

4.1 Общее

Логическая структура *Расширенной ФБН* определена в пунктах 4.2 - 4.21.

4.2 Синтаксис

Синтаксис языка состоит из одного или более *синтаксических-правил*.

4.3 Синтаксическое-правило

Синтаксическое-правило состоит из *мета-идентификатора* (имя нетерминального символа, который был определён) за которым следует *символ-определения*, за ним идёт *список-определений* и завершается всё *символом-разделителя*.

4.4 Список-определений

Список-определений состоит из упорядоченного списка одного или более *единичных-определений*, разделённых между собой *символом-разделителя-определений*.

4.5 Единичное-определение

Единичное-определение состоит из упорядоченного списка одного или более *синтаксических-терминов*, разделённых между собой *символом-объединения*.

4.6 Синтаксический-термин

Синтаксический-термин состоит из:

a) *Синтаксического-фактора*, или
b) *Синтаксического-фактора*, за которым следует *символ-исключения* за которым, в свою очередь, следует *синтаксическое-исключение*.

4.7 Синтаксическое-исключение

Синтаксическое-исключение состоит из *синтаксического-фактора*, являющегося предметом ограничения последовательностей символов, представленных *синтаксическим-исключением*, которое могло бы быть представлено идентичным образом с помощью *синтаксического-фактора*, не содержащего *мета-идентификаторов*.

ПРИМЕЧАНИЕ: Если бы *синтаксическому-исключению* разрешено было быть произвольным *синтаксическим-фактором*, то *Расширенная ФБН* могла бы определить более широкий класс языков, чем контекстно-независимые грамматики, включая попытки, которые приводят к парадоксам, подобным парадоксу Рассела, например:

```
xx = "A" - xx;
```

Является ли “A” примером xx? Такое разрешение является нежелательным, и поэтому форма *синтаксического-исключения* ограничена, оправдывая данное ограничение предоставлением безопасности подобных случаев. Т.о., несмотря на то, что *синтаксический-фактор* в общем является эквивалентом некоторой контекстно-независимой грамматики, *синтаксическое-исключение* всегда является эквивалентом некоторой регулярной грамматики. Может показаться, что различием между контекстно-свободной грамматикой и регулярной грамматикой всегда является другая контекстно-независимая грамматика; следовательно *синтаксический-термин* (и следовательно любая грамматика, определённая согласно данному стандарту) эквивалентна некоторой контекстно-независимой грамматике.

4.8 Синтаксический-фактор

Синтаксический фактор состоит из:

a) *целого числа*, за которым следует *символ-повторений*, за которым, в свою очередь, идёт *синтаксическая-основа*, или
b) *синтаксической-основы*.

4.9 Целое число

Целое число формируется из упорядоченного списка, состоящего из одного или более десятичных цифр.

4.10 Синтаксическая-основа

Синтаксическая-основа состоит из одного из следующих вариантов:

a) последовательность-альтернатив;
b) повторяющаяся-последовательность;
c) сгруппированная-последовательность;
d) мета-идентификатор;
e) терминальная-строка;
f) специальная-последовательность;
g) пустая-последовательность.

4.11 Последовательность-альтернатив

Последовательность-альтернатив состоит из *символа-начала-альтернативы*, за которым следует *список-определений*, и завершается всё *символом-конца-альтернативы*.

4.12 Повторяющаяся-последовательность

Повторяющаяся-последовательность состоит из *символа-начала-повторений*, за которым следует *список-определений* и завершается всё *символом-конца-повторений*.

4.13 Сгруппированная-последовательность

Сгруппированная-последовательность состоит из *символа-начала-группы*, за которым следует *список-определений*, и завершается всё *символом-конца-группы*.

4.14 Мета-идентификатор

Мета-идентификатор – это упорядоченный список, состоящий из одного или более *символов-мета-идентификатора*, при условии, что первый *символ-мета-идентификатор* является буквой.

4.15. Символ-мета-идентификатора

Символ-мета-идентификатора является *буквой* или *десятичной-цифрой*.

4.16 Терминальная-строка

Терминальная строка состоит из:

a) *Символа-одинарной-кавычки*, за которым следует один или более *Первых-терминальных-символов*, и завершается всё *символом-одинарной-кавычки*, или
b) *Символа-двойной-кавычки*, за которым следует один или более *Вторых-терминальных-символов*, и завершается всё *символом-двойной-кавычки*.

4.17 Первый-терминальный-символ

Первый-терминальный-символ – это любой *терминальный-символ*, за исключением *символа-одинарной-кавычки*.

4.18 Второй-терминальный-символ

Второй-терминальный-символ – это любой *терминальный-символ*, за исключением *символа-двойной-кавычки*.

4.19 Специальная-последовательность

Специальная-последовательность состоит из *символа-специальной-последовательности*, за которым следует (возможно пустая) последовательность из *символов-специальной-последовательности* и завершается всё *символом-специальной-последовательности*.

4.20 Символ-специальной-последовательности

Символ-специальной-последовательности – это любой *терминальный-символ*, за исключением *символа-специальной-последовательности*.

4.21 Пустая-последовательность

Пустая-последовательность состоит из пустой последовательности *терминальных-символов*.

4.22 Дополнительные примеры

Следующий пример является синтаксическим правилом продолжения строки в Fortran 77, согласно которому такая строка начинается с 5-ти пробелов, шестой символ не должен быть пробелом или нулём, и в целом, это не должно быть более чем 72 (=5+1+66) символов:

```
Fortran 77 continuation line = 5 * " ",
(character - (" " | "0")), 66 * [character] ;
```

В Fortran 66 определение строки продолжения является более сложным. Следующий пример является синтаксическим правилом, которое устанавливает, что строка продолжения не должна начинаться с символа ‘C’, должна содержать не менее шести символов, причём шестой символ не должен быть пробелом или нулём, а так же длина такой строки в сумме не должна превышать 72 (=1+4+1+66) символов:

```
Fortran 66 continuation line = character -
"C", 4 * character, character - (" " |
"0"), 66 * [character] ;
```

5 Символы, представленные каждым синтаксическим элементом

5.1 Общий

Каждое *синтаксическое-правило* является синтаксическим правилом, определяющим (возможно пустые) последовательности *терминальных* и *нетерминальных-символов*. Каждая такая последовательность символов представлена *нетерминальным-символом*, называемым *мета-идентификатором*, который указывается вначале *синтаксического-правила*. Пункты 5.2 – 5.12 определяют последовательности символов которые представлены любым *списком-определений*.

ПРИМЕЧАНИЯ

1. Когда представлен полный синтаксис языка, в нём присутствуют:
   a. начальный символ и
   b. как минимум, одно *синтаксическое-правило*, начинающееся с любого *мета-идентификатора*, используемого в качестве *синтаксической-основы*.
2. Труднее понять такой язык, в котором есть несколько *синтаксических-правил*, начинающихся с *мете-идентификаторов* и при этом не указывающих, что каждое такое определение лишь частично определяет некоторый *нетерминальный-символ*.

5.2 Терминальная-строка

Терминальная-строка представлена одним из следующих вариантов:

a) Последовательность *первых-терминальных-символов*, размещённых между *символами-одинарной-кавычки*, или
b) Последовательность *вторых-терминальных-символов*, размещённых между *символами-двойной-кавычки*.

5.3 Мета-идентификатор

Мета-идентификатор используемый как *синтаксическая-основа*, представляет собой любую последовательность символов, определённых посредством *списка-определения* любого *синтаксического-правила*, начинающегося с данного *мета-идентификатора*.

5.4 Сгруппированная-последовательность

Сгруппированная-последовательность представляет любую последовательность символов, определённых посредством *списка-определений*, заключённого между *символом-начала-группы* и *символом-конца-группы*.

5.5 Последовательность-альтернатив

Последовательность-альтернатив представлена одним из следующих вариантов:

a) *Пустая-последовательность-символов*, или
b) Любая последовательность символов, определённая *списком-определений*, заключённым между его *символом-начала-альтернативы* и *символом-конца-альтернативы*.

5.6 Повторяющаяся-последовательность

Повторяющаяся-последовательность представляет (возможно пустую) последовательность подпоследовательностей, где каждая подпоследовательность является любой последовательностью символов, определённых посредством *списка-определений*. *Повторяющаяся-последовательность* всегда заключается между *символом-начала-повторений* и *символом-конца-повторений*.

5.7 Синтаксический-фактор

Синтаксический-фактор представляет точное количество подпоследовательностей, где каждая подпоследовательность является последовательностью символов, представленных *синтаксической-основой*, являющейся, в свою очередь, частью *синтаксического-фактора*. Необходимое количество подпоследовательностей равно единице, если целочисленное значение не указано или же равно указанному целочисленному значению.

В качестве примера, следующие *синтаксические-правила* иллюстрируют возможности для повторяющихся выражений.

```
aa = "A";
bb = 3 * aa, "B";
cc = 3 * [aa], "C";
dd = {aa}, "D";
ee = aa, {aa}, "E";
ff = 3 * aa, 3 * [aa], "F";
gg = 3 * {aa}, "D";
```

Терминальные-строки, определённые этими правилами, будут подобны следующим:

```
aa: A
bb: AAAB
cc: C AC AAC AAAC
dd: D AD AAD AAAD AAAAD etc.
ee: AE AAE AAAE AAAAE AAAAAE etc.
ff: AAAF AAAAF AAAAAF AAAAAAF
```

ПРИМЕЧАНИЕ – Определение для `gg` , несмотря на то, что является синтаксически верным, не является разумным. Последовательности символов, представленных `gg`, являются идентичными `dd`, но не могут быть проанализированы однозначно.

5.8 Синтаксический-термин

Когда *синтаксический-термин* является единственным *синтаксическим-фактором*, то это представляет любую последовательность символов, представленных этим *синтаксическим-фактором.*

Когда *синтаксический-термин* является *синтаксическим-фактором,* за которым следует *синтаксическое-исключение*, это представляет любую последовательность символов, которая удовлетворяет обоим условиям:

a) Это последовательность символов, представленная *синтаксическим-фактором*.
b) Это не последовательность символов, представленная *синтаксическим-исключением.*

В качестве примера, следующие *синтаксические-правила* иллюстрируют возможности, предоставленные *символом-исключения.*

```
letter = "A" | "B" | "C" | "D" | "E" | "F"
| "G" | "H" | "I" | "J" | "K" | "L" | "M"
| "N" | "O" | "P" | "Q" | "R" | "S" | "T"
| "U" | "V" | "W" | "X" | "Y" | "Z";
vowel = "A" | "E" | "I" | "O" | "U";
consonant = letter - vowel;
ee = {"A"}-, "E";
```

Терминальные-строки, определённые этими правилами, следующие:

```
letter: A B C D E F G H I J etc.
vowel: A E I O U
consonant: B C D F G H J K L M etc.
ee: AE AAE AAAE AAAAE AAAAAE etc.
```

ПРИМЕЧАНИЕ – `{"A"}` представляет последовательность из одного или более `"A"`, потому что это *синтаксический-термин* с пустым *синтаксическим-исключением.*

5.9 Единичное-определение

Единичное-определение представляет собой последовательность из одной или более подпоследовательности, где каждая подпоследовательность в свою очередь является последовательностью символов, представленных соответствующим *синтаксическим-термином* в том *единичном-определении.*

5.10 Список-определений

Список-определений представляет любую последовательность символов, представленную любым из *единичных-определений*, формирующих этот *список-определений.*

5.11 Специальная-последовательность

Последовательность символов, представленных *специальной-последовательностью*, выходящих за рамки настоящего Международного Стандарта. Только **формат** *специальной-последовательности* определён в этом Международном Стандарте. *Специальная-последовательность* обеспечивает нотацию для расширений, которые могут потребоваться пользователю.

5.12 Пустая-последовательность

Пустая-последовательность является пустой последовательностью символов.

6 Размещение и Комментарии

6.1 Общее

Размещение синтаксиса на странице является почти полностью произвольным. Пункты 6.2-6.4 определяют, что непечатаемые символы, такие как пробел и переход на новую строку не влияют на синтаксис, если эти символы находятся вне *терминальной-строки* или пары символов, формирующих единственный *терминальный-символ*. Пункты 6.5-6.7 определяют, где в синтаксисе может быть в качестве комментариев размещён произвольный текст.

ПРИМЕЧАНИЯ

1. Человеку намного легче читать и понимать синтаксис, если каждое *синтаксиче-*

ское-правило начинается с новой строки и различные символы метаязыка размещены с заметным интервалом.

2. Язык, определённый с помощью *Расширенной ФБН* может иметь лексические правила, полностью отличающуюся от собственных правил *Расширенной ФБН*.
3. Комментарии предоставляют поясняющий текст, добавленный в синтаксис и т.о. помогающий человеку в понимании синтаксиса. Например, *синтаксические-правила* могут быть пронумерованы, а каждый *мета-идентификатор* сопровождаться комментарием, определяющим позицию этого *синтаксического-правила*. Рекомендуется, чтобы всякий комментарий, относящийся к *синтаксическому-правилу*, размещался перед *символом-разделителя* данного правила.
4. Комментарии формально не влияют на язык, определяемый посредством синтаксиса.

6.2 Терминальный-символ

Терминальным-символом Расширенной ФБН является один из следующих символов:

a) буква;
b) десятичная-цифра;
c) *символ-объединения;*
d) *символ-определения;*
e) *символ-разделителя-определений;*
f) *символ-конца-комментария;*
g) *символ-конца-группы;*
h) *символ-конца-альтернативы;*
i) *символ-конца-повторений;*
j) *символ-исключения;*
k) *символ-одинарной-кавычки;*
l) *символ-повторения;*
m) *символ-двойной-кавычки;*
n) *символ-специальной-последовательности;*
o) *символ-начала-комментария;*
p) *символ-начала-группы;*
q) *символ-начала-альтернативы;*
r) *символ-начала-повторений;*
s) *символ-разделителя;*
t) *другой-символ;*

6.3 Символ-без-пробелов

Символом-без-пробелов является либо

a) *терминальный-символ*, не являющийся ни *символом-одинарной-кавычки*, ни *символом-двойной-кавычки*, либо
b) *терминальная-строка.*

6.4 Разрывающий-разделитель

Разрывающий-разделитель – это один из непечатаемых символов: *пробел, горизонтальная-табуляция, вертикальная-табуляция* или *подача-бумаги.*

Один или более *разрывающих-разделителей* может быть размещён:

a) перед синтаксисом, и
b) между двух *символов-без-пробела* в синтаксисе, и
c) после синтаксиса

без влияния на язык, определённый посредством данного синтаксиса.

6.5 Не-комментируемый-символ

Не-комментируемый-символ – это любой из следующий символов:

a) *Терминальный-символ*, не являющийся ни *буквой*, ни *десятичной-цифрой*, ни *символом-одинарной-кавычки*, ни *символом-двойной-кавычки*, ни *символом-начала-комментария*, ни *символом-конца-комментария*, ни *символом-специальной-последовательности*;
b) *Мета-идентификатор;*
c) *Целое-число;*
d) Терминальная-строка;
e) Специальная-последовательность;

6.6 Комментируемый-символ

Комментируемый-символ – это один из следующих:

a) *Текстовый-комментарий-в-скобках;*
b) *Не-комментируемый-символ;*
c) *Другой-символ;*

6.7 Текстовый-комментарий-в-скобках

Текстовый-комментарий-в-скобках – это *символ-начала-комментария,* за которым следует (возможно пустая) последовательность *комментируемых-символов*, и в завершении идёт *символ-конца-комментария.*

Один или более *текстовых-комментариях-в-скобках* может быть размещено:

a) перед синтаксисом, и
b) между двумя *не-комментируемыми-символами*, и
c) после синтаксиса

без влияния на язык, определённый посредством синтаксиса.

ПРИМЕЧАНИЕ – пункты 6.5-6.7 подразумевают, что *текстовый-комментарий-в-скобках* не может появляться в любом из следующих:

a) *мета-идентификатор;*
b) *целое-число;*
c) *специальная-последовательность;*
d) *терминальная-строка.*

7 Представление каждого терминального-символа в расширенной ФБН

7.1 Общее

Представление каждого *терминального-символа* и *разрывающего-разделителя* в *Расширенной ФБН* использует символы 7-ми битного набора символов (ISO/IEC 646:1991 International Reference Version), определённые в пунктах 7.2-7.8.

7.2 Буквы и цифры

Каждая *буква* и *цифра* представлены соответствующим символом.

7.3 Другие терминальные символы

Таблица 1 определяет представление символов для каждого *терминального-символа*, который не является ни буквой, ни цифрой, ни *другим-символом.*

Таблица 1 – представление *терминальных-символов*

Символ метаязыка	**Обычное представление**
Символ-объединения	, запятая
Символ-определения	= знак равенства
Символ разделителя-определений	| вертикальная линия
Символ-конца-комментария	*) звёздочка, правая скобка
Символ-конца-группы	) правая скобка
Символ-конца-альтернативы	] правая прямоугольная скобка
Символ-конца-повторений	} правая фигурная скобка
Символ-исключения	- дефис-минус
Символ-одиночной-кавычки	´ апостроф
Символ-повторений	* звёздочка
Символ-двойной-кавычки	" кавычка
Символ-специальной-последовательности	? знак вопроса
Символ-начала-комментария	(* левая скобка, звёздочка
Символ-начала-группы	(левая скобка
Символ-начала-альтернативы	[левая прямоугольная скобка
Символ-начала-повторений	{ левая фигурная скобка
Символ-разделителя	; точка с запятой

7.4 Альтернативные представления

Таблица 2 определяет альтернативные представления для некоторых *терминальных-символов*.

Таблица 2 – альтернативное представление *терминальных-символов*

Символ метаязыка	**Альтернативное представление**
Символ-разделителя-определений	/ знак деления
Символ-разделителя-определений	! восклицательный знак
символ-конца-альтернативы	/) знак деления, правая скобка
Символ-конца-повторений	:) двоеточие, правая скобка
Символ-начала-альтернативы	(/ левая скобка, знак деления
Символ-начала-повторений	(: левая скобка, двоеточие
Символ-разделителя	. точка

ПРИМЕЧАНИЯ

1. Основная причина указания альтернативных представлений заключается в том, что не все компьютеры и печатающий машинки имеют символы, перечисленные в таблице 1.
2. Во избежание путаницы, представление *терминальных-символов* в любом конкретном документе должно быть единообразным.
3. Пункты 7.2-7.4 подразумевают, что символами, необходимыми для *Расширенной ФБН* являются:

   буквы цифры = , - * ( ) ?
   | или / или !
   / или обе скобки [ ]
   : или обе скобки { }
   ´ или " (Оба символа необходимы, если был определён любой терминальный символ языка)

7.5 Другой-символ

Другой-символ – это любой символ из набора, определённого в ISO/IEC

646:1991, который:

a) Не является управляющим символом
b) Не требуется для представления любого другого *терминального-символа*.

ПРИМЕЧАНИЯ – когда *терминальные-символы* представлены как указано в `таблице 1`, то *другими-символами* являются:

```
  пробел
. точка
: двоеточие
! восклицательный знак
+ знак сложения
_ нижняя линия
% знак процентов
@ коммерческая «А»
& амперсанд
# знак числа
$ знак доллара
< знак меньше
> знак больше
/ знак деления (слеш)
\ обратный слеш
^ циркумфлекс
` диакритический знак
~ тильда
```

7.6 Разрывающий-разделитель

Разрывающий-разделитель представлен такими символами как:

a) *пробел*, представленный символом пробела
b) *горизонтальная-табуляция* представленная символом горизонтальной табуляции
c) *новая-строка* представлена (возможно пустой) последовательностью из символа возврата каретки и символа перехода на новую строку, и (возможно пустой) последовательности из символа возврата каретки,
d) *вертикальная-табуляция* представлена символом вертикальной табуляции,
e) *подача-бумаги* представлен символом подачи бумаги.

7.7 Терминальные-символы, представленные парой символов

Каждая пара символов, указанных в `таблице 3`, в *синтаксическом-правиле* всегда представляет собой единый *терминальный—символ*, за исключением случае, когда они находятся внутри *терминальной-строки*, или же внутри *специальной-последовательности*.

Таблица 3 – Парные символы, представляющие собой единый терминальный-символ

(*
*)
(:
:)
(/
/)

ПРИМЕЧАНИЕ – Данное ограничение является необходимым, т.к. такие последовательности символов являются двусмысленными. Например /) может означать как *символ-разделителя-определений*, так и *символ-конца-группы*.

7.8 Неправильные последовательности символов

Каждая запись в `таблице 4` указывает последовательность символов, которая не появляется в *синтаксическом-правиле* за рамками *терминальной-строки* или *специальной-последовательности.*

Таблица 4 – неправильные последовательности символов

(*)
(:)
(/)

ПРИМЕЧАНИЕ – данное ограничение необходимо, т.к. эти последовательности символов являются двусмысленными. Например, (*) может быть *символом-начала-комментария*, за которым следует *символ-конца-группы*, или же это может быть *символ-начала-группы*, за которым следует *символ-конца-комментария*.

Вставка *разрывающего-разделителя* позволяет избежать двусмысленности. Например, (*) является *символом-начала-комментария*, за которым следует *символ-конца-группы*. В то же время (*) является *символом-начала-группы*, за которым следует *символ-конца-комментария*.

8 Примеры

8.1 Синтаксис Расширенной ФБН

```
(*
Синтаксис Расширенной ФБН может определить сам себя. В этом примере представлены четыре части:

первая часть именует символы;

вторая часть определяет удаление ненужных непечатаемых символов;

третья часть определяет удаление текстовых комментариев;

заключительная часть определяет саму структуру Расширенной ФБН.

Каждое синтаксическое правило в этом примере начинается с комментария, который указывает на соответствующий пункт данного стандарта.

Значения специальных-последовательностей не определены в стандарте. В этом примере (см. п.7.6) они представляют функции управления, определённых в ISO/IEC 6429:1992. Другие специальные-последовательности определяют синтаксическое-исключение (см. п.4.7)
*)

(*
Первая часть лексического синтаксиса определяет символы в 7-битном наборе символов (ISO/IEC 646:1991), которыми в Расширенной ФБН представлен каждый терминальный-символ и разрывающий-разделитель.
*)

(* см. 7.2 *) letter
= 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h'
| 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p'
| 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x'
| 'y' | 'z'
| 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H'
| 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P'
| 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X'
| 'Y' | 'Z';

(* см. 7.2 *) decimal digit
= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7'
| '8' | '9';

(*
Представление следующих терминальных-символов определено в пунктах 7.3, 7.4 и в таблицах 1, 2.
*)
concatenate symbol = ',';
defining symbol = '=';
definition separator symbol = '|' | '/' | '!';
end comment symbol = '*)';
end group symbol = ')';
end option symbol = ']' | '/)';
end repeat symbol = '}' | ':)';
except symbol = '-';
first quote symbol = "'";
repetition symbol = '*';
second quote symbol = '"';
special sequence symbol = '?';
start comment symbol = '(*';
start group symbol = '(';
start option symbol = '[' | '(/';
start repeat symbol = '{' | '(:';
terminator symbol = ';' | '.';

(* см. п.7.5 *) other character
= ' ' | ':' | '+' | '_' | '%' | '@'
| '&' | '#' | '$' | '<' | '>' | '\'
| '^' | '`' | '~';

(* см. П.7.6 *) space character = ' ';

horizontal tabulation character
= ? ISO 6429 символ горизонтальной табуляции ?
;

new line
= { ? ISO 6429 символ перевода каретки ? },
? ISO 6429 символ перехода на новую строку ?,
{ ? ISO 6429 символ перевода каретки ? };

vertical tabulation character
= ? ISO 6429 символ вертикальной табуляции ? ;

form feed
= ? ISO 6429 символ подачи бумаги ? ;

(*
Вторая часть синтаксиса определяет удаление из синтаксиса ненужных непечатаемых символов.
*)
(* см п.6.2 *) terminal character
= letter
| decimal digit
| concatenate symbol
| defining symbol
| definition separator symbol
| end comment symbol
| end group symbol
| end option symbol
| end repeat symbol
| except symbol
| first quote symbol
| repetition symbol
| second quote symbol
| special sequence symbol
| start comment symbol
| start group symbol
| start option symbol
| start repeat symbol
| terminator symbol
| other character;

(* см. п.6.3 *) gap free symbol
= terminal character - (first quote symbol |
second quote symbol) | terminal string;

(* см. п.4.16 *) terminal string
= first quote symbol, first terminal character,
{first terminal character}, first quote symbol
| second quote symbol,
second terminal character, {second terminal
character}, second quote symbol;

(* см. п.4.17 *) first terminal character
= terminal character - first quote symbol;

(* см. п.4.18 *) second terminal character
= terminal character - second quote symbol;

(* см. п.6.4 *) gap separator
= space character
```

```
| horizontal tabulation character
| new line
| vertical tabulation character
| form feed;

(* см.6.5 *) syntax
= {gap separator},
gap free symbol, {gap separator},
{gap free symbol, {gap separator}};

(*
Третья часть синтаксиса определяет удаление комментариев, ограниченных скобками, из не содержащих разрывов символов, образующих синтаксис.
*)
(* см. п.6.6 *) commentless symbol
= terminal character
- (letter
| decimal digit
| first quote symbol
| second quote symbol
| start comment symbol
| end comment symbol
| special sequence symbol
| other character)
| meta identifier
| integer
| terminal string
| special sequence;

(* см. п.4.9 *) integer
= decimal digit, {decimal digit};

(* см. п.4.14 *) meta identifier
= letter, {meta identifier character};

(* см. п.4.15 *) meta identifier character
= letter
| decimal digit;

(* см. п.4.19 *) special sequence
= special sequence symbol,
{special sequence character},
special sequence symbol;

(* см п.4.20 *) special sequence character
= terminal character - special sequence symbol;

(* см. п.6.7 *) comment symbol
= bracketed textual comment
| other character
| commentless symbol;

(* см. п.6.8 *) bracketed textual comment
= start comment symbol, {comment symbol},
end comment symbol;

(* см. п.6.9 *) syntax
= {bracketed textual comment},
commentless symbol,
{bracketed textual comment},
{commentless symbol,
{bracketed textual comment}};

(*
Финальная часть синтаксиса описывает абстрактный синтаксис Расширенной ФБН, т.е. структуру в терминах символов, свободных от комментариев.
*)
(* см.п.4.2 *) syntax
= syntax rule, {syntax rule};

(* см.п.4.3 *) syntax rule
= meta identifier, defining symbol,
definitions list, terminator symbol;

(* см. п.4.4 *) definitions list
= single definition,
{definition separator symbol,
single definition};

(* см. п.4.5 *) single definition
= syntactic term,
{concatenate symbol, syntactic term};

(* см. п.4.6 *) syntactic term
= syntactic factor,
[except symbol, syntactic exception];

(* см. п.4.7 *) syntactic exception
= ? синтаксический-фактор, кторый мог бы быть представлен синтаксическим-фактором, не содержащим мета-идентификаторов ? ;

(* см. п.4.8 *) syntactic factor
= [integer, repetition symbol],
syntactic primary;

(* см. п.4.10 *) syntactic primary
= optional sequence
| repeated sequence
| grouped sequence
| meta identifier
| terminal string
| special sequence
| empty sequence;

(* см. п.4.11 *) optional sequence
= start option symbol, definitions list,
end option symbol;

(* см. п.4.12 *) repeated sequence
= start repeat symbol, definitions list,
end repeat symbol;

(* см. п.4.13 *) grouped sequence
= start group symbol, definitions list,
end group symbol;

(* см. п.4.21 *) empty sequence
= ;
```

8.2 Расширенная ФБН для неформального определения самой себя.

```
(*
Этот пример определяет Расширенную ФБН неформально. Многие синтаксические правила включают комментарий для пояснения своего назначения; внутрь комментария, мета-идентификатор вложен в угловые скобки < и >, дабы избежать путаницы с подобными английскими словами. Нетерминальные символы <letter>, <decimal digit> и <character> не определены. Позиция <comments> заявлена в комментарии, но формально не определена.
*)
syntax = syntax rule, {syntax rule};

syntax rule
```

```
= meta identifier, '=', definitions list,
';'
(* <syntax rule> определяет последователь-
ности символов, представленных через <meta
identifier> *);

definitions list
= single definition, {'|',
single definition}
(* | разделяет альтернативные
<single definitions> *);

single definition = term, {',', term}
(* , separates successive <terms> *);

term = factor, ['-', exception]
(* <term> представляет любую последова-
тельность символов, которые определены с
помощью  <factor>, но не определены по-
средством <exception> *);

exception = factor
(*<factor> может быть использован как
<exception>, если это может быть заменено
на <factor>, не содержащий <meta identifi-
ers> *);

factor = [integer, '*'], primary
(* <integer> указывает число повторений
<primary> *);

primary
= optional sequence | repeated sequence
| special sequence | grouped sequence
| meta identifier | terminal string |
empty;

empty = ;

optional sequence = '[', definitions list,
']'
(* Скобки [ и ] содержат необязательные
символы *);

repeated sequence = '{', definitions list,
'}'
(* Скобки { и } содержат символы, которые
могут повторяться некоторое количество раз
*);

grouped sequence = '(', definitions list,
')'
(* Скобки ( и ) позволяют любому
<definitions list> быть как <primary> *);

terminal string
= "'", character - "'", {character - "'"},
"'"
| '"', character - '"', {character - '"'},
'"'
(* <terminal string> представляет
<characters> между символами кавычек '_'
или "_" *);

meta identifier = letter, {letter |
decimal digit}
(* <meta identifier> - это имя синтаксиче-
ского элемента языка, который будет опре-
делён *);

integer = decimal digit, {decimal digit};

special sequence = '?', {character - '?'},
'?'
(* Значение <special sequence> не опреде-
лено в стандарте метаязыка. *);

comment = '(*', {comment symbol}, '*)'
(* Комментарий может находиться где угодно
вне <terminal string>, <meta identifier>,
<integer> или <special sequence> *);

comment symbol
= comment | terminal string |
special sequence | character;
```

8.3 Расширенная ФБН, определённая неформально

```
(*
ЭТОТ ПРИМЕР ИСПОЛЬЗУЕТ ПРЕДСТАВЛЕНИЕ,
ОПРЕДЕЛЁННОЕ В ТАБЛИЦЕ 2
*)
SYNTAX = SYNTAX RULE, (: SYNTAX RULE :).
SYNTAX RULE
= META IDENTIFIER, '=', DEFINITIONS LIST,
'.'.
DEFINITIONS LIST = SINGLE DEFINITION,
(: '/', SINGLE DEFINITION :).
SINGLE DEFINITION = TERM, (: ',', TERM :).
TERM = FACTOR, (/ '-', EXCEPTION /).
EXCEPTION = FACTOR.
FACTOR = (/ INTEGER, '*' /), PRIMARY.
PRIMARY =
OPTIONAL SEQUENCE / REPEATED SEQUENCE
/ SPECIAL SEQUENCE / GROUPED SEQUENCE
/ META IDENTIFIER / TERMINAL / EMPTY.
EMPTY = .
OPTIONAL SEQUENCE = '(/', DEFINITIONS
LIST, '/)'.
REPEATED SEQUENCE = '(:', DEFINITIONS
LIST, ':)'.
GROUPED SEQUENCE = '(', DEFINITIONS LIST,
')'.
TERMINAL
= "'", CHARACTER - "'",
(: CHARACTER - "'" :), "'"
/ '"', CHARACTER - '"',
(: CHARACTER - '"' :), '"'.
META IDENTIFIER = LETTER, (: LETTER /
DIGIT :).
INTEGER = DIGIT, (: DIGIT :).
SPECIAL SEQUENCE = '?', (: CHARACTER - '?'
:), '?'.
COMMENT = '(*', (: COMMENT SYMBOL :),
'*)'.
COMMENT SYMBOL
= COMMENT / TERMINAL / SPECIAL SEQUENCE
/ CHARACTER.
```

Приложение A
(информативное)
Двухуровневая грамматика

A.1 Для большинства пользователей, тех возможностей, которые описаны в данном Международном Стандарте, более чем достаточно. Тем не менее, некоторые пользователи захотят создать более мощные расширения. Это приложение иллюстрирует возможности советуя, каким именно образом *Расширенная ФБН* может быть расширена для определения двухуровневой грамматики. Этот вид грамматики использовавшийся, например, в Algol 68 (*van Wijngaarden, 1975*), предоставляет более точный, но менее прямой способ определения языков. Хотя нотация (так же известная как *грамматика Wijngaarden*, или как *W-грамматика*) является более мощной, она более сложна для понимания и, как признались авторы Algol 68: «может оказаться трудной для неподготовленного читателя».

В двухуровневой грамматике существует два вида правил. Некоторые, именуемые *гипер-правилами*, подобны *синтаксическим-правилам* в *Расширенной ФБН*, за исключением того, что они могут включать специальные слова, известные как *метапонятия*. Другие правила, именуемые *мета-порождающими-правилами*, определяют последовательности символов, которые соответствуют каждому *метапонятию*. *Синтаксические-правила* языка сгенерированы путём соответствующей замены каждого *метапонятия* в *гипер-правиле*. Если *метапонятие* встречается в *гипер-правиле* более одного раза, то в процессе генерации *синтаксического-правила* замена будет выполнена на каждом повторе. *Метапонятия* внутри *терминальных-строк* и комментариев так же систематически заменяются.

A.2 Небольшое дополнительное примечание, необходимое для расширения *Расширенной ФБН* в двухуровневую грамматику:

a) Введите *мета-порождающий-символ*, например ==, т.о. можно будет отличить *мета-порождающее-правило* от *гипер-правила*.
b) Отделите *метапонятия* от других *гипер-понятий* путём использования прописных букв для *метапонятий* и *символов-нижнего-регистра* для *гипер-понятий* и *мета-идентификаторов*.

Например, язык, определённый с помощью двухуровневой грамматики:

```
metaproduction rule:
INTREAL == integer | real;

hyper-rules:
program = {statement}, 'end';

statement = INTREAL statement;

INTREAL statement
= 'print INTREAL', INTREAL expression;

INTREAL expression = INTREAL value,
{('+' | '-' | '*' | '/'), INTREAL value};

integer value = digit, {digit};

real value
= digit, '.', digit, {digit}, '@', digit;
```

эквивалентен языку (определённому с помощью *Расширенной ФБН*):

```
program = {statement}, 'end';

statement = integer statement;

statement = real statement;

integer statement
= 'print integer', integer expression;

real statement = 'print real', real
expression;

integer expression = integer value,
{('+' | '-' | '*' | '/'), integer value};

real expression = real value,
{('+' | '-' | '*' | '/'), real value};

integer value = digit, {digit};

real value
= digit, '.', digit, {digit}, '@', digit;
```

A.3 Синтаксис *Расширенной ФБН* должен был бы быть изменён следующим образом:

a) Добавьте следующие дополнительные правила:

```
metaproduction rule
= metanotion, metaproduction
defining symbol, hypernotion,
{definition separator symbol,
hypernotion}, terminator symbol;

metanotion
= upper case letter, {upper case
letter};

metaproduction defining symbol =
"==";

hypernotion
= letter, {letter | decimal digit};

upper case letter
= 'A' | 'B' | 'C' | 'D' | 'E' | 'F'
| 'G' | 'H' | 'I' | 'J' | 'K' | 'L'
| 'M' | 'N' | 'O' | 'P' | 'Q' | 'R'
| 'S' | 'T' | 'U' | 'V' | 'W' | 'X'
| 'Y' | 'Z';

lower case letter
= 'a' | 'b' | 'c' | 'd' | 'e' | 'f'
| 'g' | 'h' | 'i' | 'j' | 'k' | 'l'
| 'm' | 'n' | 'o' | 'p' | 'q' | 'r'
| 's' | 't' | 'u' | 'v' | 'w' | 'x'
| 'y' | 'z';
```

b) Измените существующие правила:

```
syntax = hyper rule, {hyper rule};

hyper rule = hypernotion, defining
symbol, definitions list, terminator
symbol;

syntactic primary
= optional sequence | repeated se-
quence | grouped sequence | hyperno-
tion | terminal string | special se-
quence | empty sequence;

letter = upper case letter | lower
case letter;

meta identifier
= lower case letter, {lower case
letter | decimal digit};
```

Тем не менее, это простое определение оставило бы некоторые проблемы нерешёнными: замена на *метапонятия* может быть определена не уникально и в этом месте возможно бесконечное количество порождающих правил которые, в свою очередь, могут порождать правила бесконечной длины.

Приложение B
(информативное)
Библиография

Следующие стандарты и исследовательские работы упомянуты во введении.

ISO 1539:1980, *Endorsement of ANSI X3.9-1978,* American National Standard — Programming Language FORTRAN. *American National Standards Institute, New York, USA. 1978.*

ISO/IEC 1539:1991, *Information technology — Programming languages — FORTRAN.*

ISO 1989:1985, *Endorsement of ANSI X3.23-1985,* American National Standard — Programming Language COBOL. *American National Standards Institute, New York, USA. 1985.*

ISO/IEC 9945-2 : 1993, *Information technology — Portable Operating System Interface (POSIX) — Part 2: Shell and utilities.*

BS 5904, *Programming languages - RTL/2, British Standards Institution, 1979.*

BS 5905, *Programming languages - CORAL 66, British Standards Institution, 1979.*

(Naur, 1960), *P Naur (Editor), Revised Report on the Algorithmic Language ALGOL 60, Computer Journal, Vol 5, No 4, pp349-367, Jan 1963.*

(van Wijngaarden, 1975), *A van Wijngaarden, B J Mailloux, J E L Peck, C H A Koster, M Sintzoff, C H Lindsey, L G L T Meertens, R G Fisker, Revised report on the Algorithmic Language ALGOL 68, Acta Informatica, Vol 5, parts 1-3, 1975 (also published in SIGPLAN Notices, Vol 12, No 5, pp1-70, May 1977).*

(Wirth, 1977), *N Wirth, What can we do about the unnecessary diversity of notation for syntactic definitions? Comm ACM, Vol 20, No 11, Nov 1977, p822.*

Переводчик: Андрей Бушман.
Обо всех обнаруженных ошибках и неточностях просьба сообщать в google-группу: https://groups.google.com/forum/?hl=ru#!forum/localizations
Там же вы всегда сможете найти последнюю отредактированную версию данного документа.