ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 13584-20— 2006

Системы автоматизации производства и их интеграция

БИБЛИОТЕКА ДЕТАЛЕЙ

Часть 20

Логический ресурс. Логическая модель выражений

ISO 13584-20:1998
Industrial automation systems and integration — Parts library — Part 20: Logical resource: Logical model of expressions
(IDT)

Издание официальное

БЗ 1—2006/450

Москва
Стандартинформ
2007

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Государственным научным учреждением «Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики» на основе собственного аутентичного перевода стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 459 «Информационная поддержка жизненного цикла изделий»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 27 декабря 2006 г. № 484-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 13584-20:1998 «Системы автоматизации производства и их интеграция. Библиотека деталей. Часть 20. Логический ресурс. Логическая модель выражений» (ISO 13584-20:1998 «Industrial automation systems and integration — Parts library — Part 20: Logical resource: Logical model of expressions»). При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении G

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2007

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения и сокращения . . . 1
3.1 Термины, определенные в ИСО10303-11 . . . 1
3.2 Термины, определенные в ИСО10303-44 . . . 2
3.3 Другие термины и определения . . . 2
3.4 Сокращения . . . 3
4 Основные понятия и допущения . . . 3
4.1 Статические и динамические данные . . . 3
4.2 Синтаксис выражений . . . 3
4.3 Семантика выражений . . . 3
4.4 Уровни абстракции при моделировании выражений . . . 4
4.5 Моделирование переменной . . . 4
4.6 Отображаемость на языке SQL . . . 5
5 ISO13584_generic_expressions_schema . . . 5
5.1 Введение . . . 5
5.2 Определения объектов ISO13584_generic_expressions_schema . . . 6
5.3 Определения функций ISO13584_generic_expressions_schema . . . 9
6 ISO13584_expressions_schema . . . 11
6.1 Введение . . . 11
6.2 Определения общих объектов ISO13584_expressions_schema . . . 12
6.3 ISO13584_expressions_schema: определения объектов для числовых выражений . . . 13
6.4 Определения объектов булевых выражений . . . 25
6.5 Определения объектов строковых выражений . . . 33
6.6 Функции для определения свойств выражения . . . 37
Приложение A (обязательное) Сокращенные наименования объектов . . . 44
Приложение B (обязательное) Регистрация информационного объекта . . . 45
Приложение C (справочное) Диаграммы на языке EXPRESS-G . . . 46
Приложение D (справочное) Использование схемы ISO13584_expressions_schema . . . 56
Приложение E (справочное) Конкретизация схем . . . 59
Приложение F (справочное) Статический анализ выражений . . . 63
Приложение G (справочное) Сведения о соответствии национальных стандартов Российской Федерации ссылочным международным стандартам . . . 64

Введение

Настоящий стандарт входит в комплекс стандартов ИСО 13584, устанавливающих требования к компьютерному представлению и обмену данными из библиотеки деталей. Цель комплекса — обеспечить нейтральный механизм передачи данных из библиотеки деталей независимо от того, какое приложение использует эти данные. Поэтому этот комплекс пригоден не только для обмена файлами, содержащими сведения о деталях, но также является основой для реализации и совместного использования баз данных, содержащих библиотеки деталей.

Каждая часть комплекса устанавливает требования, относящиеся к одной из тематических групп: концептуальные описания, логические ресурсы, ресурсы реализации, методология описания, аттестационное тестирование, протокол обмена представлениями и стандартизованное содержимое. Описания тематических групп приведены в ИСО 13584-1. Настоящий стандарт устанавливает требования, относящиеся к тематической группе логических ресурсов.

В настоящем стандарте установлены универсальные конструкции ресурсов на языке EXPRESS, необходимые для моделирования выражений, детализация которых приведена в других частях комплекса стандартов ИСО 13584. Эти конструкции могут также использоваться вне рамок этого комплекса там, где информационные модели выражений на языке EXPRESS могут быть полезными.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы автоматизации производства и их интеграция

БИБЛИОТЕКА ДЕТАЛЕЙ

Часть 20
Логический ресурс. Логическая модель выражений

Industrial automation systems and integration. Parts library.
Part 20. Logical resource. Logical model of expressions

Дата введения — 2007—07—01

1 Область применения

Настоящий стандарт устанавливает:

- EXPRESS-схему для обобщенных выражений;
- EXPRESS-схему для выражений, которая моделирует подмножество допустимых выражений на языке EXPRESS, требования к которому установлены в ИСО 10303-11, соответствующее целому, вещественному, булевому и строковому типам данных. В этой схеме используются ресурсы, определенные в схеме обобщенных выражений.

Область применения настоящего стандарта распространяются на:

- обмен выражениями, которые содержат как константы, так и переменные;
- функцию, которая проверяет, должно ли числовое выражение быть приведено к целому значению;
- константы, которые обеспечивают семантическую корректность выражения;
- вычисление переменных или функций, используемых в выражении;
- функцию, которая проверяет, может ли выражение быть отображено на языке запросов SQL.

Область применения настоящего стандарта не распространяется на:

- присвоение значений переменных в пределах некоторого контекста;
- переключающий механизм, который вычисляет значение выражения в заданном контексте.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие международные стандарты:

ИСО/МЭК 9075:1992 Информационные технологии. Языки баз данных. SQL

ИСО 10303-11:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS

ИСО 10303-44:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 44. Интегрированные обобщенные ресурсы. Конфигурация структуры изделия

3 Термины, определения и сокращения

3.1 Термины, определенные в ИСО 10303-11

В настоящем стандарте использованы следующие термины с соответствующими определениями.

3.1.1 **тип данных** (data type): Область значений.

3.1.2 **объект** (entity): Класс информации, определенный общими свойствами.

Издание официальное

3.1.3 **тип данных объекта** (entity data type): Представление объекта. Тип данных объекта устанавливает область значений, определяемую общими атрибутами и ограничениями.

3.1.4 **экземпляр объекта (типа данных)** (entity (data type) instance): Обозначенный блок данных, который представляет блок информации внутри класса, определенного объектом, и принадлежит к области значений, установленной типом данных объекта.

3.2 Термины, определенные в ИСО 10303-44

В настоящем стандарте использованы следующие термины с соответствующими определениями.

3.2.1 **узел-предок** (ancestor node): Любой узел, к которому можно прийти из данного узла путем последовательного обхода его связей в обратном направлении. Для заданного узла его узлами-предками являются все узлы-родители, все узлы-родители этих узлов-родителей и т.д.

3.2.2 **подчиненный узел** (child node): Узел, на который непосредственно указывает связь.

3.2.3 **узел-потомок** (descendent node): Любой узел, к которому можно прийти из данного узла путем последовательного обхода его связей. Для заданного узла узлами-потомками являются все подчиненные узлы, все подчиненные узлы этих подчиненных узлов и т.д.

3.2.4 **ориентированный ациклический граф** (directed acyclic graph): Набор узлов и связей, в котором ни один узел не является узлом-предком (или узлом-потомком) по отношению к самому себе.

3.2.5 **связь** (link): Однонаправленное отношение одного узла с другим внутри ориентированного ациклического графа.

3.2.6 **узел** (node): Элемент ориентированного ациклического графа, соединенный с другим подобным элементом посредством связей.

3.2.7 **узел-родитель** (parent node): Узел, из которого исходит связь.

3.3 Другие термины и определения

В настоящем стандарте использованы следующие термины с соответствующими определениями.

3.3.1 **арность оператора** (arity of an operator): Максимальное число операндов, которые должны быть связаны с оператором.

3.3.2 **бинарный оператор** (binary operator): Оператор, арность которого равна двум.

3.3.3 **среда** (environment): Связь между синтаксисом и семантикой.

Примечание — В контексте настоящего стандарта объект **environment** привязывает к объекту **generic_variable** (синтаксису) соответствующее ему значение (семантику), представленное объектом **variable_semantics**.

3.3.4 **оценка** (evaluation): Вычисление значения, представленного в виде выражения.

3.3.5 **выражение** (expression): Набор переменных и/или констант, которые объединяются операторами.

Примечания

1 Выражение определяет функцию, чьими аргументами являются переменные, встречающиеся в выражении.

2 Структурой, представляющей выражение, является ориентированный ациклический граф, узлами которого являются операторы, константы и переменные. Связи в этом графе представляют собой однонаправленное отношение каждого оператора со своими аргументами, которые также могут быть выражениями.

3.3.6 **тип данных выражения** (expression data type): Область, в которой должны находиться значения результата выражения.

Примечание — В настоящем стандарте контроль типа и синтез типа проводят только для схемы **ISO13584_expressions_schema**. Контроль типа обеспечивается структурой и правилами этой EXPRESS-схемы. Синтез типа выполняется EXPRESS-функцией TYPEOF, которая определяет, принадлежит ли объект **expression** к **numeric_expression**, **Boolean_expression** или **string_expression**, а также функцией **is_int**, которая определяет, относится ли значение **numeric_expression** к типу INTEGER.

3.3.7 **интерпретация** (interpretation): Функция, которая возвращает семантику, связанную с каждой переменной.

Примечание — Данная функция использует среду, которая связывает переменную с соответствующей ей семантикой (и, может быть, с ее возможным значением).

3.3.8 **оператор большой арности** (multiple arity operator): Оператор, арность которого больше двух.

3.3.9 **оператор** (operator): Функция, которая соединяет одно или несколько значений, называемых операндами, чтобы выработать значение, называемое результатом.

Примечание — Определение оператора включает в себя определение типов данных его операндов и результата.

3.3.10 **семантика** (semantics): Смысл заданного понятия.

Пример — Семантикой переменной является смысл, выраженный посредством этой переменной.

Примечание — В контексте настоящего стандарта семантика представлена объектом **variable_semantics**. Типом этого объекта является ABSTRACT SUPERTYPE, который может быть использован для передачи конкретных смыслов и значений.

3.3.11 **синтаксическое представление** (syntactic representation): Последовательность символов, которая представляет заданное понятие.

Примечания

1 В общепринятых языках программирования последовательность символов, используемая для представления различных понятий, должна подчиняться набору правил, называемому синтаксисом языка.

2 В контексте настоящего стандарта синтаксическое представление представляет собой имя экземпляра объекта (типа данных).

3.3.12 **контроль типа** (type control): Операция, которая позволяет определить, правильно или нет данное выражение отнесено к определенному типу.

Примечания

1 Выражение отнесено к определенному типу правильно, если тип данных каждого операнда каждого оператора этого выражения соответствует требуемому типу данных.

2 В настоящем стандарте контроль типа обеспечивается ограничениями EXPRESS-схемы.

3.3.13 **синтез типа** (type synthesis): Определение типа данных выражения.

Примечание — В схеме **ISO13584_expressions_schema** тип данных каждой константы, переменной или результата оператора передается объектом, который ее представляет, поэтому тип данных выражения **expression** может быть выведен из результата EXPRESS-функции TYPEOF, примененной к узлу, который представляет это выражение.

3.3.14 **унарный оператор** (unary operator): Оператор, арность которого равна единице.

3.3.15 **переменная** (variable): Представление значения, которое должно принадлежать к определенному типу данных.

3.4 Сокращения

В настоящем стандарте используется следующее сокращение, установленное в ИСО/МЭК 9075:1992:

SQL — Structured Query Language (структурированный язык запросов).

4 Основные понятия и допущения

4.1 Статические и динамические данные

При обмене информацией из библиотеки деталей возникает необходимость обмениваться не только статическими свойствами, которые могут быть смоделированы в виде данных, но также и динамическими характеристиками, которые выражают, например, как значение свойства может быть выведено из значений других свойств. Выражения являются одной из структур, которые способны моделировать динамические характеристики.

Настоящий стандарт устанавливает форму однозначного представления и обмена машинно-интерпретируемыми выражениями.

4.2 Синтаксис выражений

Синтаксис выражений состоит из ряда символов, которые являются константами, переменными и операторами этого выражения. В текстовых языках эти символы подчиняются набору правил, обычно определяемому в грамматике.

В настоящем стандарте константы, переменные и операторы представлены как типы данных объектов. Правила, которым должны подчиняться эти объекты, моделируются в EXPRESS-схеме, которая определяет типы данных объектов.

4.3 Семантика выражений

4.3.1 Семантика выражений

Выражения состоят из операторов и операндов. Семантика выражения определяется:

- диапазоном функции, выполняемой каждым оператором;
- функцией интерпретации, которая связывает соответствующее значение с каждым операндом;
- функцией оценки, которая вычисляет результат каждого оператора в случае ее применения к его операндам.

В настоящем стандарте функция интерпретации моделируется определением подтипа объекта **variable_semantics**. Функция оценки в настоящем стандарте не рассматривается.

4.3.2 Период обмена и период оценки

Обработка выражений включает в себя следующие два типа процедур, соответствующих периоду обмена и периоду оценки:

1) В период обмена выражение представлено своей структурой. Соответствующий ориентированный ациклический граф моделируется и обменивается. При этом не требуется, чтобы переменные были ограничены какими-либо значениями, но их семантика известна. На этом уровне могут быть выполнены различные виды статического анализа (ацикличность графа, проверка типов, отображаемость на языке SQL).

2) В период оценки выражению может быть присвоено значение. На этом этапе все переменные, встречающиеся в выражении, должны получить какое-либо значение. Динамическая оценка выражения, тестирование и отладка могут быть выполнены на этом уровне.

В настоящем стандарте рассматривается только статический анализ, который соответствует периоду обмена. Обзор различных анализов выражений приведен в приложении F.

4.4 Уровни абстракции при моделировании выражений

Операторы, использованные в **ISO_generic_expressions_schema**, позволяют определять абстрактные обобщенные выражения, которые могут быть конкретизированы для различных целей и типов данных.

4.4.1 Конкретизация ISO13584_generic_expressions_schema

ISO13584_expressions_schema является конкретизацией схемы **ISO13584_generic_expressions_schema**. Операторы, используемые в **ISO13584_expressions_schema**, принадлежат к подмножеству операторов, определенных в ИСО 10303-11. Их диапазон и функция оценки должны соответствовать спецификации, установленной в ИСО 10303-11.

Операнды являются либо константами, представленными в виде литеральных значений, либо переменными, либо другими выражениями. Переменные имеют строго определенный тип и предполагается, что, когда выражение оценивается в некотором контексте, функция интерпретации обеспечивает каждую переменную значением, соответствующим ее типу данных. Если это условие не соблюдается, то результат оценки выражения будет ошибочным.

В приложении E рассмотрены детали процесса конкретизации и приведен метод создания других возможных конкретизаций **ISO13584_generic_expressions_schema**.

4.4.2 Конкретизация ISO13584_expressions_schema

ISO13584_expressions_schema может быть конкретизирована, чтобы соответствовать требованиям, которые не ограничиваются операторами с числовыми, строковыми и булевыми значениями, определенными в руководстве по языку EXPRESS в ИСО 10303-11.

Конкретизация **ISO13584_expressions_schema** заключается либо в расширении схемы за счет добавления определений новых объектов, которые выражают семантику функций, возвращая целое, вещественное, булево или строковое значение, либо в определении новых подтипов **variable_semantics**.

Объекты, которые выражают семантику функций, должны быть определены как подтипы соответствующего объекта **defined_function**. Действительно, функция, возвращающая целое, вещественное, булево или строковое значение, соответственно должна быть определена как подтип **integer_defined_function**, **real_defined_function**, **boolean_defined_function** и **string_defined_function**.

Такая конкретизация совместима со строгой проверкой типа данных, что следует из правил схемы **ISO13584_expressions_schema**.

В приложении E установлены детали процесса конкретизации и приведена методика конкретизации **ISO13584_expressions_schema**.

4.5 Моделирование переменной

Переменная имеет три свойства:

- представляет собой символ (синтаксический), который может быть использован для построения выражения;
- ассоциируется с типом данных, который определяет область ее значений;
- ассоциируется с семантикой, которая определяет ее смысл и, следовательно, ее значение в период оценки.

4.5.1 Синтаксическое представление

В настоящем стандарте переменная является экземпляром объекта **generic_variable**. Экземпляр такого типа данных связывается с идентификатором, определяющим индивидуальность экземпляра, который вводит в действие символ соответствующей переменной, когда она используется в выражении.

4.5.2 Область значений переменной

Область значений переменной моделируется определением подтипа объекта **generic_variable**. Этот подход, известный как строгое определение типа, обеспечивает, что тип данных любого выражения может быть вычислен (синтезирован) в период обмена.

4.5.3 Семантика переменной

В **ISO13584_generic_expressions_schema** должны входить переменные, связанные с **variable_semantics**. Этот объект определен как ABSTRACT SUPERTYPE и должен быть переопределен через подтип при использовании конкретной семантики. Описание подтипа объекта **variable_semantics** должно содержать описание контекста, в котором эта переменная должна использоваться. Это описание должно сопровождаться описанием функции интерпретации, которая связывает значение с этой переменной.

Пример — Запись на языке EXPRESS SELF.a является примером variable_semantics. Такая переменная может быть использована только в контексте некоторого экземпляра некоторого класса, в котором появляется ее синтаксическое представление. В рамках этого контекста функция интерпретации является функцией, которая связывает с этой переменной значение атрибута «а» этого экземпляра.

В приложении D приведены различные методы, которые могут быть использованы для определения подтипов объекта **variable_semantics**, соответствующих некоторым конкретным требованиям. Это обеспечивает конкретизацию определения функции интерпретации через объект **variable_semantics**.

4.6 Отображаемость на языке SQL

В языках программирования выражения обычно используются для доступа к базам данных. Язык SQL включает только небольшое число операторов и ограничивает использование выражений до заданной категории выражений. Для того чтобы сделать понятным различие между выражениями, которые являются SQL-отображаемыми, и остальными выражениями, специально введенный булев атрибут указывает, является или нет выражение отображаемым на языке SQL.

5 ISO13584_generic_expressions_schema

В данном разделе определены требования к схеме **ISO13584_generic_expressions_schema**. Приведенное ниже описание на языке EXPRESS представляет собой блок **ISO13584_generic_expressions_schema**.

EXPRESS-спецификация

```
*)
SCHEMA ISO13584_generic_expressions_schema;
(*
```

5.1 Введение

ISO13584_generic_expressions_schema представляет собой абстрактный ресурс, который обеспечивает общие рамки для набора всех возможных выражений.

Пример — В параметрической модели фигуры вещественное значение может быть представлено как выражение, операндами которого являются элементы геометрического представления, например distance (point_1, point_2). Геометрический элемент может быть определен геометрическим оператором, например centre_of (circle_1).

Основной структурой является ациклический граф, чьими узлами являются операторы, литеральные значения и переменные. Дуги графа изображают связь операторов со своими аргументами, которые сами являются выражениями.

Схема **ISO13584_generic_expressions_schema** предоставляет следующие возможности:

- обобщенные константы, соответствующие значениям обрабатываемого типа данных;
- обобщенные переменные, обозначающие значения, интерпретируемые из контекста;
- унарные выражения для унарных операторов;
- бинарные выражения для бинарных операторов;
- выражения большой арности для операторов большой арности.

Каждый конкретный вид выражений получается путем конкретизации (фактически, путем задания подтипа) констант, переменных и операторов, т.е. определения для них обоснованного типа (или типов) данных.

5.2 Определения объектов ISO13584_generic_expressions_ schema

5.2.1 Объект generic_expression

Объект **generic_expression** является информационной моделью обобщенного выражения. Он представляет собой ABSTRACT SUPERTYPE (абстрактный супертип) всех возможных выражений. Для того чтобы обеспечить ацикличность выражений, его разбивают на подтипы в соответствии с его арностью. Если подтип **generic_expression** не является подтипом **simple_generic_expression** либо **unary_generic_expression**, либо **binary_generic_expression**, либо **multiple_arity_generic_expression**, то он не должен содержать ни одной переменной.

EXPRESS-спецификация

```
*)
ENTITY generic_expression
ABSTRACT SUPERTYPE OF(ONEOF(simple_generic_expression,
                            unary_generic_expression,
                            binary_generic_expression,
                            multiple_arity_generic_expression));
WHERE
    WR1: is_acyclic(SELF);
END_ENTITY;
(*
```

Формальное утверждение

WR1 — граф, связанный с описанным выражением, должен быть ациклическим.

Неформальное утверждение

IP1 — либо объект **generic_expression** не должен содержать ни одной переменной **generic_variable**, либо он должен быть подтипом **simple_generic_expression**, **unary_generic_expression**, либо **binary_generic_expression**, либо **multiple_arity_generic_expression**.

Примечание — Это неформальное утверждение обеспечивает то, что набор переменных, вычисленных функцией **used_variable** (см. 5.3.2), включает в себя все переменные, имеющиеся в **generic_expression**.

5.2.2 Объект simple_generic_expression

Объект **simple_generic_expression** является обобщенным выражением, которое олицетворяет либо обобщенную переменную, либо обобщенный литерал.

EXPRESS-спецификация

```
*)
ENTITY simple_generic_expression
ABSTRACT SUPERTYPE OF (ONEOF(generic_literal, generic_variable))
SUBTYPE OF (generic_expression);
END_ENTITY;
(*
```

5.2.3 Объект generic_literal

Объект **generic_literal** является абстрактной константой, которая может быть включена в обобщенное выражение.

EXPRESS-спецификация

```
*)
ENTITY generic_literal
ABSTRACT SUPERTYPE
SUBTYPE OF (simple_generic_expression);
END_ENTITY;
(*
```

5.2.4 Объект generic_variable

Объект **generic_variable** является абстрактной переменной, которая может быть включена в обобщенное выражение. Для объекта **generic_variable** должен быть определен подтип, чтобы установить тип данных допустимых значений этой переменной. Переменная должна быть связана со средой, которая представляет собой функцию интерпретации, связывающую значение с этой переменной.

Примечание — **real_numeric_variable**, **integer_numeric_variable**, **boolean_numeric_variable** и **string_numeric_variable** являются примерами подтипов объекта **generic_variable**. Эти подтипы определены в разделе 6.

EXPRESS-спецификация

```
*)
ENTITY generic_variable
ABSTRACT SUPERTYPE
SUBTYPE OF (simple_generic_expression);
INVERSE
        interpretation :
                environment FOR syntactic_representation;
END_ENTITY;
(*
```

Определение атрибута

interpretation — объект **environment** (среда), который позволяет связывать значение с переменной.

5.2.5 Объект variable_semantics

Объект **variable_semantics** используется для представления смысла **generic_variable**. Он относится к абстрактному супертипу ABSTRACT SUPERTYPE, для которого должны быть определены подтипы при использовании **variable_semantics**. Объект **variable_semantics** должен определять контекст, в рамках которого переменная должна использоваться вместе с функцией интерпретации, связывающей значение с этой переменной.

Пример — При моделировании класса компонентов, которые имеют свойства «a» и «b» с целыми значениями, ограничение, устанавливающее, что для каждого экземпляра этого класса значение «a» должно быть больше значения «b», может быть смоделировано посредством:

- двух экземпляров real_variable;

- двух экземпляров self_property_semantics, являющегося подтипом variable_semantics, которые относятся к свойству «a» и свойству «b», соответственно;

- выражения boolean_expression между двумя объектами real_variable, которое указывает, что первый объект больше, чем второй.

EXPRESS-спецификация

```
*)
ENTITY variable_semantics
ABSTRACT SUPERTYPE;
END_ENTITY;
(*
```

5.2.6 Объект environment

Объект **environment** обеспечивает связь между синтаксисом и семантикой переменной. Он представляет собой функцию интерпретации, которая связывает значение с **generic_variable**.

EXPRESS-спецификация

```
*)
ENTITY environment;
    syntactic_representation: generic_variable;
    semantics: variable_semantics;
END_ENTITY;
(*
```

Определения атрибутов

syntactic_representation — объект **generic_variable**, который обозначает значение переменной;

semantics — смысл переменной, который включает в себя механизм для доступа к ее значению.

5.2.7 Объект unary_generic_expression

Объект **unary_generic_expression** является абстрактным супертипом ABSTRACT SUPERTYPE для всех унарных операторов.

EXPRESS-спецификация

```
*)
ENTITY unary_generic_expression
ABSTRACT SUPERTYPE
SUBTYPE OF (generic_expression);
    operand: generic_expression;
END_ENTITY;
(*
```

Определение атрибута

operand — обобщенное выражение, которое представляет собой операнд унарного оператора.

5.2.8 Объект binary_generic_expression

Объект **binary_generic_expression** является абстрактным супертипом ABSTRACT SUPERTYPE для всех бинарных операторов.

EXPRESS-спецификация

```
*)
ENTITY binary_generic_expression
ABSTRACT SUPERTYPE
SUBTYPE OF (generic_expression);
    operands: LIST [2:2] OF generic_expression;
END_ENTITY;
(*
```

Определение атрибута

operands — список из двух обобщенных выражений, которые представляют собой два операнда бинарного оператора.

5.2.9 Объект multiple_arity_generic_expression

Объект **multiple_arity_generic_expression** является абстрактным супертипом ABSTRACT SUPERTYPE для всех операторов большой арности.

EXPRESS-спецификация

```
*)
ENTITY multiple_arity_generic_expression
ABSTRACT SUPERTYPE
```

```
SUBTYPE OF(generic_expression);
        operands: LIST [2:?] OF generic_expression;
END_ENTITY;
(*
```

Определение атрибута

operands — список обобщенных выражений, которые представляют собой операнды оператора большой арности.

5.3 Определения функций ISO13584_generic_expressions_schema

В данном подразделе определены функции схемы **ISO13584_generic_expressions_schema**.

5.3.1 Функция is_acyclic

Функция **is_acyclic** проверяет наличие цикла в рассматриваемом графе выражения. В ней определены две функции. Первая (**is_acyclic**) имеет в качестве аргумента объект **generic_expression**, который должен быть проверен. Затем эта функция вызывает вторую (**acyclic**) с двумя аргументами: **generic_expression** и пустой набор. Этот набор должен быть модифицирован в результате выполнения функции таким образом, чтобы он содержал набор всех узлов, уже прослеженных по цепочке рекурсивных вызовов вниз до данного узла.

Функция **is_acyclic** возвращает значение TRUE (истина), если представленная аргументом **arg** структура является ориентированным ациклическим графом, в противном случае она возвращает значение FALSE (ложь).

Примечание — В приложении F определена роль такой структуры в статическом анализе (например, контроль типа, сбор переменных и т.д.) и в динамическом анализе (например, оценка) выражения

EXPRESS-спецификация

```
*)
FUNCTION is_acyclic (arg: generic_expression): BOOLEAN;
RETURN (acyclic (arg, []));
END_FUNCTION ; -- is_acyclic

FUNCTION acyclic (arg1: generic_expression;
                  arg2: SET OF generic_expression): BOOLEAN;

LOCAL
    result: BOOLEAN;
END_LOCAL;

IF ('ISO13584_GENERIC_EXPRESSIONS_SCHEMA.SIMPLE_GENERIC_EXPRESSION'
   IN TYPEOF (arg1))
THEN
   RETURN (TRUE);

END_IF;

IF arg1 IN arg2
THEN
   RETURN (FALSE);
END_IF;

IF 'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.UNARY_GENERIC_EXPRESSION'
   IN TYPEOF (arg1)
THEN
    RETURN
    (acyclic(arg1\unary_generic_expression.operand,arg2+[arg1]));
END_IF;
```

```
        IF 'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.BINARY_GENERIC_EXPRESSION'
        IN TYPEOF (arg1)
THEN
        RETURN
        (acyclic(arg1\binary_generic_expression.operands[1],arg2+[arg1])
        AND
        acyclic(arg1\binary_generic_expression.operands[2],arg2+[arg1]));
END_IF;

IF
'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.MULTIPLE_ARITY_GENERIC_EXPRESSION'
     IN TYPEOF (arg1)
THEN
        result := TRUE;
        REPEAT i := 1 TO
                SIZEOF (arg1\multiple_arity_generic_expression.operands);
            result := result AND
            acyclic(arg1\multiple_arity_generic_expression.operands[i], arg2+[arg1]);
        END_REPEAT;

        RETURN (result);
END_IF;

END_FUNCTION; -- acyclic
(*
```

5.3.2 Функция used_variables

Функция **used_variables** просматривает весь граф обобщенного выражения, собирая все использованные переменные, и после этого возвращает их. Данная функция проходит весь ориентированный ациклический граф, представляющий **generic_expression**.

Примечание — Эта функция не используется в схеме **ISO13584_generic_expressions_schema**. Она является ресурсом, предназначенным для использования в схемах, которые используют **ISO13584_generic_expressions_schema** или ее возможные конкретизации. Например, эта функция используется в некоторых стандартах комплекса ИСО 13584 для записи ограничений на переменные, встречающиеся в выражениях.

EXPRESS-спецификация

```
*)
FUNCTION used_variables (arg : generic_expression) :
                SET OF generic_variable;

LOCAL
        result : SET OF generic_variable := [];
END_LOCAL;

IF 'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.GENERIC_VARIABLE'
        IN TYPEOF (arg)
THEN
        RETURN ([arg]);
END_IF;

IF 'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.UNARY_GENERIC_EXPRESSION'
    IN TYPEOF (arg)
THEN
    RETURN (used_variables (arg\unary_generic_expression.operand));

END_IF;
```

```
IF 'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.BINARY_GENERIC_EXPRESSION'
   IN TYPEOF (arg))
THEN
   RETURN(used_variables(arg\binary_generic_expression.operands[1])
      + used_variables (arg\binary_generic_expression.operands[2]));
END_IF;

IF
'ISO13584_GENERIC_EXPRESSIONS_SCHEMA.MULTIPLE_ARITY_GENERIC_EXPRESSION'
    IN TYPEOF (arg)
THEN
      REPEAT i := 1 TO
          SIZEOF(arg\multiple_arity_generic_expression.operands);
          result := result + used_variables(
              arg\multiple_arity_generic_expression.operands[i]);
      END_REPEAT;

      RETURN (result);
END_IF;
RETURN ([ ]);    -- в этом случае подтип не должен содержать
                 -- никаких переменных (см. IP1 в generic_expression)
END_FUNCTION; -- used_variables
END_SCHEMA; -- ISO13584_generic_expressions_schema
(*
```

6 ISO13584_expressions_schema

В данном разделе установлены требования к **ISO13584_expressions_schema**. Следующее описание на языке EXPRESS представляет блок **ISO13584_expressions_schema** и устанавливает необходимые внешние ссылки.

EXPRESS-спецификация

```
*)
SCHEMA ISO13584_expressions_schema;
REFERENCE FROM ISO13584_generic_expressions_schema(
                generic_expression,
                simple_generic_expression,
                generic_variable,
                generic_literal ,
                unary_generic_expression,
                binary_generic_expression,
                multiple_arity_generic_expression);
(*
```

6.1 Введение

ISO13584_expressions_schema обеспечивает моделирование числовых, булевых и строковых выражений, построенных из операторов и функций языка EXPRESS. Имена операторов и функций — по ИСО 10303-11. Они имеют ту же семантику, что и соответствующие элементы в ИСО 10303-11.

Основная структура — по **ISO13584_generic_expressions_schema**. Она представляет собой ориентированный ациклический граф, узлами которого являются операторы, литеральные значения и переменные, а связи представляют собой взаимосвязь операторов со своими аргументами, которые сами являются выражениями.

ISO13584_expressions_schema обеспечивает строгое определение типа выражения. Любое выражение обозначает либо числовое, либо строковое, либо булево значение. Этот тип данных может быть вычислен по структуре выражения. Чтобы дать возможность включить в выражение либо **generic_expression**, либо функцию, определенную в приложении, тип данных результата которой является либо числом, либо строкой, либо булевым значением, **ISO13584_expressions_schema** содержит

следующие абстрактные объекты: **numeric_defined_function**, **integer_defined_function**, **real_defined_function**, **string_defined_function** и **boolean_defined_function**. Значение, передаваемое любым из этих объектов, должно соответствовать типу данных значения, связанного с этим объектом. Предполагается, что **generic_expression**, которое не является подтипом одного из этих объектов, имеет значение, тип данных которого не является одним из типов, которые моделируются **ISO13584_expressions_schema**.

ISO13584_expressions_schema содержит информацию о том, может или нет выражение быть отображено на языке SQL, определение которого приведено в ИСО/МЭК 9075. Чтобы дать возможность включить в выражение либо **generic_expression,** либо функцию, определенную в приложении, которая может быть отображена на языке SQL, **ISO13584_expressions_schema** содержит абстрактный объект **SQL_mappable_defined_function**. Объект **generic_expression**, который является подтипом этого объекта, должен быть отображаемым на языке SQL. Предполагается, что объект **generic_expression**, который не является подтипом этого объекта, имеет структуру, которая не может быть отображена на языке SQL.

Выражения определяются введением подтипа объекта **generic_expression** из **ISO13584_generic_expressions_schema**. Все константы, переменные и операторы, определенные в **ISO13584_expressions_schema**, являются подтипами соответствующих объектов **ISO13584_generic_expressions_schema**. Это соответствие определяется арностью каждого оператора.

6.2 Определения общих объектов ISO13584_expressions_schema

6.2.1 Объект expression

Объект **expression** является частным случаем объекта **generic_expression**, ограниченным числовой, булевой или строковой областью значений.

EXPRESS-спецификация

```
*)
ENTITY expression
ABSTRACT SUPERTYPE OF (ONEOF (numeric_expression,
                              boolean_expression,
                              string_expression))
SUBTYPE OF (generic_expression);
END_ENTITY;
(*
```

6.2.1.1 Объект **variable** обозначает значение, интерпретированное из некоторого контекста.

EXPRESS-спецификация

```
*)
ENTITY variable
ABSTRACT SUPERTYPE OF (ONEOF (numeric_variable,
        boolean_variable,
        string_variable))
SUBTYPE OF(generic_variable);
END_ENTITY;
(*
```

6.2.1.2 Объект **defined_function** является функцией (строго определенного типа), которая предназначена для создания на ее основе подтипов в информационных моделях, в которых используется **ISO13584_expressions_schema**.

EXPRESS-спецификация

```
*)
ENTITY defined_function
ABSTRACT SUPERTYPE OF ((ONEOF (numeric_defined_function,
                               string_defined_function,
                                    boolean_defined_function)
```

```
                                    )
                        ANDOR SQL_mappable_defined_function);
END_ENTITY;
(*
```

6.2.1.3 Объект **SQL_mappable_defined_function** является функцией, которая предназначена для создания на ее основе подтипов в информационных моделях, в которых используется **ISO13584_expressions_schema**, для указания того, что некоторое выражение **generic_expression** является отображаемым на языке SQL.

EXPRESS-спецификация

```
*)
ENTITY SQL_mappable_defined_function
ABSTRACT SUPERTYPE
SUBTYPE OF (defined_function);
END_ENTITY;
(*
```

6.3 ISO13584_expressions_schema: определения объектов для числовых выражений

6.3.1 Объект numeric_expression

Объект **numeric_expression** представляет собой выражение, диапазоном которого является тип данных NUMBER, определенный в 8.1.1 ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY numeric_expression
ABSTRACT SUPERTYPE OF (ONEOF (simple_numeric_expression,
        unary_numeric_expression,
        binary_numeric_expression,
        multiple_arity_numeric_expression,
          length_function,
          value_function,
           numeric_defined_function))
SUBTYPE OF (expression);
DERIVE
    is_int: BOOLEAN := is_int_expr (SELF);
    sql_mappable: BOOLEAN := is_SQL_mappable (SELF);
END_ENTITY;
(*
```

Определения атрибутов

is_int — булев атрибут, который показывает, имеет ли оценка выражения целое значение или нет;

sql_mappable — булев атрибут, который показывает, является ли выражение отображаемым на языке SQL.

6.3.2 Объект simple_numeric_expression

Объект **simple_numeric_expression** представляет собой либо числовой литерал, либо числовую переменную.

EXPRESS-спецификация

```
*)
ENTITY simple_numeric_expression
ABSTRACT SUPERTYPE OF (ONEOF (literal_number, numeric_variable))
SUBTYPE OF (numeric_expression, simple_generic_expression);
END_ENTITY;
(*
```

6.3.3 Объект literal_number

Объект **literal_number** представляет собой литерал языка EXPRESS числового типа (NUMBER). Объект **literal_number** может быть либо целым литералом, либо вещественным литералом.

Примечание — Областью значений типа данных NUMBER языка EXPRESS являются числовые значения.

EXPRESS-спецификация

```
*)
ENTITY literal_number
ABSTRACT SUPERTYPE OF (ONEOF (int_literal, real_literal))
SUBTYPE OF (simple_numeric_expression, generic_literal);
    the_value: NUMBER;
END_ENTITY;
(*
```

Определение атрибута

the_value — значение литерала NUMBER.

6.3.4 Объект int_literal

Объект **int_literal** представляет собой литерал языка EXPRESS типа INTEGER.

Примечание — Литерал языка EXPRESS типа INTEGER представляет собой целое число VALUE, которое состоит из одной или более цифр.

EXPRESS-спецификация

```
*)
ENTITY int_literal
SUBTYPE OF (literal_number);
    SELF\literal_number.the_value: INTEGER;
END_ENTITY;
(*
```

Определение атрибута

the_value — значение литерала INTEGER.

Объект real_literal

Объект **real_literal** представляет собой литерал языка EXPRESS типа REAL.

Примечание — Литерал языка EXPRESS типа REAL представляет собой целое число VALUE, которое состоит из мантиссы и необязательной экспоненты; мантисса должна содержать десятичную точку.

EXPRESS-спецификация

```
*)
ENTITY real_literal
SUBTYPE OF (literal_number);
    SELF\literal_number.the_value: REAL;
END_ENTITY;
(*
```

Определение атрибута

the_value — значение литерала REAL.

6.3.6 Объект numeric_variable

Объект **numeric_variable** является переменной, которая представляет собой значение числа.

EXPRESS-спецификация

```
*)
ENTITY numeric_variable
```

```
SUPERTYPE OF (ONEOF (int_numeric_variable,
                     real_numeric_variable))
SUBTYPE OF (simple_numeric_expression, variable);
WHERE
     WR1:   ('ISO13584_EXPRESSIONS_SCHEMA .INT_NUMERIC_VARIABLE'
            IN TYPEOF (SELF) ) OR
            ('ISO13584_EXPRESSIONS_SCHEMA .REAL_NUMERIC_VARIABLE'
             IN TYPEOF (SELF));
END_ENTITY;
(*
```

Формальное утверждение

WR1 — объект **numeric_variable** может быть только целой или вещественной переменной.

6.3.7 Объект int_numeric_variable

Объект **int_numeric_variable** является переменной, которая представляет собой значение целого числа.

EXPRESS-спецификация

```
*)
ENTITY int_numeric_variable
SUBTYPE OF (numeric_variable);
END_ENTITY;
(*
```

6.3.8 Объект real_numeric_variable

Объект **real_numeric_variable** является переменной, которая представляет собой значение вещественного числа.

EXPRESS-спецификация

```
*)
ENTITY real_numeric_variable
SUBTYPE OF (numeric_variable);
END_ENTITY;
(*
```

6.3.9 Объект unary_numeric_expression

Объект **unary_numeric_expression** является унарным оператором, диапазоном которого является тип данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY unary_numeric_expression
ABSTRACT SUPERTYPE OF (ONEOF (unary_function_call))
SUBTYPE OF (numeric_expression, unary_generic_expression);
     SELF\unary_generic_expression.operand : numeric_expression;
END_ENTITY;
(*
```

Определение атрибута

operand — числовое выражение, которое представляет собой параметр унарного оператора.

6.3.10 Объект binary_numeric_expression

Объект **binary_numeric_expression** является бинарным оператором, диапазоном которого является тип данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY binary_numeric_expression
```

```
ABSTRACT SUPERTYPE OF (ONEOF (minus_expression,
                              div_expression,
                              mod_expression,
                              slash_expression,
                              power_expression,
                              binary_function_call))
SUBTYPE OF (numeric_expression, binary_generic_expression);
      SELF\binary_generic_expression.operands : LIST [2:2] OF
                                                numeric_expression;
END_ENTITY;
(*
```

Определение атрибута

operands — список, содержащий два параметра бинарного оператора.

6.3.11 Объект multiple_arity_numeric_expression

Объект **multiple_arity_numeric_expression** является оператором большой арности, диапазоном которого является тип данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY multiple_arity_numeric_expression
ABSTRACT SUPERTYPE OF (ONEOF (plus_expression,
                              mult_expression,
                              multiple_arity_function_call))
SUBTYPE OF (numeric_expression, multiple_arity_generic_expression);
      SELF\multiple_arity_generic_expression.operands : LIST [2:?] OF
                                                numeric_expression
END_ENTITY;
(*
```

Определение атрибута

operands — список, содержащий параметры оператора большой арности. Длина этого списка равна арности оператора.

6.3.12 Объект length_function

Объект **length_function** включает в себя семантику функции LENGTH по ИСО 10303-11.

Примечание — Функция LENGTH возвращает число символов в строке. Ее параметром является объект **string_expression**, содержащий значение строкового типа данных. Результатом выполнения функции является возвращенное значение, которое представляет собой число символов в строке; оно должно быть больше или равно нулю.

EXPRESS-спецификация

```
*)
ENTITY length_function
SUBTYPE OF (numeric_expression, unary_generic_expression);
      SELF\unary_generic_expression.operand: string_expression;
END_ENTITY;
(*
```

Определение атрибута

SELF\unary_generic_expression.operand — строка, длина которой вычисляется с помощью функции **length_function**.

6.3.13 Объект value_function

Объект **value_function** включает в себя семантику функции VALUE по ИСО 10303-11. Если значение, представленное объектом **string_expression**, который соответствует унаследованному атрибуту **operand**, является целым числом, должен быть использован подтип **int_value_function**.

Примечание — Функция VALUE возвращает числовое представление строки. Ее параметром является объект **string_expression**, содержащий значение строкового типа данных. Результатом выполнения функции является число, соответствующее представлению строки.

EXPRESS-спецификация

```
*)
ENTITY value_function
SUPERTYPE OF (int_value_function)
SUBTYPE OF (numeric_expression, unary_generic_expression);
      SELF\unary_generic_expression.operand: string_expression;
END_ENTITY;
(*
```

Определение атрибута

SELF\unary_generic_expression.operand — строковое выражение, значение которого вычисляется функцией **value_function**.

6.3.14 Объект int_value_function

Объект **int_value_function** включает в себя семантику функции VALUE по ИСО 10303-11 и констатирует, что объект **string_expression**, который соответствует унаследованному атрибуту **operand**, представляет собой целое число.

Примечание — Функция INTEGER VALUE возвращает целочисленное представление строки. Ее параметром является объект **string_expression**, содержащий значение строкового типа данных. Результатом выполнения функции является целое число, соответствующее представлению строки.

EXPRESS-спецификация

```
*)
ENTITY int_value_function
SUPERTYPE OF (value_function);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — объект **string_expression**, который соответствует атрибуту **operand**, должен быть выражен строкой, представляющей собой целое число.

6.3.15 Объект numeric_defined_function

Объект **numeric_defined_function** является функцией, областью значений которой является тип данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY numeric_defined_function
ABSTRACT SUPERTYPE OF (ONEOF (integer_defined_function,
                              real_defined_function))
SUBTYPE OF (numeric_expression, defined_function);
END_ENTITY;
(*
```

6.3.16 Объект plus_expression

Объект **plus_expression** включает в себя семантику оператора '+' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — Объект **plus_expression** является выражением, которое вычисляет сумму всех своих операндов. В **ISO13584_expressions_schema** все такие операнды должны быть выражены числами, относящимися к области числового типа данных.

EXPRESS-спецификация

```
*)
ENTITY plus_expression
SUBTYPE OF (multiple_arity_numeric_expression);
END_ENTITY;
(*
```

6.3.17 Объект minus_expression

Объект **minus_expression** включает в себя семантику оператора '-' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — **Minus_expression** является выражением, которое вычисляет разность своих операндов. В **ISO13584_expressions_schema** все такие операнды должны быть выражены числами, относящимися к области числового типа данных.

EXPRESS-спецификация

```
*)
ENTITY minus_expression
SUBTYPE OF (binary_numeric_expression);
END_ENTITY;
(*
```

6.3.18 Объект mult_expression

Объект **mult_expression** включает в себя семантику оператора '□' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — Объект **mult_expression** является выражением, которое вычисляет произведение всех своих операндов. В **ISO13584_expressions_schema** все такие операнды должны быть выражены числами, относящимися к области числового типа данных.

EXPRESS-спецификация

```
*)
ENTITY mult_expression
SUBTYPE OF (multiple_arity_numeric_expression);
END_ENTITY;
(*
```

6.3.19 Объект div_expression

Объект **div_expression** включает в себя семантику оператора языка EXPRESS 'DIV' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — Объект **div_expression** является выражением, которое вычисляет целое частное своих операндов. В **ISO13584_expressions_schema** все такие операнды должны быть выражены числами, относящимися к области числового типа данных. Результатом является целое число.

EXPRESS-спецификация

```
*)
ENTITY div_expression
SUBTYPE OF (binary_numeric_expression);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — оценкой **SELF\binary_generic_expression.operands[2]** не должен быть 0.

6.3.20 Объект mod_expression

Объект **mod_expression** включает в себя семантику оператора языка EXPRESS 'MOD' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — Объект **mod_expression** является выражением, которое вычисляет значение первого аргумента по модулю значения второго аргумента. Его операнды должны быть выражены числами, относящимися к области числового типа данных. Результатом является целое число.

EXPRESS-спецификация

```
*)
ENTITY mod_expression
SUBTYPE OF (binary_numeric_expression);
END_ENTITY;
(*
```

6.3.21 Объект slash_expression

Объект **slash_expression** включает в себя семантику оператора языка EXPRESS '/' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — Объект **slash_expression** является выражением, которое вычисляет вещественный результат деления первого аргумента на второй. Его операнды должны быть выражены числами, относящимися к области числового типа данных. Результатом является число.

EXPRESS-спецификация

```
*)
ENTITY slash_expression
SUBTYPE OF (binary_numeric_expression);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — оценкой **SELF\binary_generic_expression.operands[2]** не должен быть 0.

6.3.22 Объект power_expression

Объект **power_expression** включает в себя семантику оператора языка EXPRESS '/' по ИСО 10303-11, относящуюся к типу данных NUMBER.

Примечание — Объект **power_expression** является выражением, которое вычисляет возведение первого аргумента в степень, определяемую значением второго операнда. Его операнды должны быть выражены числами, относящимися к области числового типа данных. Результатом является число.

EXPRESS-спецификация

```
*)
ENTITY power_expression
SUBTYPE OF (binary_numeric_expression);
END_ENTITY;
(*
```

6.3.23 Объект unary_function_call

Объект **unary_function_call** является унарным оператором, относящимся к типу данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY unary_function_call
```

```
ABSTRACT SUPERTYPE OF (ONEOF (abs_function,
                              minus_function,
                              sin_function,
                              cos_function,
                              tan_function,
                              asin_function,
                              acos_function,
                              exp_function,
                              log_function,
                              log2_function,
                              log10_function,
                              square_root_function))
SUBTYPE OF (unary_numeric_expression);
END_ENTITY;
(*
```

6.3.24 Объект binary_function_call

Объект **binary_function_call** является бинарным оператором, относящимся к типу данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY binary_function_call
ABSTRACT SUPERTYPE OF (ONEOF (atan_function))
SUBTYPE OF (binary_numeric_expression);
END_ENTITY;
(*
```

6.3.25 Объект multiple_arity_function_call

Объект **multiple_arity_function_call** является оператором большой арности, относящимся к типу данных NUMBER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY multiple_arity_function_call
ABSTRACT SUPERTYPE OF (ONEOF (maximum_function,
                              minimum_function))
SUBTYPE OF (multiple_arity_numeric_expression);
END_ENTITY;
(*
```

6.3.26 Объект abs_function

Объект **abs_function** включает в себя семантику функции 'ABS' по ИСО 10303-11, относящуюся к типу данных NUMBER по ИСО 10303-11.

Примечание — Функция ABS возвращает абсолютное значение числа. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных. Результатом выполнения функции является абсолютное значение параметра. Возвращаемый тип данных идентичен типу данных параметра.

EXPRESS-спецификация

```
*)
ENTITY abs_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.27 Объект minus_function

Объект **minus_function** включает в себя семантику унарного оператора языка EXPRESS '-' по ИСО 10303-11, относящуюся к типу данных NUMBER по ИСО 10303-11.

Примечание — Функция MINUS возвращает инвертированное значение числа. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных. Результатом выполнения функции является инвертированное значение параметра. Возвращаемый тип данных идентичен типу данных параметра.

EXPRESS-спецификация

```
*)
ENTITY minus_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.28 Объект sin_function

Объект **sin_function** включает в себя семантику функции языка EXPRESS 'SIN' по ИСО 10303-11.

Примечание — Функция SIN возвращает значение синуса угла. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой угол, выраженный в радианах. Результатом выполнения функции является синус параметра ($-1,0 \le$ результат $\le 1,0$).

EXPRESS-спецификация

```
*)
ENTITY sin_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.29 Объект cos_function

Объект **cos_function** включает в себя семантику функции языка EXPRESS 'COS' по ИСО 10303-11.

Примечание — Функция COS возвращает значение косинуса угла. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой угол, выраженный в радианах. Результатом выполнения функции является косинус параметра ($-1,0 \le$ результат $\le 1,0$).

EXPRESS-спецификация

```
*)
ENTITY cos_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.30 Объект tan_function

Объект **tan_function** включает в себя семантику функции языка EXPRESS 'TAN' по ИСО 10303-11.

Примечание — Функция TAN возвращает значение тангенса угла. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой угол, выраженный в радианах. Результатом выполнения функции является тангенс параметра.

EXPRESS-спецификация

```
*)
ENTITY tan_function
SUBTYPE OF (unary_function_call);
```

```
END_ENTITY;
(*
```

6.3.31 Объект asin_function

Объект **asin_function** включает в себя семантику функции языка EXPRESS 'ASIN' по ИСО 10303-11.

Примечание — Функция ASIN возвращает угол, соответствующий заданному значению синуса. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой синус угла, выраженного в радианах. Результатом выполнения функции является угол, выраженный в радианах ($-\pi/2 \le$ результат $\le \pi/2$), синус которого задан параметром.

EXPRESS-спецификация

```
*)
ENTITY asin_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.32 Объект acos_function

Объект **acos_function** включает в себя семантику функции языка EXPRESS 'ACOS' по ИСО 10303-11.

Примечание — Функция ACOS возвращает угол, соответствующий заданному значению косинуса. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой косинус угла, выраженного в радианах. Результатом выполнения функции является угол, выраженный в радианах ($-\pi/2 \le$ результат $\le \pi/2$), косинус которого задан параметром.

EXPRESS-спецификация

```
*)
ENTITY acos_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.33 Объект exp_function

Объект **exp_function** включает в себя семантику функции языка EXPRESS 'EXP' по ИСО 10303-11.

Примечание — Функция EXP возвращает значение *e* (основание системы натуральных логарифмов), возведенное в степень, заданную значением параметра. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее значение параметра. Результатом выполнения функции является значение *e*, возведенное в степень, заданную значением параметра.

EXPRESS-спецификация

```
*)
ENTITY exp_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.34 Объект log_function

Объект **log_function** включает в себя семантику функции языка EXPRESS 'LOG' по ИСО 10303-11.

Примечание — Функция LOG возвращает натуральный логарифм числа. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой значение параметра, которое должно быть больше нуля. Результатом выполнения функции является вещественное число, которое является натуральным логарифмом значения параметра.

EXPRESS-спецификация

```
*)
ENTITY log_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.35 Объект log2_function

Объект **log2_function** включает в себя семантику функции языка EXPRESS 'LOG2' по ИСО 10303-11.

Примечание — Функция LOG2 возвращает логарифм числа по основанию 2. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой значение параметра, которое должно быть больше нуля. Результатом выполнения функции является вещественное число, которое является логарифмом по основанию 2 значения параметра.

EXPRESS-спецификация

```
*)
ENTITY log2_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.36 Объект log10_function

Объект **log10_function** включает в себя семантику функции языка EXPRESS 'LOG10' по ИСО 10303-11.

Примечание — Функция LOG10 возвращает десятичный логарифм числа. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой значение параметра, которое должно быть больше нуля. Результатом выполнения функции является вещественное число, которое является десятичным логарифмом значения параметра.

EXPRESS-спецификация

```
*)
ENTITY log10_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.37 Объект square_root_function

Объект **square_root_function** включает в себя семантику функции языка EXPRESS 'SQRT' по ИСО 10303-11.

Примечание — Функция SQRT (квадратный корень) возвращает неотрицательное значение квадратного корня числа. Ее параметром является объект **numeric_expression**, содержащий значение числового типа данных, представляющее собой значение параметра, которое должно быть больше нуля. Результатом выполнения функции является неотрицательное вещественное число, которое является квадратным корнем значения параметра.

EXPRESS-спецификация

```
*)
ENTITY square_root_function
SUBTYPE OF (unary_function_call);
END_ENTITY;
(*
```

6.3.38 Объект atan_function

Объект **atan_function** включает в себя семантику функции языка EXPRESS 'ATAN' по ИСО 10303-11.

Примечание — Функция ATAN возвращает угол, соответствующий заданному значению тангенса, представленного параметрами. Ее параметрами являются два объекта **numeric_expression**, представляющие собой значение **SELF\binary_generic_expression[1]/SELF/binary_generic_expression[2]**. Результатом выполнения функции является угол, выраженный в радианах ($-\pi/2 \leq$ результат $\leq \pi/2$), тангенс которого задан параметрами. Если значением второго выражения, соответствующего **SELF\binary_generic_expression[2]** является 0, то результат равен $-\pi/2$ или $\pi/2$ в зависимости от знака **SELF\binary_generic_expression[1]**.

EXPRESS-спецификация

```
*)
ENTITY atan_function
SUBTYPE OF (binary_function_call);
END_ENTITY;
(*
```

6.3.39 Объект maximum_function

Объект **maximum_function** возвращает максимальное значение из списка значений его операндов, которые должны выражаться числами, относящимися к области типа данных NUMBER.

EXPRESS-спецификация

```
*)
ENTITY maximum_function
SUBTYPE OF (multiple_arity_function_call);
END_ENTITY;
(*
```

6.3.40 Объект minimum_function

Объект **minimum_function** возвращает минимальное значение из списка значений его операндов, которые должны выражаться числами, относящимися к области типа данных NUMBER.

EXPRESS-спецификация

```
*)
ENTITY minimum_function
SUBTYPE OF (multiple_arity_function_call);
END_ENTITY;
(*
```

6.3.41 Объект integer_defined_function

Объект **integer_defined_function** представляет собой любую функцию, областью значений которой является тип данных INTEGER по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY integer_defined_function
ABSTRACT SUPERTYPE
SUBTYPE OF (numeric_defined_function);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — подтипом объекта **integer_defined_function** может быть либо объект **generic_expression**, либо такой подтип, который не содержит какой-либо объект **generic_variable**.

Примечание — Объект **integer_defined_function** является ресурсом, позволяющим конкретизировать эту схему путем включения любой определяемой приложением функции, которая возвращает значение, принадлежащее области данных целого типа.

6.3.42 Объект real_defined_function

Объект **real_defined_function** представляет собой любую функцию, областью значений которой является тип данных REAL по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY real_defined_function
ABSTRACT SUPERTYPE
SUBTYPE OF (numeric_defined_function);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — подтипом объекта **real_defined_function** может быть либо объект **generic_expression**, либо такой подтип, который не содержит какой-либо объект **generic_variable**.

Примечание — Объект **real_defined_function** является ресурсом, позволяющим конкретизировать эту схему путем включения любой определяемой приложением функции, которая возвращает значение, принадлежащее области данных вещественного типа.

6.4 Определения объектов булевых выражений

Объект **boolean_expression** представляет собой выражение, областью значений которого является тип данных BOOLEAN по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY boolean_expression
ABSTRACT SUPERTYPE OF (ONEOF (simple_boolean_expression,
                              unary_boolean_expression,
                              binary_boolean_expression,
                              multiple_arity_boolean_expression,
                              comparison_expression,
                              interval_expression,
                              boolean_defined_function))
SUBTYPE OF (expression);
END_ENTITY;
(*
```

6.4.1 Объект simple_boolean_expression

Объект **simple_boolean_expression** представляет собой унарный оператор, областью значений которого является тип данных BOOLEAN по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY simple_boolean_expression
ABSTRACT SUPERTYPE OF (ONEOF (boolean_literal,
                              boolean_variable))
SUBTYPE OF (boolean_expression, simple_generic_expression);
END_ENTITY;
(*
```

6.4.2 Объект boolean_literal

Объект **boolean_literal** представляет собой литерал типа BOOLEAN языка EXPRESS.

Примечание — Областью значений типа данных Boolean языка EXPRESS является набор, содержащий два литерала TRUE (истина) и FALSE (ложь).

EXPRESS-спецификация

```
*)
ENTITY boolean_literal
SUBTYPE OF (simple_boolean_expression, generic_literal);
      the value: BOOLEAN;
END_ENTITY;
(*
```

Определение атрибута

the value — значение литерала BOOLEAN.

6.4.3 Объект boolean_variable

Объект **boolean_variable** представляет собой переменную, которая обозначает булево значение.

EXPRESS-спецификация

```
*)
ENTITY boolean_variable
SUBTYPE OF (simple_boolean_expression, variable);
END_ENTITY;
(*
```

6.4.4 Объект unary_boolean_expression

Объект **unary_boolean_expression** представляет собой унарный оператор, областью значений которого является тип данных BOOLEAN по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY unary_boolean_expression
ABSTRACT SUPERTYPE OF (ONEOF (not_expression, odd_function))
SUBTYPE OF (boolean_expression, unary_generic_expression);
END_ENTITY;
(*
```

6.4.5 Объект not_expression

Объект **not_expression** включает в себя семантику отрицания 'NOT', применяемого к оператору булевого типа данных по ИСО 10303-11.

Примечание — Для оператора NOT требуется один булевый операнд и его результатом является булево значение.

EXPRESS-спецификация

```
*)
ENTITY not_expression
SUBTYPE OF (unary_boolean_expression);
SELF\unary_generic_expression.operand: boolean_expression;
END_ENTITY;
(*
```

Определение атрибута

SELF\unary_generic_expression.operand — выражение **boolean_expression**, которое представляет собой операнд оператора 'NOT'.

6.4.6 Объект odd_function

Объект **odd_function** включает в себя семантику оператора языка EXPRESS 'ODD' по ИСО 10303-11.

Примечание — Функция ODD возвращает значение TRUE или FALSE в зависимости от того, является число нечетным или четным. Ее параметром является числовое выражение, значение которого принадлежит к целому типу данных. В качестве результата функция возвращает значение TRUE, если значение параметра по модулю 2 равно 1, и FALSE — в противном случае.

EXPRESS-спецификация

```
*)
ENTITY odd_function
SUBTYPE OF (unary_boolean_expression);
      SELF\unary_generic_expression.operand: numeric_expression;
WHERE
      WR1: is_int_expr(SELF\numeric_expression);
END_ENTITY;
(*
```

Определение атрибута

SELF\unary_generic_expression.operand — выражение **numeric_expression**, которое представляет собой операнд функции 'ODD'.

Формальное утверждение

WR1 — операнд представляет собой выражение **numeric_expression**, областью значений которого является тип данных INTEGER языка EXPRESS.

6.4.7 Объект binary_boolean_expression

Объект **binary_boolean_expression** представляет собой бинарный оператор, областью значений которого является тип данных BOOLEAN языка EXPRESS по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY binary_boolean_expression
ABSTRACT SUPERTYPE OF (ONEOF (xor_expression, equals_expression))
SUBTYPE OF (boolean_expression, binary_generic_expression);
END_ENTITY;
(*
```

6.4.8 Объект multiple_arity_boolean_expression

Объект **multiple_arity_boolean_expression** представляет собой оператор большой арности, областью значений которого является тип данных BOOLEAN языка EXPRESS по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY multiple_arity_boolean_expression
ABSTRACT SUPERTYPE OF (ONEOF (and_expression, or_expression))
SUBTYPE OF (boolean_expression, multiple_arity_generic_expression);
      SELF\multiple_arity_generic_expression.operands :
                                    LIST [2:?] OF boolean_expression;
END_ENTITY;
(*
```

Определение атрибута

operands — экземпляры объекта **boolean_expression**, которые представляют собой параметры оператора большой арности.

6.4.9 Объект xor_expression

Объект **xor_expression** включает в себя семантику оператора 'XOR' по ИСО 10303-11.

Примечание — Для оператора XOR требуются два булевых операнда и он вычисляет булево значение, которое является исключающей дизъюнкцией двух операндов.

EXPRESS-спецификация

```
*)
ENTITY xor_expression
SUBTYPE OF (binary_boolean_expression);
        SELF\binary_generic_expression.operands:
                                LIST [2:2] OF boolean_expression;
END_ENTITY;
(*
```

Определение атрибута

operands — два операнда **boolean_expression** оператора XOR, которые должны представлять собой булевы выражения.

6.4.10 Объект equals_expression

Объект **equals_expression** включает в себя семантику оператора равенства экземпляров ':=: ' по ИСО 10303-11, областью значений которого являются объекты типа данных **generic_expression**.

Примечание — Оператор равенства экземпляров сравнивает значения двух экземпляров типа **generic_expression**, и результатом сравнения является булево значение. В контексте схемы **ISO_13584_expressions_schema** два значения экземпляров типа **generic_expression** являются сравнимыми, если их типы данных совпадают либо если один тип данных является подтипом второго.

EXPRESS-спецификация

```
*)
ENTITY equals_expression
SUBTYPE OF (binary_boolean_expression);
END_ENTITY;
(*
```

6.4.11 Объект and_expression

Объект **and_expression** включает в себя семантику оператора 'AND' по ИСО 10303-11 и использует свойство ассоциативности для работы с большой арностью.

Примечание — Для оператора AND требуются, по крайней мере, два булевых выражения и он вычисляет булево значение, которое является конъюнкцией двух операндов.

EXPRESS-спецификация

```
*)
ENTITY and_expression
SUBTYPE OF (multiple_arity_boolean_expression);
END_ENTITY;
(*
```

6.4.12 Объект or_expression

Объект **or_expression** включает в себя семантику оператора 'OR' по ИСО 10303-11 и использует свойство ассоциативности для работы с большой арностью.

Примечание — Для оператора OR требуются, по крайней мере, два булевых выражения, и он вычисляет булево значение, которое является включающей дизъюнкцией двух операндов.

EXPRESS-спецификация

```
*)
ENTITY or_expression
```

```
SUBTYPE OF (multiple_arity_boolean_expression);
END_ENTITY;
(*
```

6.4.13 Объект comparison_expression

Объект **comparison_expression** включает в себя семантику операторов сравнения различных значений по ИСО 10303-11, которые ограничены типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

EXPRESS-спецификация

```
*)
ENTITY comparison_expression
ABSTRACT SUPERTYPE OF (ONEOF (comparison_equal,
                              comparison_greater,
                              comparison_greater_equal,
                              comparison_less,
                              comparison_less_equal,
                              comparison_not_equal,
                              like_expression))
SUBTYPE OF (boolean_expression, binary_generic_expression);
    SELF\binary_generic_expression.operands: LIST [2:2] OF
expression;
WHERE
    WR1:  (('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
            IN TYPEOF (SELF\binary_generic_expression.operands[1]))
        AND
           'ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
           IN TYPEOF (SELF\binary_generic_expression.operands[2]))
    OR
          (('ISO13584_EXPRESSIONS_SCHEMA.BOOLEAN_EXPRESSION'
            IN TYPEOF (SELF\binary_generic_expression.operands[1]))
        AND
           ('ISO13584_EXPRESSIONS_SCHEMA.BOOLEAN_EXPRESSION'
           IN TYPEOF (SELF\binary_generic_expression.operands[2])))
    OR
          (('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
              IN TYPEOF (SELF\binary_generic_expression.operands[1]))
    AND
           ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
           IN TYPEOF (SELF\binary_generic_expression.operands[2])));
END_ENTITY;
(*
```

Определение атрибута

operands — список объектов **generic_expressions**, содержащий выражения, подлежащие сравнению.

Формальное утверждение

WR1 — операнды **comparison_expression** должны быть совместимыми по типу.

6.4.14 Объект comparison_equal

Объект **comparison_equal** включает в себя семантику оператора равенства значений ('=') по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

EXPRESS-спецификация

```
*)
```

```
ENTITY comparison_equal
SUBTYPE OF (comparison_expression);
END_ENTITY;
(*
```

Примечание — Объект **comparison_equal** имеет значение TRUE, если оба операнда имеют одно и то же значение.

6.4.15 Comparison_greater

Объект **comparison_greater** включает в себя семантику оператора '>' (больше чем) по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

Примечание — Объект **comparison_greater** имеет значение TRUE, если значение первого операнда больше, чем значение второго операнда. Соответствующими порядками старшинства являются:

- математическое упорядочение вещественных чисел;
- FALSE < TRUE;
- лексикографический порядок в строках.

EXPRESS-спецификация

```
*)
ENTITY comparison_greater
SUBTYPE OF (comparison_expression);
END_ENTITY;
(*
```

6.4.16 Объект comparison_greater_equal

Объект **comparison_greater_equal** включает в себя семантику оператора '>=' (больше или равно) по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

Примечание — Объект **comparison_greater_equal** имеет значение TRUE, если значение первого операнда больше или равно значению второго операнда. Соответствующими порядками старшинства являются:

- математическое упорядочение вещественных чисел;
- FALSE < TRUE;
- лексикографический порядок в строках.

EXPRESS-спецификация

```
*)
ENTITY comparison_greater_equal
SUBTYPE OF (comparison_expression);
END_ENTITY;
(*
```

6.4.17 Объект comparison_less

Объект **comparison_less** включает в себя семантику оператора '<' (меньше чем) по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

Примечание — Объект **comparison_less** имеет значение TRUE, если первый операнд имеет меньшее значение, чем второй операнд. Соответствующими порядками старшинства являются:

- математическое упорядочение вещественных чисел;
- FALSE < TRUE;
- лексикографический порядок в строках.

EXPRESS-спецификация

```
*)
ENTITY comparison_less
```

```
SUBTYPE OF (comparison_expression);
END_ENTITY;
(*
```

6.4.18 Объект comparison_less_equal

Объект **comparison_less_equal** включает в себя семантику оператора языка EXPRESS '<=' (меньше или равно) по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

Примечание — Объект **comparison_less_equal** имеет значение TRUE, если значение первого операнда меньше или равно значению второго операнда. Соответствующими порядками старшинства являются:

- математическое упорядочение вещественных чисел;
- FALSE < TRUE;
- лексикографический порядок в строках.

EXPRESS-спецификация

```
*)
ENTITY comparison_less_equal
SUBTYPE OF (comparison_expression);
END_ENTITY;
(*
```

6.4.19 Объект comparison_not_equal

Объект **comparison_not_equal** включает в себя семантику оператора '< >' (значения не равны) по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

Примечание — Объект **comparison_not_equal** имеет значение TRUE, если первый операнд имеет значение, отличающееся от значения второго операнда. Соответствующими порядками старшинства являются:

- математическое упорядочение вещественных чисел;
- FALSE < TRUE;
- лексикографический порядок в строках.

EXPRESS-спецификация

```
*)
ENTITY comparison_not_equal
SUBTYPE OF (comparison_expression);
END_ENTITY;
(*
```

6.4.20 Объект like_expression

Объект **like_expression** включает в себя семантику оператора соответствия строк LIKE по ИСО 10303-11. Первый операнд является целевой строкой; второй операнд — образцовой строкой. Специальные символы (часто называемые безразличными символами (wild cards)) в образцовой строке определены в ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY like_expression
SUBTYPE OF (comparison_expression);
WHERE
    WR1: ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
        IN TYPEOF (SELF\comparison_expression.operands[1])) AND
        ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
         IN TYPEOF (SELF\comparison_expression.operands[2]));
```

```
END_ENTITY;
(*
```

Формальное утверждение

WR1 — список операндов должен содержать два выражения **generic_expression**, которые являются выражениями типа **string_expression**.

6.4.21 Объект interval_expression

Объект **interval_expression** включает в себя семантику выражения интервала по ИСО 10303-11, ограниченного типами данных NUMBER, BOOLEAN и STRING языка EXPRESS. Оба операнда должны иметь один и тот же тип данных.

Примечание — Выражение **interval_expression** проверяет, находится или нет значение в пределах заданного интервала. Оно принимает значение TRUE, если **interval_low** <= **interval_item** <= **interval_high**. Соответствующими порядками старшинства являются:

- математическое упорядочение вещественных чисел;
- лексикографический порядок в строках.

EXPRESS-спецификация

```
*)
ENTITY interval_expression
SUBTYPE OF (boolean_expression, multiple_arity_generic_expression);
DERIVE
    interval_low: generic_expression
                  : = SELF\multiple_arity_generic_expression.operands[1];
    interval_item:generic_expression
                  : = SELF\multiple_arity_generic_expression.operands[2];
    interval_high:generic_expression
                  : = SELF\multiple_arity_generic_expression.operands[3];
WHERE
    WR1:('ISO13584_EXPRESSIONS_SCHEMA.EXPRESSION'
                                        IN TYPEOF (interval_low))
        AND ('ISO13584_EXPRESSIONS_SCHEMA.EXPRESSION'
                                        IN TYPEOF (interval_item))
        AND ('ISO13584_EXPRESSIONS_SCHEMA.EXPRESSION'
                                        IN TYPEOF (interval_high));
    WR2: (('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
                                        IN TYPEOF (SELF.interval_low))
            AND ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
                                        IN TYPEOF (SELF.interval_high))
            AND ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
                                        IN TYPEOF (SELF.interval_item)))
          OR
          (('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
                                        IN TYPEOF (SELF.interval_low))
            AND ('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
                                        IN TYPEOF (SELF.interval_item))
            AND ('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
                                        IN TYPEOF (SELF.interval_high)));
END_ENTITY;
(*
```

Определения атрибутов

interval_low — операнд, задающий нижнюю границу (bound_1) интервального выражения (пункт 12.2.4 ИСО 10303-11);

interval_high — операнд, задающий верхнюю границу (bound_2) интервального выражения (пункт 12.2.4 ИСО 10303-11);

interval_item — операнд interval_item интервального выражения (пункт 12.2.4 ИСО 10303-11).

Формальные утверждения

WR1 — операнды должны иметь тип данных **expression**.

WR2 — типы сравниваемых в **interval_expression** выражений должны быть выражены в сравнимых выражениях.

6.4.22 Объект boolean_defined_function

Объект **boolean_defined_function** представляет собой любой оператор, определенный в прикладной программе, областью значений которого является тип данных BOOLEAN по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY boolean_defined_function
ABSTRACT SUPERTYPE
SUBTYPE OF (defined_function, boolean_expression);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — подтипом объекта **boolean_defined_function** может быть либо объект **generic_expression**, либо такой объект, который не содержит ни одного объекта **generic_variable**.

6.5 Определения объектов строковых выражений

Объект **string_expression** представляет собой выражение, областью значений которого является тип данных STRING языка EXPRESS по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY string_expression
ABSTRACT SUPERTYPE OF (ONEOF (simple_string_expression,
                              index_expression,
                              substring_expression,
                              concat_expression,
                              format_function,
                              string_defined_function))
SUBTYPE OF (expression);
END_ENTITY;
(*
```

6.5.1 Объект simple_string_expression

Объект **simple_string_expression** представляет собой унарный оператор, областью значений которого является тип данных STRING языка EXPRESS по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY simple_string_expression
ABSTRACT SUPERTYPE OF (ONEOF (string_literal,
                              string_variable))
SUBTYPE OF (string_expression, simple_generic_expression);
END_ENTITY;
(*
```

6.5.2 Объект string_literal

Объект **string_literal** представляет собой литерал типа STRING языка EXPRESS.

Примечание — Областью значений типа данных STRING языка EXPRESS являются последовательности символов.

EXPRESS-спецификация

```
*)
ENTITY string_literal
SUBTYPE OF (simple_string_expression, generic_literal);
        the_value: STRING;
END_ENTITY;
(*
```

Определение атрибута

the_value — значение литерала STRING.

6.5.3 Объект string_variable

Объект **string_variable** представляет собой переменную, которая обозначает значение типа STRING.

EXPRESS-спецификация

```
*)
ENTITY string_variable
SUBTYPE OF (simple_string_expression, variable);
END_ENTITY;
(*
```

6.5.4 Объект index_expression

Объект **index_expression** включает в себя семантику оператора индексирования строки ([]) по ИСО 10303-11.

Примечание — У строкового оператора **index_expression** есть два операнда: строковое значение (представленное атрибутом **operand**) индексируется значением индекса (представленного атрибутом **index**). Результирующее строковое значение представляет собой символ, расположенный на позиции, определяемой значением атрибута **index**.

EXPRESS-спецификация

```
*)
ENTITY index_expression
SUBTYPE OF (string_expression, binary_generic_expression);
DERIVE
        operand: generic_expression:=
SELF\binary_generic_expression.operands[1];
        index:generic_expression:=
SELF\binary_generic_expression.operands[2];
WHERE
        WR1: ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
                IN TYPEOF(operand))
             AND ('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
                IN TYPEOF(index));
        WR2: is_int_expr (index);
END_ENTITY;
(*
```

Определения атрибутов

operand — объект **string_expression,** который представляет литерал STRING;

index — целочисленное выражение **numeric_expression,** которое показывает значение индекса.

Формальные утверждения

WR1 — первый операнд должен быть объектом **string_expression**, а второй операнд — объектом **numeric_expression**.

WR2 — позиция, определяемая операндом **index**, должна иметь целое значение.

Неформальное утверждение

IP1 — операнд **index** должен иметь значение типа INTEGER, которые больше нуля и меньше или равно длине строки, заданной операндом STRING.

6.5.5 Объект substring_expression

Объект **substring_expression** включает в себя семантику оператора индексирования подстроки ([:]) по ИСО 10303-11.

Примечание — У строкового оператора **substring_expression** есть три операнда: строковое значение (представленное атрибутом **operand**) индексируется значением индекса (представленного атрибутами **index1** и **index2**). Оператор **substring_expression** вычисляет строковое значение длиной (**index2 — index1 + 1**). В результате строковое значение эквивалентно последовательности символов, расположенных с позиции **index1** по позицию **index2**.

EXPRESS-спецификация

```
*)
ENTITY substring_expression
SUBTYPE OF (string_expression, multiple_arity_generic_expression);
DERIVE
    operand:generic_expression:=
        SELF\multiple_arity_generic_expression.operands[1];
    index1:generic_expression:=
        SELF\multiple_arity_generic_expression.operands[2];
    index2:generic_expression:=
        SELF\multiple_arity_generic_expression.operands[3];
WHERE
    WR1: ('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION'
                                    IN TYPEOF(operand))
    AND ('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
                                    IN TYPEOF(index1))
    AND ('ISOI3584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION'
                                    IN TYPEOF(index2));
    WR2: SIZEOF (SELF\multiple_arity_generic_expression.operands)=3;
    WR3: is_int_expr (index1);
    WR4: is_int_expr (index2);
END_ENTITY;
(*
```

Определения атрибутов

operand — объект **string_expression,** из которого выделяется подстрока;

index1 — объект **numeric_expression,** который показывает нижнюю границу подстроки в **operand**;

index2 — объект **numeric_expression,** который показывает верхнюю границу подстроки в **operand**.

Формальные утверждения

WR1 — операнд должен быть объектом **string_expression**; **index1** и **index2** должны быть объектами **numeric_expression**.

WR2 — список операндов объекта **substring_expression** должен содержать три элемента.

WR3 — операнд **index1** должен иметь значение типа INTEGER.

WR3 — операнд **index2** должен иметь значение типа INTEGER.

Неформальные утверждения

IP1 — операнды **index1** и **index2** должны иметь значения типа INTEGER, которые больше нуля и меньше или равны длине строки, заданной операндом STRING.

IP2 — значение **index1** должно быть меньше или равно значению **index2**.

6.5.6 Объект concat_expression

Объект **concat_expression** включает в себя семантику оператора конкатенации (+) по ИСО 10303-11, относящуюся к типу данных STRING.

Примечание — У оператора **concat_expression** есть, по крайней мере, два операнда типа **string_expression**. Он вычисляет строковое значение, являющееся результатом конкатенации всех элементов списка **SELF\multiple_arity_generic_expression.operands**. Конкатенация этих элементов осуществляется в порядке их расположения в списке.

EXPRESS-спецификация

```
*)
ENTITY concat_expression
SUBTYPE OF (string_expression, multiple_arity_generic_expression);
      SELF\multiple_arity_generic_expression.operands:
                                        LIST [2 : ?] OF string_expression;
END_ENTITY;
(*
```

Определение атрибута

SELF\multiple_arity_generic_expression.operands — выражения типа **string_expression,** которые подлежат конкатенации.

6.5.7 Объект format_function

Объект **format_function** включает в себя семантику функции FORMAT по ИСО 10303-11.

Примечание — Функция **format_function** возвращает отформатированное строковое представление числа. Ее параметрами являются: **numeric_expression**, соответствующее операнду **value_to_format**, которое выражается в виде вещественного или целого числа, и **string_expression**, соответствующее операнду **format_string**, которое выражается в виде строки, содержащей команды форматирования. Результатом функции является строковое представление значения операнда **value_to_format**, соответствующее значению операнда **format_string**. Строка форматирования содержит специальные символы, которые указывают, какой формат должен быть у результата. Различные способы описания строки форматирования установлены в ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY format_function
SUBTYPE OF (string_expression, binary_generic_expression);
DERIVE
      value_to_format: generic_expression:=
            SELF\binary_generic_expression.operands[1];
      format_string:generic_ expression:=
            SELF\binary_generic_expression.operands[2];
WHERE
      WR1: (('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_EXPRESSION')
                                          IN TYPEOF(value_to_format))
      AND (('ISO13584_EXPRESSIONS_SCHEMA.STRING_EXPRESSION')
                                          IN TYPEOF(format_string));
END_ENTITY;
(*
```

Определения атрибутов

value_to_format — объект **numeric_expression,** который должен быть отформатирован;

format_string — команды форматирования, которые определяют вид результата.

Формальное утверждение

WR1 — операнд **value_to_format** должен принадлежать к типу **numeric_expression**, а операнд **format_string** — к типу **string_expression**.

6.5.8 Объект string_defined_function

Объект **string_defined_function** представляет собой любой оператор, определенный в прикладной программе, областью значений которого является тип данных STRING по ИСО 10303-11.

EXPRESS-спецификация

```
*)
ENTITY string_defined_function
```

```
ABSTRACT SUPERTYPE
SUBTYPE OF (defined_function, string_expression);
END_ENTITY;
(*
```

Неформальное утверждение

IP1 — подтип объекта **string_defined_function** может быть либо объектом **generic_expression**, либо он не должен содержать ни одного объекта **generic_variable**.

6.6 Функции для определения свойств выражения

Следующие декларации на языке EXPRESS устанавливают различные функции, которые позволяют определять свойства выражений.

6.6.1 Функция is_int_expr

Функция **is_int_expr** проверяет, является ли результатом синтаксически правильного выражения целочисленное значение или нет.

EXPRESS-спецификация

```
*)
FUNCTION is_int_expr (arg: numeric_expression) : BOOLEAN;

LOCAL
      i: INTEGER;
END_LOCAL;

IF 'ISO13584_EXPRESSIONS_SCHEMA.INT_LITERAL' IN TYPEOF(arg)
THEN
   RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.REAL_LITERAL' IN TYPEOF(arg)
THEN
   RETURN (FALSE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.INT_NUMERIC_VARIABLE' IN TYPEOF(arg)
THEN
   RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.REAL_NUMERIC_VARIABLE' IN TYPEOF(arg)
THEN
   RETURN (FALSE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.ABS_FUNCTION' IN TYPEOF(arg)
THEN
   RETURN (is_int_expr(arg\unary_numeric_expression.operand));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.MINUS_FUNCTION' IN TYPEOF(arg)
THEN
   RETURN (is_int_expr(arg\unary_numeric_expression.operand));
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.SIN_FUNCTION' IN TYPEOF (arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.COS_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.TAN_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.ASIN_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.ACOS_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.ATAN_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.EXP_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.LOG_FUNCTION' IN TYPEOF(arg))
```

```
      OR ('ISO13584_EXPRESSIONS_SCHEMA.LOG2_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.LOG10_FUNCTION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.SQUARE_ROOT_FUNCTION'
                                                        IN TYPEOF (arg))
THEN
      RETURN (FALSE);
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.PLUS_EXPRESSION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.MULT_EXPRESSION'
                                                        IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.MAXIMUM_FUNCTION'
                                                        IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.MINIMUM_FUNCTION'
                                                        IN TYPEOF(arg))
THEN
      REPEAT i :=1 TO SIZEOF (
            arg\multiple_arity_numeric_expression.operands);
      IF NOT
        is_int_expr(arg\multiple_arity_numeric_expression.operands[i])
      THEN
        RETURN (FALSE);
      END_IF;
      END_REPEAT;
       RETURN (TRUE);
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.MINUS_EXPRESSION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.POWER_EXPRESSION'
                                                        IN TYPEOF(arg))
THEN
     RETURN (is_int_expr(arg\binary_numeric_expression.operands[1])
        AND is_int_expr(arg\binary_numeric_expression.operands[2]));
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.DIV_EXPRESSION' IN TYPEOF(arg))
    OR ('ISO13584_EXPRESSIONS_SCHEMA.MOD_EXPRESSION' IN TYPEOF(arg))
THEN
      RETURN (TRUE); (*всегда выдает результат типа INTEGER*)
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.SLASH_EXPRESSION' IN TYPEOF(arg)
THEN
      RETURN (FALSE);          (*всегда выдает результат типа REAL*)
END_IF;
 IF 'ISO13584_EXPRESSIONS_SCHEMA.LENGTH_FUNCTION' IN TYPEOF(arg)
THEN
      RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.VALUE_FUNCTION' IN TYPEOF(arg)
THEN
      IF 'ISO13584_EXPRESSIONS_SCHEMA.INT_VALUE_FUNCTION'
                                                        IN TYPEOF(arg)
      THEN
        RETURN (TRUE);
      ELSE
        RETURN (FALSE);
      END_IF;
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.INTEGER_DEFINED_FUNCTION'
                                                        IN TYPEOF(arg)
THEN
```

```
        RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.REAL_DEFINED_FUNCTION' IN TYPEOF(arg)
THEN
        RETURN (FALSE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.BOOLEAN_DEFINED_FUNCTION'
                                                                IN TYPEOF(arg)
THEN
   RETURN (FALSE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.STRING_DEFINED_FUNCTION'
                                                                IN TYPEOF(arg)
THEN
        RETURN (FALSE);
END_IF;
(* Если встречается другое generic_expression, которое не является подтипом integer_defined_function,
то его результат не является целым числом *)
RETURN (FALSE);

END_FUNCTION; -- is_int_expr
(*
```

6.6.2 Функция is_SQL_mappable

Функция **is_SQL_mappable** проверяет, содержит ли ациклический граф, представляющий выражение, только элементы, отображаемые на языке SQL.

EXPRESS-спецификация

```
*)
FUNCTION is_SQL_mappable (arg: expression) : BOOLEAN;

LOCAL
        i: INTEGER;
END_LOCAL;

IF 'ISO13584_EXPRESSIONS_SCHEMA.SIMPLE_NUMERIC_EXPRESSION'
                                                                IN TYPEOF(arg)
THEN
    RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.SQL_MAPPABLE_DEFINED_FUNCTION'
                                                                IN TYPEOF(arg)
THEN
     RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.MINUS_FUNCTION' IN TYPEOF(arg)
THEN
      RETURN (is_SQL_mappable(arg\unary_numeric_expression.operand));
END_IF;
IF  ('ISO13584_EXPRESSIONS_SCHEMA .ABS_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.SIN_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.COS_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.TAN_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.ASIN_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.ACOS_FUNCTION' IN TYPEOF(arg))
        OR ('1SO13584_EXPRESSIONS_SCHEMA.ATAN_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.EXP_FUNCTION' IN TYPEOF(arg))
```

```
        OR ('ISO13584_EXPRESSIONS_SCHEMA.LOG_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.LOG2_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.LOG10_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.SQUARE_ROOT_FUNCTION'
                                                        IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.VALUE_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.LENGTH_FUNCTION'
                                                        IN TYPEOF(arg))
THEN
   RETURN (FALSE);
END_IF;
IF  ('ISO13584_EXPRESSIONS_SCHEMA.PLUS_EXPRESSION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.MULT_EXPRESSION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.MAXIMUM_FUNCTION'
                                                        IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.MINIMUM_FUNCTION'
                                                        IN TYPEOF(arg))
THEN
        REPEAT i :=1 TO SIZEOF (
                arg\multiple_arity_numeric_expression.operands);
             IF NOT is_SQL_mappable(
                arg\multiple_arity_numeric_expression.operands[i])
        THEN
                 RETURN (FALSE);
        END_IF;
 END_REPEAT;
RETURN (TRUE);
END_IF;
IF  ('ISO13584_EXPRESSIONS_SCHEMA.MINUS_EXPRESSION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.SLASH_EXPRESSION' IN
                                                        TYPEOF(arg))
THEN
        RETURN (is_SQL_mappable(
              arg\binary_numeric_expression.operands[1])
        AND is_SQL_mappable(arg\binary_numeric_expression.operands[2]));
END_IF;
IF  ('ISO13584_EXPRESSIONS_SCHEMA.DIV_EXPRESSION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.MOD_EXPRESSION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.POWER_EXPRESSION'
                                                        IN TYPEOF(arg))
THEN
        RETURN (FALSE);          (* операторы, не поддерживаемые в языке SQL *)
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.SIMPLE_BOOLEAN_EXPRESSION'
                                                        IN TYPEOF(arg)
THEN
        RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.NOT_EXPRESSION' IN TYPEOF(arg)
THEN
        RETURN (is_SQL_mappable (arg\UNARY_GENERIC_EXPRESSION.OPERAND));
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.ODD_FUNCTION' IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.XOR_EXPRESSION'
                                                        IN TYPEOF(arg))
THEN
        RETURN (FALSE);
```

```
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.AND_EXPRESSION' IN TYPEOF(arg))
            OR ('ISO13584_EXPRESSIONS_SCHEMA.OR_EXPRESSION' IN TYPEOF(arg))
THEN
      REPEAT i:=1 TO SIZEOF (
                arg\MULTIPLE_ARITY_BOOLEAN_EXPRESSION.OPERANDS);
          IF NOT is_SQL_mappable (
                arg\MULTIPLE_ARITY_BOOLEAN_EXPRESSION.OPERANDS[i])
 THEN
                RETURN (FALSE);
          END_IF;
     END_REPEAT;
     RETURN (TRUE);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.EQUALS_EXPRESSION' IN TYPEOF(arg)
THEN
   RETURN (is_SQL_mappable (
            arg\BINARY_GENERIC_EXPRESSION.OPERANDS[1])
      AND is_SQL_mappable(
            arg\BINARY_GENERIC_EXPRESSION.OPERANDS[2]));
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_EQUAL' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_GREATER'
                                                      IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_GREATER_EQUAL'
                                                      IN TYPEOF(arg))
       OR ('ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_LESS'
                                                      IN TYPEOF(arg))
       OR ('ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_LESS_EQUAL'
                                                       IN TYPEOF(arg))
       OR ('ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_NOT_EQUAL'
                                                        IN TYPEOF(arg))
        OR ('ISO13584_EXPRESSIONS_SCHEMA.LIKE_EXPRESSION'
                                                       IN TYPEOF(arg))
THEN
     RETURN (is_SQL_mappable (arg\COMPARISON_EXPRESSION.OPERANDS[1])
        AND is_SQL_mappable (arg\COMPARISON_EXPRESSION.OPERANDS[2]));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.INTERVAL_EXPRESSION' IN TYPEOF(arg)
THEN
      RETURN (is_SQL_mappable(arg\interval_expression.interval_low)
        AND is_SQL_mappable(arg\interval_expression.interval_high)
        AND is_SQL_mappable(arg\interval_expression.interval_item));
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.NUMERIC_DEFINED_FUNCTION'
                                                      IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.BOOLEAN_DEFINED_FUNCTION'
                                                      IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.STRING_DEFINED_FUNCTION'
                                                      IN TYPEOF(arg))
THEN
      RETURN (FALSE);
END_IF;
(* Предполагается, что все определенные функции не являются отображаемыми на языке SQL *)
IF 'ISO13584_EXPRESSIONS_SCHEMA.SIMPLE_STRING_EXPRESSION'
                                                       IN TYPEOF(arg)
THEN
      RETURN (TRUE);
```

```
END_IF;
IF ('ISO13584_EXPRESSIONS_SCHEMA.INDEX_EXPRESSION' IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.SUBSTRING_EXPRESSION'
                                                        IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.CONCAT_EXPRESSION'
                                                        IN TYPEOF(arg))
      OR ('ISO13584_EXPRESSIONS_SCHEMA.FORMAT_FUNCTION'
                                                        IN TYPEOF(arg))
THEN
      RETURN (FALSE);
END_IF;
(* Если встречается другое generic_expression, которое не является подтипом SQL_mappable_defined_
function, то оно не является отображаемым на языке SQL *)
      RETURN (FALSE);
END_FUNCTION; -- is_SQL_mappable
(*
```

6.6.3 Функция used_functions

Функция **used_functions** просматривает весь граф выражения, собирая все функции **application_defined**, и в результате возвращает их. Данная функция просматривает ориентированный ациклический граф, представляющий выражение.

Примечание — Эта функция является ресурсом для схем, использующих **ISO13584_expressions_schema** или ее возможные конкретизации. Например, она используется в некоторых частях комплекса стандартов ИСО 13584 для записи ограничений на функции, встречающиеся в выражениях.

EXPRESS-спецификация

```
*)
FUNCTION used_functions (arg : expression) : SET OF defined_function;

LOCAL
      result : SET OF defined_function := [];
END_LOCAL;

IF ('ISO13584_EXPRESSIONS_SCHEMA.DEFINED_FUNCTION' IN TYPEOF(arg))
THEN
      RETURN ( [arg] );
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.UNARY_NUMERIC_EXPRESSION' IN
   TYPEOF (arg)
THEN
       RETURN (used_functions (arg\unary_numeric_expression.operand));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.BINARY_NUMERIC_EXPRESSION' IN
       TYPEOF (arg)
THEN
      RETURN (used_functions (arg\binary_numeric_expression.operands[1])
         + used_functions (arg\binary_numeric_expression.operands[2])));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.MULTIPLE_ARITY_NUMERIC_EXPRESSION' IN
       TYPEOF (arg)
THEN
       REPEAT i := 1 TO SIZEOF (
                arg\multiple_arity_numeric_expression.operands);
          result := result + used_functions (
                arg\multiple_arity_numeric_expression.operands[i]);
       END_REPEAT;
       RETURN (result);
END_IF;
```

```
IF 'ISO13584_EXPRESSIONS_SCHEMA.UNARY_GENERIC_EXPRESSION'
                                                        IN TYPEOF (arg)
THEN
   RETURN (used_functions (arg\unary_generic_expression.operand));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.BINARY_BOOLEAN_EXPRESSION'
                                                        IN TYPEOF (arg)
THEN
   RETURN (used_functions (arg\binary_generic_expression.operands[1])
      + used_functions (
            arg\binary_generic_expression.operands[2]));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.MULTIPLE_ARITY_BOOLEAN_EXPRESSION'
                                                        IN TYPEOF (arg)
THEN
      REPEAT i := 1 TO
          SIZEOF (arg\multiple_arity_Boolean_expression.operands);
          result := result + used_functions (
               arg\multiple_arity_Boolean_expression.operands[i]);
      END_REPEAT;
      RETURN (result);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.COMPARISON_EXPRESSION'
                                                        IN TYPEOF (arg)
THEN
      RETURN (used_functions (arg\comparison_expression.operands[1])
                + used_functions (arg\comparison_expression.operands[2]));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.INTERVAL_EXPRESSION' IN TYPEOF (arg)
THEN
   RETURN (used_functions(arg\interval_expression.interval_low)
            + used_functions(arg\interval_expression.interval_high)
            + used_functions(arg\interval_expression.interval_item));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.INDEX_EXPRESSION' IN TYPEOF (arg)
THEN
    RETURN (used_functions (arg\index_expression.operand)
      + used_functions (arg\index_expression.index));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.SUBSTRING_EXPRESSION' IN TYPEOF (arg)
THEN
    RETURN (used_functions (arg\substring_expression.operand)
      + used_functions (arg\substring_expression.index1)
      + used_functions (arg\substring_expression.index2));
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.CONCAT_EXPRESSION' IN TYPEOF (arg)
THEN
   REPEAT i := 1 TO SIZEOF (arg\concat_expression.operands);
      result := result + used_functions (
            arg\concat_expression.operands[i]);
       END_REPEAT;
      RETURN (result);
END_IF;
IF 'ISO13584_EXPRESSIONS_SCHEMA.FORMAT_FUNCTION' IN TYPEOF (arg)
THEN
   RETURN (used_functions (arg\format_function.value_to_format)
      + used_functions (arg\format_function.format_string));
END_IF;
```

```
IF 'ISO13584_EXPRESSIONS_SCHEMA.LIKE_EXPRESSION' IN TYPEOF (arg)
THEN
   RETURN (used_functions (arg\like_expression.operands[1])
       + used_functions (arg\like_expression.operands[2]));
END_IF;
RETURN ([ ]);

END_FUNCTION; -- used_functions

END_SCHEMA; -- ISO13584_expressions_schema
(*
```

Приложение А
(обязательное)

Сокращенные наименования объектов

Сокращенные наименования объектов, установленных в настоящем стандарте, приведены в таблице А.1. Требования к использованию сокращенных наименований указаны в методах реализации, приведенных в стандартах комплекса ИСО 10303.

Таблица А.1 —Сокращенные наименования объектов

Наименование объекта	Сокращенное наименование
ABS_FUNCTION	ABSFNC
ACOS_FUNCTION	ACSFNC
AND_EXPRESSION	ANDEXP
ASIN_FUNCTION	ASNFNC
ATAN_FUNCTION	ATNFNC
BINARY_BOOLEAN_EXPRESSION	BNBLEX
BINARY_FUNCTION_CALL	BNFNCL
BINARY_GENERIC_EXPRESSION	BNGNEX
BINARY_NUMERIC_EXPRESSION	BNNMEX
BOOLEAN_DEFINED_FUNCTION	BLDFFN
BOOLEAN_EXPRESSION	BLNEXP
BOOLEAN_LITERAL	BLNLTR
BOOLEAN_VARIABLE	BLNVRB
COMPARISON_EQUAL	CMPEQL
COMPARISON_EXPRESSION	CMPEXP
COMPARISON_GREATER	CMPGRT
COMPARISON_GREATER_EQUAL	CMGREQ
COMPARISON_LESS	CMPLSS
COMPARISON_LESS_EQUAL	CMLSEQ
COMPARISON_NOT_EQUAL	CMNTEQ
CONCAT_EXPRESSION	CNCEXP
COS_FUNCTION	CSFNC
DEFINED_FUNCTION	DFNFNC
DIV_EXPRESSION	DVEXP
ENVIRONMENT	ENVRNM
EQUALS_EXPRESSION	EQLEXP
EXPRESSION	EXPRSS

Окончание таблицы А.1

Наименование объекта	Сокращенное наименование
EXP_FUNCTION	EXPFNC
FORMAT_FUNCTION	FRMFNC
GENERIC_EXPRESSION	GNREXP
GENERIC_LITERAL	GNRLTR
GENERIC_VARIABLE	GNRVRB
SQUARE_ROOT_FUNCTION	SQRTFN
STRING_DEFINED_FUNCTION	STDFFN
STRING_EXPRESSION	STREXP
STRING_LITERAL	STRLTR
STRING_VARIABLE	STRVRB
SUBSTRING_EXPRESSION	SBSEXP
TAN_FUNCTION	TNFNC
UNARY_BOOLEAN_EXPRESSION	UNBLEX
UNARY_FUNCTION_CALL	UNFNCL
UNARY_GENERIC_EXPRESSION	UNGNEX
UNARY_NUMERIC_EXPRESSION	UNNMEX
VALUE_FUNCTION	VLFNC
VARIABLE	VRBL
VARIABLE_SEMANTICS	VRBSMN
XOR_EXPRESSION	XREXP

Приложение В
(обязательное)

Регистрация информационного объекта

В.1 Обозначение документа

Для обеспечения однозначного обозначения информационного объекта в открытой системе настоящему стандарту присвоен следующий идентификатор объекта:

{ ISO standard 13584 part (20) version(1) }

Значение этого обозначения установлено в ИСО 8824-1 и описано в ИСО 13584-1.

В.2 Обозначение схем

В.2.1 Обозначение ISO_generic_expressions_schema

Схеме **ISO_generic_expressions_schema** (раздел 4) присвоен следующий идентификатор объекта:

{ ISO standard 13584 part (20) version(1) object(1) ISO13584-generic-expressions-schema(1) }

В.2.2 Обозначение ISO_generic_expressions_schema

Схеме **ISO_expressions_schema** (раздел 5) присвоен следующий идентификатор объекта:

{ ISO standard 13584 part (20) version(1) object(1) ISO13584-expressions-schema(2) }

Приложение С
(справочное)

Диаграммы на языке EXPRESS-G

Рисунки С.1 — С.13 соответствуют текстам на языке EXPRESS, приведенным в разделах 4 и 5. В диаграммах использована графическая нотация EXPRESS-G для языка EXPRESS. Язык EXPRESS-G определен в приложении А ИСО 10303-11.

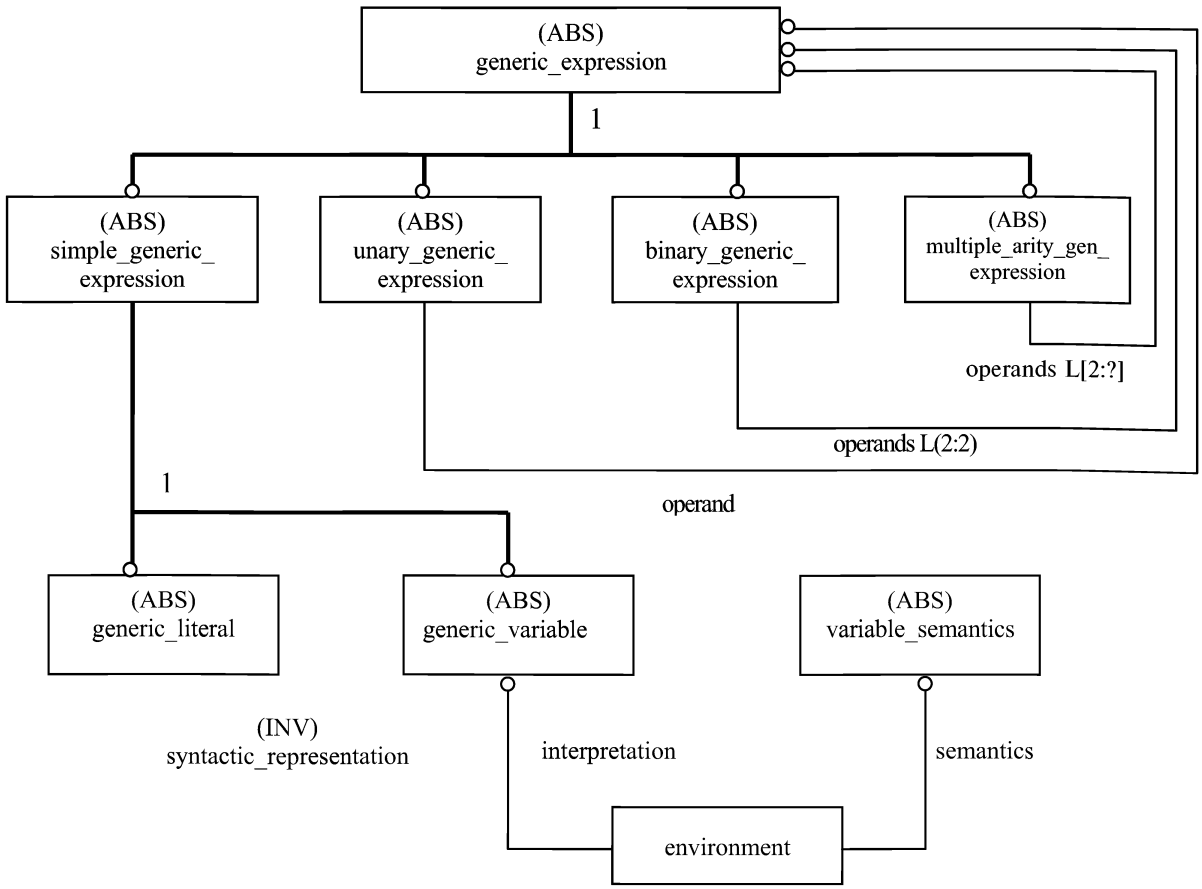

Рисунок С.1 — **ISO13584_generic_expressions_schema** — EXPRESS-G диаграмма 1 из 1

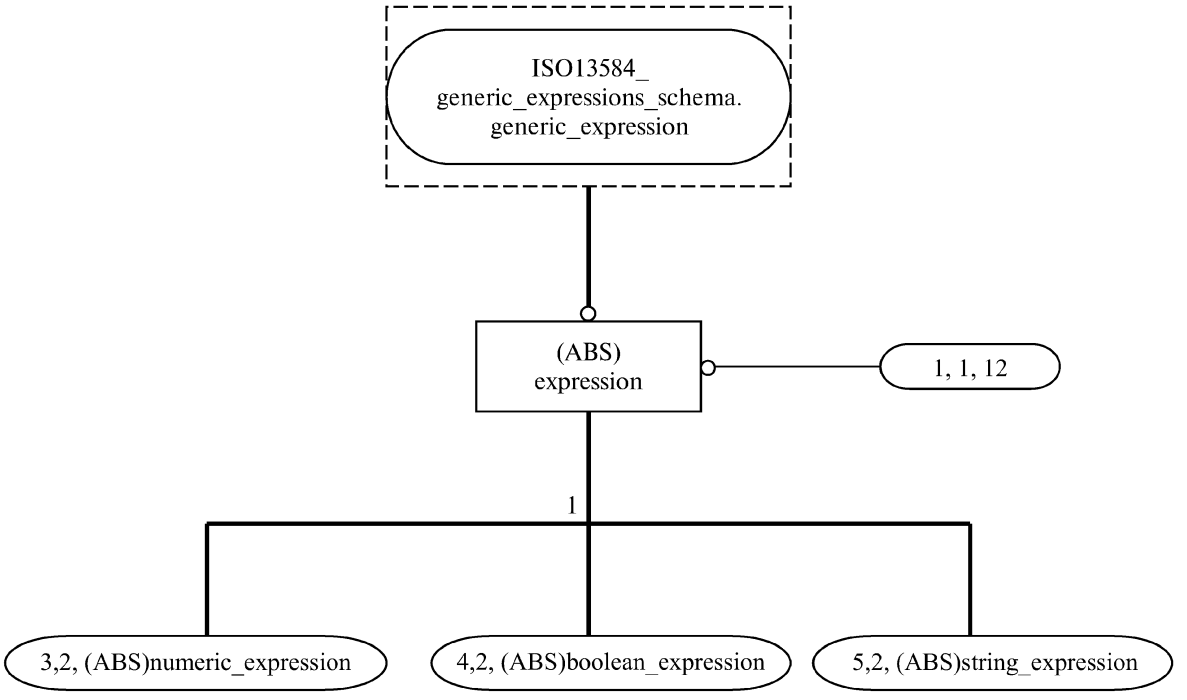

Рисунок С.2 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 1 из 12.
Строковая типизация выражений

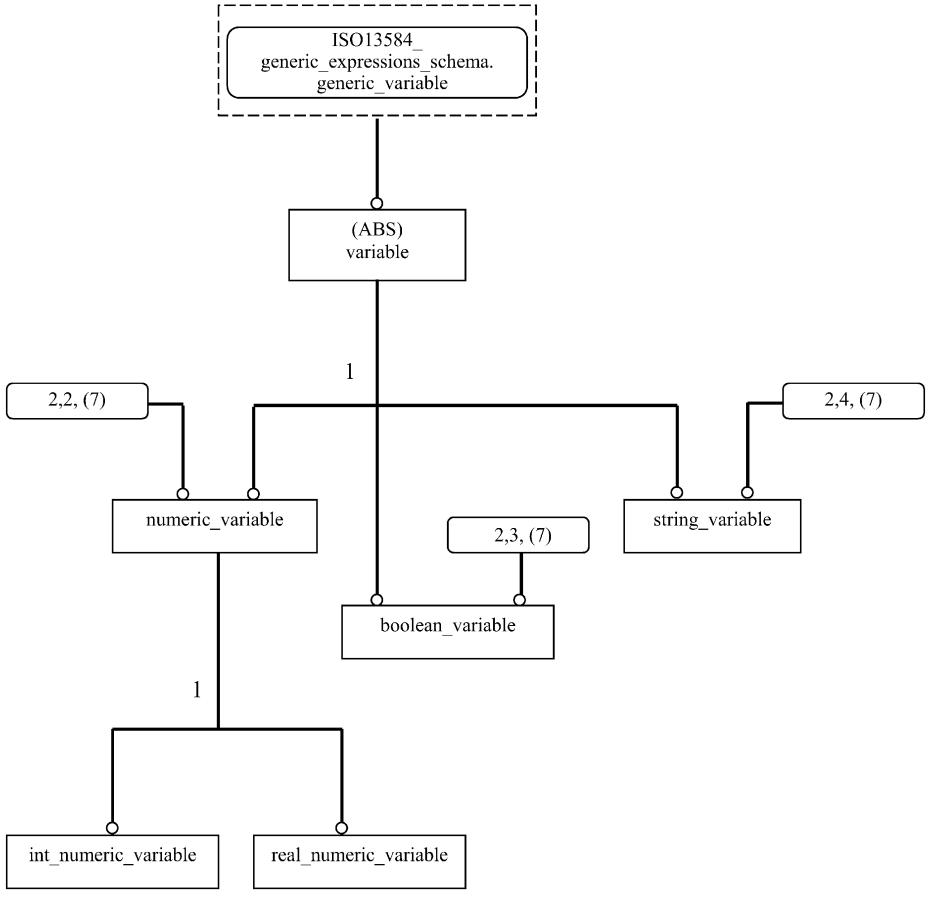

Рисунок С.3 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 2 из 12.
Переменные

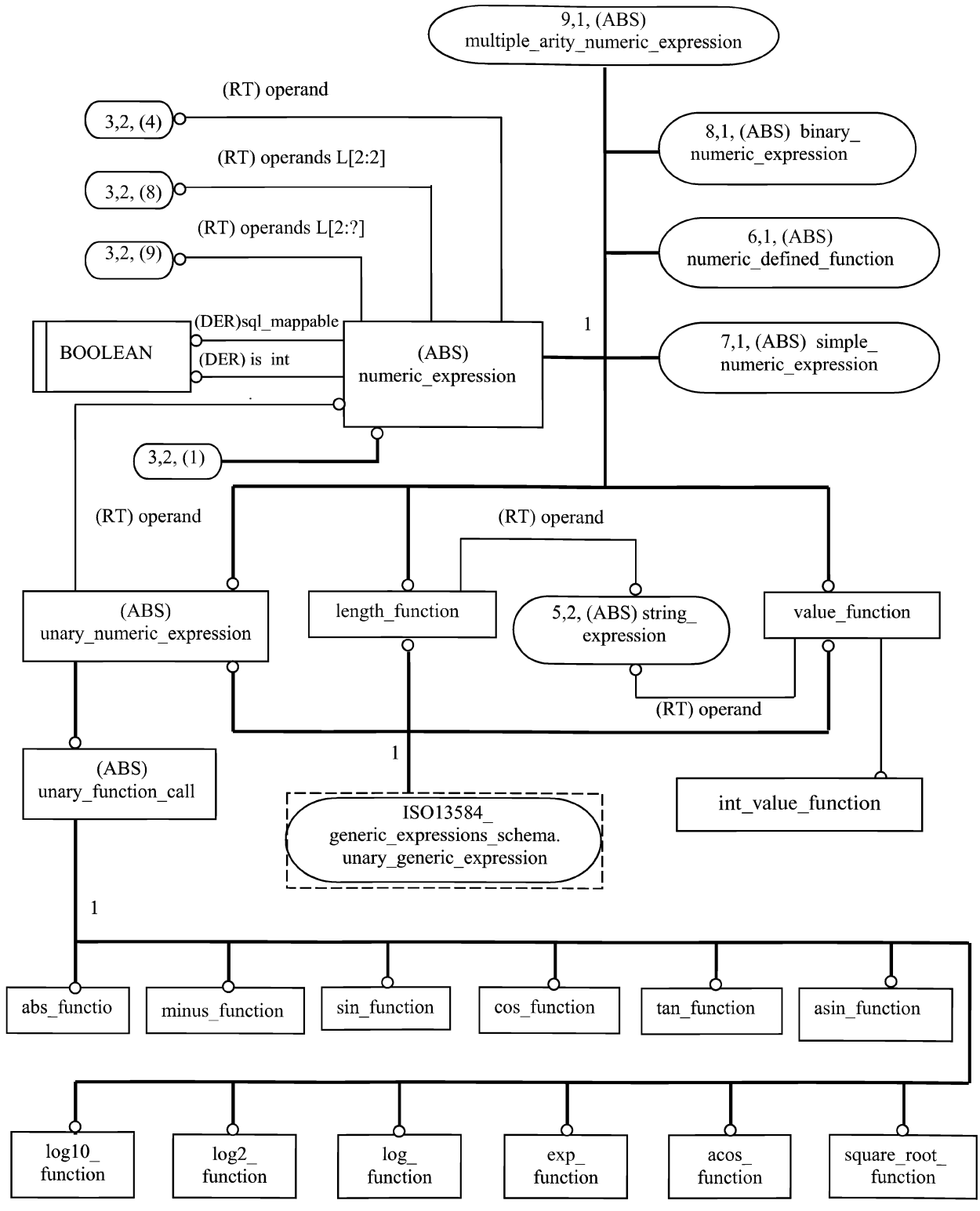

Рисунок С.4 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 3 из 12.
Числовые выражения

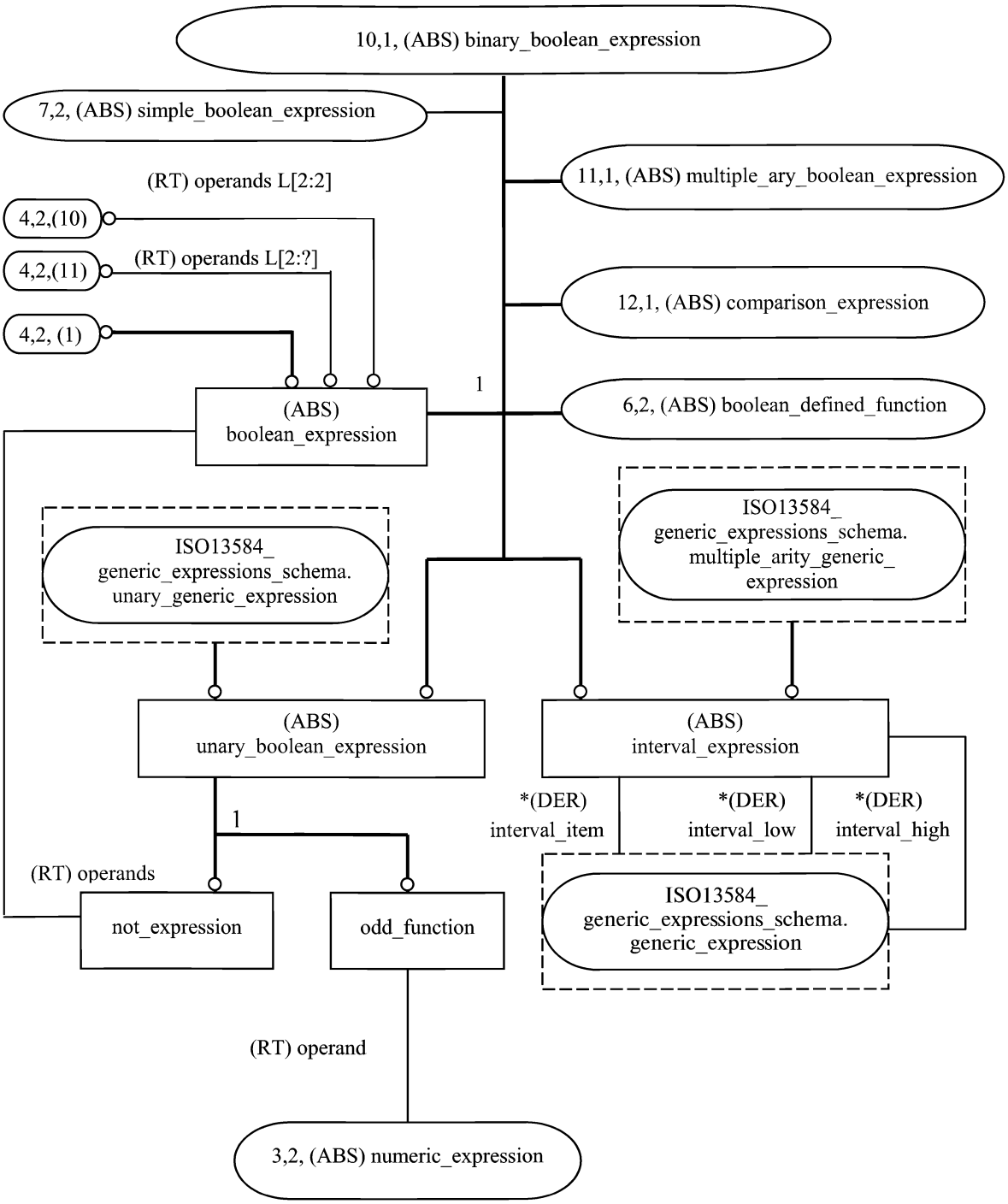

Рисунок С.5 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 4 из 12.
Булевы выражения

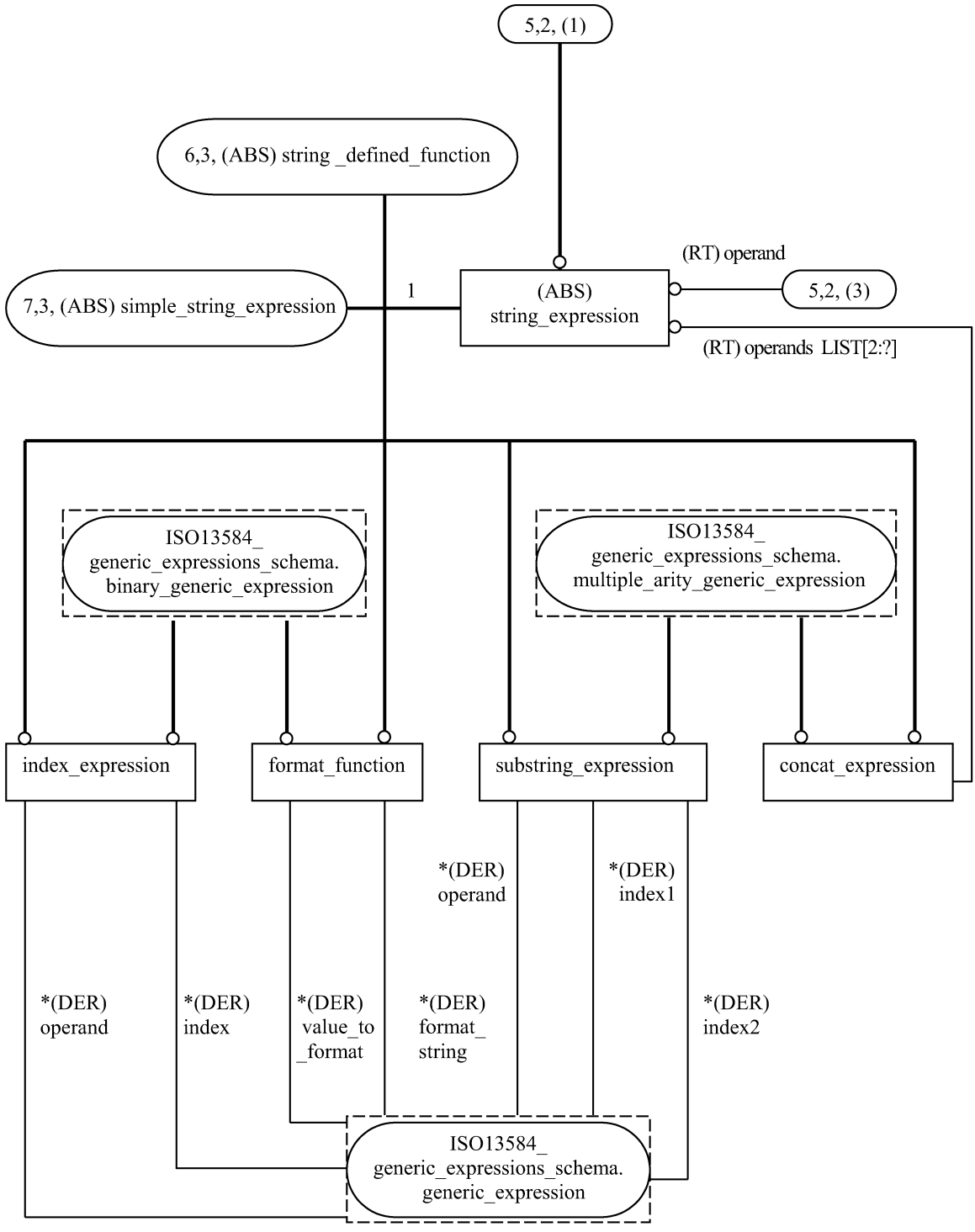

Рисунок С.6 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 5 из 12.
Строковые выражения

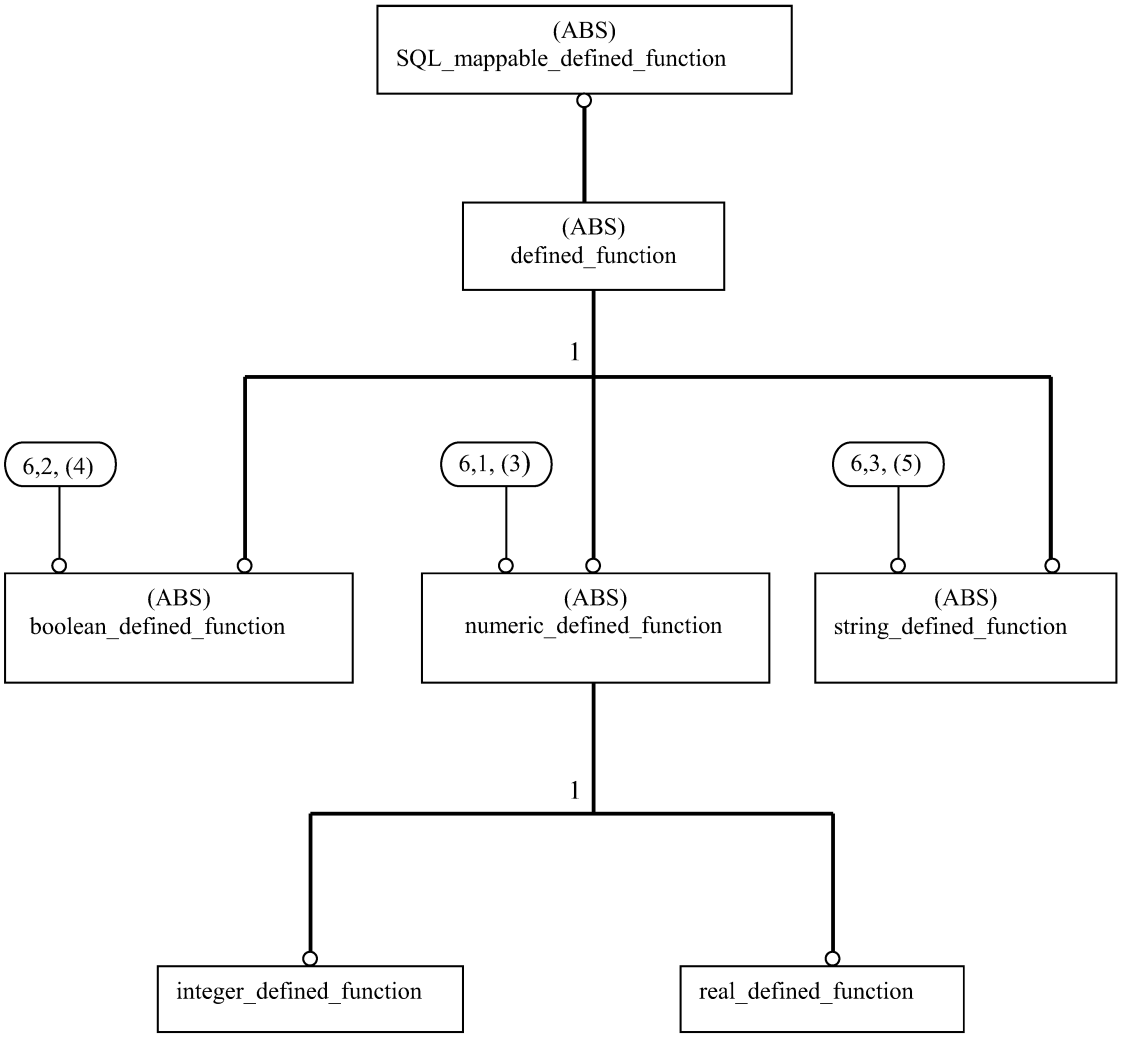

Рисунок С.7 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 6 из 12.
defined_function

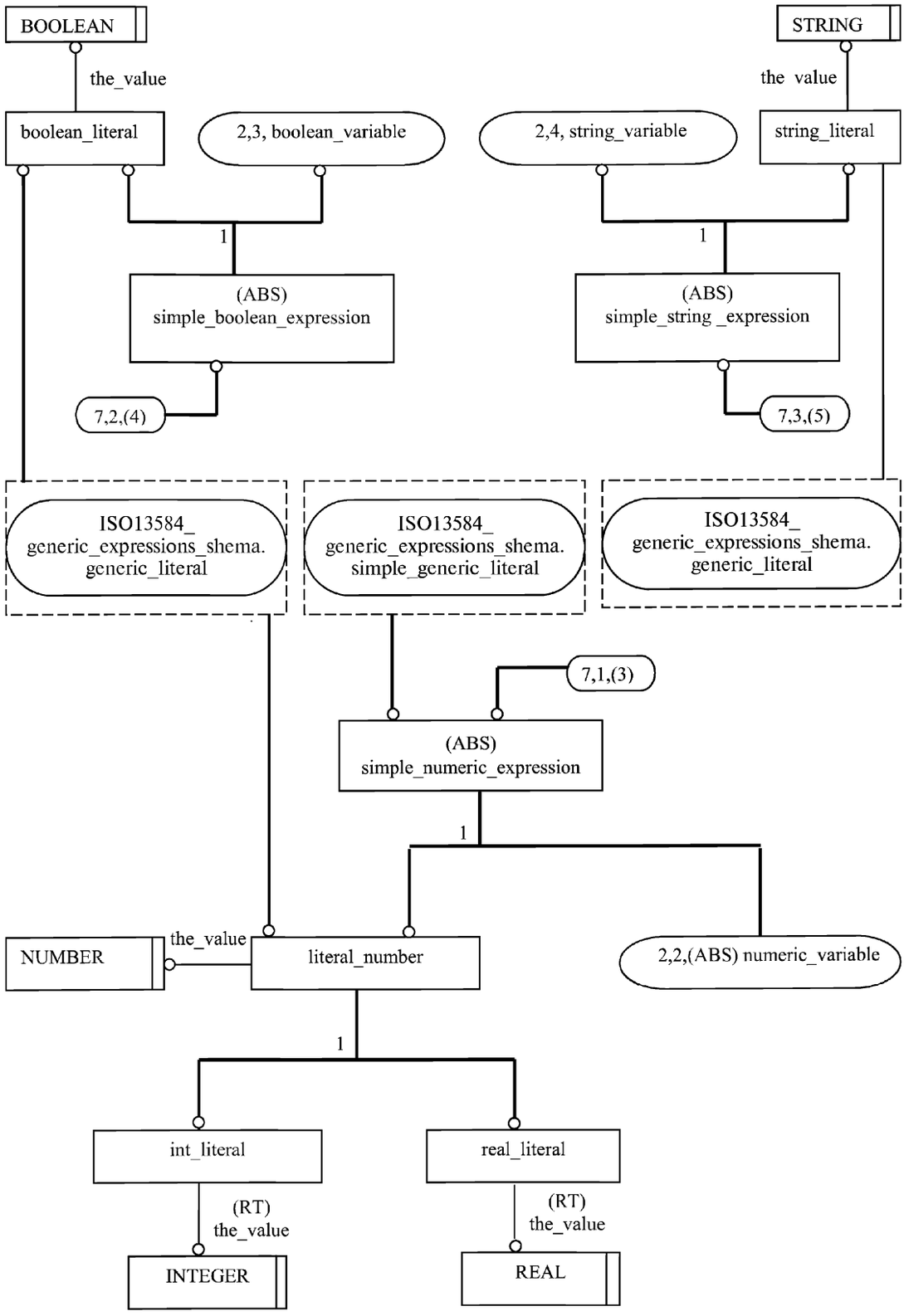

Рисунок С.8 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 7 из 12.
Подтипы **simple_generic_expression**

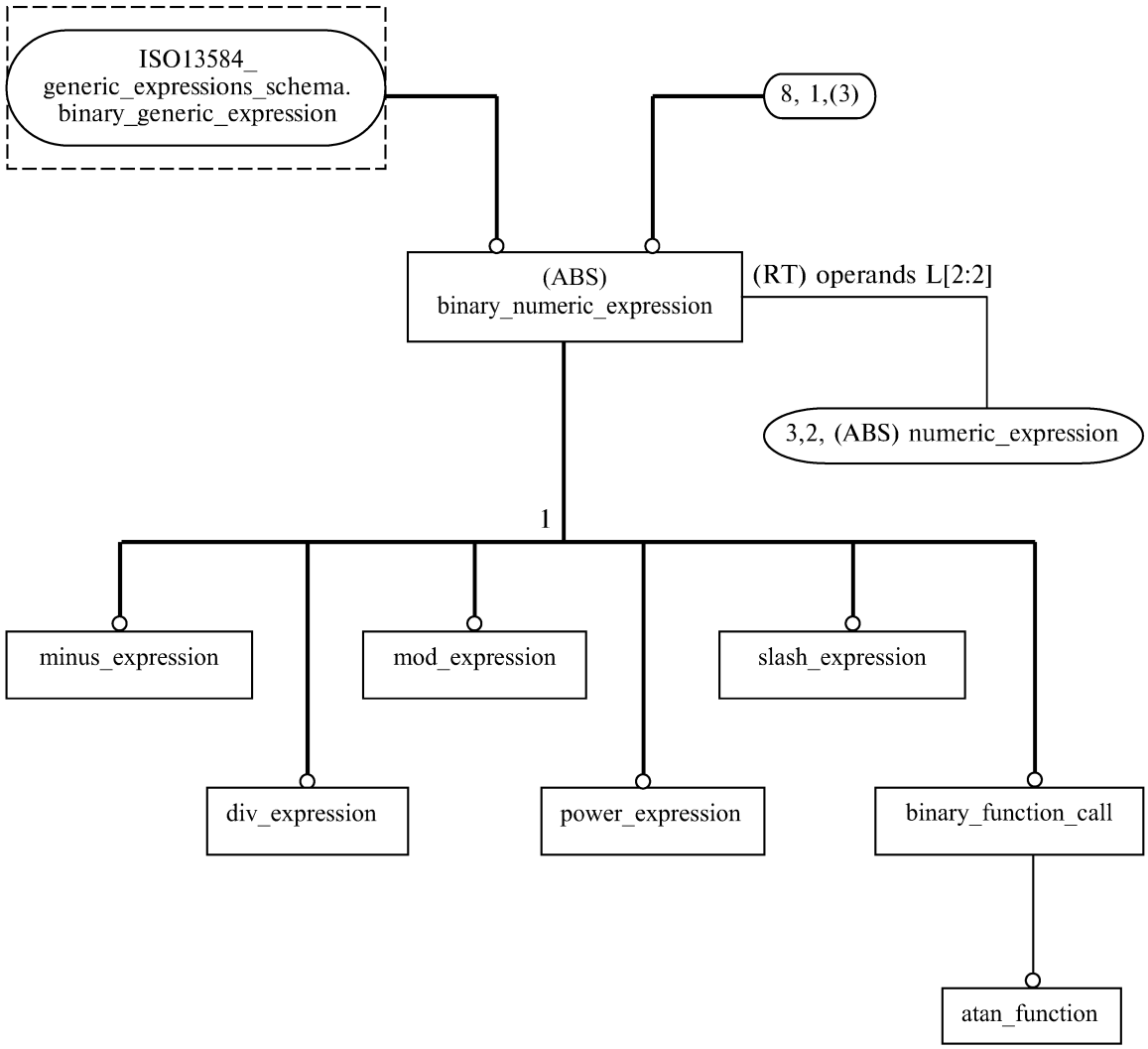

Рисунок С.9 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 8 из 12.
Бинарные числовые выражения

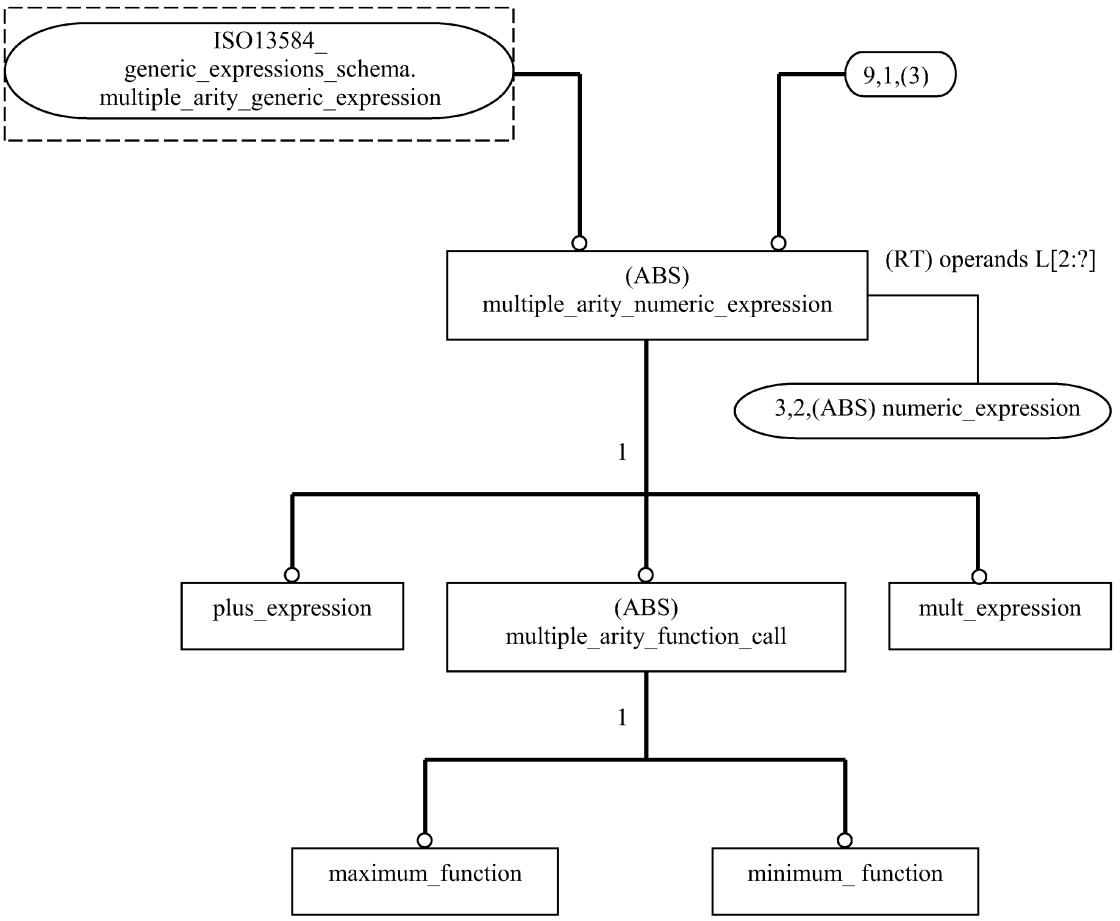

Рисунок С.10 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 9 из 12. Большая арность

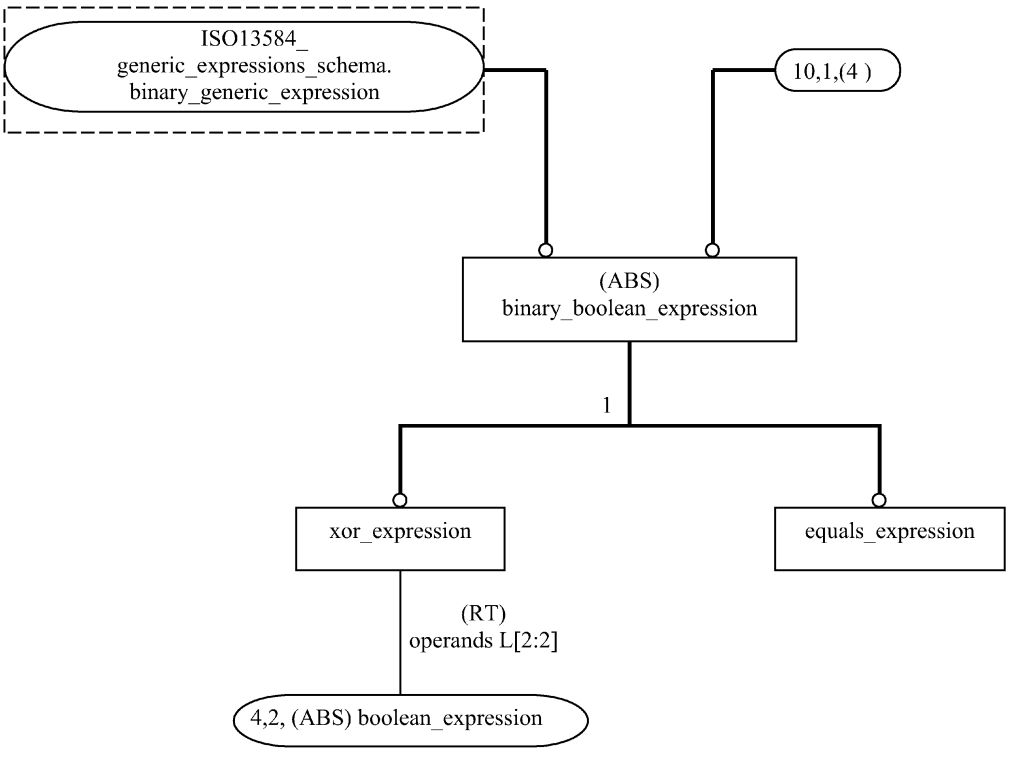

Рисунок С.11 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 10 из 12. Бинарные булевы выражения

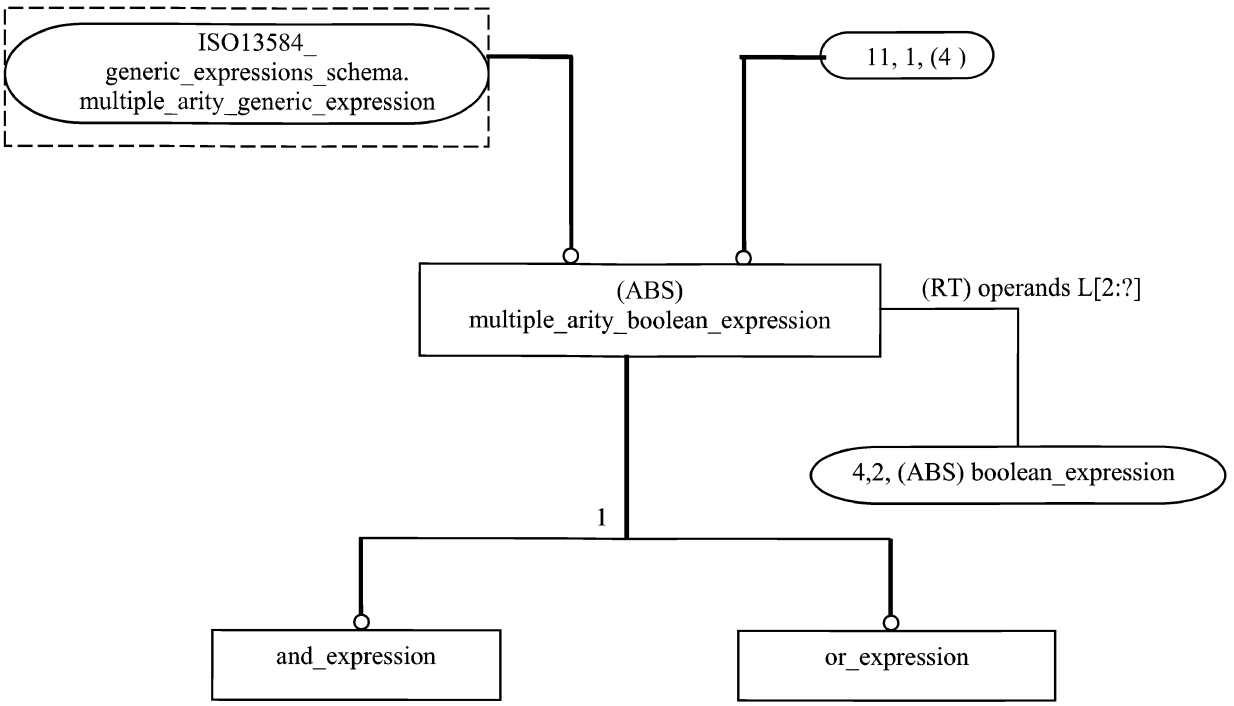

Рисунок С.12 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 11 из 12.
Булевы выражения большой арности

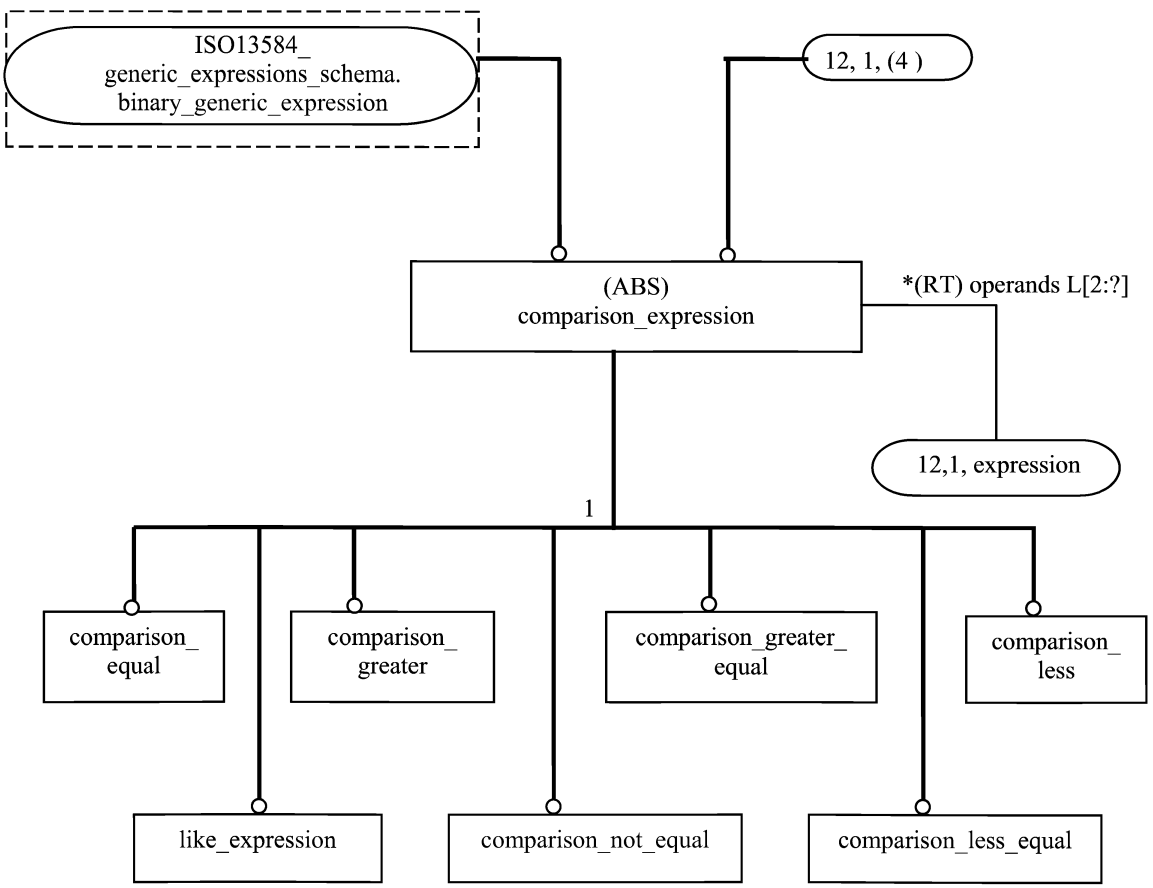

Рисунок С.13 — **ISO13584_expressions_schema** — EXPRESS-G диаграмма 12 из 12. Выражения сравнения

Приложение D
(справочное)

Использование схемы ISO13584_expressions_schema

D.1 Введение

Схемы, установленные в настоящем стандарте, являются обобщенными ресурсами, предназначенными для использования во всех случаях, когда выражения должны быть промоделированы на языке EXPRESS.

В настоящем приложении рассмотрены возможные варианты использования и применения этих обобщенных ресурсов.

ISO13584_expressions_schema кодирует выражения, которые могут быть построены из комбинации операторов простых выражений (взятых из языка EXPRESS), констант и переменных.

Для каждого из этих элементов (операторов, констант и переменных) семантика должна быть задана для каждого использования выражения.

Семантика операторов устанавливается в соответствии с тем, как она определена для каждого оператора в руководстве по языку EXPRESS.

При использовании выражений единственной семантикой, которая должна быть определена, является семантика переменных, т.е. должна быть определена функция интерпретации, которая ограничивает значения этих переменных.

D.2 Функция интерпретации и семантика переменной

Семантика операторов и констант определена в настоящем стандарте.

Константы имеют простую семантику, поскольку они интерпретируются значением, которое они представляют.

В выражениях переменные обозначают понятия, которые связаны со значениями. Выражения, использующие эти переменные, обозначают взаимосвязи между их текущими значениями.

При использовании **ISO13584_expressions_schema** для каждой переменной должно быть определено следующее:

- какое понятие обозначает переменная;
- в каком контексте это понятие может быть использовано;
- каков механизм (функция интерпретации), который ограничивает значение этой переменной.

В настоящем стандарте понятие, которое обозначает переменная (ее семантика), представлено объектом **variable_semantics**. Поэтому три указанные выше позиции должны быть заданы определением подтипа типа данных этого абстрактного объекта.

D.3 Представление функции интерпретации в настоящем стандарте

Как указано в D.2, переменные должны иметь функцию интерпретации для того, чтобы задать их семантику. В комплексе стандартов ИСО 13584 эта цель достигается путем комбинации трех объектов: **variable**, **environment** и **variable_semantics**. Объект **variable** определяет синтаксическое представление семантического понятия, которое представлено объектом **variable_semantics**. Объект **environment** связывает эти два объекта, чтобы точно ассоциировать одну семантику с данной переменной.

На рисунке D.1 приведена схема определения этих объектов.

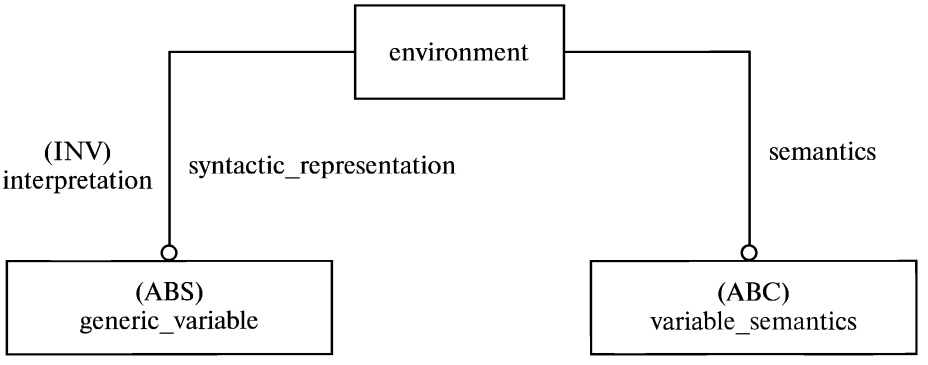

Рисунок D.1 — Связь синтаксиса и семантики переменной

D.4 Использование объекта variable_semantics для определения семантики новых переменных

Предполагается, что понятие, которое должно быть представлено переменной, уже существует в виде объекта модели данных. Поэтому на уровне модели данных возможны следующие два подхода, чтобы связать это понятие с переменной:

- при первом подходе используется отдельный подтип объекта **variable_semantics**, который представляет этот объект посредством атрибута(ов);

- при втором подходе определяется общий подтип для объекта **variable_semantics** и типа данных этого объекта.

Данная информационная модель определяет только первую из упомянутых выше позиций (см. D.2). Другая позиция должна быть определена в документации на эту модель.

Иллюстрация этих двух подходов приведена в D.4.1 и D.4.2.

D.4.1 Использование отдельного подтипа объекта variable_semantics

Данный подход к определению семантики заключается во введении подтипа объекта **variable_semantics**, который представляет собой понятие, которое обозначает переменная, посредством атрибута(ов).

Следующий пример показывает, как этот подход работает. Выбран реальный пример, чтобы показать осуществимость этого подхода на практике.

Пример — Определение переменной SELF:

Напомним, что переменная SELF является особой переменной, встречающейся в объектно-ориентированных подходах. Она используется в контексте экземпляра класса для представления этого отдельного экземпляра и, возможно, значения одного отдельного атрибута этого отдельного экземпляра.

В ИСО 13584-24, в котором моделирование семейств деталей осуществляется на основе объектно-ориентированного подхода, показана необходимость представления переменных SELF, чтобы допустить использование выражений, включающих этот вид переменных. На рисунке D.2 приведена EXPRESS-G диаграмма, демонстрирующая способ представления этой семантики на основе простого наследования объекта variable_semantics.

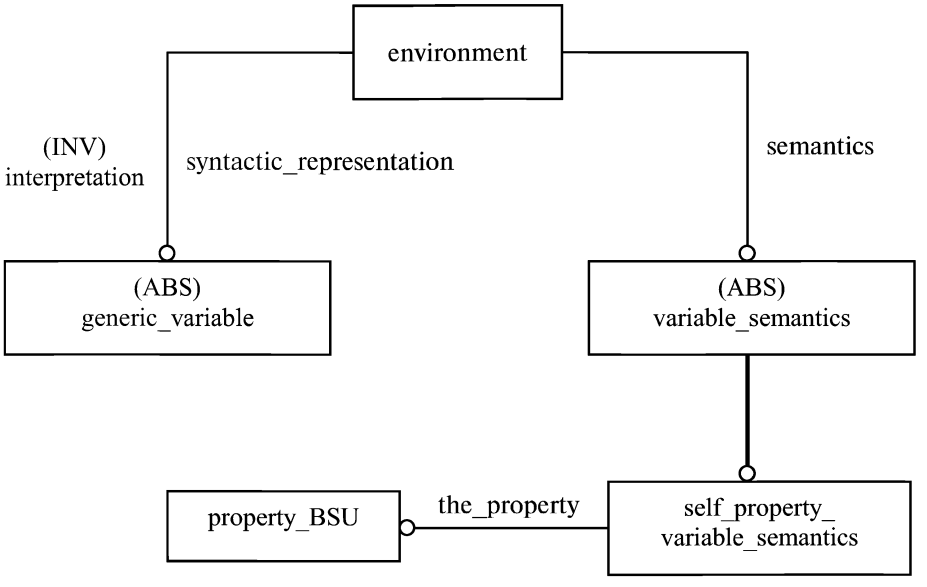

Рисунок D.2 — Установление семантики определением подтипа объекта **variable_semantics**

Примечание — Объект **property_BSU** определен в ИСО 13584-42. Он позволяет идентифицировать одно отдельное свойство.

Объект **self_property_variable_semantics** определен как подтип объекта **variable_semantics**. Он имеет атрибут, называемый *the_property*, который идентифицирует отдельное свойство вложенного класса (представленного объектом **property_BSU**). Поэтому этот объект несет в себе семантику *the_property* экземпляра SELF.

Заметим, что в документации на объект **self_property_variable_semantics** определено, что:

- соответствующая переменная должна использоваться в контексте экземпляра класса PLib;
- переменная обозначает значение релевантного свойства экземпляра этого класса.

D.4.2 Множественное наследование объекта variable_semantics и другого объекта

Множественное наследование позволяет собрать в общем объекте свойства различных супертипов.

Беря за основу концепцию совместного использования, можно определить семантику переменной как совместное использование объекта **variable_semantics** (чтобы показать, что мы имеем дело с семантикой переменной) и любого другого объекта, который мы хотим представить через эту переменную.

Пример — Определение объекта variable_semantics, связанного с объектом material_property_representation, описанным в ИСО 10303-45.

Путем определения подтипов конструкций ресурсов из **material_property_representation_schema**, установленной в ИСО 10303-45, значение объекта **material_property** может быть связано с выражением **numeric_expression**, которое определяет зависимость между значением этого **material_property** и другими **property_definitions**, на которые ссылается их **material_property_representation**.

Способ реализации вышеуказанного представлен на рисунке D.3

Правило WHERE (где) в объекте **material_property_representation_and_function** необходимо обеспечить, чтобы выражение **functional_specification numeric_expression** включало в качестве переменных только те переменные, которые связаны через объект **environment** с **material_property_semantics**, и для которых все эти **material_property_semantics** соответствуют набору свойств, моделирующих среду (т.е. на каждую из этих **material_property_semantics** ссылается **SELF.dependent_environment.elements[i].definition** через некоторый индекс i).

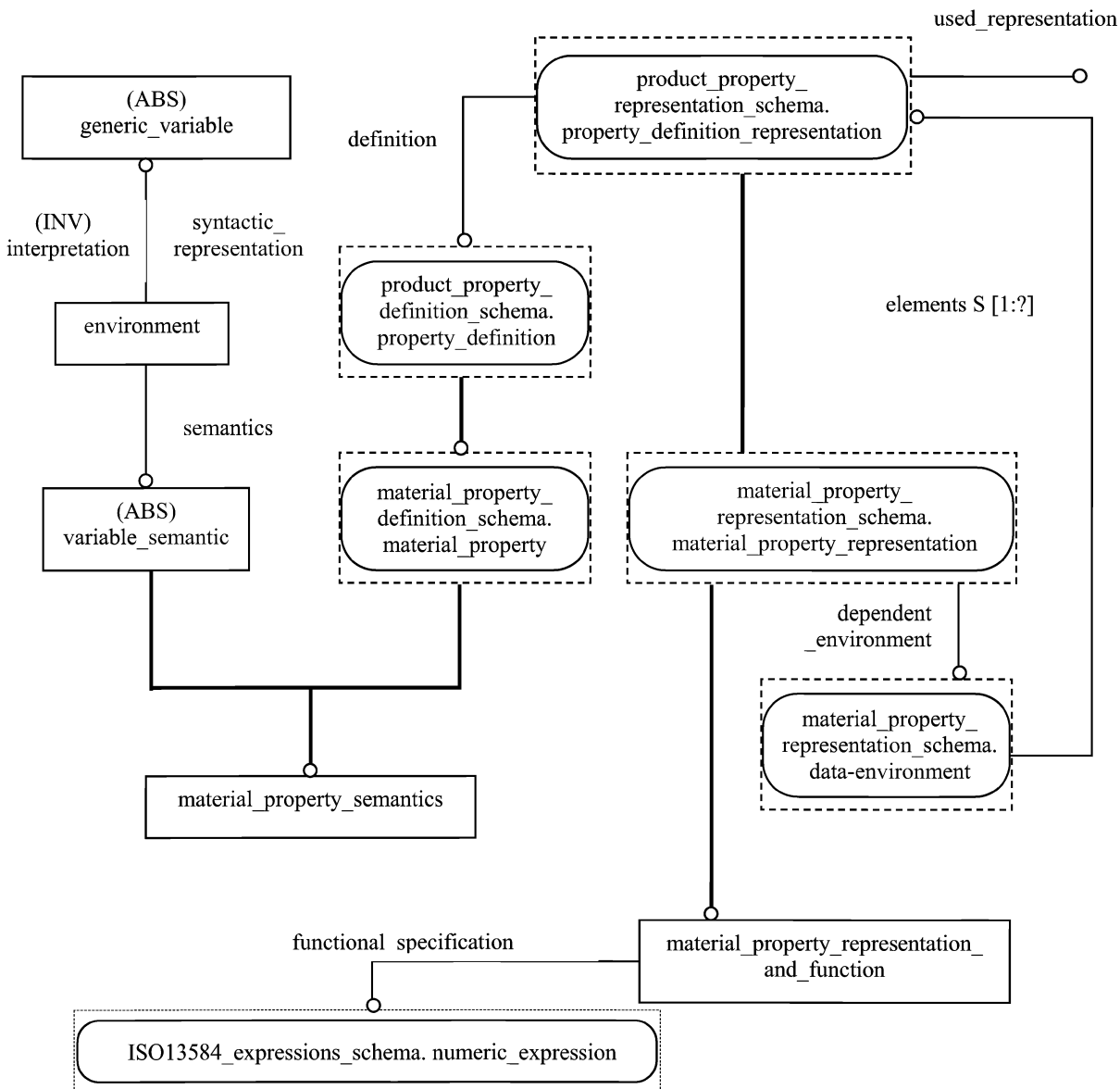

Рисунок D.3 — Конкретизация семантики путем определения подтипа объекта **variables_semantics** и другого объекта

Используя подтип **material_property_representation_and_function**, объект **material_ property** связывается как со значением, которое соответствует некоторым отдельным значениям свойств, от которых они зависят, так и с функцией, определяющей функциональную зависимость между этими значениями.

Отметим, что этот подход не требует использования дополнительного атрибута у объекта **material_property_semantics**. Механизм наследования распространяет все атрибуты объекта **material_property** (т.е. идентификацию свойства).

Отметим также, что в этом примере в документации должны быть описания контекста использования переменной (для одного отдельного изделия, для любого изделия из некоторого набора и т.д.) и функции интерпретации.

D.4.3 Определение понятия, не представленного в модели

Семантика некоторой переменной может соответствовать понятиям, которые еще не представлены в модели в явном виде. В этом случае полная информационная модель этого понятия должна быть построена как подтип **variable_semantics**.

На рисунке D.4 показано, как могут быть определены две переменные, которые представляют координаты *x* и *y* в некотором отдельном двумерном пространстве. Такие переменные могут быть включены в любое числовое выражение, которое определяет точную (математическую) форму конкретного объекта **curve**.

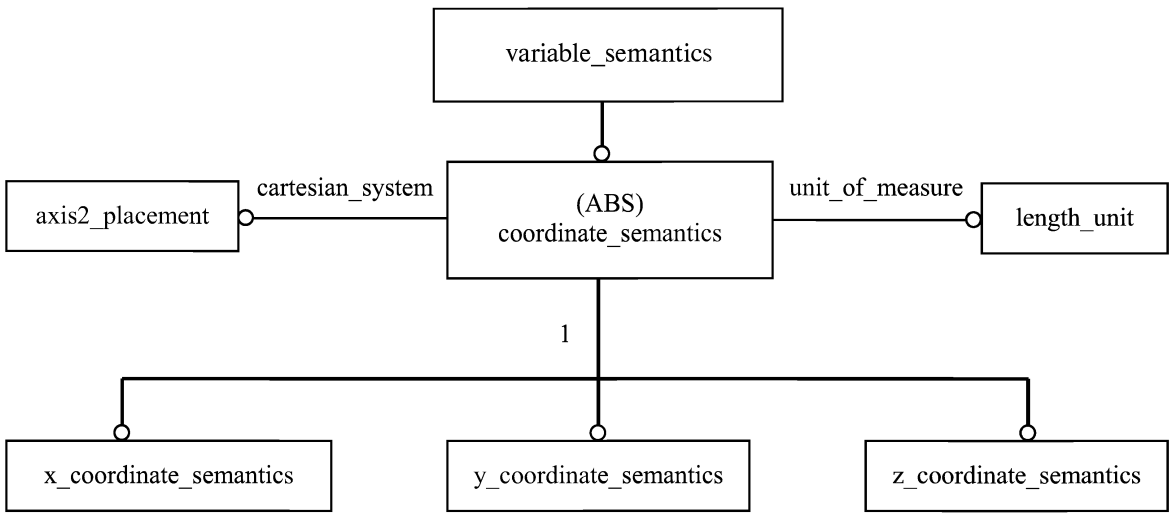

Рисунок D.4 — Пример определения понятия, не представленного в модели: координаты

Приложение E
(справочное)

Конкретизация схем

E.1 Введение

Определения схем **ISO13584_generic_expressions_schema** и **ISO13584_expressions_schema** являются обобщенными ресурсами, так как они могут быть конкретизированы без оказания какого-либо воздействия на уже определенные ресурсы. Эти схемы были определены для того, чтобы они могли быть конкретизированы или расширены для введения и моделирования выражений любого вида. Целью настоящего приложения является показать, как может быть выполнена конкретизация этих двух схем.

E.2 Конкретизация схемы ISO13584_generic_expressions_schema

ISO13584_generic_expressions_schema определяет структуру выражения любого вида без введения какого-либо конкретного типа данных.

Конкретизация **ISO13584_generic_expressions_schema** требует введения конкретного(ых) типа(ов) данных. Такой тип данных должен быть введен путем определения подтипа существующих EXPRESS-ресурсов и может указываться как источник новыми операторами, которые манипулируют этим нововведенным типом данных.

Пример — В схеме ISO13584_expressions_schema в качестве конкретных типов данных определены только integer, real, Boolean и string. Поэтому только эти типы связаны с операторами, которые вычисляют значения.

При введении нового типа данных в процесс конкретизации вовлекаются следующие объекты языка EXPRESS:

- **generic_expression**;
- **generic_variable**;
- **variable_semantics**;
- **generic_literal**;
- **unary_generic_expression**;
- **binary_generic_expression**;
- **multiple_arity_generic_expression**.

Е.3 Конкретизация схемы ISO13584_expressions_schema

Конкретизация схемы **ISO13584_expressions_schema** представляет возможность расширения набора операторов теми, которые либо объединяют значения, принадлежащие к базовым типам данных этой схемы (т.е. NUMBER, REAL, INTEGER, BOOLEAN и STRING), либо производят значения, принадлежащие к этим базовым типам данных (например, расстояние между двумя точками).

Такая специализация должна обеспечить прочность и корректность результирующих выражений и не должна влиять на корректность обобщенных ресурсов, представленных как EXPRESS-функции. Кроме того, эти ресурсы должны быть согласованы с их абстрактными описаниями (например, **is_int_expr** должно позволить определить, принадлежит ли тип данных выражения к типу данных INTEGER).

Следующие EXPRESS-объекты были специально введены, чтобы поддерживать такой процесс конкретизации:

- **boolean_defined_function**;
- **integer_defined_function**;
- **numeric_defined_function**;
- **real_defined_function**;
- **string_defined_function**;
- **SQL_mappable_defined_function**.

Е.4 Методология для конкретизации настоящего стандарта

В данном разделе установлена последовательность действий для определения различных конкретизаций схем, установленных в настоящем стандарте.

Е.4.1 Идентифицировать конкретный тип данных, который должен быть обработан выражением. Если должны использоваться только типы данных INTEGER, REAL, BOOLEAN и STRING, следует конкретизировать схему **ISO13584_expressions_schema**, в противном случае следует конкретизировать схему **ISO13584_generic_expressions_schema**.

Е.4.2 Идентифицировать семантические понятия, которые должны соответствовать определенным выражениям. Определить требуемые подтипы объекта **variable_semantics** (см. приложение D).

Е.4.3 Конкретизация схемы ISO13584_generic_expressions_schema

Е.4.3.1 Определить корень всех выражений нового типа данных путем определения подтипа объекта **generic_expression** (или **numeric_expression**, **Boolean_expression**, либо **string_expression** соответственно).

Е.4.3.2 Определить переменные этих выражений путем определения подтипа объекта **generic_variable**, который связан через объект **environment** с семантикой, определенной по Е.4.2.

Е.4.3.3 Определить константы этих выражений путем определения подтипа объекта **generic_literal**.

Е.4.3.4 Определить операторы, которые объединяют переменные и константы нового типа данных, путем определения подтипов объектов **unary_generic_expression**, **binary_generic_expression** и **multiple_arity_generic_expression** в соответствии с их арностью. Присоединить правила, которые обеспечивают семантическую прочность и корректность этих операторов.

Е.4.4 Интеграция со схемой ISO13584_expressions_schema

Е.4.4.1 Если существует какой-либо оператор, который производит значения нового типа данных из выражений **expression** (например, комплексное число, определяемое двумя вещественными числами), то:

- снабдить ссылками схему **ISO13584_expressions_schema**;
- определить этот оператор как общий тип данных корневого объекта, определенного по Е.4.3.1, и один из следующих объектов: **unary_generic_expression**, **binary_generic_expression** или **multiple_arity_generic_expression**, в соответствии с арностью этого оператора. Добавить правила, которые обеспечивают семантическую корректность этого оператора.

Е.4.4.2 Если существует какой-либо оператор на новом типе данных, который генерирует значения одного из базовых типов данных схемы **ISO13584_expressions_schema**, то для каждого оператора или функции, возвращающих типы данных INTEGER, REAL, BOOLEAN или STRING, необходимо ввести подтип соответствующего объекта

defined_function, чтобы определить соответствующую функцию и один из объектов **unary_generic_expression**, **binary_generic_expression** или **multiple_arity_generic_expression** в соответствии с арностью этого оператора.

Этот процесс обеспечивает следующее:

- любое выражение, согласующееся с конкретизированной схемой, является семантически корректным;
- все функции, определенные настоящим стандартом как ресурсы (например, **is_int_expr**, **used_variables**, **is_SQL_mappable** и т. д.) согласуются со своими абстрактными описаниями.

Примечание — При конкретизации схемы **ISO13584_expressions_schema** действия по Е.4.3 не проводят.

Е.5 Пример конкретизации схемы ISO13584_generic_expressions_schema

В данном примере фрагмент описания на языке EXPRESS выбран настолько простым, насколько это возможно, чтобы сделать понятной конкретизацию данной схемы.

Рассмотрим конкретизацию схемы **ISO13584_generic_expressions_schema** для обработки выражений, оперирующих с комплексными числами. Эти выражения получают путем определения подтипа схемы **ISO13584_generic_expressions_schema**.

В следующих EXPRESS-спецификациях показана суть такой схемы. При этом будут использоваться лишь несколько выражений:

Шаг 1 — заданным типом данных являются комплексные числа.

Шаг 2 — семантическое понятие, которое должно быть определено комплексными выражениями.

Предположим, что необходимо связать эти выражения с парой координат двумерных объектов **cartesian_point** (по ИСО 10303-42).

Можно определить следующую семантику для обеспечения связи с комплексными переменными.

EXPRESS-спецификация

```
*)
ENTITY position_variable_semantics
SUBTYPE OF (variable_semantics, cartesian_point);
WHERE
    SELF\geometric_representation_item.dimension=2;
END_ENTITY;
(*
```

Семантикой должна быть точка в двумерном пространстве, а в описании должно быть указано, что действительная часть является координатой *x*, а мнимая часть — координатой *y*.

Предполагается, что в данном примере не нужны выражения **point_expressions**.

Шаг 3 — определение **complex_expression** путем конкретизации схемы **ISO13584_generic_expressions_schema**.

EXPRESS-спецификация

```
*)
ENTITY complex_expression
SUBTYPE OF (generic_expression)
ASTRACT SUPERTYPE OF (ONEOF (                simple_complex_expression,
                                             unary_complex_expression,
                                             binary_complex_expression,
                                             multiple_arity_complex_expression));
END_ENTITY;

ENTITY simple_complex_expression
ABSTRACT SUPERTYPE ( OF ONEOF(               complex_variable,
                                                        complex_literal))
SUBTYPE OF (complex_expression, simple_generic_expression);
END_ENTITY;

ENTITY complex_variable;
SUBTYPE OF (simple_complex_expression, generic_variable);
END_ENTITY;

ENTITY complex_literal
```

```
SUBTYPE OF (simple_complex_expression, generic_literal);
  real_part, imag_part : REAL;
END_ENTITY;

ENTITY unary_complex_expression
.......
END_ENTITY;

ENTITY binary_complex_expression
.....
END_ENTITY;

ENTITY multiple_arity_complex_expression
......
END_ENTITY;

ENTITY position_cartesian_constructor
SUBTYPE OF (complex_expression, binary_generic_expression);
  x, y : REAL
END_ENTITY;

ENTITY position_polar_constructor
SUBTYPE OF (complex_expression, binary_generic_expression);
  ro, theta : REAL;
END_ENTITY;
(*
```

Шаг 4 — Специализация схемы **ISO13584_generic_expressions_schema** задается определением следующих объектов:

EXPRESS-спецификация

```
*)
ENTITY real_part_defined_function
SUBTYPE OF (real_defined_function, unary_generic_expression);
   the_complex : complex_expression;
END_ENTITY;

ENTITY imaginary_part_defined_function
SUBTYPE OF (real_defined_function, unary_generic_expression)
  the_complex : complex_expression;
END_ENTITY;

ENTITY angle_defined_function
SUBTYPE OF (real_defined_function, unary_generic_expression)
END_ENTITY;

ENTITY module_defined_function
SUBTYPE OF (real_defined_function, unary_generic_expression);
END_ENTITY;
(*
```

Отметим, что эта схема осталась неполной, так как целью данного приложения не является установка стандартной схемы для комплексных выражений.

Е.6 Пример специализации схемы ISO13584_expressions_schema

Данный пример соответствует случаю, когда шаг 3, приведенный в Е.5, опущен.

Необходимо ввести оператор, который вычисляет расстояние между двумя точками в декартовом пространстве. Эта функция возвращает вещественное число. Объект, соответствующий этой функции, получают путем конкретизации. Вводят подтип функции **real_defined_function**. Определяют следующий объект:

EXPRESS-спецификация

```
*)
ENTITY Distance_function
SUBTYPE OF(real_defined_function);
  pt1, pt2 : point;
END_ENTITY;
(*
```

Приложение F
(справочное)

Статический анализ выражений

F.1 Введение

В данном приложении рассматриваются выражения в контексте настоящего стандарта. Могут быть выделены две возможные точки зрения в соответствии с введенными понятиями и предположениями, сделанными в разделе 4: период обмена и период оценки.

В настоящем стандарте рассматривается только обмен выражениями. Только статические свойства выражений могут быть выведены на этом этапе. На нем не моделируются ни функция оценки, ни то, как эта оценка проводится.

В данном приложении рассматривается анализ выражений, который может быть выполнен в период обмена, и даются некоторые пояснения по этим процессам.

На уровне обобщенных выражений возможны вычисления значений используемых переменных и функций. Более того, ацикличность графа, представляющего выражение, также может быть проверена.

На уровне выражений можно вычислить тип данных выражения.

F.2 Функция is_acyclic_function

Напомним, что:

- ориентированный граф представляет собой набор узлов, соединенных некоторыми однонаправленными связями;

- ориентированный граф является ациклическим, если не существует пути из узла, приводящего обратно в этот же узел.

Как установлено в разделе 4 настоящего стандарта, структурой, лежащей в основе выражения, является ориентированный ациклический граф. Эта структура является обязательной для обхода выражения и обеспечивает, что все функции, которые выполняют обход такого графа, завершаются, когда они достигают узла, из которого нет исходящей связи, приводящей к этому узлу.

Функция **is_acyclic** проверяет, соответствует ли выражение этому условию.

Заметим, что дерево является частным случаем ориентированного ациклического графа.

F.3 Функции used_variables и used_functions

Функции **used_variables** и **used_functions** являются функциями, которые в качестве результата возвращают все объекты, представляющие переменные или функции, встречающиеся в данном выражении.

Эти функции осуществляют обход ориентированного ациклического графа, который представляет собой выражение, для оценки и синтеза типа данных и возвращают переменные и функции, встречающиеся в этом выражении.

F.4 Функция is_SQL_mappable

Функция **is_SQL_mappable** проверяет, является ли данное выражение отображаемым на языке SQL для баз данных с запросами.

Эта функция осуществляет обход ориентированного ациклического графа, который представляет собой выражение, для оценки и синтеза типа данных и возвращает значение TRUE, если выражение является отображаемым.

В ИСО 13584-24 эта функция была использована для обработки структур данных, связанных с таблицами и реляционными базами данных, а также для проверки отображаемости нескольких выражений, встречающихся в ограничениях.

F.5 Контроль типа и синтез типа

Контроль типа является процессом, который позволяет проверить корректность типа выражения.

Синтез типа является процессом, который позволяет синтезировать тип данных выражения.

В **ISO13584_expressions_schema**, определенной в настоящем стандарте, контроль типа обеспечивается ограничениями этой схемы. Синтез типа является результатом применения EXPRESS-функции TYPEOF к объекту, который является носителем типа данных.

Приложение G
(справочное)

Сведения о соответствии национальных стандартов Российской Федерации ссылочным международным стандартам

Таблица G.1

Обозначение ссылочного международного стандарта	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 9075:1992	ГОСТ Р ИСО/МЭК 9075—93 Информационная технология. Язык баз данных SQL с расширением целостности
ИСО 10303-11:1994	ГОСТ Р ИСО 10303-11—2000 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS
ИСО 10303-44:1994	ГОСТ Р ИСО 10303-44—2002 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 44. Интегрированные обобщенные ресурсы. Конфигурация структуры изделия

УДК 331.103.25 (006.34):004.65:006.354 ОКС 25.040.40 П87 ОКСТУ 4002

Ключевые слова: библиотека деталей, логический ресурс, логическая модель выражений, EXPRESS-схемы, схема выражений, схема обобщенных выражений, объекты, функции, выражения

Редактор *Т.А. Леонова*
Технический редактор *Л.А. Гусева*
Корректор *В.И. Варенцова*
Компьютерная верстка *А.Н. Золотаревой*

Сдано в набор 26.02.2007. Подписано в печать 05.04.2007. Формат 60 × 84 1/8. Бумага офсетная. Гарнитура Ариал.
Печать офсетная. Усл. печ. л. 7,90. Уч.-изд. л. 7,30. Тираж 200 экз. Зак. 306. С 3904.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru
Набрано во ФГУП «СТАНДАРТИНФОРМ» на ПЭВМ.
Отпечатано в филиале ФГУП «СТАНДАРТИНФОРМ» — тип. «Московский печатник», 105062 Москва, Лялин пер., 6.

gost.e-ecolog.ru