ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 18629-41 — 2011

Системы промышленной автоматизации и интеграция

ЯЗЫК СПЕЦИФИКАЦИЙ ПРОЦЕССА

Часть 41

Дефинициональные расширения: расширения действий

ISO 18629-41:2006
Industrial automation systems and integration — Process specification language — Part 41: Definitional extension: Activity extensions
(IDT)

Издание официальное

Москва
Стандартинформ
2014

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 Научно-техническим центром «ИНТЕК» на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 100 «Стратегический и инновационный менеджмент»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 22 декабря 2011 г. № 1610-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 18629-41:2006 «Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 41. Дефинициональные расширения: расширения действий» (ISO 18629-41:2006 «Industrial automation systems and integration — Process specification language — Part 41: Definitional extension: Activity extensions»).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «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 (gost.ru)

© Стандартинформ, 2014

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 . . . 2
3.1 Термины и определения . . . 2
3.2 Сокращения . . . 4
4 Общая информация об ИСО 18629 . . . 4
5 Структура настоящего стандарта . . . 5
6 Недетерминированные действия: перестановочная структура ветвей . . . 6
6.1 Примитивная лексика перестановочной структуры ветвей . . . 6
6.2 Определяемая лексика понятий перестановочной структуры ветвей . . . 6
6.3 Теории ядра, обусловленные перестановочной структурой ветвей . . . 6
6.4 Дефинициональные расширения, обусловленные перестановочной структурой ветвей . . . 6
6.5 Определения понятий для перестановочной структуры ветвей . . . 6
6.6 Грамматика соотношений перестановочной структуры ветвей . . . 7
7 Недетерминированные действия: складная структура ветвей . . . 8
7.1 Примитивная лексика складной структуры ветвей . . . 8
7.2 Определяемая лексика понятий для складной структуры ветвей . . . 8
7.3 Теории, обусловленные складной структурой ветвей . . . 8
7.4 Дефинициональные расширения, обусловленные складной структурой ветвей . . . 8
7.5 Определения складной структуры ветвей . . . 8
7.6 Грамматика описаний процесса для складной структуры ветвей . . . 9
8 Недетерминированные действия: структура ветвей и упорядочивание . . . 10
8.1 Примитивная лексика структуры ветвей и упорядочивания . . . 10
8.2 Определяемая лексика структуры ветвей и упорядочивания . . . 10
8.3 Теории, обусловленные структурой ветвей и упорядочиванием . . . 10
8.4 Дефинициональные расширения, обусловленные структурой ветвей и упорядочиванием . . . 10
8.5 Определения структуры ветвей и упорядочивания . . . 10
8.6 Грамматика структуры ветвей и упорядочивания . . . 12
9 Недетерминированные действия: повторяющаяся структура ветвей . . . 12
9.1 Примитивная лексика повторяющейся структуры ветвей . . . 12
9.2 Определяемые соотношения повторяющейся структуры ветвей . . . 12
9.3 Теории, обусловленные повторяющейся структурой ветвей . . . 13
9.4 Дефинициональные расширения, обусловленные повторяющейся структурой ветвей . . . 13
9.5 Определения повторяющейся структуры ветвей . . . 13
9.6 Грамматика повторяющейся структуры ветвей . . . 14
10 Спектр действий: перестановочные деревья действий . . . 15
10.1 Примитивная лексика перестановочных деревьев действий . . . 15
10.2 Определяемые соотношения перестановочных деревьев действий . . . 15
10.3 Теории, обусловленные перестановочными деревьями действий . . . 15
10.4 Дефинициональные расширения, обусловленные перестановочными деревьями действий . . . 15
10.5 Определения перестановочных деревьев действий . . . 15
10.6 Грамматика описаний процесса для перестановочных деревьев действий . . . 16
11 Спектр действий: уплотняющая структура ветвей . . . 16
11.1 Примитивная лексика уплотняющей структуры ветвей . . . 17
11.2 Определяемая лексика уплотняющей структуры ветвей . . . 17
11.3 Теории, обусловленные уплотняющей структурой ветвей . . . 17
11.4 Дефинициональные расширения, обусловленные уплотняющей структурой ветвей . . . 17
11.5 Определения уплотняющей структуры ветвей . . . 17
11.6 Грамматика уплотняющей структуры ветвей . . . 18
12 Спектр действий: деревья действий и переупорядочивание . . . 18
12.1 Примитивная лексика деревьев действий и переупорядочивания . . . 18
12.2 Определяемая лексика деревьев действий и переупорядочивания . . . 18

12.3 Теории, обусловленные деревьями действий и переупорядочиванием . . . 19
12.4 Дефинициональные расширения, обусловленные деревьями действий и переупорядочиванием . . . 19
12.5 Определения деревьев действий и переупорядочивания . . . 19
12.6 Грамматика деревьев действий и переупорядочивания . . . 20
13 Спектр и включение поддерева . . . 20
13.1 Примитивная лексика спектра и включения поддерева . . . 20
13.2 Определяемая лексика спектра и включения поддерева . . . 20
13.3 Теории, обусловленные спектром и включением поддерева . . . 20
13.4 Дефинициональные расширения, обусловленные спектром и включением поддерева . . . 20
13.5 Определения спектра и включения поддерева . . . 20
13.6 Грамматика перестановочной структуры ветвей . . . 21
14 Встраивающие ограничения для действий . . . 22
14.1 Примитивная лексика встраивающих ограничений для действий . . . 22
14.2 Определяемая лексика встраивающих ограничений для действий . . . 22
14.3 Теории, обусловленные встраивающими ограничениями для действий . . . 22
14.4 Дефинициональные расширения, обусловленные встраивающими ограничениями для действий . . . 22
14.5 Определения встраивающих ограничений для действий . . . 22
14.6 Грамматика встраивающих ограничений для действий . . . 24
15 Скелетные деревья действий . . . 24
15.1 Примитивная лексика скелетных деревьев действий . . . 24
15.2 Определяемая лексика скелетных деревьев действий . . . 24
15.3 Теории, обусловленные скелетными деревьями действий . . . 24
15.4 Дефинициональные расширения, обусловленные скелетными деревьями действий . . . 24
15.5 Определения скелетных деревьев действий . . . 24
15.6 Грамматика скелетных деревьев действий . . . 26
16 Неделимые действия: параллелизм вверх . . . 26
16.1 Примитивная лексика неделимых действий: параллелизм вверх . . . 26
16.2 Определяемая лексика неделимых действий: параллелизм вверх . . . 26
16.3 Теории, обусловленные неделимыми действиями: параллелизм вверх . . . 26
16.4 Дефинициональные расширения, обусловленные неделимыми действиями: параллелизм вверх 26
16.5 Определения неделимых действий: параллелизм вверх . . . 26
16.6 Грамматика неделимых действий: параллелизм вверх . . . 27
17 Неделимые действия: параллелизм вниз . . . 27
17.1 Примитивная лексика неделимых действий: параллелизм вниз . . . 27
17.2 Определяемая лексика неделимых действий: параллелизм вниз . . . 27
17.3 Теории, обусловленные неделимыми действиями: параллелизм вниз . . . 28
17.4 Дефинициональные расширения, обусловленные неделимыми действиями: параллелизм вниз . . . 28
17.5 Определения неделимых действий: параллелизм вниз . . . 28
17.6 Грамматика неделимых действий: параллелизм вниз . . . 29
18 Спектр неделимых действий . . . 29
18.1 Примитивная лексика спектра неделимых действий . . . 29
18.2 Определяемая лексика спектра неделимых действий . . . 29
18.3 Теории, обусловленные спектром неделимых действий . . . 29
18.4 Дефинициональные расширения, обусловленные спектром неделимых действий . . . 29
18.5 Определения спектра неделимых действий . . . 29
18.6 Грамматика спектра неделимых действий . . . 30
19 Входные условия для действий . . . 30
19.1 Примитивная лексика входных условий для действий . . . 30
19.2 Определяемая лексика входных условий для действий . . . 30
19.3 Теории, обусловленные входными условиями для действий . . . 30

19.4 Дефинициональные расширения, обусловленные спектром неделимых действий 30
19.5 Определения входных условий для действий 31
19.6 Грамматика входных условий для действий 31
Приложение А (справочное) ASN.1 Идентификатор настоящего стандарта 32
Приложение В (справочное) Пример описания технологического процесса в соответствии с настоящим стандартом 33
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации 40
Библиография 41

Введение

ИСО 18629 — это комплекс стандартов на компьютерно-интерпретируемый обмен данными обеспечения технологического процесса. Все части комплекса стандартов ИСО 18629 устанавливают групповой язык программирования для описания конкретного технологического процесса, рассматриваемого как часть всего процесса изготовления изделия либо внутри одной промышленной компании, либо сразу в нескольких промышленных секторах (компаниях) вне его связи с какой-либо моделью компьютерного представления. Природа данного языка программирования такова, что он обеспечивает доступ к спецификациям технологического процесса и технологическим данным изделия на всех стадиях процесса его изготовления.

В настоящем стандарте установлены описания дефинициональных расширений языка программирования, относящихся к расширениям действий в соответствии с ИСО 18629.

Все части комплекса ИСО 18629 не связаны с какой-либо конкретной моделью компьютерного представления технологического процесса. Все вместе указанные части ИСО 18629 обеспечивают структурную технологическую взаимосвязь процессов производства для улучшения оперативной совместимости рассматриваемых технических приложений.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

ЯЗЫК СПЕЦИФИКАЦИЙ ПРОЦЕССА

Часть 41
Дефинициональные расширения: расширения действий

Industrial automation systems and integration. Process specification language.
Part 41. Definitional extension. Activity extensions

Дата введения — 2012—09—01

1 Область применения

В настоящем стандарте определена спецификация непримитивных понятий языка с использованием набора определений, установленных в ИСО 18629. Данные определения обеспечивают аксиоматизацию семантики терминологии, приведенной в международных стандартах комплекса ИСО 18629.

Область применения настоящего стандарта распространяется на определения независимых от времени и состояния понятий (с использованием понятий, установленных в ИСО 18629-11 и ИСО 18629-12), связанных с классами действий в соответствии с разделом 5.

Область применения настоящего стандарта не распространяется на определения зависящих от времени и состояния понятий (с использованием понятий, установленных в ИСО 18629-11 и ИСО 18629-12).

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие международные стандарты, которые необходимо учитывать при использовании настоящего стандарта. В случае ссылок на документы, у которых указана дата утверждения, необходимо пользоваться только указанной редакцией. В случае, когда дата утверждения не приведена, следует пользоваться последней редакцией ссылочных документов, включая любые поправки и изменения к ним:

ИСО/МЭК 8824-1 Информационные технологии. Нотация абстрактного синтаксиса версии 1 (ASN.1). Часть 1. Спецификация базовой нотации (ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation)

ИСО 10303-1 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 1. Обзор и основные принципы (ISO 10303-1, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles)

ИСО 15531-1 Системы промышленной автоматизации и интеграция. Управляющая информация промышленным производством. Часть 1. Общий обзор (ISO 15531-1, Industrial automation systems and integration — Industrial manufacturing management data — Part 1: General overview)

ИСО 18629-1:2004 Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 1. Обзор и основные принципы (ISO 18629-1: 2004, Industrial automation systems and integration — Process specification language — Part 1: Overview and basic principles)

ИСО 18629-11:2005 Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 11. Ядро PSL (ISO 18629-11:2005, Industrial automation systems and integration — Process specification language — Part 11: PSL core)

ИСО 18629-12:2005 Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 12. Внешнее ядро (ISO 18629-12:2005, Industrial automation systems and integration — Process specification language — Part 12: Outer core)

Издание официальное

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

3.1.1 **аксиома** (axiom): Точно сформулированное аналитическое выражение на формальном языке, устанавливающее ограничения к интерпретации символов в словаре языка.

[ИСО 18629-1]

3.1.2 **данные** (data): Представление информации в формальном виде, подходящем для ее передачи, интерпретации или обработки людьми или на электронно-вычислительных машинах.

[ИСО 10303-1]

3.1.3 **установленная лексика** (defined lexicon): Набор символов в нелогической лексике, обозначающих установленные понятия.

Примечание — Установленная лексика состоит из констант, функций и отношений.

Пример — Термины с консервативным определением

[ИСО 18629-1]

3.1.4 **дефинициональное расширение** (definitional extension): Расширение ядра PSL, представляющее новые лингвистические понятия, которые могут быть определены с помощью терминов ядра PSL.

Примечание — Дефинициональные расширения не добавляют выразительную силу ядру PSL и используются для подробного описания семантики и терминологии в области применения.

[ИСО 18629-1]

3.1.5 **дискретное изготовление** (discrete manufacturing): Производство дискретной продукции.

Пример — Автомашины, приборы или компьютер.

[ИСО 15531-1]

3.1.6 **расширение** (extension): Расширение ядра PSL, содержащее дополнительные аксиомы.

Примечание 1 — Ядро PSL представляет собой относительно простой набор аксиом, достаточный для представления широкого круга основных процессов. Однако для представления более сложных процессов требуются дополнительные ресурсы, отсутствующие в ядре PSL. Ядро PSL с каждым понятием следует использовать для описаний того или иного процесса, а для описания разнообразных модульных расширений следует использовать расширение и дополнения ядра PSL. В этом случае пользователь может использовать такой язык, который соответствует требованиям к выразительности.

Примечание 2 — Все расширения являются теориями ядра или дефинициональными расширениями.

[ИСО 18629-1]

3.1.7 **грамматика** (grammar): Правила совместного использования логических символов и словарных терминов для составления точно сформулированных аналитических выражений.

[ИСО 18629-1]

3.1.8 **информация** (information): Факты, концепции или инструкции.

[ИСО 10303-1]

3.1.9 **язык** (language): Сочетание лексики и грамматики.

[ИСО 18629-1]

3.1.10 **производство** (manufacturing): Функция или действие, предусматривающие перевод или превращение материала из сырья или заготовки в завершенное состояние.

[ИСО 15531-1]

3.1.11 **производственный процесс** (manufacturing process): Структурированный комплекс видов деятельности или работ, выполняемых с материалом для перевода его из сырья или заготовки в завершенное состояние.

Примечание — Производственные процессы могут быть представлены в виде технологической схемы процесса, схемы движения продукта, в виде табличной схемы или схемы фиксированного расположения. К планируемым производственным процессам могут относиться изготовление продукта для складирования, на заказ и для сборки на заказ и т. д., основанным на стратегическом использовании и размещении материально-производственных запасов.

[ИСО 15531-1]

3.1.12 **модель** (model): Сочетание набора элементов и истинного назначения, удовлетворяющее всем правильно построенным формулировкам в теории.

Примечание 1 — В настоящем стандарте определение термина «модель» отличается от используемого в научной и другой литературе: если предложение является верным в определенной интерпретации, то можно сказать, что интерпретация – это модель предложения. Виды семантик, представленных в настоящем стандарте, часто называют теоретически смоделированными семантиками.

Примечание 2 — Модель обычно представляют в виде совокупности дополнительных структур (частично упорядоченных, в качестве структурного или векторного пространства). В этом случае модель определяет значения для терминологии и понятия истины для предложений языка в условиях данной модели. Задавая модель, основной набор аксиом математических структур, используемый в наборе аксиом, используют как основу для определения понятий, представленных в терминах языка, и их логических взаимосвязей, в результате чего набор моделей создает формальные семантики онтологии.

[ИСО 18629-1]

3.1.13 **онтология** (ontology): Лексика специализированной терминологии, дополненная необходимой спецификацией значений терминов.

Примечание 1 — Структурированный набор относительных терминов, представленный с описанием значений терминов на формальном языке. Описание значения объясняет, как и почему термины соотносятся, и определяет условия сегментирования и структурирования набора терминов.

Примечание 2 — Основополагающим компонентом языка технологических спецификаций ИСО 18629 является онтология. Примитивные концепции в онтологии, соответствующей определению ИСО 18629, достаточны для описания основных производственных, инженерных и бизнес-процессов.

Примечание 3 — Основное внимание онтологии направлено не только на термины, но и на их значения. Произвольный набор терминов включен в онтологию, но эти термины могут приниматься только в том случае, если их значения согласованы. Такие предполагаемые семантики терминов могут быть утверждены и использованы.

Примечание 4 — Любой термин, используемый без точного определения, может быть причиной неясности и путаницы. Сложность для онтологии в том, что структура нуждается в создании терминов, имеющих точное значение. Для онтологии, соответствующей определению ИСО 18629, необходимо предоставить математически строгую характеристику информационного процесса, а также четкое выражение основных логических свойств этой информации на языке, указанном в ИСО 18629.

Примечание 5 — Расширения включают аксиомы внешнего ядра (оболочка ядра базового языка).

[ИСО 18629-1]

3.1.14 **примитивная концепция** (primitive concept): Лексический термин, не имеющий консервативного определения.

[ИСО 18629-1]

3.1.15 **примитивная лексика** (primitive lexicon): Набор символов в нелогическом словаре, обозначающих элементарные понятия.

Примечание — Примитивная лексика включает в себя постоянные, функциональные и реляционные символы.

[ИСО 18629-1]

3.1.16 **процесс** (process): Структурированный ряд видов деятельности, включающий в себя различные сущности предприятия, предназначенный и организованный для достижения конкретной цели.

Примечание — Данное определение аналогично определению, приведенному в ИСО 10303-49. Тем не менее ИСО 15531 нуждается в понятии структурированного набора деятельностей без какого-либо предопределенного отношения ко времени или этапам. С точки зрения управления потоком некоторые свободные процессы могут требовать синхронизации в отношении цели, хотя в действительности они ничего не выполняют (задачи-призраки).

[ИСО 15531-1]

3.1.17 **продукт** (product): Изделие, материал или вещество, изготовленное в процессе производства.

[ИСО 10303-1]

3.1.18 **ресурс** (resource): Любые устройство, инструмент и средства, за исключением сырья и компонентов конечной продукции, имеющиеся в расположении предприятия для производства товаров и услуг.

Примечание 1 — Рассматриваемое понятие ресурса адаптировано по отношению к ИСО 15531-1. Понятие ресурса, введенное в ИСО 15531-1, не включает сырьевые материалы, продукты и компоненты, являющиеся (с точки зрения системной теории) элементами окружающей среды и, таким образом, не являющиеся частью системы. В настоящем стандарте данное допущение снято. Более того, определение, принятое в ИСО 15531-1, во многом использует определение, принятое в ИСО 10303-49, при этом оно включается в определение, принятое в настоящей части ИСО 18629. В дополнение к понятию ресурса, принятому в ИСО 15531, понятие ресурса, принятое в настоящем стандарте, включает сырьевые и расходуемые материалы в соответствии с ИСО 18629-14.

Примечание 2 — Ресурсы в соответствии с приведенным выше определением включают также рабочую силу, рассматриваемую как особое средство с заданными возможностями и заданной производительности. Указанные средства рассматриваются как целесообразные для использования в процессе производства на основании технического задания. Данное определение не включает какого-либо моделирования индивидуального или группового поведения человеческого ресурса, за исключением его способности выполнять заданную работу в процессе производства (например, преобразование сырого материала или полуфабриката, обеспечение логистических услуг и т. п.). Это означает, что человеческие ресурсы, как и другие, рассматриваются с точки зрения их функций, их возможностей и их состояния (например, занят, свободен). При этом исключается какое-либо моделирование или представление какого-либо аспекта индивидуального или группового социального поведения.

[ИСО 15531-1]

3.1.19 **теория** (theory): Набор аксиом и определений, относящийся к данному понятию или набору понятий.

Примечание — Данное определение отражает подход искусственного интеллекта, где теория — это набор предположений, на которых основано значение соответствующего понятия.

[ИСО 18629-1]

3.2 Сокращения

KIF — формат обмена знаниями — Knowledge Interchange Format.

4 Общая информация об ИСО 18629

Части с 41 по 49 комплекса международных стандартов ИСО 18629 определяют дефинициональные расширения, необходимые для формулировки точных определений и родственных аксиом непримитивных понятий ИСО 18629. Дефинициональные расширения определены ИСО 18629-11 и ИСО 18629-12, где введены новые элементы лексики. Данные элементы дефинициональных расширений могут быть полностью определены в соответствии с ИСО 18629-11 и ИСО 18629-12. Дефинициональные расширения дают точные семантические определения элементов, используемых в спецификациях индивидуальных технических приложений или типов технических приложений, при обеспечении совместных работ. Дефинициональные расширения существуют в следующих категориях:

- расширения действий;
- временные расширения и расширения, основанные на состоянии;
- упорядочивание действий и расширение продолжительности;
- назначения ресурса,
- наборы ресурсов;
- расширения действий процессора.

Индивидуальным (групповым) пользователям ИСО 18629 может потребоваться расширение ИСО 18629 для спецификации понятий, отсутствующих в настоящее время в частях с 41 по 49 комплекса международных стандартов ИСО 18629. Для этих целей они должны использовать элементы, определенные в ИСО 18629. Пользовательские расширения и их определения устанавливают дефинициональные расширения, которые не должны быть включены в части с 41 по 49 ИСО 18629.

Примечание — Пользовательские расширения должны удовлетворять требованиям ИСО 18629 и соответствовать ИСО 18629-1:2004 (подразделы 5.1 и 5.2).

Предметом рассмотрения частей с 41 по 49 комплекса стандартов ИСО 18629 являются:

- семантические определения (на основе понятий, установленных в ИСО 18629-11 и ИСО 18629-12), элементы которых являются характерными для шести понятий, определенных выше;
- набор аксиом, ограничивающих использование элементов в дефинициональных расширениях.

Предметом рассмотрения частей с 41 по 49 комплекса стандартов ИСО 18629 не являются:
- определения и аксиомы для понятий, определенных ИСО 18629-11 и ИСО 18629-12;
- элементы, не определенные в соответствии с ИСО 18629-11 и ИСО 18629-12;
- пользовательские расширения.

5 Структура настоящего стандарта

Фундаментальные теории, описанные в настоящем стандарте:
- недетерминированные действия: перестановочная структура ветвей;
- недетерминированные действия: складная структура ветвей;
- недетерминированные действия: структура ветвей и упорядочивание;
- недетерминированные действия: повторяющаяся структура ветвей;
- спектр действий: перестановочные деревья действий;
- спектр действий: уплотняющая структура ветвей;
- спектр действий: деревья действий и переупорядочивание;
- спектр и включение составляющего дерева;
- встраивающие ограничения для действий;
- скелетные деревья действий;
- неделимые действия: параллелизм вверх;
- неделимые действия: параллелизм вниз;
- спектр неделимых действий;
- входные условия для действий.

На рисунке 1 показаны соотношения между указанными расширениями и теориями ядра (см. ИСО 18629-12), используемыми для спецификации дефинициональных расширений в настоящем стандарте. Стрелки указывают на зависимости между расширениями настоящего стандарта и ИСО 18629-12. Все теории, определенные в настоящем стандарте, являются расширениями для ИСО 18629-12, который сам является расширением ИСО 18629-11.

Примечание — Полную диаграмму взаимозависимости между расширениями ИСО 18629-12 см. в ИСО 18629-12:2005 (подраздел 5.1).

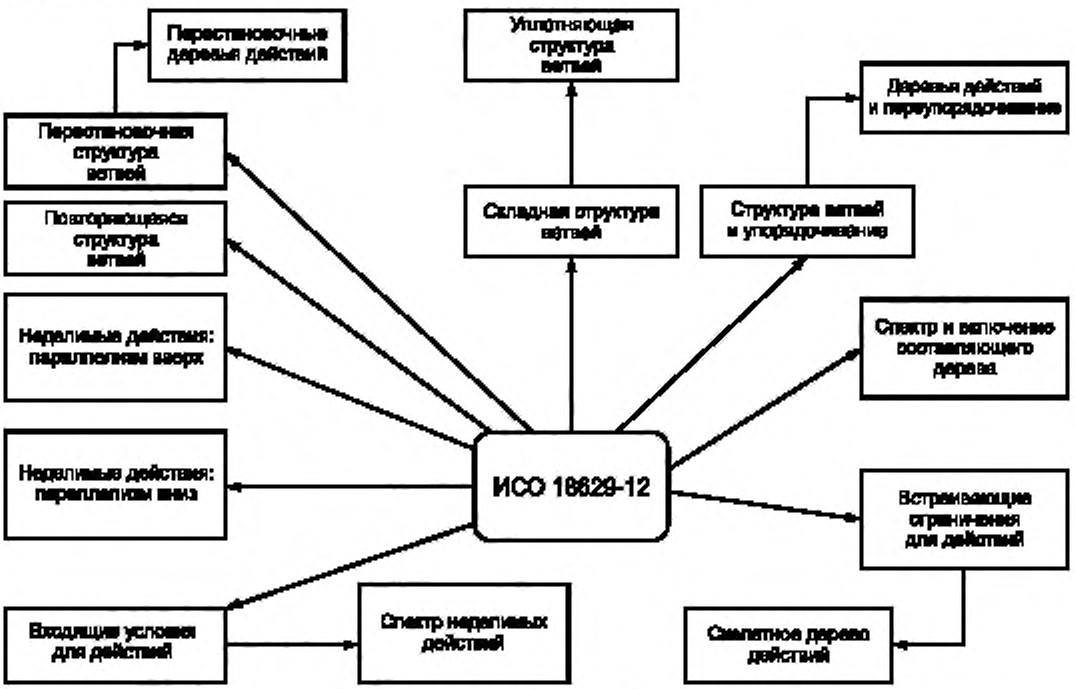

Рисунок 1 — Дефинициональные расширения ИСО 18629-41

6 Недетерминированные действия: перестановочная структура ветвей

Данный раздел характеризует все определения, обусловленные недетерминированными действиями: перестановочная структура ветвей.

6.1 Примитивная лексика перестановочной структуры ветвей

Лексика перестановочной структуры ветвей не требует примитивных соотношений.

6.2 Определяемая лексика понятий перестановочной структуры ветвей

В данном разделе определены нижеследующие соотношения:
- (branch_monomorphic ?a ?occ1 ?occ2);
- (branch_automorphic ?a ?occ1 ?occ2);
- (permuted ?a);
- (nondet_permuted ?a);
- (partial_permuted ?a);
- (simple ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

6.3 Теории ядра, обусловленные перестановочной структурой ветвей

Для данных расширений необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

6.4 Дефинициональные расширения, обусловленные перестановочной структурой ветвей

Для перестановочной структуры ветвей дефинициональные расширения не требуются.

6.5 Определения понятий для перестановочной структуры ветвей

Для перестановочной структуры ветвей определены нижеследующие понятия.

6.5.1 Branch_monomorphic

Одна ветвь минимального дерева действий является мономорфической для другой ветви тогда и только тогда, когда набор событий на одной ветви может быть встроен внутрь набора событий на другой ветви.

```
(forall (?occ1 ?occ2) (iff (branch_monomorphic ?occ1 ?occ2)
(forall (?s1 ?a)
    (implies (and (occurrence_of ?occ1 ?a)
                  (subactivity_occurrence ?s1 ?occ1))
             (exists (?s2)
                 (and (subactivity_occurrence ?s2 ?occ2)
                      (mono ?s1 ?s2 ?a)))))))
```

6.5.2 Branch_automorphic

Две ветви минимального дерева действий являются автоморфическими по отношению друг к другу тогда и только тогда, когда набор событий на одной ветви является перестановкой набора событий на другой ветви.

```
(forall (?occ1 ?occ2) (iff (branch_automorphic ?occ1 ?occ2)
(and (branch_monomorphic ?occ1 ?occ2)
     (branch_monomorphic ?occ2 ?occ1))))
```

6.5.3 Permuted

Событие является перестановочным тогда и только тогда, когда набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на всех прочих ветвях.

```
(forall (?occ) (iff (permuted ?occ)
(forall (?occ1 ?occ2)
```

```
(implies (and (same_grove ?occ1 ?occ)
        (same_grove ?occ2 ?occ))
    (branch_automorphic ?occ1 ?occ2))))))
```

6.5.4 Nondet_permuted

Событие является недетерминированно перестановочным тогда и только тогда, когда набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на некоторых других ветвях.

```
(forall (?occ) (iff (nondet_permuted ?occ)
(forall (?occ1)
    (implies (same_grove ?occ1 ?occ)
        (exists (?occ2)
            (and (same_grove ?occ1 ?occ2)
                (not (= ?occ1 ?occ2))
                (branch_automorphic ?occ1 ?occ2)))))))
```

6.5.5 Partial_permuted

Событие является частично перестановочным тогда и только тогда, когда существует ветвь, для которой набор событий является перестановкой набора событий на некоторых других ветвях. При этом существует ветвь, не являющаяся перестановкой какой-либо другой ветви.

```
(forall (?occ) (iff (partial_permuted ?occ)
(exist (?occ1 ?occ2 ?occ3)
    (and (same_grove ?occ1 ?occ)
        (same_grove ?occ2 ?occ)
        (not (= ?occ1 ?occ2))
        (branch_automorphic ?occ1 ?occ2)
        (same_grove ?occ3 ?occ)
        (forall (?occ4)
            (implies (and (same_grove ?occ3 ?occ4 ?a)
                    (branch_automorphic ?occ3 ?occ4))
                (= ?occ3 ?occ4)))))))))
```

6.5.6 Simple

Событие является простым тогда и только тогда, когда ни одна из ветвей какого-либо дерева действий не является автоморфической.

```
(forall (?occ) (iff (simple ?occ)
(forall (?occ1 ?occ2)
    (implies (and (same_grove ?occ1 ?occ)
            (same_grove ?occ2 ?occ)
            (branch_automorphic ?occ1 ?occ2))
        (= ?occ1 ?occ2)))))
```

6.6 Грамматика соотношений перестановочной структуры ветвей

Нижеследующие грамматические утверждения устанавливают описания технологического процесса и вспомогательные правила, определенные в KIF для перестановочной структуры ветвей.

Примечание — Назначение и важность использования грамматических утверждений ИСО 18629 поясняется в ИСО 18629-1:2004 (пункты 3.3.8, 4.2.4 и 5.1).

```
< permuted_spec > ::= (forall (< variable >)
            (implies (same_tree < variable > ?occ)
                < occurrence_sentence >))
< nondet_permuted_spec > ::= (forall (< variable >*)
            (implies (same_tree < variable > ?occ)
                (or < occurrence_sentence >*)))
< partial_permuted_spec > ::= (or < nondet_permuted_spec >
            < simple_spec >)
< simple_spec > ::= (exists (< variable >*)
            (and < occurrence_formula >
                < branch_spec >))
```

```
< occurrence_literal > ::= (occurrence < variable > < term >)
                (subactivity_occurrence < variable > < variable >)
< occurrence_formula > ::= < occurrence_literal > |
                (and < occurrence_literal >*)
< occurrence_sentence > := (exists (< variable >*)
                    < occurrence_formula >)
< branch_spec > ::= (and (same_tree < variable > ?occ)+
                (not (= < variable > ?occ))+)
```

7 Недетерминированные действия: складная структура ветвей

Данный раздел характеризует все определения, обусловленные недетерминированными действиями: складная структура ветвей.

7.1 Примитивная лексика складной структуры ветвей

Лексика складной структуры ветвей не требует примитивных соотношений.

7.2 Определяемая лексика понятий для складной структуры ветвей

В данном подразделе определены нижеследующие соотношения:
- (branch_homomorphic ?occ1 ?occ2);
- (folded ?a);
- (nondet_folded ?a);
- (partial_folded ?a);
- (rigid ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

7.3 Теории, обусловленные складной структурой ветвей

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

7.4 Дефинициональные расширения, обусловленные складной структурой ветвей

Для складной структуры ветвей дефинициональные расширения не требуются.

7.5 Определения складной структуры ветвей

Для складной структуры ветвей определены нижеследующие понятия.

7.5.1 Branch_homomorphic

Одна ветвь минимального дерева действий является гомоморфической для другой ветви тогда и только тогда, когда существует отображение набора событий на одной ветви внутрь заданного набора событий на другой ветви.

```
(forall (?occ1 ?occ2) (iff branch_homomorphic ?occ1 ?occ2)
(and (same_grove ?occ1 ?occ2)
     (forall (?s1 ?a)
          (implies (and (occurrence_of ?occ1 ?a)
                        (subactivity_occurrence ?s1 ?occ1))
                   (exists (?s2)
                           (subactivity_occurrence ?s2 ?occ2)
                           (hom ?s1 ?s2 ?a)))))))
```

7.5.2 Folded

Некоторое действие является сложенным тогда и только тогда, когда существует единственная ветвь, для которой набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на указанной единственной ветви.

```
(forall (?occ) (iff (folded ?occ)
(exists (?occ1)
        (and (same_grove ?occ1 ?occ)
                (forall (?occ2)
                        (implies (same_grove ?occ2 ?occ)
                                (branch_homomorphic ?occ1 ?occ2)))))))
```

7.5.3 Nondet_folded

Некоторое действие является недетерминированно сложенным тогда и только тогда, когда существует ветвь такая, что набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на первой ветви.

```
(forall (?occ) (iff (nondet_folded ?occ)
(forall (?occ1)
        (implies (same_grove ?occ1 ?occ)
                (exists (?occ2)
                        (and (same_grove ?occ2 ?occ)
                                (not (= ?occ1 ?occ2))
                                (branch_homomorphic ?occ1 ?occ2)))))))
```

7.5.4 Partial_folded

Некоторое действие является частично сложенным тогда и только тогда, когда существует ветвь такая, что набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на первой ветви.

```
(forall (?occ) (iff (partial_folded ?occ)
(exists (?occ1 ?occ2 ?occ3)
        (and (same_grove ?occ1 ?occ)
             (same_grove ?occ2 ?occ)
             (not (= ?occ1 ?occ2))
             (branch_homomorphic ?occ1 ?occ2)
             (same_grove ?occ3 ?occ)
             (forall (?occ4)
                     (implies (and (same_grove ?occ4 ?occ)
                                   (not (= ?occ3 ?occ4)))
                              (not (branch_homomorphic ?occ3 ?occ4)))))))))
```

7.5.5 Rigid

Некоторое действие является жестким тогда и только тогда, когда ни одна из ветвей некоторого дерева действий не является гомоморфической.

```
(forall (?occ) (iff (rigid ?occ)
(forall (?occ1 ?occ2)
        (implies (and (same_grove ?occ1 ?occ)
                      (same_grove ?occ2 ?occ)
                      (not (= ?occ1 ?occ2)))
                 (not (branch_homomorphic ?occ1 ?occ2))))))
```

7.6 Грамматика описаний процесса для складной структуры ветвей

Нижеследующие грамматические утверждения устанавливают описания технологического процесса и вспомогательные правила, определенные в KIF для складной структуры ветвей.

```
< folded_spec > ::= (exists (< variable >)
        (and (same_tree < variable > ?occ)
                < occurrence_axiom >))
< nondet_folded_spec > ::= (exists (< variable >)
        (and (same_tree < variable > ?occ)
                (or < occurrence_axiom >*)))
< partial_folded_spec > ::= (or < nondet_folded_spec >
                < rigid_spec >)
```

```
< rigid_spec > ::= (exists (< variable >*)
        (and < occurrence_formula >
                < branch_spec >))
< occurrence_disjunct > ::= < occurrence_literal > |
        (or < occurrence_literal >*)
< occurrence_axiom > ::= (exists (< variable >*)
                < occurrence_disjunct >)
```

8 Недетерминированные действия: структура ветвей и упорядочивание

Данный раздел характеризует все определения, обусловленные недетерминированными действиями: структура ветвей и упорядочивание.

8.1 Примитивная лексика структуры ветвей и упорядочивания

Лексика структуры ветвей и упорядочивания не требует примитивных соотношений.

8.2 Определяемая лексика структуры ветвей и упорядочивания

В данном подразделе определены нижеследующие соотношения:
- (order_tree ?s1 ?s2 ?a);
- (root_automorphic ?occ1 ?occ2);
- (ordered ?occ);
- (nondet_ordered ?occ);
- (broken_ordered ?occ);
- (ordered ?occ).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

8.3 Теории, обусловленные структурой ветвей и упорядочиванием

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

8.4 Дефинициональные расширения, обусловленные структурой ветвей и упорядочиванием

Для структуры ветвей и упорядочивания дефинициональные расширения не требуются.

8.5 Определения структуры ветвей и упорядочивания

Для структуры ветвей и упорядочивания определены нижеследующие понятия.

8.5.1 Mono_tree

Указанное поддерево, имеющее корень в ?s1, может быть мономорфически встроено внутрь поддерева, имеющего корень в ?s2.

```
(forall (?s1 ?s2 ?a) (iff (mono_tree ?s1 ?s2 ?a)
(forall (?s3 ?s4 ?s5)
        (implies (and (min_precedes ?s1 ?s3 ?a)
                (min_precedes ?s2 ?s4 ?a)
                (mono ?s3 ?s4 ?a)
                (min_precedes ?s3 ?s5 ?a))
            (exists (?s6)
                (and (mono ?s5 ?s6 ?a)
                        (min_precedes ?s4 ?s6 ?a)))))))
```

8.5.2 Order_tree

Указанное соотношение сохраняет значение, если существует некоторый автоморфизм порядка, отображающий поддерево с корнем в ?s1 в поддерево с корнем в ?s2.

```
(forall (?s1 ?s2 ?a) (iff (order_tree ?s1 ?s2 ?a)
(and (mono_tree ?s1 ?s2 ?a)
     (forall (?s3 ?s4 ?s5 ?s6)
          (implies (and (min_precedes ?s1 ?s3 ?a)
                        (min_precedes ?s2 ?s4 ?a)
                        (cousin ?s3 ?s4 ?a)
                        (min_precedes ?s3 ?s5 ?a)
                        (cousin ?s5 ?s6 ?a))
                   (iff (iso_occ ?s3 ?s5)
                        (iso_occ ?s4 ?s6)))))))
```

8.5.3 Root_automorphic

Два события на одном и том же дереве автоморфны корнями тогда и только тогда, когда для каждого события существует набор событий, являющихся корнями изоморфных поддеревьев.

```
(forall (?occ1 ?occ2) (iff (root_automorphic ?occ1 ?occ2)
(exists (?a ?s1 ?s2)
     (and (occurrence_of ?occ1 ?a)
          (occurrence_of ?occ2 ?a)
          (subactivity_occurrence ?s1 ?occ1)
          (subactivity_occurrence ?s2 ?occ2)
          (order_tree ?s1 ?s2 ?a)
          (order_tree ?s2 ?s1 ?a)))))
```

8.5.4 Ordered

Событие является упорядоченным тогда и только тогда, когда оно автоморфно корнями со всеми прочими корневыми подсобытиями на этом же дереве.

```
(forall (?occ1 ?occ2) (iff (root_automorphic ?occ1 ?occ2)
(exists (?a ?s1 ?s2)
     (and (occurrence_of ?occ1 ?a)
          (occurrence_of ?occ2 ?a)
          (subactivity_occurrence ?s1 ?occ1)
          (subactivity_occurrence ?s2 ?occ2)
          (order_tree ?s1 ?s2 ?a)
          (order_tree ?s2 ?s1 ?a)))))
```

8.5.5 Nondet_ordered

Событие является недетерминировано упорядоченным тогда и только тогда, когда каждое событие на дереве автоморфно корнями с некоторым другим корневым событием на дереве.

```
(forall (?occ1) (iff (nondet_ordered ?occ1)
(forall (?occ2)
     (implies (same_grove ?occ1 ?occ2)
          (exists (?occ3)
               (and (same_grove ?occ1 ?occ3)
                    (not (= ?occ3 ?occ2))
                    (root_automorphic ?occ2 ?occ3)))))))
```

8.5.6 Broken_ordered

Некоторое действие имеет сломанный (нарушенный) порядок тогда и только тогда, когда на одном и том же дереве существуют события, автоморфные корнями некоторым другим корневым подсобытиям на нем, а также существуют корневые события, неавтоморфные корнями каким-либо другим корневым подсобытиям.

```
(forall (?occ1) (iff (broken_ordered ?occ1)
(exists (?occ2)
     (and (same_grove ?occ1 ?occ2)
          (not (= ?occ1 ?occ2))
          (root_automorphic ?occ1 ?occ2)
          (forall (?occ3)
               (implies (and (same_grove ?occ3 ?occ1)
                             (not (= ?occ3 ?occ2))
                        (not (root_automorphic ?occ3 ?occ2)))))))))
```

8.5.7 Ordered

Событие является упорядоченным тогда и только тогда, когда ни одно из событий на этом же дереве не автоморфно с ним корнями.

```
(forall (?occ1) (iff (unordered ?occ1)
(forall (?occ2)
   (implies (and (same_grove ?occ1 ?occ2)
                 (not (= ?occ1 ?occ2)))
            (not (root_automorphic ?occ1 ?occ2))))))
```

8.6 Грамматика структуры ветвей и упорядочивания

Нижеследующие грамматические утверждения устанавливают описания технологического процесса и вспомогательные правила, определенные в KIF для структуры ветвей и упорядочивания.

```
< ordered_spec > ::= (forall (< variable >)
                       (implies (same_tree < variable > ?occ)
                                < ordered_sentence >)) |
                     (forall (< variable >)
                       (implies (same_tree < variable > ?occ)
                                < ordered_formula >))

< nondet_ordered_spec > ::= (forall (< variable >)
                              (implies (same_tree < variable > ?occ)
                                       (or < ordered_sentence >*)))

< broken_ordered_spec > ::= (forall (< variable >)
                              (implies (same_tree < variable > ?occ)
                                       (or {< ordered_sentence >* |
                                            < ordered_list >*})))

< unordered_spec > ::= (forall (< variable >)
                         (implies (same_tree < variable > ?occ)
                                  (or < ordered_list >*)))

< ordered_literal > ::= (min_precedes < variable > < variable > <term >) |
                        (next_subocc < variable > < variable > < term >)

< ordered_list > ::= < ordered_literal > |
                     (and < ordered_literal >*)

< conditional_occurrence > ::= (implies < occurrence_formula >
                                < ordered_list >)

< ordered_sentence > ::= (exists (< variable >)
                           (and (root_occ < variable > ?occ)
                                < occurrence_disjunct >
                                < conditional_occurrence >)))

< ordered_conjunct > ::= < ordered_literal > |
                         (and < ordered_formula >*)

< ordered_formula > ::=          (exists (< variable >)
                                   (and (root_occ < variable > ?occ)
                                        < occurrence_disjunct >
                                   < ordered_conjunct >)) |
                         (or < ordered_formula >*)
```

9 Недетерминированные действия: повторяющаяся структура ветвей

Данный раздел характеризует все определения, обусловленные недетерминированными действиями: повторяющаяся структура ветвей.

9.1 Примитивная лексика повторяющейся структуры ветвей

Лексика повторяющейся структуры ветвей не требует примитивных соотношений.

9.2 Определяемые соотношения повторяющейся структуры ветвей

В данном подразделе определены нижеследующие соотношения:

- (branch_mono ?s1 ?s2 ?s3 ?s4 ?a);
- (reptree ?s ?occ);
- (repetitive ?occ);
- (nondet_repetitive ?occ);
- (partial_repetitive ?occ);
- (amorphous ?occ).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

9.3 Теории, обусловленные повторяющейся структурой ветвей

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

9.4 Дефинициональные расширения, обусловленные повторяющейся структурой ветвей

Для повторяющейся структуры ветвей дефинициональные расширения не требуются.

9.5 Определения повторяющейся структуры ветвей

Для повторяющейся структуры ветвей определены нижеследующие понятия.

9.5.1 Branch_mono

Подветвь некоторого минимального действия с начальным событием ?s1 и заключительным событием ?s2 является событийно изоморфической для подветви с начальным событием ?s3 и заключительным событием ?s4.

```
(forall (?s1 ?s2 ?s3 ?s4 ?a) (iff (branch_mono ?s1 ?s2 ?s3 ?s4 ?a)
(forall (?s5 ?s6)
    (implies (and (min_precedes ?s1 ?s5 ?a)
                  (min_precedes ?s6 ?s2 ?a)
                  (next_subocc ?s5 ?s6 ?a))
             (exists (?s7 ?s8)
                (and (min_precedes ?s3 ?s7 ?a)
                     (min_precedes ?s8 ?s4 ?a)
                     (next_subocc ?s7 ?s8 ?a)
                     (iso_occ ?s5 ?s7)
                     (iso_occ ?s6 ?s8)))))))
```

9.5.2 Reptree

Каждое событие ?occ может быть встроено в некоторую ветвь поддерева с корневым событием ?s.

```
(forall (?s ?occ) (iff (reptree ?s ?occ)
(forall (?s1 ?a)
    (implies (and (occurrence ?occ ?a)
                  (subactivity_occurrence ?s1 ?occ))
             (exists (?s2 ?s3 ?s4 ?occ1)
                 (and (same_tree ?o ?o1)
                      (leaf_occ ?s2 ?occ1)
                      (<_min ?s3 ?s1 ?a)
                      (<_min ?s1 ?s4 ?a)
                      (branch_mono ?s3 ?s4 ?s ?s2 ?a)))))))
```

9.5.3 Repetitive

Событие является повторяющимся тогда и только тогда, когда каждая ветвь на этом же дереве действий может быть встроена в уникальное поддерево. Интуитивно ясно, что это эквивалентно существованию уникального повторяющегося поддерева.

```
(forall (?occ) (iff (repetitive ?occ)
(exists (?s1 ?occ2)
```

```
(and (same_tree ?occ ?occ2)
        (subactivity_occurrence ?s ?occ2)
        (forall (?occ1)
                (implies (same_tree ?occ ?occ1)
                        (and (reptree ?s ?occ1)
                                (not (root_occ ?s ?occ1))))))))
```

9.5.4 nondet_repetitive

Некоторое действие является недетерминированно повторяющимся тогда и только тогда, когда каждая ветвь на этом же дереве действий может быть встроена в некоторое поддерево. Интуитивно ясно, что это эквивалентно существованию нескольких неизоморфических повторяющихся поддеревьев.

```
(forall (?occ) (iff (nondet_repetitive ?occ)
(forall (?occ1)
        (implies (same_tree ?occ ?occ1)
                (exists (?s1 ?occ2)
                        (and (same_tree ?occ ?occ2)
                                (subactivity_occurrence ?s ?occ2)
                                (reptree ?s ?occ1)
                                (not (root_occ ?s ?occ1))))))))
```

9.5.5 partial_repetitive

Событие является частично повторяющимся тогда и только тогда, когда существуют повторяющиеся поддеревья. При этом существуют ветви на этом же дереве, которые не могут быть отображены на какое-либо повторяющееся поддерево.

```
(forall (?occ) (iff (partial_repetitive ?occ)
(and (exists (?occ1 ?occ2 ?s1)
        (and (same_tree ?occ ?occ1)
                (same_tree ?occ ?occ2)
                (subactivity_occurrence ?s ?occ2)
                (rep_tree ?s ?occ1)
                (not (root_occ ?s ?occ1))))
(exists (?occ2)
        (forall (?s2)
                (implies (reptree ?s1 ?occ1)
                        (root_occ ?s1 ?occ1)))))))
```

9.5.6 Amorphous

Событие является аморфным тогда и только тогда, когда не существует повторяющихся поддеревьев на этом же дереве действий.

```
(forall (?occ) (iff amorphous ?occ)
(forall (?occ1 ?s)
        (implies (and (same_tree ?occ1 ?occ)
                        (reptree ?s ?occ1))
                (root_occ ?s ?occ1)))))
```

9.6 Грамматика повторяющейся структуры ветвей

Нижеследующие грамматические утверждения устанавливают описания технологического процесса и вспомогательные правила, определенные в KIF для повторяющейся структуры ветвей.

```
< repetitive_spec > ::= (forall (< term > ?s1 ?s2)
        (iff (do < term > ?s1 ?s2)
                < rep_formula >))
< nondet_rep_spec > ::= (forall (< term > ?s1 ?s2)
        (iff (do < term > ?s1 ?s2)
                (or < rep_formula >*)))
< partial_rep_spec > ::= (forall (< term > ?s1 ?s2)
        (iff (do < term > ?s1 ?s2)
                < partial_rep_formula >
< rep_formula > ::= (exists (< term >)
```

```
        (and (subactivity < term > < term >)
                (forall (?s3)
                        (implies (do < term > ?s1 ?s3)
                                (or (= ?s2 ?s3)
                                        (do < term > ?s3 ?s2))))))
< partial_rep_formula > ::= (exists (< term > ?s3)
        (and (subactivity < term > < term >)
                (do < term > ?s1 ?s3)
                (or (= ?s2 ?s3)
                        (do < term > ?s3 ?s2))))
```

10 Спектр действий: перестановочные деревья действий

Данный раздел характеризует все определения, обусловленные спектром действий: перестановочное дерево действий.

10.1 Примитивная лексика перестановочных деревьев действий

Лексика перестановочных деревьев действий не требует примитивных соотношений.

10.2 Определяемые соотношения перестановочных деревьев действий

В данном подразделе определены нижеследующие соотношения:
- (reordered ?a);
- (partial_reordered ?a);
- (nondet_reorder ?a);
- (nonorderable ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

10.3 Теории, обусловленные перестановочными деревьями действий

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

10.4 Дефинициональные расширения, обусловленные перестановочными деревьями действий

Для данных расширений необходима перестановочная структура ветвей.

10.5 Определения перестановочных деревьев действий

Для перестановочных деревьев действий определены нижеследующие понятия.

10.5.1 Reordered

Некоторое действие является переупорядоченным тогда и только тогда, когда набор событий на каждой ветви одного минимального дерева действий является перестановкой набора событий на ветви другого дерева действий для ?a.

```
(forall (?a) (iff (reordered ?a)
(forall (?occ1 ?occ2)
        (implies (and (occurrence_of ?occ1 ?a)
                      (occurrence_of ?occ2 ?a))
                 (branch_automorphic ?occ1 ?occ2)))))
```

10.5.2 Nondet_reordered

Некоторое действие является недетерминированно переупорядоченным тогда и только тогда, когда набор событий на каждой ветви одного минимального дерева действий для ?a является перестановкой набора событий на некоторой ветви другого минимального дерева действий для ?a.

```
(forall (?a) (iff (nondet_reordered ?a)
(forall (?occ1)
        (implies (occurrence_of ?occ1 ?a)
                 (exists (?occ2)
                         (and (occurrence_of ?occ2 ?a)
                              (not (same_grove ?occ1 ?occ2))
                              (branch_automorphic ?occ1 ?occ2)))))))
```

10.5.3 Partial_reordered

Некоторое действие является частично переупорядоченным тогда и только тогда, когда существует ветвь такая, что ее набор событий является перестановкой набора событий на некоторой другой ветви. При этом существует некоторая ветвь, не являющаяся перестановкой событий на какой-либо другой ветви.

```
(forall (?a) (iff (partial_reordered ?a)
(exist (?occ1 ?occ2 ?occ3)
        (and (occurrence_of ?occ1 ?a)
             (occurrence_of ?occ2 ?a)
             (not (same_grove ?occ1 ?occ2))
             (branch_automorphic ?occ1 ?occ2)
             (occurrence_of ?occ3 ?a)
             (same_grove ?occ1 ?occ3)
             (forall (?occ4)
                     (implies (and (occurrence_of ?occ4 ?a)
                                   (not (= ?occ2 ?occ4))
                                   (same_grove ?occ2 ?occ4))
                              (not (branch_automorphic ?occ2 ?occ4)))))))))
```

10.5.4 Unorderable

Некоторое действие является неупорядоченным тогда и только тогда, когда ни одна из ветвей некоторого дерева действий не является автоморфической для какой-либо другой ветви на других деревьях указанного спектра действий.

```
(forall (?a) (iff (unorderable ?a)
(forall (?occ1 ?occ2)
        (implies (and (occurrence_of ?occ1 ?a)
                      (occurrence_of ?occ2 ?a)
                      (not (same_grove ?occ1 ?occ2)))
                 (not (branch_automorphic ?occ1 ?occ2))))))
```

10.6 Грамматика описаний процесса для перестановочных деревьев действий

Нижеследующие грамматические утверждения устанавливают описания технологического процесса, определенные в KIF для перестановочных деревьев действий (аксиомы перестановочных действий для ветвей).

```
< reordered_spec > ::= (forall (< variable >)
        (implies (occurrence < variable > < term >)
                 < occurrence_sentence >))

< nondet_reordered_spec > ::= (forall (< variable >*)
        (implies (occurrence < variable > < term >)
                 (or < occurrence_sentence >*)))

< partial_reordered_spec > ::= (or < nondet_reordered_spec >
        < nonorderable_spec >)
< nonorderable_spec > ::= (exists (< variable >*)
        (and < occurrence_formula >
             < branch_spec >))
```

11 Спектр действий: уплотняющая структура ветвей

Данный раздел характеризует все определения, обусловленные спектром действий: уплотняющая структура ветвей.

11.1 Примитивная лексика уплотняющей структуры ветвей

Лексика уплотняющей структуры ветвей не требует примитивных соотношений.

11.2 Определяемая лексика уплотняющей структуры ветвей

В данном подразделе определены нижеследующие соотношения:
- (compacted ?a);
- (nondet_compacted ?a);
- (partial_compacted ?a);
- (stiff ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

11.3 Теории, обусловленные уплотняющей структурой ветвей

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

11.4 Дефинициональные расширения, обусловленные уплотняющей структурой ветвей

Для данных расширений необходима перестановочная структура ветвей.

11.5 Определения уплотняющей структуры ветвей

Для уплотняющей структуры ветвей определены нижеследующие понятия.

11.5.1 Compacted

Некоторое действие является уплотненным тогда и только тогда, когда существует ветвь такая, что набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на его первой ветви.

```
(forall (?a) (iff (compacted ?a)
(exists (?occ1)
        (and (occurrence_of ?occ1 ?a)
                (forall (?occ2)
                        (implies (and (occurrence_of ?occ2 ?a)
                                (not (same_grove ?occ1 ?occ2)))
                                (branch_homomorphic ?occ1 ?occ2))))))))
```

11.5.2 Nondet_compacted

Некоторое действие является недетерминированно уплотненным тогда и только тогда, когда существует ветвь такая, что набор событий на каждой ветви минимального дерева действий является перестановкой набора событий на его первой ветви.

```
(forall (?a) (iff (nondet_compacted ?a)
(forall (?occ1)
        (implies (occurrence_of ?occ1 ?a)
                (exists (?occ2)
                        (and (occurrence_of ?occ2 ?a)
                                (not (same_grove ?occ1 ?occ2))
                                (branch_homomorphic ?occ1 ?occ2)))))))
```

11.5.3 Partial_compacted

Некоторое действие является частично уплотненным тогда и только тогда, когда существует ветвь такая, что набор событий на некоторой ветви (но не на всех) минимального дерева действий является перестановкой набора событий на его первой ветви.

```
                (forall (?a) (iff (partial_compacted ?a)
(exists (?occ1 ?occ2 ?occ3)
        (and (occurrence_of ?occ1 ?a)
            (occurrence_of ?occ2 ?a)
```

```
(not (same_grove ?occ1 ?occ2))
(branch_homomorphic ?occ1 ?occ2)
(occurrence_of ?occ3 ?a)
(same_grove ?occ3 ?occ1)
(forall (?occ4)
    (implies (and (occurrence_of ?occ4 ?a)
                  (not (= ?occ2 ?occ4))
                  (same_grove ?occ2 ?occ4))
             (not (branch_homomorphic ?occ2 ?occ4)))))))
```

11.5.4 Stiff

Некоторое действие ?a является несгибаемым тогда и только тогда, когда ни одна из ветвей некоторого дерева действий для ?a не является эпиморфической для ветвей какого-либо другого дерева действий для ?a.

```
(forall (?a) (if (stiff ?a)
(forall (?occ1 ?occ2)
    (implies (and (occurrence_of ?occ1 ?a)
                  (occurrence_of ?occ2 ?a)
                  (not (same_grove ?occ1 ?occ2)))
             (not (branch_homomorphic ?occ1 ?occ2))))))
```

11.6 Грамматика уплотняющей структуры ветвей

Нижеследующие грамматические утверждения устанавливают описания технологического процесса, определенные в KIF для уплотняющей структуры ветвей.

```
< compacted_spec > ::= (forall (< variable >)
    (implies (occurrence < variable > < term >)
             (exists (< variable >)
                     (and (occurrence < variable > < term >)
                          < occurrence_axiom >)))))

< nondet_compacted_spec > ::= (forall (< variable >)
  (implies (occurrence < variable > < term >)
      (exists (< variable >)
              (and (occurrence < variable > < term >)
                   (or < occurrence_axiom >*))))))

< partial_compacted_spec > ::= (or < nondet_compacted_spec >
                < stiff_spec >)

< stiff_spec > ::= (exists (< variable >*)
                (and < occurrence_formula >
                     < branch_spec >))
```

12 Спектр действий: деревья действий и переупорядочивание

Данный раздел характеризует все определения, обусловленные спектром действий: деревья действий и переупорядочивание.

12.1 Примитивная лексика деревьев действий и переупорядочивания

Лексика деревьев действий и переупорядочивания не требует примитивных соотношений.

12.2 Определяемая лексика деревьев действий и переупорядочивания

В данном подразделе определены нижеследующие соотношения:

- (treeordered ?a);
- (partial_treeordered ?a);
- (nondet_treeordered ?a);
- (scrambled ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

12.3 Теории, обусловленные деревьями действий и переупорядочиванием

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

12.4 Дефинициональные расширения, обусловленные деревьями действий и переупорядочиванием

Для данных расширений необходимы структура ветвей и упорядочивание.

12.5 Определения деревьев действий и переупорядочивания

Для деревьев действий и переупорядочивания определены нижеследующие понятия.

12.5.1 Treeordered

Некоторое действие упорядочено на дереве тогда и только тогда, когда для некоторого события одно корневое событие автоморфно корнями всем прочим корневым событиям для других событий данного действия.

```
(forall (?a) (iff (treeordered ?a)
(forall (?occ1 ?occ2)
    (implies (and (occurrence_of ?occ1 ?a)
            (occurrence_of ?occ2 ?a)
            (not (same_grove ?occ1 ?occ2))
            (root_automorphic ?occ1 ?occ2)))))
```

12.5.2 Nondet_treeordered

Событие недетерминированно упорядочено на дереве тогда и только тогда, когда каждое корневое событие минимального дерева действий автоморфно корнями некоторому другому корневому событию на дереве событий.

```
forall (?a) (iff (nondet_treeordered ?a)
(forall (?occ1)
    (implies (occurrence_of ?occ1 ?a)
        (exists (?occ2)
            (and (occurence_of ?occ2 ?a)
                (not (same_grove ?occ1 ?occ2))
                (root_automorphic ?occ1 ?occ2)))))))
```

12.5.3 Partial_treeordered

Некоторое действие частично упорядочено на дереве тогда и только тогда, когда существуют корневые события минимального дерева действий, автоморфные корнями некоторым другим корневым событиям на дереве событий. При этом существуют корневые события, не автоморфные корнями какому-либо другому корневому событию.

```
(forall (?a) (iff (partial_treeordered ?a)
(exists (?occ1 ?occ2)
    (and (occurrence_of ?occ1 ?a)
      (occurrence_of ?occ2 ?a)
        (not (same_grove ?occ1 ?occ2))
        (root_automorphic ?occ1 ?occ2)
        (forall (?occ3)
            (implies (and (occurrence_of ?occ3 ?a)
                (not (same_grove ?occ3 ?occ2)))
                (not (root_automorphic ?occ3 ?occ2)))))))))
```

12.5.4 Scrambled

Некоторое действие является зашифрованным тогда и только тогда, когда ни одна из ветвей некоторого дерева действий не автоморфна корнями какому-либо другому дереву действий.

```
(forall (?a) (iff (scrambled ?a)
(forall (?occ1 ?occ2)
    (implies (and (occurrence_of ?occ1 ?a)
                  (occurrence_of ?occ2 ?a)
                  (not (same_grove ?occ1 ?occ2)
                  (not (root_automorphic ?occ1 ?occ2))))))
```

12.6 Грамматика деревьев действий и переупорядочивания

Нижеследующие грамматические утверждения устанавливают описания технологического процесса, определенные в KIF для деревьев действий и переупорядочивания.

```
< treeordered_spec > ::= (forall (< variable >)
                  (implies (occurrence < variable > < term >)
                           < ordered_sentence >)) |
    (forall (< variable >)
                  (implies (occurrence < variable > < term >)
                           < ordered_formula >))
< nondet_treeordered_spec > ::= (forall (< variable >)
                  (implies (occurrence < variable > < term >)
                           (or < ordered_sentence >*)))
< partial_treeordered_spec > ::= (forall (< variable >)
                  (implies (occurrence < variable > < term >)
                           (or {< ordered_sentence >* |
                                < ordered_list >*})))
< scrambled_spec > ::= (forall (< variable >)
                  (implies (occurrence < variable > < term >)
                           (or < ordered_list >*)))
```

13 Спектр и включение поддерева

Данный раздел характеризует все определения, обусловленные спектром и включением поддерева.

13.1 Примитивная лексика спектра и включения поддерева

Лексика спектра и включения поддерева не требует примитивных соотношений.

13.2 Определяемая лексика спектра и включения поддерева

В данном подразделе определены нижеследующие соотношения:
- (subtree_embed ?occ1 ?occ2 ?a);
- (multiple_outcome ?a);
- (weak_outcome ?a);
- (nondet_outcome ?a);
- (immiscible ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

13.3 Теории, обусловленные спектром и включением поддерева

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

13.4 Дефинициональные расширения, обусловленные спектром и включением поддерева

Для спектра и включения поддерева дефинициональные расширения не требуются.

13.5 Определения спектра и включения поддерева

Для спектра и включения поддерева определены нижеследующие понятия.

13.5.1 Subtree_embed

Одно минимальное дерево действий для ?a может быть встроено в другое поддерево тогда и только тогда, когда какой-либо набор событий на дереве с корнями в ?s1 может быть отображен на какой-либо набор событий на дереве с корнями в ?s2 так, что упорядочивание сохраняется.

```
(forall (?s1 ?s2 ?a) (iff (subtree_embed ?s1 ?s2 ?a)
(forall (?s3)
    (implies (min_precedes ?s1 ?s3 ?a)
        (exists (?s4)
            (and (min_precedes ?s2 ?s4 ?a)
            (iso_occ ?s4 ?s3)))))))
```

13.5.2 Multiple_outcome

Некоторое действие является многозначным тогда и только тогда, когда какие-либо два дерева действий могут быть встроены одно в другое.

```
(forall (?a) (iff (multiple_outcome ?a)
(forall (?s1 ?s2)
    (implies (and (root ?s1 ?a)
        (root ?s2 ?a))
      (or (subtree_embed ?s1 ?s2 ?a)
        (subtree_embed ?s2 ?s1 ?a))))))
```

13.5.3 Weak_outcome

Некоторое действие ?a является слабым тогда и только тогда, когда существует минимальное дерево действий, которое может быть встроено во все прочие минимальные деревья действий для ?a.

```
(forall (?a) (iff (weak_outcome ?a)
(exists (?s1)
    (and (root ?s1 ?a)
        (forall (?s2)
            (implies (root ?s2 ?a)
                (subtree_embed ?s2 ?s1 ?a)))))))
```

13.5.4 Nondet_outcome

Некоторое действие ?a имеет недетерминированный результат тогда и только тогда, когда существует минимальное дерево действий, которое может быть встроено внутрь других минимальных деревьев действий для ?a. При этом существуют отдельные невстраиваемые ветви.

```
(forall (?a) (iff (nondet_outcome ?a)
(exists (?s1 ?s2 ?s3)
    (and (root ?s1 ?a)
        (root ?s2 ?a)
        (subtree_embed ?s2 ?s1 ?a)
        (root ?s3 ?a)
        (forall (?s4)
            (implies (root ?s4 ?a)
                (not (or (subtree_embed ?s3 ?s4 ?a)
                    (subtree_embed ?s4 ?s3 ?a)))))))))
```

13.5.5 immiscible

Некоторое действие является несмешиваемым тогда и только тогда, когда никакое из деревьев спектра не может быть встроено в какое-либо другое дерево.

```
(forall (?a) (iff (imiscible ?a)
(forall (?s1 ?s2)
    (implies (and (root ?s1 ?a)
        (root ?s2 ?a))
      (not (or (subtree_embed ?s1 ?s2 ?a)
        (subtree_embed ?s2 ?s1 ?a)))))))
```

13.6 Грамматика перестановочной структуры ветвей

Грамматика описаний процесса для перестановочной структуры ветвей эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

14 Встраивающие ограничения для действий

Данный раздел характеризует все определения, обусловленные встраивающими ограничениями для действий.

14.1 Примитивная лексика встраивающих ограничений для действий

Лексика встраивающих ограничений для действий не требует примитивных соотношений.

14.2 Определяемая лексика встраивающих ограничений для действий

В данном подразделе определены нижеследующие соотношения:
- (live_branch ?s1 ?s2 ?a);
- (embedded ?s1 ?s2 ?a);
- (dead_branch ?s1 ?s2 ?a);
- (dead_occurrence ?s1 ?s2 ?a);
- (embed_tree ?s1 ?s2 ?s3 ?a);
- (subocc_equiv ?s1 ?s2 ?s3 ?a);
- (unrestricted ?occ).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

14.3 Теории, обусловленные встраивающими ограничениями для действий

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

14.4 Дефинициональные расширения, обусловленные встраивающими ограничениями для действий

Для встраивающих ограничений для действий дефинициональные расширения не требуются.

14.5 Определения встраивающих ограничений для действий

Для встраивающих ограничений для действий определены нижеследующие понятия.

14.5.1 Live_branch

Событие ?s2 находится на живой ветви некоторого встроенного дерева действий для ?a с корнем ?s1 тогда и только тогда, когда оно является событием на указанной ветви.

```
(forall (?s1 ?s2 ?a) (iff (live_branch ?s1 ?s2 ?a)
(exists (?occ)
    (and (occurrence_of ?occ ?a)
        (root_occ ?s1 ?occ)
        (min_precedes ?s1 ?s2 ?a)))))
```

14.5.2 Embedded

Событие является элементом положительного максимального встроенного поддерева тогда и только тогда, когда оно является внешним событием, находящимся между двумя событиями на ветке дерева событий.

```
(forall (?s1 ?s2 ?a) (iff (embedded ?s1 ?s2 ?a)
(exists (?s3)
    (and (root ?s2 ?a)
        (min_precedes ?s2 ?s3 ?a)
        (precedes ?s2 ?s1)
        (precedes ?s1 ?s3)
        (not (min_precedes ?s1 ?s3 ?a)))))
```

14.5.3 Dead_branch

Некоторое событие ?s2 находится на мертвой ветви по отношению к некоторому дереву действий для ?a с корневым событием ?s1 тогда и только тогда, когда событие ?s2 находится на ветви, изоморфной некоторой ветви встроенного дерева действий. Событие ?s2 является самым ранним мертвым событием по отношению к дереву действий для ?a с корневым событием ?s1 тогда и только тогда, когда оно является последующим элементом события, не являющегося листом на дереве действий. При этом на дереве не существует событий, происходящих позже, чем ?s2.

```
(forall (?s1 ?s2 ?a) (iff (dead_branch ?s1 ?s2 ?a)
(exists (?a1 ?s3)
        (and (root ?s1 ?a)
                (min_precedes ?s1 ?s3 ?a)
                (not (leaf ?s3 ?a))
                (= ?s2 (successor ?a1 ?s3))
                (not (exists (?s4)
                        (and (precedes ?s3 ?s4)
                                (min_precedes ?s3 ?s4 ?a))))))))
```

14.5.4 Dead_occurence

Событие для внешнего действия ?s2 является мертвым событием по отношению к дереву действий ?a с корневым событием ?s1 тогда и только тогда, когда ?s2 находится между двумя событиями действий на мертвой ветви ?a.

```
(forall (?s1 ?s2 ?a) (iff (dead_occurrence ?s1 ?s2 ?a)
(exists (?s3)
        (and (dead_branch ?s1 ?s3 ?a)
                (precedes ?s1 ?s2)
                (precedes ?s2 ?s3)))))
```

14.5.5 Embed_tree

?s1 и ?s2 являются событиями поддействий ?a, которые являются элементами того же встроенного дерева событий для ?a с корнем ?s3.

```
(forall (?s1 ?s2 ?s3 ?a) (iff (embed_tree ?s1 ?s2 ?s3 ?a)
(exists (?a1 ?a2)
        (and (subactivity ?a1 ?a)
                (subactivity ?a2 ?a)
                (occurrence_of ?s1 ?a1)
                (occurrence_of ?s2 ?a2)
                (or (live_branch ?s3 ?s1 ?a)
                        (dead_branch ?s3 ?s1 ?a))
                (or (live_branch ?s3 ?s2 ?a)
                        (dead_branch ?s3 ?s2 ?a))))))
```

14.5.6 Subocc_equiv

Два события ?s1, ?s2 являются эквивалентными на уровне подсобытия в отношении ?a тогда и только тогда, когда они являются элементами того же максимально встроенного поддерева ?a с корневым событием ?s3, и они являются событиями поддействий и согласуются независимо от того, являются ли они событиями поддействий события ?a.

```
(forall (?s1 ?s2 ?s3 ?a) (iff (subocc_equiv ?s1 ?s2 ?s3 ?a)
(and (embed_tree ?s1 ?s2 ?s3 ?a)
(iff (min_precedes ?s3 ?s1 ?a)
        (min_precedes ?s3 ?s2 ?a)))))
```

14.5.7 Unrestricted

Событие ?occ является несвязанным тогда и только тогда, когда каждая ветвь максимального встроенного поддерева является живой.

```
(forall (?occ) (iff unrestricted ?occ)
(forall (?a ?s1 ?s2 ?s3)
        (implies (and (occurrence_of ?occ ?a)
                (root_occ ?s3 ?occ)
                (embed_tree ?s1 ?s2 ?s3 ?a))
            (subocc_equiv ?s1 ?s2 ?s3 ?a)))))
```

14.6 Грамматика встраивающих ограничений для действий

Грамматика описаний процесса для встраивающих ограничений для действий эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

15 Скелетные деревья действий

Данный раздел характеризует все определения, обусловленные скелетными деревьями действий.

15.1 Примитивная лексика скелетных деревьев действий

Лексика скелетных деревьев действий не требует примитивных соотношений.

15.2 Определяемая лексика скелетных деревьев действий

В данном подразделе определены нижеследующие соотношения:
- (fused ?occ);
- (embed_occ ?occ);
- (free ?occ);
- (assisted ?occ);
- (helpless ?occ);
- (unbound ?occ);
- (bound ?occ);
- (strict ?occ).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

15.3 Теории, обусловленные скелетными деревьями действий

Для данной теории необходимы следующие вспомогательные теории: act_occ.th; complex.th; atomic.th; subactivity.th; occtree.th; psl_core.th.

15.4 Дефинициональные расширения, обусловленные скелетными деревьями действий

Для данных расширений необходимы встраивающие ограничения для действий.

15.5 Определения скелетных деревьев действий

Для скелетных деревьев действий определены нижеследующие понятия.

15.5.1 Fused

Событие ?occ является расплавленным тогда и только тогда, когда каждое событие, расположенное между корневым и листовым событием, также является событием ?occ (но уже для поддействия).

```
(forall (?occ) (iff (fused ?occ)
(forall (?s1 ?s2 ?s3)
    (implies (and (root_occ ?s1 ?occ)
                  (subactivity_occurrence ?s2 ?occ)
                  (precedes ?s1 ?s3)
                  (precedes ?s3 ?s2))
             (subactivity_occurrence ?s3 ?occ)))))
```

15.5.2 Embedd_occ

Ветвь дерева событий, содержащая событие ?occ, может быть встроена в мертвую ветвь на том же максимальном встроенном поддереве.

```
(forall (?occ) (iff (embed_occ ?occ)
(forall (?a ?s ?s1 ?s2 ?s3)
(implies (and (occurrence_of ?occ ?a)
              (root_occ ?s ?occ)
              (next_subocc ?s1 ?s2 ?occ)
              (precedes ?s1 ?s3)
              (precedes ?s3 ?s2))
         (exists (?s4 ?s5 ?s6)
           (and (precedes ?s4 ?s5)
```

```
        (precedes ?s5 ?s6)
        (iso_occ ?s4 ?s1)
        (iso_occ ?s5 ?s3)
        (iso_occ ?s6 ?s3)
        (dead_branch ?s1 ?s4 ?a)
        (dead_branch ?s1 ?s6 ?a)))))))
```

15.5.3 Free

Событие является свободным тогда и только тогда, когда на этом же дереве существуют как расплавленные события, так и нерасплавленные события. Это все равно что не существуют необходимые внешние события.

```
(forall (?occ) (iff (free ?occ)
(exists (?occ1 ?occ2)
    (and (same_tree ?occ ?occ1)
        (same_tree ?occ ?occ2)
        (not (fused ?occ1))
        (fused ?occ2)))))
```

15.5.4 Assisted

Событие является ассистированным тогда и только тогда, когда каждое событие на этом же дереве не расплавлено. Это все равно что существуют необходимые внешние события.

```
    (forall (?occ) (iff (assisted ?occ)
  (forall (?occ1)
    (implies (same_tree ?occ ?occ1)
        (not (fused ?occ1))))))
```

15.5.5 Helpless

Некоторое действие является беспомощным тогда и только тогда, когда между какими-либо двумя событиями существует внешнее событие. Это все равно когда внешние действия всегда необходимы.

```
(forall (?occ) (iff (helpless ?occ)
(forall (?s1 ?s2)
    (implies (next_subocc ?s1 ?s2 ?occ)
        (exists (?s3)
            (and (precedes ?s1 ?s3)
                (precedes ?s3 ?s2)
                (not (subactivity_occurrence ?s3 ?occ))))))))
```

15.5.6 Unbound

Событие является несвязанным тогда и только тогда, когда ни одно из событий на этом же дереве не может быть встроено внутрь мертвой ветви. Это все равно когда запрещенные внешние действия отсутствуют.

```
(forall (?occ) (iff (unbound ?occ)
(forall (?occ1)
    (implies (same_tree ?occ1 ?occ)
        (not (embed_occ ?occ1))))))
```

15.5.7 Bound

Событие является связанным тогда и только тогда, когда на этом же дереве существуют события, которые могут быть встроены внутрь мертвой ветви. Это все равно что существуют запрещенные внешние действия.

```
(forall (?occ) (iff (bound ?occ)
(exists (?occ1 ?occ2)
    (and (same_tree ?occ1 ?occ)
        (not (embed_occ ?occ1))
        (same_tree ?occ2 ?occ)
        (embed_occ ?occ2)))))
```

15.5.8 Strict

Событие является строгим тогда и только тогда, когда все события на этом же дереве расплавлены. Это все равно что все внешние события запрещены.

```
(forall (?occ) (iff (strict ?occ)
(forall (?occ1)
        (implies (same_tree ?occ ?occ1)
                 (fused ?occ1)))))
```

15.6 Грамматика скелетных деревьев действий

Грамматика описаний процесса для скелетных деревьев действий эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

16 Неделимые действия: параллелизм вверх

Данный раздел характеризует все определения, обусловленные неделимыми действиями с параллелизмом вверх.

16.1 Примитивная лексика неделимых действий: параллелизм вверх

Лексика неделимых действий (параллелизм вверх) не требует примитивных соотношений.

16.2 Определяемая лексика неделимых действий: параллелизм вверх

В данном подразделе определены нижеследующие соотношения:
- (natural ?a ?s);
- (artificial ?a ?s);
- (performed ?a ?s);
- (up_ghost ?a ?s);
- (up_conflict ?a ?s);
- (quark ?a ?s).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

16.3 Теории, обусловленные неделимыми действиями: параллелизм вверх

Для данной теории необходимы вспомогательные теории: act_occ.th; complex.th; atomic.th; subactivity.th; occtree.th; psl_core.th.

16.4 Дефинициональные расширения, обусловленные неделимыми действиями: параллелизм вверх

Для неделимых действий (параллелизм вверх) дефинициональные расширения не требуются.

16.5 Определения неделимых действий: параллелизм вверх

Для неделимых действий (параллелизм вверх) определены нижеследующие понятия.

16.5.1 Natural

Некоторое действие является естественным тогда и только тогда, когда оно является поддействием для какого-либо действия, выполнение которого возможно когда-либо.

```
(forall (?a ?s) (iff (natural ?a ?s)
(forall (?a1)
        (implies (and (poss ?a ?s)
                      (poss ?a1 ?s))
                 (subactivity ?a ?a1)))))
```

16.5.2 Artificial

Некоторое действие является искусственным тогда и только тогда, когда его выполнение возможно совместно с каким-либо другим действием.

```
 (forall (?a ?s) (iff (artificial ?a ?s)
(forall (?a1)
 (implies (and (poss ?a ?s)
          (poss ?a1 ?s)
          (not (subactivity ?a1 ?a)))
         (poss (conc ?a ?a1) ?s)))))
```

16.5.3 Performed

Некоторое действие является выполняемым тогда и только тогда, когда существуют поддействия, невозможные для выполнения, если возможно указанное действие.

```
(forall (?a ?s) (iff (performed ?a ?s)
(exists (?a1 ?a2)
  (and (subactivity ?a1 ?a)
       (subactivity ?a2 ?a)
       (poss ?a ?s)
       (not (poss (conc ?a1 ?a2) ?s))))))
```

16.5.4 Up_ghost

Некоторое действие является паразитным тогда и только тогда, когда при возможности супердействия оно также возможно.

```
(forall (?a ?s) (iff (up_ghost ?a ?s)
(forall (?a1)
  (implies (and (poss ?a1 ?s)
       (subactivity ?a ?a1))
       (poss ?a ?s)))))
```

16.5.5 Up_conflict

Некоторое действие является конфликтным тогда и только тогда, когда при возможности его совместной композиции с каким-либо другим действием также возможно одно из их поддействий.

```
(forall (?a ?s) (iff (up_conflict ?a ?s)
(forall (?a1)
       (implies (poss (conc ?a ?a1) ?s)
              (or (poss ?a ?s)
                     (poss ?a1 ?s)
                     (subactivity ?a ?a1))))))
```

16.5.6 Quark

Некоторое действие является кварком тогда и только тогда, когда само оно невозможно в изоляции, однако возможны его супердействия.

```
(forall (?a ?s) (iff (quark ?a ?s)
(exists (?a1 ?s)
       (and (atomic ?a1)
       (subactivity ?a ?a1)
       (poss ?a1 ?s)
       (not (poss ?a ?s))))))
```

16.6 Грамматика неделимых действий: параллелизм вверх

Грамматика описаний процесса для неделимых действий (параллелизм вверх) эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

17 Неделимые действия: параллелизм вниз

Данный раздел характеризует все определения, обусловленные неделимыми действиями с параллелизмом вниз.

17.1 Примитивная лексика неделимых действий: параллелизм вниз

Лексика неделимых действий (параллелизм вниз) не требует примитивных соотношений.

17.2 Определяемая лексика неделимых действий: параллелизм вниз

В данном подразделе определены нижеследующие соотношения.

- (superpose ?a ?s);
- (assistance ?a ?s);
- (team ?a ?s);
- (ghost ?a ?s);

- (conflict ?a ?s);
- (dysfunction ?a ?s).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

17.3 Теории, обусловленные неделимыми действиями: параллелизм вниз

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

17.4 Дефинициональные расширения, обусловленные неделимыми действиями: параллелизм вниз

Для данного расширения дефинициональные расширения не требуются.

17.5 Определения неделимых действий: параллелизм вниз

Для неделимых действий (параллелизм вниз) определены нижеследующие понятия.

17.5.1 Superpose

Некоторое действие является суперпозицией тогда и только тогда, когда возможно каждое его поддействие, когда возможно само действие.

```
(forall (?a ?s) (iff (superpose ?a ?s)
(forall (?a1)
  (implies (and (subactivity ?a1 ?a)
       (poss ?a ?s))
      (poss ?a1 ?s)))))
```

17.5.2 Assistance

Некоторое действие является ассистирующим тогда и только тогда, когда оно возможно, возможна и любая совместная композиция его поддействий.

```
(forall (?a ?s) (iff (assistance ?a ?s)
(forall (?a1 ?a2)
  (implies (and (subactivity ?a1 ?a)
       (subactivity ?a2 ?a)
       (poss ?a ?s))
      (poss (conc ?a1 ?a2) ?s)))))
```

17.5.3 Team

Некоторое действие является командным тогда и только тогда, когда оно возможно, ни одно из его поддействий невозможно в изоляции.

```
(forall (?a ?s) (iff (team ?a ?s)
(exists (?a1 ?a2)
       (and (subactivity ?a1 ?a)
            (subactivity ?a2 ?a)
            (poss ?a ?s)
            (not (poss (conc ?a1 ?a2) ?s))))))
```

17.5.4 Ghost

Некоторое действие является паразитным тогда и только тогда, когда оно невозможно, ни одно из его поддействий также невозможно.

```
(forall (?a ?s) (iff (ghost ?a ?s)
(forall (?a1)
  (implies (and (subactivity ?a1 ?a)
       (poss ?a1 ?s))
      (poss ?a ?s)))))
```

17.5.5 Conflict

Некоторое действие является конфликтным при условии, что, как только оно невозможно, становятся возможны некоторые из его поддействий.

```
 (forall (?a ?s) (iff (conflict ?a ?s)
(forall (?a1 ?a2)
   (implies (and (subactivity ?a1 ?a)
         (subactivity ?a2 ?a)
         (not (poss ?a1 ?s))
         (poss (conc ?a1 ?a2) ?s))
        (poss ?a ?s))))))
```

17.5.6 Dysfunction

Некоторое действие является дисфункцией тогда и только тогда, когда существуют поддействия, которые по отдельности невозможны, но возможна их совместная композиция.

```
(forall (?a ?s) (iff (dysfunction ?a ?s)
(exists (?a1 ?a2)
         (and (subactivity ?a1 ?a)
                  (subactivity ?a2 ?a)
                  (not (poss ?a1 ?s))
                  (not (poss ?a ?s))
                  (poss (conc ?a1 ?a2) ?s)))))
```

17.6 Грамматика неделимых действий: параллелизм вниз

Грамматика описаний процесса для неделимых действий (параллелизм вниз) эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

18 Спектр неделимых действий

Данный раздел характеризует все определения, обусловленные спектром неделимых действий.

18.1 Примитивная лексика спектра неделимых действий

Лексика спектра неделимых действий не требует примитивных соотношений.

18.2 Определяемая лексика спектра неделимых действий

В данном подразделе определены нижеследующие соотношения:
- (global_ideal ?a);
- (global_nonideal ?a);
- (global_filter ?a);
- (global_nonfilter ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

18.3 Теории, обусловленные спектром неделимых действий

Для данной теории необходимы:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- occtree.th;
- psl_core.th.

18.4 Дефинициональные расширения, обусловленные спектром неделимых действий

Для данных расширений необходимы входные условия для действий.

18.5 Определения спектра неделимых действий

Определены нижеследующие понятия для спектра неделимых действий.

18.5.1 Global_ideal

Некоторое действие является глобальным идеальным фильтром тогда и только тогда, когда каждое супердействие имеет эквивалентные входные условия и указанное действие возможно.

```
(forall (?a) (iff (global_ideal ?a)
(forall (?a1 ?s1 ?s2)
```

```
(implies (and (subactivity ?a ?a1)
              (poss ?a ?s1)
              (poss ?a ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

18.5.2 global_nonideal

Некоторое действие является глобальным неидеальным фильтром тогда и только тогда, когда каждое супердействие имеет эквивалентные входные условия, когда указанное действие невозможно.

```
(forall (?a) (iff (global_nonideal ?a)
(forall (?a1 ?s1 ?s2)
  (implies (and (subactivity ?a ?a1)
                (not (poss ?a ?s1))
                (not (poss ?a ?s2)))
           (poss_equiv ?a1 ?s1 ?s2)))))
```

18.5.3 global_filter

Некоторое действие является глобальным фильтром тогда и только тогда, когда каждое поддействие имеет эквивалентные входные условия, когда указанное действие возможно.

```
(forall (?a) (iff (global_filter ?a)
(forall (?a1 ?s1 ?s2)
  (implies (and (subactivity ?a1 ?a)
                (poss ?a ?s1)
                (poss ?a ?s2))
           (poss_equiv ?a1 ?s1 ?s2)))))
```

18.5.4 global_nonfilter

Некоторое действие не является глобальным фильтром тогда и только тогда, когда каждое поддействие имеет эквивалентные входные условия, когда указанное действие невозможно.

```
(forall (?a) (iff (global_nonfilter ?a)
(forall (?a1 ?s1 ?s2)
  (implies (and (subactivity ?a1 ?a)
                (not (poss ?a ?s1))
                (not (poss ?a ?s2)))
           (poss_equiv ?a1 ?s1 ?s2)))))
```

18.6 Грамматика спектра неделимых действий

Грамматика описаний процесса для спектра неделимых действий эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

19 Входные условия для действий

Данный раздел характеризует все определения, обусловленные входными условиями для действий.

19.1 Примитивная лексика входных условий для действий

Лексика входных условий для действий не требует примитивных соотношений.

19.2 Определяемая лексика входных условий для действий

В данном подразделе определены нижеследующие соотношения:

- (poss_equiv ?a ?occ1 ?occ2);
- (trunc ?occ);
- (unconstrained ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

19.3 Теории, обусловленные входными условиями для действий

Для данной теории необходимы вспомогательные теории: act_occ.th; complex.th; atomic.th; subactivity.th; occtree.th; psl_core.th.

19.4 Дефинициональные расширения, обусловленные спектром неделимых действий

Данное расширение не требует каких-либо дефинициональных расширений.

19.5 Определения входных условий для действий

Для входных условий для действий определены нижеследующие понятия.

19.5.1 poss_equiv

Два события эквивалентны по входным условиям по отношению к некоторому действию, если они согласованы по допустимости указанных событий для данного действия.

```
(forall (?a ?occ1 ?occ2) (iff (poss_equiv ?a ?occ1 ?occ2)
implies (and (occurrence_of ?s1 ?a)
                (occurrence_of ?s2 ?a))
   (iff (legal ?occ1)
        (legal ?occ2)))))
```

19.5.2 trunc

Данное соотношение истинно, если событие ?occ1 является последующим элементом некоторого допустимого события.

```
(forall (?occ) (iff (trunc ?occ)
(forall (?occ1 ?a)
    (implies (= ?occ (successor ?a ?occ1))
            (legal ?occ1)))))
```

19.5.3 Unconstrained

Некоторое действие является безусловным, если все события на дереве событий эквивалентны по входным условиям.

```
(forall (?a) (iff (unconstrained ?a)
(forall (?s1 ?s2)
    (implies (and (activity_occurrence ?s1)
            (activity_occurrence ?s2))
        (poss_equiv ?a ?s1 ?s2)))))
```

19.6 Грамматика входных условий для действий

Грамматика описаний процесса для входных условий для действий эквивалентна грамматике утверждений общего описания технологического процесса в соответствии с ИСО 18629-11.

Приложение А
(справочное)

ASN.1 Идентификатор настоящего стандарта

Для однозначной идентификации информационного объекта в открытой системе настоящему стандарту присвоен следующий идентификатор:

iso standard 18629 part 41 version 1

Значение данного идентификатора определено в ИСО/МЭК 8824-1 и детально описано в ИСО 18629-1.

Приложение В
(справочное)

Пример описания технологического процесса в соответствии с настоящим стандартом

В данном приложении рассмотрен подробный сценарий использования языка спецификаций процесса PSL (Process Specification Language) в соответствии с ИСО 18629. Далее рассмотрен частный случай использования программного описания технологического процесса.

Данный сценарий включает совместное выполнение нескольких операций. Целью является повышение эффективности использования данных о технологическом процессе при изготовлении изделия. Подчеркнем, что данный язык программирования прежде всего позволяет повысить эффективность компьютерного обмена данными между сотрудниками планового отдела и руководством производственного подразделения.

В данном приложении рассмотрено расширение примера, использованного в ИСО 18629-11:2005 (приложение С). Пример иллюстрирует технические приложения внешних понятий для спецификации процесса изготовления изделия GT-350.

В.1 Процесс изготовления изделия GT-350

В данном разделе различные производственные процессы объединены в набор действий высокого уровня, необходимых для создания изделия GT-350. В соответствии с технологической картой изделия GT-350 (см. ИСО 18629-11:2005, таблица С.1) компоненты данного изделия либо покупают по контракту, либо изготовляют внутри самого предприятия. Рассматриваемые описания технологических процессов связаны с конкретными действиями, выполняемыми внутри предприятия для изготовления компонентов изделия. Данное рассмотрение технологического процесса в направлении «сверху — вниз» дает общую картину происходящего как комплексного действия по производству изделия GT-350 вплоть до составляющих его действий, выполняемых на уровне более мелких подразделений предприятия.

В соответствии с рисунком В.1 весь процесс изготовления изделия GT-350 организован в шести основных секторах. В первых пяти из них (изготовление интерьера, изготовление привода, изготовление кузова, изготовление двигателя и изготовление шасси) работы могут быть выполнены независимо друг от друга. Одно условие: они должны быть закончены к моменту начала общей сборки изделия.

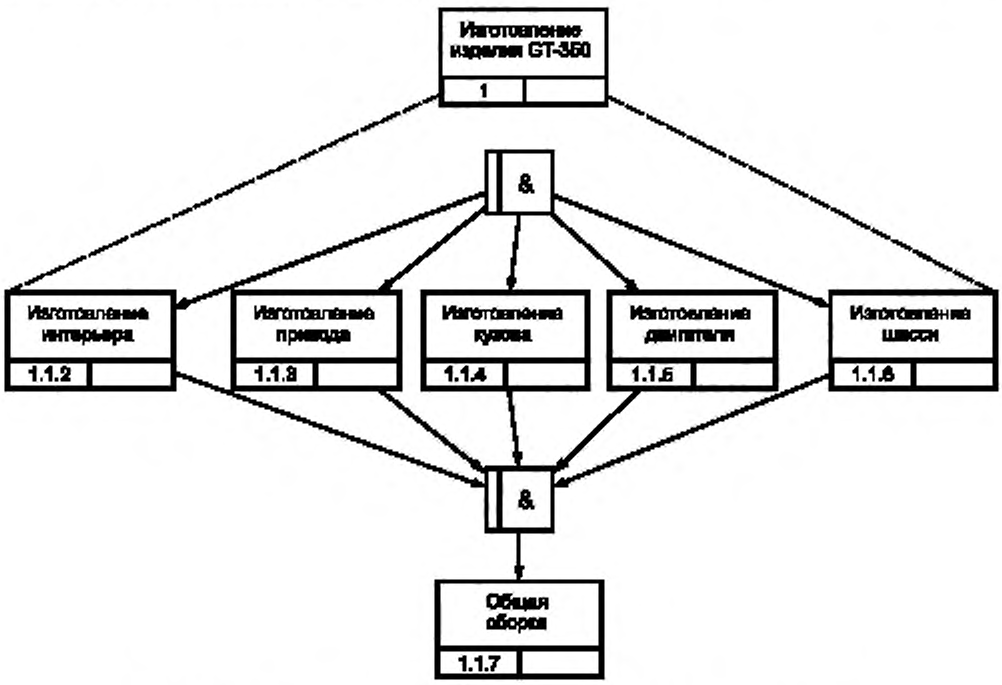

Рисунок В.1 — Верхний уровень процесса изготовления изделия GT-350 [5]

Представление верхнего уровня технологического процесса на языке PSL имеет вид:

```
(subactivity make-chassis make_gt350)
(subactivity make-interior make_gt350)
(subactivity make-drive make_gt350)
(subactivity make-trim make_gt350)
(subactivity make-engine make_gt350)
(subactivity final-assembly make_gt350)
(artificial make_chassis)
(artificial make_interior)
(artificial make_drive)
(artificial make_trim)
(artificial make_enngine)
(forall (?occ)
        (implies (occurrence_of ?occ make_gt350)
                 (permuted ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_gt350)
                 (ordered ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_gt350)
                 (folded ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_gt350)
                 (amorphous ?occ)))
(not (uniform make_gt350))
(unrestricted make_gt350)
(not (atomic make_gt350))
(forall (?occ)
        (iff (occurrence_of ?occ make_gt350)
                 (exists (?occ1 ?occ2 ?occ3 ?occ4 ?occ5 ?occ6)
                          (and (occurrence_of ?occ1 make_chassis)
                               (occurrence_of ?occ2 make_interior)
                               (occurrence_of ?occ3 make_drive)
                               (occurrence_of ?occ4 make_trim)
                               (occurrence_of ?occ5 make_engine)
                               (occurrence_of ?occ6 final_assembly)
                               (subactivity_occurrence ?occ1 ?occ)
                               (subactivity_occurrence ?occ2 ?occ)
                               (subactivity_occurrence ?occ3 ?occ)
                               (subactivity_occurrence ?occ4 ?occ)
                               (subactivity_occurrence ?occ5 ?occ)
                               (subactivity_occurrence ?occ6 ?occ)
(forall (?s1 ?s2 ?s3 ?s4 ?s5 ?s6)
                                    (implies (and (leaf_occ ?s1 ?occ1)
                                                  (leaf_occ ?s2 ?occ2)
                                                  (leaf_occ ?s3 ?occ3)
                                                  (leaf_occ ?s4 ?occ4)
                                                  (leaf_occ ?s5 ?occ5)
                                                  (root_occ ?s6 ?occ6))
                                             (and (min_precedes ?s1 ?s6 make_gt350)
                                                  (min_precedes ?s2 ?s6 make_gt350)
                                                  (min_precedes ?s3 ?s6 make_gt350)
                                                  (min_precedes ?s4 ?s6 make_gt350)
                                                  (min_precedes ?s5 ?s6 make_gt350) ))))))
```

Данное представление формализует технологический процесс в соответствии с рисунком В.1.

Все события, заключающиеся в выполнении действия «make_gt350», являются перестановочными, так как все поддействия имеют место, как только выполняется действие «make_gt350».

Все события для действия «make_gt350» являются упорядоченными, так как его поддействия «make_chassis» (изготовление шасси), «make_interior» (изготовление интерьера), «make_drive» (изготовление привода), «make_engine» (изготовление двигателя) и «make_trim» (изготовление шасси) должны иметь место до поддействия «final_assembly» (заключительная сборка).

Любое поддействие может происходить совместно с каким-либо другим поддействием. Таким образом, все неделимые поддействия являются искусственными, а все события, заключающиеся в выполнении действия «make_gt350» (изготовление изделия GT-350), являются сложенными.

Внутри действия «make_gt350» итераций нет, поэтому все события аморфны.

Поддействие «make_engine» (изготовление двигателя) не выполняется, если двигатель уже существует. Таким образом, не все события для действия «make_gt350» содержат события с одинаковым набором поддействий и с одинаковыми ограничениями упорядочивания. Поэтому действие «make_gt350» не является однородным.

Не существует внешних событий (заключающихся в выполнении действия), необходимых или запрещенных по отношению к действию «make_gt350». Следовательно, оно является несвязанным.

Каждое указанное абстрактное действие может быть рассмотрено в деталях. Однако в примере данного приложения все эти действия не рассматриваются.

На базе представления IDEF3 (в терминах представления технологического процесса) для краткого описания действий, встречающихся на различных стадиях процесса изготовления изделия, в настоящем стандарте приведены некоторые примеры использования языка программирования PSL-Outercore в соответствии с ИСО 18629-12.

B.2 Абстрактное действие «make_engine» (изготовление двигателя)

Двигатель изделия GT-350 собирается из агрегатов, изготовленных в нескольких подразделениях предприятия. Схема процесса изготовления дана на рисунке B.2. Агрегат состоит из двигательного блока, жгутов и кабелей. Составляющие процессы детально рассмотрены в подразделах ниже. Двигатель изделия GT-350 собирается на сборочном стенде A004. Сборка одного двигателя требует 5 мин.

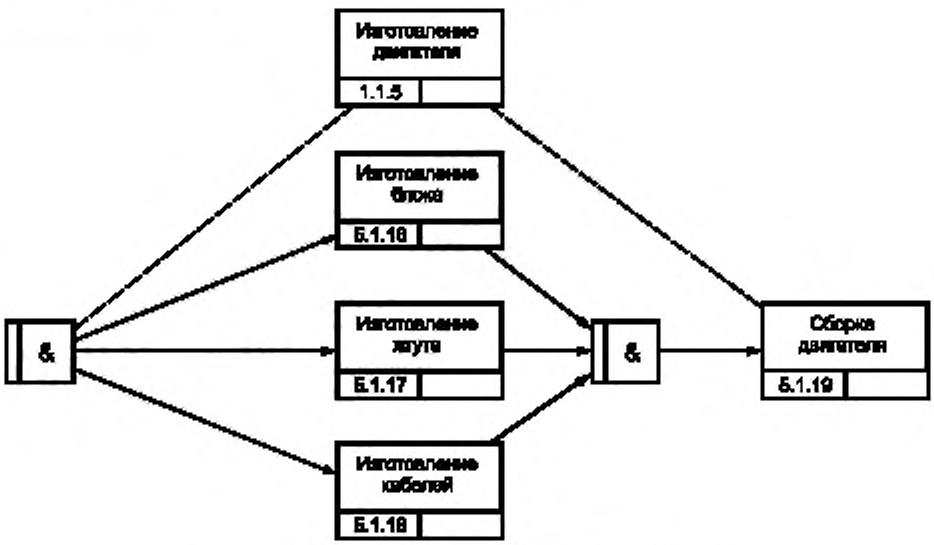

Рисунок B.2 — Процесс изготовления двигателя GT-350 [5]

Ниже представлены некоторые действия и данные технологического процесса изготовления двигателя на языке PSL:

```
(subactivity make_block make_engine)
(subactivity make-harness make_engine)
(subactivity make-wires make_engine)
(subactivity assemble_engine make_engine)
(artificial make_block)
(artificial make_harness)
(artificial make_wires)
(forall (?occ)
        (implies (occurrence_of ?occ make_engine)
                 (permuted ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_engine)
                 (ordered ?occ)))
```

```
(forall (?occ)
        (implies (occurrence_of ?occ make_engine)
                 (folded ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_engine)
                 (amorphous ?occ)))
(uniform make_engine)
(unrestricted make_engine)
(not (atomic make_engine))
(forall (?occ)
        (iff (occurrence_of ?occ make_engine)
             (exists (?occ1 ?occ2 ?occ3 ?occ4)
                     (and (occurrence_of ?occ1 make_block)
                          (occurrence_of ?occ2 make_harness)
                          (occurrence_of ?occ3 make_wires)
                          (occurrence_of ?occ4 assemble_engine)
                          (subactivity_occurrence ?occ1 ?occ)
                          (subactivity_occurrence ?occ2 ?occ)
                          (subactivity_occurrence ?occ3 ?occ)
                          (subactivity_occurrence ?occ4 ?occ)
                          (forall (?s1 ?s2 ?s3 ?s4)
                            (implies (and (leaf_occ ?s1 ?occ1)
                                          (leaf_occ ?s2 ?occ2)
                                          (leaf_occ ?s3 ?occ3)
                                                (root_occ ?s4 ?occ4))
                                          (and (min_precedes ?s1 ?s4 make_engine)
                                                (min_precedes ?s2 ?s4 make_engine)
                                                (min_precedes ?s3 ?s4 make_engine)))))))
```

Данное представление формализует технологический процесс на рисунке В.2.

Все события действия «make_engine» являются перестановочными, так как все поддействия имеют место, как только действие «make_engine» имеет место.

Все события действия «make_engine» упорядочены, так как поддействия «make_block» (изготовление блока), «make_harness» (изготовление жгута) и «make_wires» (изготовление проводов) должны закончиться до поддействия «assemble_engine» (сборка двигателя).

Любое поддействие может происходить совместно с каким-либо другим поддействием. Таким образом, все неделимые поддействия являются искусственными, а все события действия «make_engine» являются сложенными.

Внутри действия «make_engine» итераций нет, поэтому все события аморфны.

Каждое событие действия «make_engine» содержит события с тем же набором поддействий и с теми же упорядочивающими ограничениями. Таким образом, действие «make_gt350» однородно. Оно может иметь место в любое время и при любых условиях.

Отсутствуют внешние события, необходимые или запрещенные по отношению к действию «make_engine». Следовательно, оно является несвязанным.

В.3 Действие «make_block» (изготовление блока)

Блок изделия GT-350 выполняется как агрегат для сборки двигателя изделия GT-350. Изготовление блока требует выполнения всех технологических операций, начиная от литья заготовки и ее механической обработки (см. рисунок В.3).

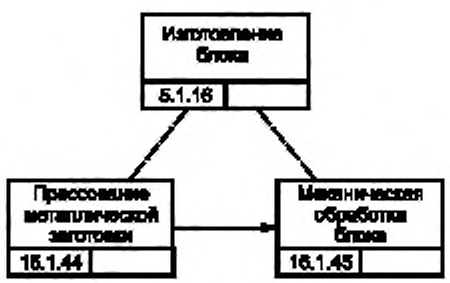

Рисунок В.3 — Процесс изготовления блока изделия GT-350

Представление некоторых действий и технологических данных на языке PSL:

```
(subactivity produce_molded_metal make_block)
(subactivity machine_block make_block)
(primitive machine_block)
(primitive produce_molded_metal)
(performed machine_block)
(performed produce_molded_metal)
(forall (?occ)
        (implies (occurrence_of ?occ make_block)
                 (permuted ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_block)
                 (ordered ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_block)
                 (rigid ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_block)
                 (amorphous ?occ)))
(uniform make_block)
(unrestricted make_block)
(not (atomic make_block))
(forall (?occ)
        (iff (occurrence_of ?occ make_block)
                 (exists (?occ1 ?occ2)
                          (and (occurrence_of ?occ1 produce_molded_metal)
                                   (occurrence_of ?occ2 machine_block)
                                   (min_precedes ?occ1 ?occ2 make_block)))))
```

Данное представление формализует технологический процесс на рисунке В.3.

Все события действия «make_block» являются перестановочными, так как все поддействия имеют место, как только действие «make_block» имеет место.

Все события действия «make_engine» упорядочены, так как поддействие «produce_molded_metal» (прессование металлической заготовки) должно иметь место до поддействия «machine_block» (механическая обработка блока).

Ни одно из указанных поддействий не может происходить совместно с каким-либо другим поддействием. Таким образом, все неделимые поддействия являются выполнимыми, а все события действия «make_block» являются жесткими.

Внутри действия «make_block» итерации отсутствуют. Следовательно, все события аморфны.

Любое событие действия «make_block» содержит события с тем же набором поддействий и с теми же упорядочивающими ограничениями. Таким образом, действие «make_block» является однородным.

Отсутствуют внешние события, необходимые или запрещенные по отношению к действию «make_block». Следовательно, оно является несвязанным.

В.4 Действие «make_harness» (изготовление жгута)

Жгут изделия GT-350 (см. рисунок В.4) изготовляют как сборочный агрегат двигателя изделия GT-350. Данный технологический процесс организован в цехе кабелей и проводов. Рисунок В.5 представляет процесс изготовления провода жгута. Жгут изделия GT-350 собирают на особом стенде из проводов и кабелей. Сборка одного жгута требует 10 мин.



Рисунок В.4 — Процесс изготовления жгута изделия GT-350

Ниже дано представление некоторых действий и соответствующих технологических данных на языке PSL:

```
(subactivity make_harness_wire make_harness)
(subactivity assemble_harness make_harness)
(primitive assemble_harness)
(performed make_harness_wire)
(performed assemble_harness)
(forall (?occ)
        (implies (occurrence_of ?occ make_harness)
                 (permuted ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_harness)
                 (ordered ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_harness)
                 (rigid ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_harness)
                 (amorphous ?occ)))
(uniform make_harness)
(unrestricted make_harness)
(not (atomic make_harness))
(forall (?occ)
        (iff (occurrence_of ?occ make_harness)
             (exists (?occ1 ?occ2 ?occ3)
                     (and (occurrence_of ?occ1 make_harness_wire)
                          (occurrence_of ?occ2 assemble_harness)
                          (leaf_occ ?occ3 ?occ1)
                          (min_precedes ?occ3 ?occ2 make_harness)))))
```

Данное представление формализует технологический процесс на рисунке В.5.

Все события действия «make_harness» (изготовление жгута) являются перестановочными, так как все поддействия имеют место, как только действие «make_harness» имеет место.

Все события действия «make_harness» упорядочены, так как поддействие «make_harness_wire» (изготовление провода жгута) выполняется до действия (assemble_harness) сборка жгута.

Ни одно из поддействий не может выполняться совместно с какими-либо другими поддействиями. Таким образом, все неделимые поддействия являются выполнимыми, а все события действия «make_harness» являются жесткими.

Так как внутри действия «make_harness» итераций нет, то все события являются аморфными.

Каждое событие действия «make_harness» содержит события с тем же набором поддействий и с теми же упорядочивающими ограничениями. Таким образом, действие «make_harness» является однородным.

Отсутствуют внешние события, необходимые или запрещенные по отношению к действию «make_harness». Следовательно, оно является несвязанным.

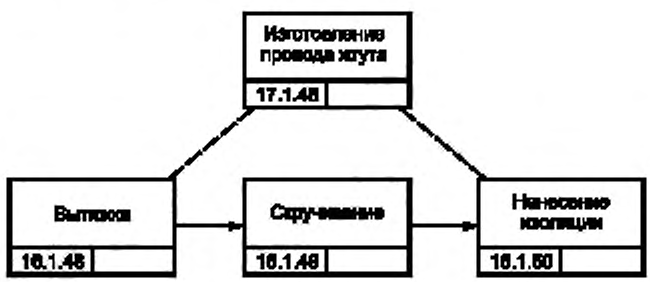

Рисунок В.5 — Процесс изготовления провода жгута

В.5 Действие «make_harness_wire» (изготовление провода жгута)

Набор проводов изделия GT-350 изготовляют как сборочный агрегат изделия GT-350. Технологический процесс организуют в цехе проводов и кабелей.

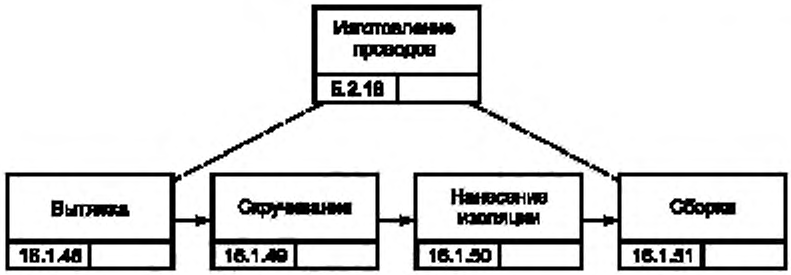

Рисунок В.6 — Процесс изготовления проводов изделия GT-350

Ниже дано представление некоторых действий и соответствующих технологических данных на языке программирования PSL:

```
(subactivity extrude make_harness_wire)
(subactivity twist make_harness_wire)
(subactivity jacket make_harness_wire)
(primitive extrude)
(primitive twist)
(primitive jacket)
(performed extrude)
(performed twist)
(performed jacket)
(forall (?occ)
        (implies (occurrence_of ?occ make_harness_wire)
                 (permuted ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_harness_wire)
                 (ordered ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_harness_wire)
                 (rigid ?occ)))
(forall (?occ)
        (implies (occurrence_of ?occ make_harness)
                 (amorphous ?occ)))
(uniform make_harness_wire)
(not (unrestricted make_harness_wire))
(not (atomic make_harness_wire))
(forall (?occ)
        (iff (occurrence_of ?occ make_harness_wire)
                 (exists (?occ1 ?occ2 ?occ3)
                          (and (occurrence_of ?occ1 extrude)
                               (occurrence_of ?occ2 twist)
                               (occurrence_of ?occ3 jacket)
                               (min_precedes ?occ1 ?occ2 make_harness_wire)
                               (min_precedes ?occ2 ?occ3 make_harness_wire))))
```

Данное представление формализует технологический процесс на рисунке В.5.

Все события действия «make_harness_wires» являются перестановочными, так как все поддействия имеют место, как только действие «make_harness_wires» имеет место.

Все события действия «make_harness_wires» упорядочены, так как поддействия «extrude» (вытяжка), «twist» (скручивание) и «jacket» (нанесение изоляции) должны происходить в заданном порядке.

Ни одно из поддействий не может происходить совместно с каким-либо другим поддействием. Таким образом, все неделимые поддействия являются выполнимыми, а все события действия «make_harness_wires» являются жесткими.

Внутри действия «make_harness_wires» итерации отсутствуют, поэтому все события являются аморфными.

Каждое событие действия «make_harness_wires» содержит события с тем же набором поддействий и с теми же упорядочивающими ограничениями. Таким образом, действие «make_harness_wires» является однородным.

Установка и переключение действий должно происходить между какими-либо двумя событиями при выполнении действия «make_harness». Следовательно, оно является несвязанным.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8824-1	IDT	ГОСТ Р ИСО/МЭК 8824-1—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации»
ИСО 10303-1	IDT	ГОСТ Р ИСО 10303-1—99 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы»
ИСО15531-1	IDT	ГОСТ Р ИСО 15531-1—2008 «Промышленные автоматизированные системы и интеграция. Данные по управлению промышленным производством. Часть 1. Общий обзор»
ИСО 18629-1:2004	—	*
ИСО 18629-11:2005	—	*
ИСО 18629-12:2005	—	*

* Соответствующий национальный стандарт отсутствует (в разработке). До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:

- IDT — идентичные стандарты.

Библиография

[1] ИСО 10303-49 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 49. Интегрированные групповые ресурсы: структура и свойства процесса

[2] ИСО 18629-14 Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 14. Теории ресурсов

[3] ИСО 18629-44 Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 44. Дефинициональные расширения: расширения ресурсов

УДК 65.011:56.681.3 ОКС 25.040.40 Т58

Ключевые слова: автоматизированные промышленные системы, интеграция, жизненный цикл систем, управление производством

Редактор *А. Д. Чайка*
Технический редактор *В. Н. Прусакова*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *В. Н. Романовой*

Сдано в набор 28.01.2014. Подписано в печать 31.03.2014. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 5,58. Уч.-изд. л. 4,09. Тираж 71 экз. Зак. 157

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru gost.e-ecolog.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru