ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 18629-42— 2011

Системы промышленной автоматизации и интеграция

ЯЗЫК СПЕЦИФИКАЦИЙ ПРОЦЕССА

Часть 42

Дефинициональные расширения: временные расширения и расширения состояния

ISO 18629-42:2006
Industrial automation systems and integration — Process specification language — Part 42: Definitional extension: Temporal and state extensions

(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

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. № 1611-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 18629-42:2006 «Системы промышленной автоматизации и интеграция. Язык спецификаций процесса. Часть 42. Дефинициональные расширения: временные расширения и расширения состояния» (ISO 18629-42:2006 «Industrial automation systems and integration — Process specification language — Part 42: Definitional extension: Temporal and state 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

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). 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
5.1 Введение . . . 5
5.2 Расширения, определенные в настоящем стандарте . . . 5
6 Основанные на состоянии входные условия выполнения действий . . . 5
6.1 Примитивная лексика основанных на состоянии входных условий выполнения действий . . . 5
6.2 Описываемая лексика понятий для основанных на состоянии входных условий выполнения действий . . . 5
6.3 Теории ядра, обусловленные основанными на состоянии входными условиями выполнения действий . . . 6
6.4 Дефинициональные расширения, обусловленные основанными на состоянии входными условиями выполнения действий . . . 6
6.5 Определения понятий для основанных на состоянии входных условий выполнения действий 6
6.6 Грамматика соотношений для основанных на состоянии входных условий выполнения действий . . . 7
7 Временные входные условия выполнения действий . . . 7
7.1 Примитивная лексика временных входных условий выполнения действий . . . 7
7.2 Описываемая лексика понятий для временных входных условий выполнения действий . . . 7
7.3 Теории, обусловленные временными входными условиями выполнения действий . . . 8
7.4 Дефинициональные расширения, обусловленные временными входными условиями выполнения действий . . . 8
7.5 Определения временных входных условий выполнения действий . . . 8
7.6 Грамматика описаний процесса для временных входных условий выполнения действий . . . 8
8 Временные и основанные на состоянии входные условия . . . 9
8.1 Примитивная лексика временных и основанных на состоянии входных условий . . . 9
8.2 Описываемая лексика временных и основанных на состоянии входных условий . . . 9
8.3 Теории, обусловленные временными и основанными на состоянии входными условиями . . . 9
8.4 Дефинициональные расширения, обусловленные временными и основанными на состоянии входными условиями . . . 9
8.5 Определения временных и основанных на состоянии входных условий . . . 9
8.6 Грамматика временных и основанных на состоянии входных условий . . . 10
9 Событийные входные условия выполнения действий . . . 11
9.1 Примитивная лексика событийных входных условий выполнения действий . . . 11
9.2 Описываемые соотношения событийных входных условий выполнения действий . . . 11
9.3 Теории, обусловленные событийными входными условиями выполнения действий . . . 11
9.4 Дефинициональные расширения, обусловленные событийными входными условиями выполнения действий . . . 11
9.5 Определения событийных входных условий выполнения действий . . . 11
9.6 Грамматика событийных входных условий выполнения действий . . . 12
10 Предотвращаемые условия выполнения действий . . . 12
10.1 Примитивная лексика предотвращаемых условий выполнения действий . . . 12
10.2 Описываемые соотношения предотвращаемых условий выполнения действий . . . 12
10.3 Теории, обусловленные предотвращаемыми условиями выполнения действий . . . 13
10.4 Дефинициональные расширения, обусловленные предотвращаемыми условиями выполнения действий . . . 13
10.5 Определения предотвращаемых условий выполнения действий . . . 13
10.6 Грамматика описаний процесса для предотвращаемых условий выполнения действий . . . 14

11 Периодические входные условия выполнения действий . . . 14
11.1 Примитивная лексика периодических входных условий выполнения действий . . . 14
11.2 Описываемая лексика периодических входных условий выполнения действий . . . 14
11.3 Теории, обусловленные периодическими входными условиями выполнения действий . . . 14
11.4 Дефинициональные расширения, обусловленные периодическими входными условиями выполнения действий . . . 14
11.5 Определения периодических входных условий выполнения действий . . . 14
11.6 Грамматика периодических входных условий выполнения действий . . . 15
12 Входные условия нарушения выполнения действий . . . 15
12.1 Примитивная лексика входных условий нарушения выполнения действий . . . 15
12.2 Описываемая лексика входных условий нарушения выполнения действий . . . 15
12.3 Теории, обусловленные входными условиями нарушения выполнения действий . . . 15
12.4 Дефинициональные расширения, обусловленные входными условиями нарушения выполнения действий . . . 16
12.5 Определения входных условий нарушения выполнения действий . . . 16
12.6 Грамматика входных условий нарушения выполнения действий . . . 17
13 Эффекты действий . . . 17
13.1 Примитивная лексика эффектов действий . . . 17
13.2 Описываемая лексика эффектов действий . . . 17
13.3 Теории, обусловленные эффектами действий . . . 17
13.4 Дефинициональные расширения, обусловленные эффектами действий . . . 17
13.5 Определения эффектов действий . . . 17
13.6 Грамматика эффектов действий . . . 18
14 Основанные на состоянии эффекты действий . . . 18
14.1 Примитивная лексика основанных на состоянии эффектов действий . . . 18
14.2 Описываемая лексика основанных на состоянии эффектов действий . . . 18
14.3 Теории ядра, обусловленные основанными на состоянии эффектами действий . . . 18
14.4 Дефинициональные расширения, обусловленные основанными на состоянии эффектами действий . . . 18
14.5 Определения основанных на состоянии эффектов действий . . . 18
14.6 Грамматика основанных на состоянии эффектов действий . . . 19
15 Временные эффекты действий . . . 19
15.1 Примитивная лексика временных эффектов действий . . . 19
15.2 Описываемая лексика временных эффектов действий . . . 19
15.3 Теории ядра, обусловленные временными эффектами действий . . . 19
15.4 Дефинициональные расширения, обусловленные временными эффектами действий . . . 19
15.5 Определения временных эффектов действий . . . 20
15.6 Грамматика временных эффектов действий . . . 20
16 Событийные эффекты действий . . . 20
16.1 Примитивная лексика событийных эффектов действий . . . 20
16.2 Описываемая лексика событийных эффектов действий . . . 20
16.3 Теории ядра, обусловленные событийными эффектами действий . . . 21
16.4 Дефинициональные расширения, обусловленные событийными эффектами действий . . . 21
16.5 Определения событийных эффектов действий . . . 21
16.6 Грамматика событийных эффектов действий . . . 21
17 Эффекты действий: событийные ограничения . . . 22
17.1 Примитивная лексика эффектов действий: событийные ограничения . . . 22
17.2 Описываемая лексика эффектов действий: событийные ограничения . . . 22
17.3 Теории ядра, обусловленные эффектами действий: событийные ограничения . . . 22
17.4 Дефинициональные расширения, обусловленные эффектами действий: событийные ограничения . . . 22
17.5 Определения эффектов действий: событийные ограничения . . . 22
17.6 Грамматика эффектов действий: событийные ограничения . . . 23
18 Эффекты действий: временные и событийные ограничения . . . 23
18.1 Примитивная лексика эффектов действий: временные и событийные ограничения . . . 23
18.2 Описываемая лексика эффектов действий: временные и событийные ограничения . . . 23

18.3 Теории ядра, обусловленные эффектами действий: временные и событийные ограничения . 23
18.4 Дефинициональные расширения, обусловленные эффектами действий: временные и событийные ограничения . . . 23
18.5 Определения эффектов действий: временные и событийные ограничения . . . 23
18.6 Грамматика эффектов действий: временные и событийные ограничения . . . 24
19 Дерево флюента . . . 24
19.1 Примитивная лексика дерева флюента . . . 24
19.2 Описываемые соотношения дерева флюента . . . 24
19.3 Теории ядра, обусловленные деревом флюента . . . 25
19.4 Дефинициональные расширения, обусловленные деревом флюента . . . 25
19.5 Определения дерева флюента . . . 25
19.6 Грамматика описаний процесса для дерева флюента . . . 25
20 Распределение комплексных действий . . . 25
20.1 Примитивная лексика распределения комплексных действий . . . 26
20.2 Описываемые соотношения распределения комплексных действий . . . 26
20.3 Теории ядра, обусловленные распределением комплексных действий . . . 26
20.4 Дефинициональные расширения, обусловленные распределением комплексных действий . 26
20.5 Определения распределения комплексных действий . . . 26
20.6 Грамматика описаний процесса распределения комплексных действий . . . 27
21 Основанное на состоянии распределение комплексных действий . . . 27
21.1 Примитивная лексика основанного на состоянии распределения комплексных действий . . 27
21.2 Описываемые соотношения основанного на состоянии распределения комплексных действий 28
21.3 Теории, обусловленные основанным на состоянии распределением комплексных действий . 28
21.4 Дефинициональные расширения, обусловленные основанным на состоянии распределением комплексных действий . . . 28
21.5 Определения основанного на состоянии распределения комплексных действий . . . 28
21.6 Грамматика описаний процесса для основанного на состоянии распределения комплексных действий . . . 29
22 Временное распределение комплексных действий . . . 29
22.1 Примитивная лексика временного распределения комплексных действий . . . 29
22.2 Описываемые соотношения временного распределения комплексных действий . . . 29
22.3 Теории, обусловленные временным распределением комплексных действий . . . 29
22.4 Дефинициональные расширения, обусловленные временным распределением комплексных действий . . . 29
22.5 Определения временного распределения комплексных действий . . . 29
22.6 Грамматика описаний процесса для временного распределения комплексных действий . . . 30
23 Смешанное распределение комплексных действий . . . 30
23.1 Примитивная лексика смешанного распределения комплексных действий . . . 30
23.2 Описываемые соотношения смешанного распределения комплексных действий . . . 30
23.3 Теории ядра, обусловленные смешанным распределением комплексных действий . . . 30
23.4 Дефинициональные расширения, обусловленные смешанным распределением комплексных действий . . . 31
23.5 Определения смешанного распределения комплексных действий . . . 31
23.6 Грамматика описаний процесса для смешанного распределения комплексных действий . . 31
24 Вариация комплексных действий . . . 32
24.1 Примитивная лексика вариации комплексных действий . . . 32
24.2 Описываемые соотношения вариации комплексных действий . . . 32
24.3 Теории, обусловленные вариацией комплексных действий . . . 32
24.4 Дефинициональные расширения, обусловленные вариацией комплексных действий . . . 32
24.5 Определения вариации комплексных действий . . . 32
24.6 Грамматика описаний процесса для вариации комплексных действий . . . 33
25 Основанная на состоянии вариации комплексных действий . . . 33
25.1 Примитивная лексика основанной на состоянии вариации комплексных действий . . . 33
25.2 Описываемые соотношения основанной на состоянии вариации комплексных действий . . . 33
25.3 Теории, обусловленные основанной на состоянии вариации комплексных действий . . . 34

25.4 Дефинициональные расширения, обусловленные основанной на состоянии вариации комплексных действий . . . 34
25.5 Определения основанной на состоянии вариации комплексных действий . . . 34
25.6 Грамматика описаний процесса для основанной на состоянии вариации комплексных действий . . . 35
26 Временная вариация комплексных действий . . . 35
26.1 Примитивная лексика временной вариации комплексных действий . . . 35
26.2 Описываемые соотношения временной вариации комплексных действий . . . 35
26.3 Теории, обусловленные временной вариацией комплексных действий . . . 35
26.4 Дефинициональные расширения, обусловленные временной вариацией комплексных действий . . . 35
26.5 Определения временной вариации комплексных действий . . . 35
26.6 Грамматика описаний процесса для временной вариации комплексных действий . . . 36
27 Смешанная вариация комплексных действий . . . 36
27.1 Примитивная лексика смешанной вариации комплексных действий . . . 36
27.2 Описываемые соотношения смешанной вариации комплексных действий . . . 36
27.3 Теории, обусловленные смешанной вариацией комплексных действий . . . 37
27.4 Дефинициональные расширения, обусловленные смешанной вариацией комплексных действий . . . 37
27.5 Определения смешанной вариации комплексных действий . . . 37
27.6 Грамматика описаний процесса для смешанной вариации комплексных действий . . . 38
28 Встроенные действия: планы . . . 38
28.1 Примитивая лексика встроенных действий: планы . . . 38
28.2 Описываемые соотношения встроенных действий: планы . . . 38
28.3 Теории, обусловленные встроенными действиями: планы . . . 38
28.4 Дефинициональные расширения, обусловленные встроенными действиями: планы . . . 38
28.5 Определения встроенных действий: планы . . . 38
28.6 Грамматика описаний процесса для встроенных действий: планы . . . 39
29 Встроенные действия: временной разброс . . . 39
29.1 Примитивная лексика встроенных действий: временной разброс . . . 39
29.2 Описываемые соотношения встроенных действий: временной разброс . . . 39
29.3 Теории, обусловленные встроенными действиями: временной разброс . . . 40
29.4 Дефинициональные расширения, обусловленные встроенными действиями: временной разброс . . . 40
29.5 Определения встроенных действий: временной разброс . . . 40
29.6 Грамматика описаний процесса для встроенных действий: временной разброс . . . 41
30 Вариация неделимых вверх действий . . . 41
30.1 Примитивная лексика вариации неделимых вверх действий . . . 41
30.2 Описываемые соотношения вариации неделимых вверх действий . . . 41
30.3 Теории, обусловленные вариацией неделимых вверх действий . . . 41
30.4 Дефинициональные расширения, обусловленные вариацией неделимых вверх действий . . . 41
30.5 Определения вариации неделимых вверх действий . . . 41
30.6 Грамматика описаний процесса для вариации неделимых вверх действий . . . 44
31 Вариация неделимых вниз действий . . . 45
31.1 Примитивная лексика вариации неделимых вниз действий . . . 45
31.2 Описываемые соотношения вариации неделимых вниз действий . . . 45
31.3 Теории, обусловленные вариацией неделимых вниз действий . . . 45
31.4 Дефинициональные расширения, обусловленные вариацией неделимых вниз действий . . . 45
31.5 Определения вариации неделимых вниз действий . . . 45
31.6 Грамматика описаний процесса для вариации неделимых вниз действий . . . 48
32 Входные условия совместных действий . . . 48
32.1 Примитивная лексика входных условий совместных действий . . . 48
32.2 Описываемые соотношения входных условий совместных действий . . . 48
32.3 Теории, обусловленные входными условиями совместных действий . . . 49
32.4 Дефинициональные расширения, обусловленные входными условиями совместных действий 49

32.5 Определения входных условий совместных действий . . . 49
32.6 Грамматика описаний процесса для входных условий совместных действий . . . 50
33 Эффекты совместных действий . . . 50
33.1 Примитивная лексика эффектов совместных действий . . . 50
33.2 Описываемые соотношения эффектов совместных действий . . . 50
33.3 Теории, обусловленные эффектами совместных действий . . . 50
33.4 Дефинициональные расширения, обусловленные эффектами совместных действий . . . 50
33.5 Определения эффектов совместных действий . . . 50
33.6 Грамматика описаний процесса для эффектов совместных действий . . . 51
34 Вариация препятствующих входных условий . . . 51
34.1 Примитивная лексика вариации препятствующих входных условий . . . 51
34.2 Описываемые соотношения вариации препятствующих входных условий . . . 51
34.3 Теории, обусловленные вариацией препятствующих входных условий . . . 51
34.4 Дефинициональные расширения, обусловленные вариацией препятствующих входных условий . . . 51
34.5 Определения вариации препятствующих входных условий . . . 52
34.6 Грамматика описаний процесса для вариации препятствующих входных условий . . . 53
35 Вариация эффектов затирания полезных данных . . . 53
35.1 Примитивная лексика вариации эффектов затирания полезных данных . . . 53
35.2 Описываемые соотношения вариации эффектов затирания полезных данных . . . 53
35.3 Теории, обусловленные вариацией эффектов затирания полезных данных . . . 53
35.4 Дефинициональные расширения, обусловленные вариацией эффектов затирания полезных данных . . . 53
35.5 Определения вариации эффектов затирания полезных данных . . . 53
35.6 Грамматика описаний процесса для вариации эффектов затирания полезных данных . . . 55
Приложение А (справочное) ASN.1 Идентификатор настоящего стандарта . . . 56
Приложение В (справочное) Пример описания технологического процесса в соответствии с настоящим стандартом . . . 57
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 63
Библиография . . . 64

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 42

Дефинициональные расширения: временные расширения и расширения состояния

Industrial automation systems and integration. Process specification language.
Part 42. Definitional extension. Temporal and state extensions

Дата введения — 2012—09—01

1 Область применения

Настоящий стандарт устанавливает спецификацию непримитивных понятий языка программирования. При этом используется набор определений, написанных на языке, установленном ИСО 18629. Данные определения устанавливают аксиомы для терминологии в соответствии с ИСО 18629.

Настоящий стандарт распространяется на:

- определения новых временных и основанных на состоянии понятий в соответствии с ИСО 18629-11 и ИСО 18629-12;

- событийные ограничения выполнения действий, сформулированные с использованием временных соотношений в соответствии с ИСО 18629-11 и соотношений, основанных на состоянии, в соответствии с ИСО 18629-12.

Настоящий стандарт не распространяется на определения новых понятий, соответствующих ИСО 18629-11 и ИСО 18629-12, и новых понятий, не зависимых от временных соотношений и соотношений, зависимых от состояния.

Примечание — Временные понятия и понятия, зависимые от состояния, относятся ко времени и статусу какой-либо сущности, связанной с технологическим процессом и используемой для его спецификации.

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)

Издание официальное

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

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 **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]

3.1.3 **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]

3.1.4 **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.

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

[ИСО 18629-1]

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

[ИСО 10303-1]

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

[ИСО 18629-1]

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

[ИСО 15531-1]

3.1.9 **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]

3.1.10 **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]

3.1.11 **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]

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

[ИСО 18629-1]

3.1.13 **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]

3.1.14 **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]

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

[ИСО 10303-1]

3.1.16 **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]

3.1.17 **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

5.1 Введение

Данный раздел характеризует фундаментальные теории, описанные в настоящем стандарте.

5.2 Расширения, определенные в настоящем стандарте

Фундаментальные теории, описанные в настоящем стандарте:
- основанные на состоянии входные условия выполнения действий;
- временные входные условия выполнения действий;
- временные и основанные на состоянии входные условия;
- событийные входные условия выполнения действий;
- предотвращаемые условия выполнения действий;
- периодические входные условия выполнения действий;
- входные условия нарушения выполнения действий;
- эффекты действий;
- эффекты действий: событийные ограничения;
- эффекты действий: временные и событийные ограничения;
- дерево флюента;
- распределение комплексных действий;
- основанное на состоянии распределение комплексных действий;
- временное распределение комплексных действий;
- смешанное распределение комплексных действий;
- вариация комплексных действий;
- основанная на состоянии вариация комплексных действий;
- временная вариация комплексных действий;
- смешанная вариация комплексных действий;
- встроенные действия: планы;
- встроенные действия: временной разброс;
- вариация неделимых вверх действий;
- вариация неделимых вниз действий;
- входные условия совместных действий;
- эффекты совместных действий;
- вариация препятствующих входных условий;
- вариация эффектов затирания полезных данных.

Все дефиниционные расширения, определенные в настоящем стандарте, являются расширениями ИСО 18629-12, которые, в свою очередь, являются расширениями ИСО 18629-11.

6 Основанные на состоянии входные условия выполнения действий

Данный раздел характеризует все определения, обусловленные основанными на состоянии входными условиями выполнения действий. Критерий, используемый для классификации указанных действий: зависит или нет допустимое выполнение действий от состояния системы до выполнения указанных действий.

6.1 Примитивная лексика основанных на состоянии входных условий выполнения действий

Лексика основанных на состоянии входных условий выполнения действий не требует никаких примитивных соотношений.

6.2 Описываемая лексика понятий для основанных на состоянии входных условий выполнения действий

В данном подразделе определены следующие соотношения:
- (state_equiv ?s1 ?s2);
- (markov_precond ?a);
- (partial_state ?a);
- (rigid_state ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

6.3 Теории ядра, обусловленные основанными на состоянии входными условиями выполнения действий

Для данного расширения необходимо иметь:
- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- disc_state.th;
- occtree.th;
- psl_core.th.

6.4 Дефинициональные расширения, обусловленные основанными на состоянии входными условиями выполнения действий

Для основанных на состоянии входных условий выполнения действий необходимо нижеследующее дефинициональное расширение:
- precond.def.

6.5 Определения понятий для основанных на состоянии входных условий выполнения действий

Для основанных на состоянии входных условий выполнения действий определены нижеследующие понятия.

6.5.1 state_equiv

Два события эквивалентны по состоянию, если после наступления данных событий имеет место одно и то же состояние.

```
(forall (?s1 ?s2) (iff (state_equiv ?s1 ?s2)
(forall (?f)
    (iff (holds ?f ?s1)
        (holds ?f ?s2)))))
```

6.5.2 markov_precond

Некоторое действие является марковским действием при заданных входных условиях, если из согласованности двух событий по состоянию следует их согласованность по расширению возможности действия.

```
(forall (?a) (iff (markov_precond ?a)
(forall (?s1 ?s2)
    (implies (state_equiv ?s1 ?s2)
        (poss_equiv ?a ?s1 ?s2))))
```

6.5.3 partial_state

Некоторое действие является частично ограниченным по состоянию тогда и только тогда, когда существуют сохраняющие событие перестановки переменных.

```
(forall (?a) (iff (partial_state ?a)
(and (exists (?s1)
    (forall (?s2)
        (implies (state_equiv ?s1 ?s2)
            (poss_equiv ?a ?s1 ?s2))))
  (exists (?s3 ?s4)
    (and  (state_equiv ?s3 ?s4)
        (not (poss_equiv ?a ?s3 ?s4)))))))
```

6.5.4 rigid_state

Некоторое действие является жестко зависимым от состояния тогда и только тогда, когда сохраняющая событие перестановка переменных является тривиальной.

```
(forall (?a) (iff (rigid_state ?a)
(forall (?s1)
    (exists (?s2)
        (and (state_equiv ?s1 ?s2)
            (not (poss_equiv ?a ?s1 ?s2)))))))
```

6.6 Грамматика соотношений для основанных на состоянии входных условий выполнения действий

Нижеследующие грамматические утверждения устанавливают описания процессов и вспомогательные правила, определенные в KIF для основанных на состоянии входных условий выполнения действий.

Примечание — Функция и важность грамматических утверждений поясняются в ИСО 18629-1:2004 (подразделы 3.3.8, 4.2.4 и 5.1).

```
< simple_state_precond > ::= (forall (?s)
                                (implies (and (occurrence ?s )
                                              (legal ?s))
                                         < simple_state_axiom >))) |
                             (forall (?s)
                                (implies (and (legal ?s)
                                              < simple_state_axiom >)
                                         (legal (successor < term > ?s))))
< state_precond > ::= (forall (< variable >*)
                                (implies (and (occurrence < variable > < term >)
                                              (legal < term >))
                                         < state_axiom >))) |
                             (forall (< variable >*)
                                (implies (and (legal < term >)
                                              < state_axiom >)
                                         (legal (successor < term > < term >))))
< simple_state_literal > ::=    (prior < term > ?s)
< simple_state_formula > ::=    < simple_state_literal > |
                                (not < simple_state_formula >) |
                                ({and | or} < simple_state_formula >*) |
                                ({implies | iff} < simple_state_formula >)
< simple_state_axiom > ::=      ({forall | exists} < variable >*)
                                      < simple_state_formula >)
< state_literal > ::=           (prior < term > < variable >)
< state_formula > ::=           < state_literal > |
                                (not < state_formula >) |
                                ({and | or} < state_formula >) |
                                ({implies | iff} < state_formula >*)
< state_axiom > ::=             (forall (< variable >+) < state_formula >)
```

7 Временные входные условия выполнения действий

Данный раздел характеризует все определения, обусловленные временными входными условиями выполнения действий. Критерий классификации указанных действий: зависит или нет действительное выполнение некоторого действия только от времени выполнения указанных действий.

7.1 Примитивная лексика временных входных условий выполнения действий

Лексика временных входных условий выполнения действий не требует примитивных отношений.

7.2 Описываемая лексика понятий для временных входных условий выполнения действий

В данном подразделе описываются нижеследующие отношения:
- (begin_equiv ?s1 ?s2);
- (time_precond ?a);
- (partial_time ?a);
- (rigid_time ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

7.3 Теории, обусловленные временными входными условиями выполнения действий

Для данных теорий необходимы:
- disc_state.th;
- occtree.th;
- psl_core.th.

7.4 Дефинициональные расширения, обусловленные временными входными условиями выполнения действий

Для временных входных условий действий требуются нижеследующие дефинициональные расширения:
- precond.def;
- start.def.

7.5 Определения временных входных условий выполнения действий

Нижеследующие понятия определены для временных входных условий выполнения действий.

7.5.1 begin_equiv

Два неделимых события эквивалентны по началу выполнения действия (приобретают характеристику begin_equiv) тогда и только тогда, когда они имеют одинаковое время начала отсчета.

```
(forall (?s1 ?s2) (iff (begin_equiv ?s1 ?s2)
(= (beginof ?s1) (beginof ?s2))))
```

7.5.2 time_precond

Некоторое действие является действием с временными входными условиями при ограничении: если два неделимых события на дереве событий согласуются по времени начала отсчета, то они согласуются и по расширению возможности данного действия.

```
(forall (?a) (iff (time_precond ?a)
(forall (?s1 ?s2)
(implies (begin_equiv ?s1 ?s2))
(poss_equiv ?a ?s1 ?s2)))))
```

7.5.3 partial_time_precond

Некоторое действие является частично ограниченным по времени тогда и только тогда, когда существуют сохраняющие событие перестановки.

```
(forall (?a) (iff (partial_time ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (begin_equiv ?s1 ?s2)
                        (poss_equiv ?a ?s1 ?s2))))
        (exists (?s3 ?s4)
                (and (begin_equiv ?s3 ?s4)
                        (not (poss_equiv ?a ?s3 ?s4)))))))
```

7.5.4 rigid_time

Некоторое действие является жестким временным действием тогда и только тогда, когда перестановка по временно́й шкале, сохраняющая событие, является тривиальной.

```
(forall (?a) (iff (rigid_time ?a)
(forall (?s1)
        (exists (?s2)
                (and (begin_equiv ?s1 ?s2)
                        (not (poss_equiv ?a ?s1 ?s2)))))))
```

7.6 Грамматика описаний процесса для временных входных условий выполнения действий

Нижеследующие грамматические утверждения определяют описания процессов и вспомогательные правила, установленные в KIF для временных входных условий выполнения действий.

```
< simple_time_precond > ::= (forall (?s)
                                (implies (and (occurrence ?s < term >)
                                              (legal ?s))
                                         < simple_time_axiom >))) |
```

```
                                (forall (?s)
                                    (implies (and (legal ?s)
                                                  < simple_time_axiom >)
                                             (legal (successor < term > ?s))))
< time_precond > ::=            (forall (?s)
                                    (implies (and   (occurrence ?s < term >)
                                                    (legal ?s))
                                             < time_axiom >))) |
                                (forall (?s)
                                    (implies (and (legal ?s)
                                                  < time_axiom >)
                                             (legal (successor < term > ?s))))
< simple_time_literal > ::=     (= (beginof ?s) < term >)
< simple_time_formula > ::=     (and < simple_time_literal >*)
< simple_time_axiom > ::=       ({forall | exists} < variable >*) < simple_time_formula >)
< ordered_time_literal > ::=    < simple_time_literal > |
                                (before (beginof ?s) < term >) |
                                (before < term > (beginof ?s))
< time_formula > ::=            < ordered_time_literal > |
                                (not < time_formula >) |
                                ({and | or} < time_formula >*) |
                                ({implies | iff} < time_formula >)
< time_axiom > ::=              ({forall | exists} < variable >*) < time_formula >)
```

8 Временные и основанные на состоянии входные условия

Данный раздел характеризует все определения, обусловленные временными и основанными на состоянии входными условиями. Критерий, используемый для классификации указанных действий: зависит или нет действительное выполнение действия от состояния до указанного события и от времени выполнения действия.

8.1 Примитивная лексика временных и основанных на состоянии входных условий

Лексика временных и основанных на состоянии входных условий не требует примитивных соотношений.

8.2 Описываемая лексика временных и основанных на состоянии входных условий

В данном подразделе описаны нижеследующие соотношения:

- (mixed_precond ?s);
- (partial_mixed ?s);
- (rigid_mixed ?s).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

8.3 Теории, обусловленные временными и основанными на состоянии входными условиями

Для данной теории необходимы:

- disc_state.th;
- occtree.th;
- psl_core.th.

8.4 Дефинициональные расширения, обусловленные временными и основанными на состоянии входными условиями

Для временных и основанных на состоянии входных условий дефинициональные расширения не требуются.

8.5 Определения временных и основанных на состоянии входных условий

Нижеследующие понятия определены для временных и основанных на состоянии входных условий.

8.5.1 mixed_precond

Некоторое действие является смешанным действием для заданных входных условий при ограничении: если два события согласуются по состоянию и времени начала отсчета, то они согласуются и по расширениию возможности действий.

```
(forall (?a) (iff (mixed_precond ?a)
(forall (?s1 ?s2)
    (implies (and (state_equiv ?s1 ?s2)
                  (begin_equiv ?s1 ?s2))
             (poss_equiv ?a ?s1 ?s2)))))
```

8.5.2 partial_mixed

Некоторое действие является частично смешанным ограниченным действием тогда и только тогда, когда существуют сохраняющие событие перестановки.

```
(forall (?a) (iff (partial_mixed ?a)
(and (exists (?s1)
    (forall (?s2)
        (implies (and (state_equiv ?s1 ?s2)
                      (begin_equiv ?s1 ?s2))
                 (poss_equiv ?a ?s1 ?s2))))
  (exists (?s3 ?s4)
    (and (state_equiv ?s3 ?s4)
         (begin_equiv ?s3 ?s4)
         (not (poss_equiv ?a ?s3 ?s4)))))))
```

8.5.3 rigid_mixed

Некоторое действие является жестким смешанным действием тогда и только тогда, когда сохраняющая событие перестановка (которая является сразу как текущей перестановкой, так и перестановкой по временной шкале) является тривиальной.

```
(forall (?a) (iff (rigid_mixed ?a)
(forall (?s1)
    (exists (?s2)
        (and (state_equiv ?s1 ?s2)
             (begin_equiv ?s1 ?s2)
             (not (poss_equiv ?a ?s1 ?s2)))))))
```

8.6 Грамматика временных и основанных на состоянии входных условий

Нижеследующие грамматические утверждения определяют описания процесса и вспомогательные правила, установленные в KIF для временных и основанных на состоянии входных условий.

```
< simple_mix_precond > ::= (forall (?s)
                               (implies (and (occurrence ?s < term >)
                                             (legal ?s))
                                        < simple_mix_axiom >))) |
                           (forall (?s)
                               (implies (and (legal ?s)
                                             < simple_mix_axiom >)
                                        (legal (successor < term > ?s))))
< mix_precond > ::= (forall (?s)
                        (implies (and (occurrence ?s < term >)
                                      (legal ?s))
                                 < mix_formula >)))
< simple_mix_literal > ::=   (= (beginof ?s) < term >) (prior ?s)
< simple_mix_formula > ::=   (and < simple_mix_literal >*)
< simple_mix_axiom > ::=     ({forall | exists} < variable >*) < simple_mix_formula >)
< ordered_mix_literal > ::=  < simple_mix_literal > |
                             (before (beginof ?s) < term >) (prior < term > ?s) |
                             (before < term > (beginof ?s)) (prior < term > ?s)
< mix_formula > ::=          < ordered_mix_literal > |
                             (not < mix_formula >) |
```

```
                                ({and | or} < mix_formula >*) |
                                ({implies | iff} < mix_formula >)
< mix_axiom > ::=               ({forall | exists} < variable >*) < mix_formula >)
```

9 Событийные входные условия выполнения действий

Данный раздел характеризует все определения, обусловленные событийными входными условиями выполнения действий. Критерий, используемый для классификации указанных действий: зависит или нет реальное выполнение некоторого действия от выполнения других действий.

9.1 Примитивная лексика событийных входных условий выполнения действий

В лексике событийных входных условий выполнения действий примитивные соотношения не используются.

9.2 Описываемые соотношения событийных входных условий выполнения действий

В данном подразделе определены следующие соотношения:
- (tree_equiv ?s1 ?s2);
- (occurrence_constrained ?a);
- (occurrence_dependent ?a);
- (occurrence_independent ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

9.3 Теории, обусловленные событийными входными условиями выполнения действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

9.4 Дефинициональные расширения, обусловленные событийными входными условиями выполнения действий

Для событийных входных условий выполнения действий необходимо нижеследующее дефинициональное расширение:
- precond.def.

9.5 Определения событийных входных условий выполнения действий

Для событийных входных условий выполнения действий определены нижеследующие понятия.

9.5.1 tree_equiv

Два события эквивалентны на дереве событий тогда и только тогда, когда они отображаются друг на друга путем некоторого преобразования дерева событий.

```
(forall (?a ?s1 ?s2)
 (iff   (tree_equiv (successor ?a ?s1) (successor ?a ?s2))
        (and   (legal_equiv ?s1 ?s2)
               (or     (tree_equiv ?s1 ?s2)
                       (initial ?s1)
                       (initial ?s2)))))
```

9.5.2 occurrence_constrained

Некоторое действие является событийно ограниченным тогда и только тогда, когда все преобразования дерева событий сохраняют действительное выполнение данного действия.

```
(forall (?a) (iff (occurrence_constrained ?a)
(forall (?s1 ?s2)
       (implies (and (occurrence ?s1 ?a)
                     (occurrence ?s2 ?a)
                     (tree_equiv ?s1 ?s2))
                (legal_equiv ?s1 ?s2)))))
```

9.5.3 occurrence_dependent

Некоторое действие является зависящим от события тогда и только тогда, когда существует преобразование дерева событий, сохраняющее действительное выполнение действия.

```
forall (?a) (iff (occurrence_dependent ?a)
(exists (?s1)
        (and    (occurrence ?s1 ?a)
                (forall (?s2)
                        (implies (and (occurrence ?s2 ?a)
                                      (tree_equiv ?s1 ?s2))
                                 (legal_equiv ?s1 ?s2)))))))
```

9.5.4 occurrence_independent

Некоторое действие является не зависящим от события тогда и только тогда, когда не существует нетривиального преобразования дерева событий, сохраняющего действительное выполнение действия.

```
(forall (?a) (iff (occurrence_independent ?a)
(forall (?s1)
        (implies (occurrence ?s1 ?a)
                (exists (?s2)
                        (and    (occurrence ?s2 ?a)
                                (tree_equiv ?s1 ?s2)
                                (not (legal_equiv ?s1 ?s2))))))))
```

9.6 Грамматика событийных входных условий выполнения действий

Нижеследующие грамматические утверждения определяют описания процессов и вспомогательные правила, установленные в KIF для событийных входных условий выполнения действий.

```
< occ_constrained_precond > ::=
(forall (?s)
        ({implies | iff} (and (occurrence ?s < term >)
                        (legal ?s)
                                (ubiquitous < term > < term >))
                {< leaf_constrained_axiom > | < inner_constrained_axiom}))
< leaf_constrained_formula > ::= (exists (?occ ?s1)
                                        (and    (occurrence ?occ < term >)
                                                (leaf_occ ?s1 ?occ)
                                                (= ?s (successor < term > ?s1))))
< leaf_constrained_axiom > ::= < leaf_constrained_formula > |
                                (not < leaf_constrained_formula >)
< inner_constrained_formula > ::= (exists (?occ ?s1 ?s2)
                                        (and    (occurrence ?occ < term >)
                                                (subactivity_occurrence ?s1 ?occ)
                                                (subactivity_occurrence ?s2 ?occ)
                                                (precedes ?s1 ?s)
                                                (precedes ?s ?s2)))
< inner_constrained_axiom > ::=         < inner_constrained_formula > |
                                (not < inner_constrained_formula >)
```

10 Предотвращаемые условия выполнения действий

Данный раздел характеризует все определения, обусловленные предотвращаемыми условиями выполнения действий. Критерий, используемый для классификации указанных действий: зависит или нет действительное выполнение действий от состояния системы до выполнения действий и от выполнения других действий.

10.1 Примитивная лексика предотвращаемых условий выполнения действий

Лексика предотвращаемых условий выполнения действий не требует примитивных соотношений.

10.2 Описываемые соотношения предотвращаемых условий выполнения действий

В данном подразделе определены нижеследующие соотношения:

- (preventable ?a);

- (possibly_preventable ?a);
- (unpreventable ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

10.3 Теории, обусловленные предотвращаемыми условиями выполнения действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

10.4 Дефинициональные расширения, обусловленные предотвращаемыми условиями выполнения действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

10.5 Определения предотвращаемых условий выполнения действий

Для предотвращаемых условий выполнения действий определены нижеследующие понятия.

10.5.1 preventable

Некоторое действие является предотвращаемым тогда и только тогда, когда допустимые события зависят от переменных, сохраняющих свое значение при выполнении других действий на дереве событий.

```
(forall (?a) (iff (preventable ?a)
(forall (?s1 ?s2)
    (implies (and   (occurrence ?s1 ?a)
                    (occurrence ?s2 ?a)
                    (state_equiv ?s1 ?s2)
                    (tree_equiv ?s1 ?s2))
        (legal_equiv ?s1 ?s2)))))
```

10.5.2 possibly_preventable

Некоторое действие является возможно предотвращаемым тогда и только тогда, когда допустимые события зависят от переменных, сохраняющих свое значение при выполнении других действий на дереве событий.

```
(forall (?a) (iff (possibly_preventable ?a)
(exists (?s1)
    (and (occurrence ?s1 ?a)
        (forall (?s2)
            (implies (and   (occurrence ?s2 ?a)
                            (state_equiv ?s1 ?s2)
                            (tree_equiv ?s1 ?s2))
                (legal_equiv ?s1 ?s2)))))))
```

10.5.3 unpreventable

Некоторое действие является непредотвращаемым тогда и только тогда, когда не существует соотношения между действительным событием и переменной, сохраняющей свое значение при выполнении других действий.

```
(forall (?a) (iff (unpreventable ?a)
(forall (?s1)
    (implies (occurrence ?s1 ?a)
        (exists (?s2)
            (and    (occurrence ?s2 ?a)
                    (state_equiv ?s1 ?s2)
                    (tree_equiv ?s1 ?s2)
                    (not (legal_equiv ?s1 ?s2)))))))))
```

10.6 Грамматика описаний процесса для предотвращаемых условий выполнения действий

Нижеследующие грамматические утверждения определяют описания процессов, установленные в KIF для предотвращаемых условий выполнения действий:

```
< preventable_precond > ::=
(forall (?s)
        ({implies | iff}  (and (occurrence ?s < term >)
                          (legal ?s)
                                    (ubiquitous < term > < term >)
                                    < simple_state_axiom >)
{< leaf_constrained_axiom > | < inner_constrained_axiom}))
< possibly_preventable_precond > ::=
(forall (?s)
        ({implies | iff}  (and (occurrence ?s < term >)
                          (legal ?s)
                                    (ubiquitous < term > < term >)
                                    < state_axiom >)
{< leaf_constrained_axiom > | < inner_constrained_axiom}))
```

11 Периодические входные условия выполнения действий

Данный раздел характеризует все определения, обусловленные периодическими входными условиями выполнения действий. Критерий, используемый для классификации указанных действий: зависит или нет действительное выполнение некоторого действия от времени его выполнения и от выполнения других действий.

11.1 Примитивная лексика периодических входных условий выполнения действий

Лексика периодических входных условий выполнения действий не требует примитивных соотношений.

11.2 Описываемая лексика периодических входных условий выполнения действий

В данном подразделе определены следующие соотношения:

- (periodic ?a);
- (intermittent ?a);
- (aperiodic ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

11.3 Теории, обусловленные периодическими входными условиями выполнения действий

Для данной теории необходимы:

- occtree.th;
- psl_core.th.

11.4 Дефинициональные расширения, обусловленные периодическими входными условиями выполнения действий

Для данных расширений необходимы:

- occ_precond.def;
- time_precond.def;
- precond.def.

11.5 Определения периодических входных условий выполнения действий

Для периодических входных условий выполнения действий определены нижеследующие понятия.

11.5.1 periodic

Некоторое действие является периодическим тогда и только тогда, когда допустимые события зависят от времени выполнения других действий.

```
(forall (?a) (iff (periodic ?a)
(forall (?s1 ?s2)
        (implies  (and   (occurrence ?s1 ?a)
                         (occurrence ?s2 ?a)
                         (begin_equiv ?s1 ?s2)
                         (tree_equiv ?s1 ?s2))
                  (legal_equiv ?s1 ?s2)))))
```

11.5.2 intermittent

Некоторое действие является перемежающимся тогда и только тогда, когда допустимые события зависят от времени выполнения других действий.

```
(forall (?a) (iff (intermittent ?a)
(exists (?s1)
        (and   (occurrence ?s1 ?a)
               (forall (?s2)
                       (implies (and    (occurrence ?s2 ?a)
                                        (begin_equiv ?s1 ?s2)
                                        (tree_equiv ?s1 ?s2))
                                (legal_equiv ?s1 ?s2)))))))
```

11.5.3 aperiodic

Некоторое действие является апериодическим тогда и только тогда, когда не существует связи между действительным событием и временем выполнения других действий.

```
(forall (?a) (iff (aperiodic ?a)
(forall (?s1)
        (implies (occurrence ?s1 ?a)
                 (exists (?s2)
                         (and    (occurrence ?s2 ?a)
                                 (begin_equiv ?s1 ?s2)
                                 (tree_equiv ?s1 ?s2)
                                 (not (legal_equiv ?s1 ?s2))))))))
```

11.6 Грамматика периодических входных условий выполнения действий

Следующие грамматические утверждения определяют описания процесса, установленные в KIF для периодических входных условий выполнения действий.

```
< periodic_precond > ::=
(forall (?s)
        ({implies | iff}   (and (occurrence ?s < term >)
                           (legal ?s)
                                  (ubiquitous < term > < term >)
                                  < simple_time_axiom >)
                   {< leaf_constrained_axiom > | < inner_constrained_axiom}))
< intermittent_precond > ::=
(forall (?s)
        ({implies | iff}   (and (occurrence ?s < term >)
                           (legal ?s)
                                        (ubiquitous < term > < term >)
                                        < time_axiom >)
                   {< leaf_constrained_axiom > | < inner_constrained_axiom}))
```

12 Входные условия нарушения выполнения действий

Данный раздел характеризует все определения, обусловленные входными условиями нарушения выполнения действий. Критерий, используемый для классификации указанных действий: зависит или нет действительное выполнение действий от состояния до выполнения действий, от времени выполнения действий и от выполнения других действий.

12.1 Примитивная лексика входных условий нарушения выполнения действий

Лексика входных условий нарушения выполнения действий не требует примитивных соотношений.

12.2 Описываемая лексика входных условий нарушения выполнения действий

В данном подразделе определены следующие соотношения:
- (spoilage ?a);
- (possible_spoilage ?a);
- (nonspoilage ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

12.3 Теории, обусловленные входными условиями нарушения выполнения действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

12.4 Дефинициональные расширения, обусловленные входными условиями нарушения выполнения действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- time_precond.def;
- precond.def.

12.5 Определения входных условий нарушения выполнения действий

Для входных условий нарушения выполнения действий определены следующие понятия.

12.5.1 spoilage

Для некоторого действия имеются входные условия нарушения его выполнения, если:

```
(forall (?a) (iff (spoilage ?a)
(forall (?s1 ?s2)
        (implies (and   (occurrence ?s1 ?a)
                        (occurrence ?s2 ?a)
                        (state_equiv ?s1 ?s2)
                        (begin_equiv ?s1 ?s2)
                        (tree_equiv ?s1 ?s2))
                 (legal_equiv ?s1 ?s2))))
```

12.5.2 possible_spoilage

Для некоторого действия имеются входные условия возможного нарушения его выполнения, если:

```
(forall (?a) (iff (possible_spoilage ?a)
(exists (?s1)
        (and    (occurrence ?s1 ?a)
                (forall (?s2)
                        (implies (and   (occurrence ?s2 ?a)
                                        (state_equiv ?s1 ?s2)
                                        (begin_equiv ?s1 ?s2)
                                        (tree_equiv ?s1 ?s2))
                                 (legal_equiv ?s1 ?s2)))))))
```

12.5.3 nonspoilage

Для некоторого действия существуют входные условия без нарушений выполнения действия, если:

```
(forall (?a) (iff (nonspoilage ?a)
(forall (?s1)
        (implies (occurrence ?s1 ?a)
                 (exists (?s2)
                         (and   (occurrence ?s2 ?a)
                                (state_equiv ?s1 ?s2)
                                (begin_equiv ?s1 ?s2)
                                (tree_equiv ?s1 ?s2)
                                (not (legal_equiv ?s1 ?s2)))))))))
```

12.6 Грамматика входных условий нарушения выполнения действий

Грамматика описаний процесса для входных условий нарушения выполнения действий эквивалентна грамматике утверждений общих описаний процесса, определенной в ИСО 18629-11.

13 Эффекты действий

Данный раздел характеризует все определения, обусловленные эффектами действий. Критерий, используемый для классификации указанных действий: одинаковы ли эффекты некоторого действия для всех выполняемых действий.

13.1 Примитивная лексика эффектов действий

Лексика эффектов действий не требует примитивных соотношений.

13.2 Описываемая лексика эффектов действий

В данном подразделе определены следующие соотношения:
- (effects_equiv ?a ?s1 ?s2);
- (context_free ?a);
- (null ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

13.3 Теории, обусловленные эффектами действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

13.4 Дефинициональные расширения, обусловленные эффектами действий

Для эффектов действий необходимы нижеследующие дефинициональные расширения:

13.5 Определения эффектов действий

Для утверждений Spectrum и Subtree Containement определены нижеследующие понятия.

13.5.1 effects_equiv

Два события для некоторого неделимого действия являются эквивалентными эффектами, если после данных событий состояния сохраняются неизменными.

```
(forall (?a ?s1 ?s2) (iff (effects_equiv ?a ?s1 ?s2)
(and (occurrence ?s1 ?a)
     (occurrence ?s2 ?a)
     (forall (?f)
          (iff     (holds ?f ?s1)
                   (holds ?f ?s2))))))
```

13.5.2 context_free

Некоторое действие является свободным от контекста (context_free), если какая-либо переменная, измененная в результате выполнения одного действия, изменяется в результате каждого выполнения указанного действия.

```
(forall (?a) (iff (context_free ?a)
(forall (?s1 ?s2)
     (implies  (and (occurrence ?s1 ?a)
                    (occurrence ?s2 ?a))
          (effects_equiv ?a ?s1 ?s2)))))
```

13.5.3 null

Некоторое действие является нулевым тогда и только тогда, когда оно не затрагивает и не фальсифицирует какую-либо переменную.

```
(forall (?a) (iff (null ?a)
(forall (?s)
     (implies (occurrence ?s ?a)
          (forall (?f)
               (iff (holds ?f ?s)
                    (prior ?f ?s)))))))
```

13.6 Грамматика эффектов действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для эффектов действий:

```
< context_free_effect > ::= (forall (?s)
                            (implies (occurrence ?s )
                                    < simple_holds_axiom >))
< null_effect > ::=         (forall (?s)
                            (implies (occurrence ?s )
                                    (iff < simple_holds_axiom >
                                       < simple_state_axiom >)))
< simple_holds_literal > ::= (holds < term > ?s)
< simple_holds_formula > ::= < simple_holds_literal > |
                            (not < simple_holds_formula >) |
                            ({and | or} < simple_holds_formula >*) |
                            ({implies | iff} < simple_holds_formula >)
< simple_holds_axiom > ::=  ({forall | exists} < variable >*)
                                < simple_holds_formula >)
```

14 Основанные на состоянии эффекты действий

Данный раздел характеризует все определения, обусловленные основанными на состоянии эффектами действий. Критерий, используемый для классификации указанных действий: зависят или нет эффекты некоторого действия от состояния до выполнения указанных действий.

14.1 Примитивная лексика основанных на состоянии эффектов действий

Лексика основанных на состоянии эффектов действий не требует примитивных соотношений.

14.2 Описываемая лексика основанных на состоянии эффектов действий

В данном подразделе определены следующие соотношения:
- (markov_effect ?a);
- (partial_state_effect ?a);
- (rigid_state_effect ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

14.3 Теории ядра, обусловленные основанными на состоянии эффектами действий

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

14.4 Дефинициональные расширения, обусловленные основанными на состоянии эффектами действий

Нижеследующее дефинициональное расширение необходимо для основанных на состоянии эффектов действий:
- precond.def.

14.5 Определения основанных на состоянии эффектов действий

Определены нижеследующие понятия для основанных на состоянии эффектов действий.

14.5.1 markov_effect

Некоторое действие является действием с марковским эффектом при условии, если когда-либо какие-либо два выполненных действия согласуются по состоянию (например, по состоянию, сохраняемому после выполнения действия), то они согласуются и по эффектам.

```
(forall (?a) (iff (markov_effects ?a)
(forall (?s1 ?s2)
      (implies (and (occurrence ?s1 ?a)
                    (occurrence ?s2 ?a)
                    (state_equiv ?s1 ?s2))
               (effects_equiv ?a ?s1 ?s2)))))
```

14.5.2 partial_state_effect

Некоторое действие является частично ограниченным по состоянию тогда и только тогда, когда существуют сохраняющие эффект перестановки переменных.

```
(forall (?a) (iff (partial_state_effects ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (state_equiv ?s1 ?s2)
                        (effects_equiv ?a ?s1 ?s2))))
     (exists (?s3 ?s4)
        (and (state_equiv ?s3 ?s4)
                (not (effects_equiv ?a ?s3 ?s4)))))))
```

14.5.3 rigid_state_effect

Некоторое действие является жестко зависимым от состояния тогда и только тогда, когда сохраняющая эффект перестановка переменных является тривиальной.

```
(forall (?a) (iff (rigid_state_effects ?a)
(forall (?s1)
        (exists (?s2)
                (and (state_equiv ?s1 ?s2)
                        (not (effects_equiv ?a ?s1 ?s2)))))))
```

14.6 Грамматика основанных на состоянии эффектов действий

Грамматика описаний процесса для встроенных ограничений действий:

```
< simple_state_effect > ::=    (forall (?s)
                                (implies (and (occurrence ?s )
                                                < simple_state_axiom >)
                                        < simple_holds_axiom >)))
< state_effect > ::=            (forall (< variable >*)
                                (implies (and (occurrence < variable > < term >)
                                                < state_axiom >)
                                        < simple_holds_axiom >)))
```

15 Временные эффекты действий

Данный раздел характеризует все определения, обусловленные временными эффектами действий. Критерий, используемый для классификации указанных действий: зависят или нет эффекты некоторого действия только от времени выполнения указанного действия.

15.1 Примитивная лексика временных эффектов действий

Лексика временных эффектов действий не требует примитивных соотношений.

15.2 Описываемая лексика временных эффектов действий

В данном подразделе определены следующие соотношения:
- (temporal_effect ?a);
- (partial_temporal ?a);
- (nontemporal ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

15.3 Теории ядра, обусловленные временными эффектами действий

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

15.4 Дефинициональные расширения, обусловленные временными эффектами действий

Для данных расширений необходимы временные эффекты действий.

15.5 Определения временных эффектов действий

Для временных эффектов действий определены нижеследующие понятия.

15.5.1 temporal_effect

Некоторое действие является действием с временными эффектами при условии, если когда-либо наступление каких-либо двух событий согласуется по времени начала отсчета, то эти события согласуются и по эффектам (например, по состоянию, которое сохраняется после выполнения действия).

```
(forall (?a) (iff (temporal_effects ?a)
(forall (?s1 ?s2)
        (implies (and (occurrence ?s1 ?a)
                      (occurrence ?s2 ?a)
                      (begin_equiv ?s1 ?s2))
                 (effects_equiv ?a ?s1 ?s2)))))
```

15.5.2 partial_temporal

Некоторое действие является действием с частично временными эффектами тогда и только тогда, когда существуют сохраняющие эффект перестановки по временной шкале.

```
(forall  (?a) (iff (partial_temporal ?a)
(and   (exists (?s1)
         (forall (?s2)
                 (implies (begin_equiv ?s1 ?s2)
                          (effects_equiv ?a ?s1 ?s2))))
     (exists (?s3 ?s4)
          (and (begin_equiv ?s3 ?s4)
               (not (effects_equiv ?a ?s3 ?s4)))))))
```

15.5.3 nontemporal

Некоторое действие является действием с жесткими временными эффектами тогда и только тогда, когда сохраняющая эффект перестановка по временной шкале является тривиальной.

```
(forall (?a) (iff (nontemporal ?a)
(forall (?s1)
        (exists (?s2)
                (and (begin_equiv ?s1 ?s2)
                     (not (effects_equiv ?a ?s1 ?s2)))))))
```

15.6 Грамматика временных эффектов действий

Грамматика описаний процесса временных эффектов действий:

```
< simple_time_effects > ::=   (forall (?s)
                              (implies (and (occurrence ?s < term >)
                                            (legal ?s))
                                 < simple_holds_axiom >)))
< time_effects > ::=          (forall (?s)
                              (implies (and (occurrence ?s < term >
                                            (legal ?s)
                                            < time_axiom >)
                                 < simple_holds_axiom >))
```

16 Событийные эффекты действий

Данный раздел характеризует все определения, обусловленные событийными эффектами действий. Критерий, используемый для классификации указанных действий: зависят или нет эффекты некоторого действия только от выполнения других действий.

16.1 Примитивная лексика событийных эффектов действий

Лексика событийных эффектов действий не требует примитивных соотношений.

16.2 Описываемая лексика событийных эффектов действий

В данном подразделе определены следующие соотношения:

- (occ_effect ?a ?s);

- (occ_depend_effect ?a ?s);
- (nonocc_effect ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

16.3 Теории ядра, обусловленные событийными эффектами действий

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

16.4 Дефинициональные расширения, обусловленные событийными эффектами действий

Для событийных эффектов действий дефинициональные расширения не требуются.

16.5 Определения событийных эффектов действий

Определены нижеследующие понятия для событийных эффектов действий.

16.5.1 occ_effect

Некоторое действие имеет событийные ограниченные эффекты тогда и только тогда, когда указанные эффекты действий сохраняются преобразованиями дерева событий.

```
(forall (?a) (iff (occ_effects ?a)
(forall (?s1 ?s2)
    (implies (and (occurrence ?s1 ?a)
                  (occurrence ?s2 ?a)
                  (tree_equiv ?s1 ?s2))
             (effects_equiv ?a ?s1 ?s2)))))
```

16.5.2 occ_depend_effect

Некоторое действие имеет зависящие от события эффекты тогда и только тогда, когда существуют преобразования дерева событий, сохраняющие эффекты данного действия.

```
(forall (?a) (iff (occ_depend_effects ?a)
(exists (?s1)
    (and (occurrence ?s1 ?a)
         (forall (?s2)
              (implies (and (occurrence ?s2 ?a)
                            (tree_equiv ?s1 ?s2))
                       (effects_equiv ?a ?s1 ?s2)))))))
```

16.5.3 nonocc_effect

Некоторое действие не имеет зависящих от события эффектов тогда и только тогда, когда преобразование дерева событий, сохраняющее указанные эффекты действий, является тривиальным.

```
(forall (?a) (iff (nonocc_effects ?a)
(forall (?s1)
    (implies (occurrence ?s1 ?a)
         (exists  (?s2)
              (and (occurrence ?s2 ?a)
                   (tree_equiv ?s1 ?s2)
                   (not (effects_equiv ?a ?s1 ?s2))))))))
```

16.6 Грамматика событийных эффектов действий

Грамматика описаний процесса событийных эффектов действий:

```
< occ_effect_axiom > ::=
(forall (?s)
    (implies {< leaf_constrained_axiom > | < inner_constrained_axiom}
         < simple_holds_axiom >))
< occ_depend_effect_axiom > ::=
(forall (?s)
    (implies {< leaf_constrained_axiom > | < inner_constrained_axiom}
         < holds_axiom >))
```

17 Эффекты действий: событийные ограничения

Данный раздел характеризует все определения, относящиеся к эффектам действий: событийные ограничения. Критерий, используемый для классификации указанных действий: зависят или нет эффекты некоторого действия от состояния до выполнения других действий.

17.1 Примитивная лексика эффектов действий: событийные ограничения

Лексика эффектов действий (событийные ограничения) не требует примитивных соотношений.

17.2 Описываемая лексика эффектов действий: событийные ограничения

В данном подразделе определены следующие соотношения:
- (quantum ?a);
- (semiclassical ?a);
- (classical ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

17.3 Теории ядра, обусловленные эффектами действий: событийные ограничения

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

17.4 Дефинициональные расширения, обусловленные эффектами действий: событийные ограничения

Данное расширение не требует дефинициональных расширений.

17.5 Определения эффектов действий: событийные ограничения

Для эффектов действий определены нижеследующие понятия: событийные ограничения.

17.5.1 quantum

Некоторое действие является квантом действия тогда и только тогда, когда его эффекты зависят от переменных, сохраняющих свое значение при выполнении других действий дерева, в особенности если эффекты сохраняются при преобразованиях данного дерева событий, которое также сохраняет переменные.

```
(forall (?a) (iff (quantum ?a)
(forall (?s1 ?s2)
    (implies (and (occurrence ?s1 ?a)
                  (occurrence ?s2 ?a)
                  (state_equiv ?s1 ?s2)
                  (tree_equiv ?s1 ?s2))
             (effects_equiv ?a ?s1 ?s2)))))
```

17.5.2 semiclassical

Некоторое действие является полуклассическим действием тогда и только тогда, когда существуют преобразования дерева событий, также сохраняющие переменные и эффекты действий.

```
(forall (?a) (iff (semiclassical ?a)
(exists (?s1)
    (and (occurrence ?s1 ?a)
         (forall (?s2)
              (implies (and (occurrence ?s2 ?a)
                            (state_equiv ?s1 ?s2)
                            (tree_equiv ?s1 ?s2))
                       (effects_equiv ?a ?s1 ?s2)))))))
```

17.5.3 classical

Некоторое действие является классическим тогда и только тогда, когда его эффекты не зависят от выполнения каких-либо других действий на дереве событий.

```
(forall (?a) (iff (classical ?a)
(forall (?s1)
    (implies (occurrence ?s1 ?a)
```

```
        (exists (?s2)
                (and (occurrence ?s2 ?a)
                        (state_equiv ?s1 ?s2)
                        (tree_equiv ?s1 ?s2)
                        (not (effects_equiv ?a ?s1 ?s2))))))))
```

17.6 Грамматика эффектов действий: событийные ограничения

Грамматика описаний процесса для эффектов действий: событие.

```
< quantum_axiom > ::=
(forall (?s)
        (implies (and {< leaf_constrained_axiom > | < inner_constrained_axiom}
                < simple_state_axiom >)
            < simple_holds_axiom >))
< semiclassical_axiom > ::=
(forall (?s)
        (implies (and {< leaf_constrained_axiom > | < inner_constrained_axiom}
                < state_axiom >)
            < simple_holds_axiom >))
```

18 Эффекты действий: временные и событийные ограничения

Данный раздел характеризует все определения, обусловленные эффектами действий: временные и событийные ограничения. Критерий, используемый для классификации указанных действий: зависят или нет эффекты некоторого действия только от времени выполнения других действий.

18.1 Примитивная лексика эффектов действий: временные и событийные ограничения

Лексика эффектов действий (временные и событийные ограничения) не требует примитивных соотношений.

18.2 Описываемая лексика эффектов действий: временные и событийные ограничения

В данном подразделе определены следующие соотношения:
- (relativistic ?a);
- (seminewton ?a);
- (newton ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

18.3 Теории ядра, обусловленные эффектами действий: временные и событийные ограничения

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

18.4 Дефинициональные расширения, обусловленные эффектами действий: временные и событийные ограничения

Для данных расширений необходимы входные условия выполнения действий.

18.5 Определения эффектов действий: временные и событийные ограничения

Определены нижеследующие понятия для эффектов действий: временные и событийные ограничения.

18.5.1 relativistic

Некоторое действие является релятивистским тогда и только тогда, когда его эффекты зависят от времени выполнения других действий на дереве событий. В особенности если данные эффекты сохраняются при преобразованиях дерева событий, сохраняющих также шкалу отсчета времени.

```
(forall (?a) (iff (relativistic ?a)
(forall (?s1 ?s2)
        (implies (and (occurrence ?s1 ?a)
```

(occurrence ?s2 ?a)
(begin_equiv ?s1 ?s2)
(tree_equiv ?s1 ?s2))
(effects_equiv ?a ?s1 ?s2)))))

18.5.2 seminewton

Некоторое действие является полуньютоновским действием тогда и только тогда, когда существуют преобразования дерева событий, сохраняющие также шкалу отсчета времени и эффекты данного действия.

(forall (?a) (iff (seminewton ?a)
(exists (?s1)
(and (occurrence ?s1 ?a)
(forall (?s2)
(implies (and (occurrence ?s2 ?a)
(begin_equiv ?s1 ?s2)
(tree_equiv ?s1 ?s2))
(effects_equiv ?a ?s1 ?s2)))))))

18.5.3 newtonian

Некоторое действие является ньютоновским тогда и только тогда, когда его эффекты не зависят от времени начала отсчета выполнения каких-либо других действий на дереве событий.

(forall (?a) (iff (newtonian ?a)
(forall (?s1)
(implies (occurrence ?s1 ?a)
(exists (?s2)
(and (occurrence ?s2 ?a)
(begin_equiv ?s1 ?s2)
(tree_equiv ?s1 ?s2)
(not (effects_equiv ?a ?s1 ?s2))))))))

18.6 Грамматика эффектов действий: временные и событийные ограничения

Грамматика описаний процесса для эффектов действий: временные и событийные ограничения.

< relativistic_axiom > ::=
(forall (?s)
(implies (and {< leaf_constrained_axiom > | < inner_constrained_axiom}
< simple_time_axiom >)
< simple_holds_axiom >))
< seminewton_axiom > ::=
(forall (?s)
(implies (and {< leaf_constrained_axiom > | < inner_constrained_axiom}
< time_axiom >)
< simple_holds_axiom >))

19 Дерево флюента

Данный раздел характеризует все определения, обусловленные деревом флюента. Выбор критерия, используемого для классификации указанных переменных, зависит от действий, либо задающих, либо отменяющих значение некоторой переменной (флюента).

19.1 Примитивная лексика дерева флюента

Лексика дерева флюента не требует примитивных соотношений.

19.2 Описываемые соотношения дерева флюента

В данном подразделе определены следующие соотношения:

- (achieved ?f ?occ);
- (falsified ?f ?occ);
- (irreversible ?f);

- (unachievable ?f);
- (bounded ?f).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

19.3 Теории ядра, обусловленные деревом флюента

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

19.4 Дефинициональные расширения, обусловленные деревом флюента

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

19.5 Определения дерева флюента

Для дерева флюента определены нижеследующие понятия.

19.5.1 achieved

Флюента ?f приобретает значение (архивируется) в результате события ?occ при условии, что она не сохраняется до наступления события, но реально сохраняется после события.

```
(forall (?f ?occ) (iff (achieved ?f ?occ)
(and (holds ?f ?occ)
     (not (prior ?f ?occ)))))
```

19.5.2 falsified

Флюента ?f является фальсифицированной (теряет значение) в результате события ?occ при условии, что она сохраняет значение до наступления события, но теряет его после наступления события.

```
(forall (?f ?occ) (iff (falsified ?f ?occ)
(and (not (holds ?f ?occ))
     (prior ?f ?occ))))
```

19.5.3 irreversible

Флюента является необратимой, если она никогда не фальсифицируется.

```
(forall (?f) (iff (irreversible ?f)
(not (exists (?occ)
     (falsified ?f ?occ)))))
```

19.5.4 nonachievable

Флюента является неархивируемой, если она никогда не принимает никаких значений.

```
(forall (?f) (iff (unachievable ?f)
(not (exists (?occ)
     (achieved ?f ?occ)))))
```

19.5.5 bounded

```
(forall (?f) (iff (bounded ?f)
(exists (?occ1 ?occ2)
     (and (achieved ?f ?occ1)
          (falsified ?f ?occ2)))))
```

19.6 Грамматика описаний процесса для дерева флюента

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для дерева флюента.

20 Распределение комплексных действий

Данный раздел характеризует все определения, обусловленные распределением комплексных действий. Критерий, используемый для классификации указанных действий: ограничено или нет в некотором смысле действительное событие, заключающееся в выполнении комплексного действия.

20.1 Примитивная лексика распределения комплексных действий

Лексика распределения комплексных действий не требует примитивных соотношений.

20.2 Описываемые соотношения распределения комплексных действий

В данном подразделе определены следующие соотношения:
- (profile ?occ ?a);
- (root_equiv ?a ?occ1 ?occ2);
- (universal ?a);
- (restricted ?a);
- (constrained ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

20.3 Теории ядра, обусловленные распределением комплексных действий

Для данных дефинициональных расширений необходимы:
- occtree.th;
- psl_core.th.

20.4 Дефинициональные расширения, обусловленные распределением комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

20.5 Определения распределения комплексных действий

Для распределения комплексных действий определены нижеследующие понятия.

20.5.1 profile

Событие ?occ является элементом профиля для заданного действия ?a при условии, что данное событие ?occ является начальным событием на ветви дерева событий, которое, в свою очередь, является событийно-изоморфическим для ветви некоторого дерева действий, соответствующего действию ?a.

```
(forall (?occ ?a) (iff (profile ?occ ?a)
(exists (?occ1 ?occ2 ?a1)
        (and (occurrence_of ?occ1 ?a)
                (occurrence_of ?occ ?a1)
                (occurrence_of ?occ2 ?a1)
                (root_occ ?occ2 ?occ1)
                (forall (?occ3 ?a2)
                        (implies (and (subactivity_occurrence ?occ3 ?occ1)
                                (occurrence_of ?occ3 ?a2))
                            (exists (?occ4)
                                (and (occurrence_of ?occ4 ?a2)
                                        (precedes ?occ ?occ4)))))))))
```

20.5.2 root_equiv

Два события ?occ1 и ?occ2 на дереве событий эквивалентны корнями по отношению к некоторому действию ?a тогда и только тогда, когда существуют события ?a, предшествующие как событию ?occ1, так и событию ?occ2.

```
(forall (?a ?occ1 ?occ2) (iff (root_equiv ?a ?occ1 ?occ2)
(implies (and (profile ?occ1 ?a)
        (profile ?occ2 ?a))
    (iff  (root ?occ1 ?a)
        (root ?occ2 ?a)))))
```

20.5.3 universal

Некоторое действие является универсальным, если возможно его выполнение когда-либо.

```
(forall (?a) (iff (universal ?a)
(forall (?occ1 ?occ2)
```

```
(implies (and (profile ?occ1 ?a)
              (profile ?occ2 ?a))
         (root_equiv ?a ?occ1 ?occ2)))))
```

20.5.4 local

Некоторое действие является локальным, если каждое его выполнение порождает события, не эквивалентные корнями.

```
(forall (?a) (iff (local ?a)
(forall (?a1)
    (exists  (?occ1 ?occ2)
             (and (occurrence_of ?occ1 ?a1)
                  (occurrence_of ?occ2 ?a1)
                  (profile ?occ1 ?a)
                  (profile ?occ2 ?a)
                  (not (root_equiv ?a ?occ1 ?occ2)))))))
```

20.5.5 restricted

Некоторое действие является связанным, если для него существуют другие действия, эквивалентные корнями.

```
(forall (?a) (iff (restricted ?a)
(and (exists (?a1)
       (forall (?occ1 ?occ2)
              (implies (and (occurrence_of ?occ1 ?a1)
                            (occurrence_of ?occ2 ?a1)
                            (profile ?occ1 ?a)
                            (profile ?occ2 ?a))
                       (root_equiv ?a ?occ1 ?occ2))))
     (exists (?a2 ?occ3 ?occ4)
       (and (occurrence_of ?occ3 ?a2)
              (occurrence_of ?occ4 ?a2)
              (profile ?occ3 ?a)
              (profile ?occ4 ?a)
              (not (root_equiv ?a ?occ3 ?occ4)))))))
```

20.6 Грамматика описаний процесса распределения комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для распределения комплексных действий.

```
< universal_axiom > ::= (forall (?s)
      (implies (legal ?s)
               < distribution_formula>))
< restricted_axiom > ::= (forall (?s)
               (implies < successor_axiom >
                    < distribution_formula>))
< distribution_formula > ::= (exists (?occ)
                    (and (occurrence ?occ ?a)
                         (root_occ ?s ?occ)))
```

21 Основанное на состоянии распределение комплексных действий

Данный раздел характеризует все определения, обусловленные основанным на состоянии распределением комплексных действий. Критерий, используемый для классификации указанных действий: зависит или нет действительное событие, заключающееся в выполнении комплексных действий, от состояния до выполнения указанных действий.

21.1 Примитивная лексика основанного на состоянии распределения комплексных действий

Лексика основанного на состоянии распределения комплексных действий не требует примитивных соотношений.

21.2 Описываемые соотношения основанного на состоянии распределения комплексных действий

В данном подразделе определены следующие соотношения:
- (trigger ?a);
- (partial_trigger ?a);
- (nontrigger ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

21.3 Теории, обусловленные основанным на состоянии распределением комплексных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

21.4 Дефинициональные расширения, обусловленные основанным на состоянии распределением комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

21.5 Определения основанного на состоянии распределения комплексных действий

Для основанного на состоянии распределения комплексных действий определены нижеследующие понятия.

21.5.1 trigger

Некоторое действие является инициирующим при условии, что если когда-либо два события согласуются по состоянию, то они согласуются и по событиям.

```
(forall (?a) (iff (trigger ?a)
(forall (?occ1 ?occ2)
    (implies (and (profile ?occ1 ?a)
                  (profile ?occ2 ?a)
                  (state_equiv ?occ1 ?occ2))
             (root_equiv ?a ?occ1 ?occ2)))))
```

21.5.2 partial_trigger

Некоторое действие является частично инициирующим при условии, что существуют события, согласующиеся как по состоянию, так и по событиям.

```
(forall (?a) (iff (partial_trigger ?a)
(and (exists (?occ1)
       (forall (?occ2)
                (implies (and (profile ?occ1 ?a)
                              (profile ?occ2 ?a)
                              (state_equiv ?occ1 ?occ2))
                         (root_equiv ?a ?occ1 ?occ2))))
     (exists (?occ3 ?occ4)
       (and (profile ?occ3 ?a)
            (profile ?occ4 ?a)
            (state_equiv ?occ3 ?occ4)
            (not (root_equiv ?a ?occ3 ?occ4)))))))
```

21.5.3 nontrigger

Некоторое действие является неинициирующим тогда и только тогда, когда сохраняющая события перестановка переменных является тривиальной.

```
(forall (?a) (iff (nontrigger ?a)
(forall (?occ1)
    (implies (profile ?occ1 ?a)
             (exists (?occ2)
```

```
(and (profile ?occ2 ?a)
     (state_equiv ?occ1 ?occ2)
     (not (root_equiv ?a ?occ1 ?occ2)))))))
```

21.6 Грамматика описаний процесса для основанного на состоянии распределения комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для основанного на состоянии распределения комплексных действий.

```
< trigger_activity > ::=    (forall (?s)
                            (implies < simple_state_axiom >
                                 < distribution_formula>))
< partial_trigger > ::=     (forall (?s < variable >+)
                            (implies < state_axiom >
                                 < distribution_formula>))
```

22 Временное распределение комплексных действий

Данный раздел характеризует все определения, обусловленные временным распределением комплексных действий. Критерий, используемый для классификации указанных действий: зависит или нет действительное выполнение некоторого комплексного действия от времени выполнения указанного действия.

22.1 Примитивная лексика временного распределения комплексных действий

Лексика временного распределения комплексных действий не требует примитивных соотношений.

22.2 Описываемые соотношения временного распределения комплексных действий

В данном подразделе определены следующие соотношения:
- (launch ?a);
- (partial_launch ?a);
- (rigid_launch ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

22.3 Теории, обусловленные временным распределением комплексных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

22.4 Дефинициональные расширения, обусловленные временным распределением комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

22.5 Определения временного распределения комплексных действий

Для временного распределения комплексных действий определены нижеследующие понятия.

22.5.1 launch

Некоторое действие является запускающим при условии: если и только если когда-либо два события согласуются по времени начала отсчета, то они согласуются и по выполнению указанных действий.

```
(forall (?a) (iff (launch ?a)
(forall (?occ1 ?occ2)
(implies (and (profile ?occ1 ?a)
     (profile ?occ2 ?a)
          (begin_equiv ?occ1 ?occ2))
     (root_equiv ?a ?occ1 ?occ2)))))
```

22.5.2 partial_launch

Некоторое действие является частично запускающим тогда и только тогда, когда существуют события, согласующиеся как по времени начала отсчета, так и по выполнению указанных действий.

```
(forall (?a) (iff (partial_launch ?a)
(and (exists (?occ1)
        (forall (?occ2)
                (implies (and (profile ?occ1 ?a)
                              (profile ?occ2 ?a)
                              (begin_equiv ?occ1 ?occ2))
                         (root_equiv ?a ?occ1 ?occ2))))
     (exists (?occ3 ?occ4)
        (and (profile ?occ3 ?a)
(profile ?occ4 ?a)
                (begin_equiv ?occ3 ?occ4))
        (not (root_equiv ?a ?occ3 ?occ4)))))))
```

22.5.3 rigid_launch

Некоторое действие является жестко запускающим тогда и только тогда, когда сохраняющая событие по временной шкале перестановка является тривиальной.

```
(forall (?a) (iff (rigid_launch ?a)
(forall (?occ1)
        (implies (profile ?occ1 ?a)
                 (exists (?occ2)
                         (and (begin_equiv ?occ1 ?occ2)
                              (not (root_equiv ?a ?occ1 ?occ2)))))))
```

22.6 Грамматика описаний процесса для временного распределения комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для временного распределения комплексных действий.

```
< simple_launch_activity > ::=          (forall (?s)
                                    (implies < simple_time_axiom >
                                        < distribution_formula>))
< partial_launch_activity > ::= (forall (?s < variable >+)
                                    (implies < time_axiom >
                                        < distribution_formula>))
```

23 Смешанное распределение комплексных действий

Данный раздел характеризует все определения, обусловленные смешанным распределением комплексных действий. Критерий, используемый для классификации указанных действий: зависят или нет допустимые события от состояния до выполнения действий и от времени выполнения данных действий.

23.1 Примитивная лексика смешанного распределения комплексных действий

Лексика смешанного распределения комплексных действий не требует примитивных соотношений.

23.2 Описываемые соотношения смешанного распределения комплексных действий

В данном подразделе определены следующие соотношения:

- (conditional_trigger ?a);
- (partial_conditional_trigger ?a);
- (unconditional_trigger ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

23.3 Теории ядра, обусловленные смешанным распределением комплексных действий

Для данной теории необходимы:

- occtree.th;
- psl_core.th.

23.4 Дефинициональные расширения, обусловленные смешанным распределением комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

23.5 Определения смешанного распределения комплексных действий

Определены нижеследующие понятия для смешанного распределения комплексных действий.

23.5.1 conditional_launch

Некоторое действие является условно запускающим действием при условии, что если и только если когда-либо два события согласуются по состоянию и по времени начала отсчета, то они согласуются и по событиям.

```
(forall (?a) (iff (conditional_launch ?a)
(forall (?occ1 ?occ2)
        (implies (and (profile ?occ1 ?a)
                      (profile ?occ2 ?a)
                      (state_equiv ?occ1 ?occ2)
                      (begin_equiv ?occ1 ?occ2))
                 (root_equiv ?a ?occ1 ?occ2)))))
```

23.5.2 partial_conditional_launch

Некоторое действие является частично условно запускающим действием при условии, что если и только если существуют события, согласованные по состоянию и по времени начала отсчета, то они также согласуются и по событиям.

```
(forall (?a) (iff (partial_conditional_launch ?a)
(and (exists (?occ1)
        (forall (?occ2)
                (implies (and (profile ?occ1 ?a)
                              (profile ?occ2 ?a)
                              (state_equiv ?occ1 ?occ2)
                              (begin_equiv ?occ1 ?occ2))
                         (root_equiv ?a ?occ1 ?occ2))))
     (exists (?occ3 ?occ4)
        (and (profile ?occ3 ?a)
             (profile ?occ4 ?a)
             (state_equiv ?occ3 ?occ4)
             (begin_equiv ?occ3 ?occ4)
             (not (root_equiv ?a ?occ3 ?occ4)))))))
```

23.5.3 unconditional_launch

Некоторое действие является безусловно запускающим тогда и только тогда, когда сохраняющая событие перестановка, сохраняющая также состояние и время, является тривиальной.

```
(forall (?a) (iff (unconditional_launch ?a)
(forall (?occ1)
        (implies (profile ?occ1 ?a)
                 (exists (?occ2)
                         (and (profile ?occ2 ?a)
                              (state_equiv ?occ1 ?occ2)
                              (begin_equiv ?occ1 ?occ2)
                              (not (root_equiv ?a ?occ1 ?occ2)))))))
```

23.6 Грамматика описаний процесса для смешанного распределения комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для смешанного распределения комплексных действий.

```
< conditional_launch_activitiy > ::= (forall (?s ?s2)
                                       (implies < simple_mix_axiom >
                                                < distribution_formula>))
```

```
< partial_conditional_launch > ::= (forall (?s ?s2 < variable >+)
                                    (implies < mix_formula >
                                        < distribution_formula>))
```

24 Вариация комплексных действий

Данный раздел характеризует все определения, обусловленные вариацией комплексных действий. Критерий, используемый для классификации указанных действий: ограничены или нет события некоторого комплексного действия.

24.1 Примитивная лексика вариации комплексных действий

Лексика вариации комплексных действий не требует примитивных соотношений.

24.2 Описываемые соотношения вариации комплексных действий

В данном подразделе определены следующие соотношения:
- (min_equiv ?occ1 ?occ2 ?a);
- (uniform ?a);
- (variegated ?a);
- (multiform ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

24.3 Теории, обусловленные вариацией комплексных действий

Для данных теорий необходимы:
- occtree.th;
- psl_core.th.

24.4 Дефинициональные расширения, обусловленные вариацией комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

24.5 Определения вариации комплексных действий

Для вариации комплексных действий определены нижеследующие понятия.

24.5.1 min_equiv

Два минимальных дерева действий для события ?a являются изоморфическими, если каждое из них может быть встроено в другое как составляющее дерево.

```
(forall (?occ1 ?occ2 ?a) (iff (min_equiv ?occ1 ?occ2 ?a)
(and (subtree_embed ?occ1 ?occ2 ?a)
     (subtree_embed ?occ2 ?occ1 ?a))))
```

24.5.2 uniform

Некоторое действие является однородным, если все его минимальные деревья действий являются изоморфическими.

```
(forall (?a) (iff (uniform ?a)
(forall (?occ1 ?occ2)
     (implies (and (root ?occ1 ?a)
                   (root ?occ2 ?a)
              (min_equiv ?occ1 ?occ2 ?a)))))
```

24.5.3 variegated

Некоторое действие ?a является варьированным, если все его минимальные деревья действий (с событийно-эквивалентными корневыми событиями) являются изоморфическими.

```
(forall (?a) (iff (variegated ?a)
(and (exists (?a1)
     (forall (?occ1 ?occ2)
```

```
            (implies (and (occurrence_of ?occ1 ?a1)
                          (occurrence_of ?occ2 ?a1)
                          (root ?occ1 ?a)
                          (root ?occ2 ?a))
                     (min_equiv ?occ1 ?occ2 ?a))))
    (exists (?a2 ?occ3 ?occ4)
        (and (occurrence_of ?occ3 ?a2)
             (occurrence_of ?occ4 ?a2)
             (root ?occ3 ?a)
             (root ?occ4 ?a)
             (not (min_equiv ?occ3 ?occ4 ?a)))))))
```

24.5.4 multiform

Некоторое действие является многообразным, если существуют неизоморфические деревья действий с событийно-эквивалентными корневыми событиями.

```
(forall (?a) (iff (multiform ?a)
(forall (?a1 ?occ1)
    (implies (and (root ?occ1 ?a)
                  (occurrence_of ?occ1 ?a1))
             (exists (?occ1 ?occ2)
                 (and (occurrence_of ?occ1 ?a1)
                      (occurrence_of ?occ2 ?a1)
                      (root ?occ1 ?a)
                      (root ?occ2 ?a)
                      (not (min_equiv ?occ1 ?occ2 ?a))))))))
```

24.6 Грамматика описаний процесса для вариации комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для вариации комплексных действий.

```
< uniform_axiom > ::= (forall (?s1 ?s2)
                      (iff (do < term > ?s1 ?s2)
                          < variation_formula >+))
< variegated_axiom > ::= (forall (?s ?s2 < variable >+)
                      (iff (do ?a ?s1 ?s2)
                          < occ_variation_formula >+))
< variation_formula > ::= (exists (< variable >*)
                          (and (subactivity < term > < term >)
                              (do < term > ?s1 ?s2)))
< occ_variation_formula > ::= (implies < successor_axiom >
                           < variation_formula >)
```

25 Основанная на состоянии вариации комплексных действий

Данный раздел характеризует все определения вариации комплексных действий, основанной на состоянии. Критерий, используемый для классификации указанных действий: зависят или нет события от состояния до выполнения указанных действий.

25.1 Примитивная лексика основанной на состоянии вариации комплексных действий

Лексика основанной на состоянии вариации комплексных действий не требует примитивных соотношений.

25.2 Описываемые соотношения основанной на состоянии вариации комплексных действий

В данном подразделе определены следующие соотношения:

- (conditional ?a);
- (partial_conditional ?a);

- (rigid_conditional ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

25.3 Теории, обусловленные основанной на состоянии вариации комплексных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

25.4 Дефинициональные расширения, обусловленные основанной на состоянии вариации комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

25.5 Определения основанной на состоянии вариации комплексных действий

Для основанной на состоянии вариации комплексных действий определены нижеследующие понятия.

25.5.1 conditional

Некоторое действие является условным при следующем ограничении: какие-либо два его минимальных дерева действий являются изоморфическими, если корни согласованы по состоянию.

```
(forall (?a) (iff (conditional ?a)
(forall (?occ1 ?occ2)
    (implies (and (root ?occ1 ?a)
                  (root ?occ2 ?a)
                  (state_equiv ?occ1 ?occ2))
             (min_equiv ?occ1 ?occ2 ?a)))))
```

25.5.2 partial_conditional

Некоторое действие является частично условным, если существуют изоморфические минимальные деревья действий, согласованные по состоянию.

```
(forall (?a) (iff (partial_conditional ?a)
(and (exists (?occ1)
       (forall (?occ2)
          (implies (and (root ?occ1 ?a)
                        (root ?occ2 ?a)
                        (state_equiv ?occ1 ?occ2))
                   (min_equiv ?occ1 ?occ2 ?a))))
     (exists (?occ3 ?occ4)
       (and (root ?occ3 ?a)
            (root ?occ4 ?a)
            (state_equiv ?occ3 ?occ4)
            (not (min_equiv ?occ3 ?oc4)))))))
```

25.5.3 rigid_conditional

Некоторое действие является жестко условным, если для каждого корневого события минимального дерева действий (для действия ?a) существуют другие события, согласованные по состоянию, являющемуся корнем неизоморфического минимального дерева действий.

```
(forall (?a) (iff (rigid_conditional ?a)
(forall (?occ1)
    (exists (?occ2)
        (and (root ?occ1 ?a)
             (root ?occ2 ?a)
             (state_equiv ?occ1 ?occ2)
             (not (min_equiv ?occ1 ?occ2 ?a)))))))
```

25.6 Грамматика описаний процесса для основанной на состоянии вариации комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для основанной на состоянии вариации комплексных действий.

```
< conditional_activity > ::= (forall (?s ?s2)
                        (iff (do ?a ?s ?s2)
                                < simple_conditional >))
< partial_conditional > ::= (forall (?s ?s2 < variable >+)
                        (iff (do ?a ?s ?s2)
                                < conditional_formula >))
< simple_conditional > ::= (implies < simple_state_axiom >
                        < variation_formula >)
< conditional_formula > ::= (implies < state_axiom >
                        < variation_formula >)
```

26 Временная вариация комплексных действий

Данный раздел характеризует все определения, обусловленные временной вариацией комплексных действий. Критерий, используемый для классификации указанных действий: зависят или нет события, представляющие собой выполнение составляющих действий некоторого комплексного действия, только от времени выполнения указанного действия.

26.1 Примитивная лексика временной вариации комплексных действий

Лексика временной вариации комплексных действий не требует примитивных соотношений.

26.2 Описываемые соотношения временной вариации комплексных действий

В данном подразделе определены следующие соотношения:
- (time_conditional ?a);
- (partial_time_conditional ?a);
- (rigid_time_conditional ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

26.3 Теории, обусловленные временной вариацией комплексных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

26.4 Дефинициональные расширения, обусловленные временной вариацией комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

26.5 Определения временной вариации комплексных действий

Для предотвращаемых условий выполнения действий определены нижеследующие понятия.

26.5.1 time_conditional

Комплексное действие является временным условным при следующем условии: какие-либо два его минимальных дерева действий являются изоморфическими, если их корни согласуются по времени начала отсчета.

```
(forall (?a) (iff (time_conditional ?a)
(forall (?occ1 ?occ2)
    (implies (and (root ?occ1 ?a)
                  (root ?occ2 ?a)
                  (begin_equiv ?occ1 ?occ2))
             (min_equiv ?occ1 ?occ2 ?a)))))
```

26.5.2 partial_time_conditional

Комплексное действие является частично временным условным тогда и только тогда, когда существуют сохраняющие событие перестановки.

```
(forall (?a) (iff (partial_time_conditional ?a)
(and (exists (?occ1)
       (forall (?occ2)
              (implies (and (root ?occ1 ?a)
                            (root ?occ2 ?a)
                            (begin_equiv ?occ1 ?occ2))
                       (min_equiv ?occ1 ?occ2 ?a))))
     (exists (?occ3 ?occ4)
       (and (root ?occ3 ?a)
            (root ?occ4 ?a)
            (begin_equiv ?occ3 ?occ4)
            (not (min_equiv ?occ3 ?oc4)))))))
```

26.5.3 rigid_time_conditional

Комплексное действие является жестким временным условным тогда и только тогда, когда сохраняющая событие по временной шкале перестановка является тривиальной.

```
(forall (?a) (iff (rigid_time_conditional ?a)
(forall (?occ1)
       (exists (?occ2)
              (and (root ?occ1 ?a)
                   (root ?occ2 ?a)
                   (begin_equiv ?occ1 ?occ2)
                   (not (min_equiv ?occ1 ?occ2 ?a)))))))
```

26.6 Грамматика описаний процесса для временной вариации комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для временной вариации комплексных действий.

```
< time_conditional_activity > ::= (forall (?s ?s2)
                                  (iff (do ?a ?s ?s2)
                                       < simple_time_conditional >))
< partial_time_conditional > ::= (forall (?s ?s2)
                                  (iff (do ?a ?s ?s2)
                                       < time_conditional_formula >))
< simple_time_conditional > ::= (implies < simple_time_axiom >
                                  < variation_formula >)
< time_conditional_formula > ::= (implies < time_axiom >
                                  < variation_formula >)
```

27 Смешанная вариация комплексных действий

Данный раздел характеризует все определения, обусловленные смешанной вариацией комплексных действий. Критерий, используемый для классификации указанных действий: зависят или нет события, заключающиеся в выполнении составляющих действий некоторого комплексного действия, от состояния до выполнения действий и от времени выполнения данных действий.

27.1 Примитивная лексика смешанной вариации комплексных действий

Лексика смешанной вариации комплексных действий не требует примитивных соотношений.

27.2 Описываемые соотношения смешанной вариации комплексных действий

В данном подразделе определены следующие соотношения:

- (mixed_conditional ?a);
- (partial_mixed_conditional ?a);
- (rigid_mixed_conditional ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

27.3 Теории, обусловленные смешанной вариацией комплексных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

27.4 Дефинициональные расширения, обусловленные смешанной вариацией комплексных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

27.5 Определения смешанной вариации комплексных действий

Для смешанной вариации комплексных действий определены нижеследующие понятия.

27.5.1 mixed_conditional

Некоторое действие является смешанным условным тогда и только тогда, когда какие-либо два его минимальных дерева действий являются изоморфическими и их корни согласуются по состоянию и времени начала действия.

```
(forall (?a) (iff (mixed_conditional ?a)
(forall (?occ1 ?occ2)
      (implies (and (root ?occ1 ?a)
                    (root ?occ2 ?a)
                    (begin_equiv ?occ1 ?occ2)
                    (state_equiv ?occ1 ?occ2))
               (min_equiv ?occ1 ?occ2 ?a)))))
```

27.5.2 partial_mixed_conditional

Некоторое действие является частично смешанным условным тогда и только тогда, когда существуют минимальные изоморфические деревья действий и их корни согласованы по состоянию и по времени начала действий.

```
(forall (?a) (iff (partial_mixed_conditional ?a)
(and (exists (?occ1)
       (forall (?occ2)
              (implies (and (root ?occ1 ?a)
                            (root ?occ2 ?a)
                            (begin_equiv ?occ1 ?occ2)
                            (state_equiv ?occ1 ?occ2))
                       (min_equiv ?occ1 ?occ2 ?a))))
     (exists (?occ3 ?occ4)
       (and (root ?occ3 ?a)
          (root ?occ4 ?a)
          (begin_equiv ?occ1 ?occ2)
          (state_equiv ?occ3 ?occ4)
          (not (min_equiv ?occ3 ?oc4)))))))
```

27.5.3 rigid_mixed_conditional

Некоторое действие является жестким смешанным условным тогда и только тогда, когда сохраняющие событие перестановки, моменты времени и состояние являются тривиальными.

```
(forall (?a) (iff (rigid_mixed_conditional ?a)
(forall (?occ1)
       (exists (?occ2)
              (and (root ?occ1 ?a)
                   (root ?occ2 ?a)
                   (begin_equiv ?occ1 ?occ2)
                   (state_equiv ?occ1 ?occ2)
                   (not (min_equiv ?occ1 ?occ2 ?a)))))))
```

27.6 Грамматика описаний процесса для смешанной вариации комплексных действий

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для предотвращаемых условий выполнения действий.

```
< mixed_conditional > ::= (forall (?s ?s2)
                    (iff (do ?a ?s ?s2)
                         < simple_mix_conditional >))
< partial_mixed_conditional > ::= (forall (?s ?s2 < variable >+)
                    (iff (do ?a ?s ?s2)
                         < mix_conditional_formula >))
< simple_mix_conditional > ::= (implies < simple_mix_axiom >
                    < variation_formula >)
< mixed_conditional_formula > ::= (implies < mix_axiom >
                    < variation_formula >)
```

28 Встроенные действия: планы

Данный раздел характеризует все определения, обусловленные встроенными действиями: планы. Критерий, используемый для классификации указанных действий: зависят или нет события некоторого комплексного действия от состояния в течение выполнения указанного действия.

28.1 Примитивная лексика встроенных действий: планы

Лексика встроенных действий (планов) не требует примитивных соотношений.

28.2 Описываемые соотношения встроенных действий: планы

В данном подразделе определены следующие соотношения:
- (preventable ?a);
- (possibly_preventable ?a);
- (unpreventable ?a) .

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

28.3 Теории, обусловленные встроенными действиями: планы

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

28.4 Дефинициональные расширения, обусловленные встроенными действиями: планы

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

28.5 Определения встроенных действий: планы

Для встроенных действий (планов) определены нижеследующие понятия.

28.5.1 plan

Некоторое дерево действий с корневым событием ?s является планом тогда и только тогда, когда все согласованные по состоянию события в максимальном встроенном поддереве согласуются между собой.

```
(forall (?s) (iff (plan ?s)
(forall (?a ?s1 ?s2)
    (implies (and (root ?s ?a)
                  (embed_tree ?s1 ?s2 ?s ?a)
                  (state_equiv ?s1 ?s2))
             (subocc_equiv ?s1 ?s2 ?s ?a)))))
```

28.5.2 nondet_plan

Некоторое дерево действий с корневым событием ?s является недетерминированным планом тогда и только тогда, когда существуют согласованные по состоянию события, заключающиеся в выполнении

составляющих действий в максимальном встроенном поддереве, которые также согласованы и между собой.

```
(forall (?s) (iff (nondet_plan ?s)
(exists (?a ?s1 ?s3 ?s4)
        (and (root ?s ?a)
             (forall (?s2)
                  (implies (and (embed_tree ?s1 ?s2 ?s ?a)
                                (state_equiv ?s1 ?s2))
                           (subocc_equiv ?s1 ?s2 ?s ?a))))
             (state_equiv ?s3 ?s4)
             (embed_tree ?s3 ?s4 ?s5 ?a)
             (not (subocc_equiv ?s3 ?s4 ?s5 ?a))))))
```

28.5.3 unplan

Некоторое дерево действий с корневым событием ?s является неплановым тогда и только тогда, когда для каждого согласованного по состоянию события существует другое событие, связанное с составляющим действием, но отличное от указанного.

```
(forall (?s) (iff (unplan ?s)
(forall (?s1 ?a)
        (implies (root ?s ?a)
                 (exists (?s2)
                      (and (embed_tree ?s1 ?s2 ?s ?a)
                           (state_equiv ?s1 ?s2)
                           (not (subocc_equiv ?s1 ?s2 ?s ?a))))))))
```

28.6 Грамматика описаний процесса для встроенных действий: планы

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для встроенных действий: планов.

```
< plan_axiom > ::=      (forall (?s ?occ < variable >*)
                        < simple_plan_axiom >)
< nondet_plan_axiom > ::= (forall (?s ?occ < variable >+)
                        < nondet_plan_axiom >))
< plan_formula > ::= (implies < subocc_formula >
                        < simple_state_axiom >)
< simple_plan_axiom > ::= < plan_formula> |
                        (and < simple_plan_axiom > < simple_plan_axiom >+)
< nondet_plan_formula > ::= (implies < subocc_formula >
                          < state_axiom >)
< nondet_plan_axiom > ::= < nondet_plan_formula> |
                        (and < nondet_plan_axiom > < nondet_plan_axiom >+)
```

29 Встроенные действия: временной разброс

Данный раздел характеризует все определения, обусловленные встроенными действиями: временной разброс. Критерий, используемый для классификации указанных действий: зависят или нет события некоторого комплексного действия от времени выполнения составляющих действий в процессе выполнения указанного комплексного действия.

29.1 Примитивная лексика встроенных действий: временной разброс

Лексика встроенных действий (временной разброс) не требует примитивных соотношений.

29.2 Описываемые соотношения встроенных действий: временной разброс

В данном подразделе определены следующие соотношения:

- (preventable ?a);
- (possibly_preventable ?a);
- (unpreventable ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

29.3 Теории, обусловленные встроенными действиями: временной разброс

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

29.4 Дефинициональные расширения, обусловленные встроенными действиями: временной разброс

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

29.5 Определения встроенных действий: временной разброс

Для встроенных действий (временной разброс) определены нижеследующие понятия.

29.5.1 spread

Некоторое событие ?occ является разбросом тогда и только тогда, когда каждое событие ограничено по времени.

```
(forall (?occ) (iff (spread ?occ)
(forall (?a ?s1 ?s2 ?s3)
    (implies (and (occurrence_of ?occ ?a)
            (root_occ ?s3 ?occ)
            (subactivity_occurrence ?s1 ?occ)
            (iso_occ ?s1 ?s2)
            (embed_tree ?s1 ?s2 ?s3 ?a)
            (begin_equiv ?s1 ?s2))
        (subocc_equiv ?s1 ?s2 ?s3 ?a)))))
```

29.5.2 partial_spread

Некоторое событие ?occ является частичным разбросом тогда и только тогда, когда некоторые составляющие действия события ?occ ограничены по времени.

```
(forall (?occ) (iff (partial_spread ?occ)
(exists (?a ?s ?s1 ?s3 ?s4)
    (and (occurrence_of ?occ ?a)
            (root_occ ?s ?occ)
            (subactivity_occurrence ?s1 ?occ)
            (forall (?s2)
                (implies (and (iso_occ ?s1 ?s2)
                        (embed_tree ?s1 ?s2 ?s ?a)
                        (begin_equiv ?s1 ?s2))
                    (subocc_equiv ?s1 ?s2 ?s ?a))))
            (subactivity_occurrence ?s3 ?occ)
            (iso_occ ?s3 ?s4)
            (begin_equiv ?s3 ?s4)
            (embed_tree ?s3 ?s4 ?s ?a)
            (not (subocc_equiv ?s3 ?s4 ?s ?a))))))
```

29.5.3 tight

Некоторое событие occ? является частично плотным тогда и только тогда, когда ни одно из составляющих действий события ?occ не ограничено по времени.

```
(forall (?occ) (iff (tight ?occ)
(forall (?s ?s1 ?a)
    (implies (and (occurrence_of ?occ ?a)
            (root_occ ?s ?occ)
            (subactivity_occurrence ?s1 ?occ))
        (exists (?s2)
            (and (iso_occ ?s1 ?s2)
                    (embed_tree ?s1 ?s2 ?s ?a)
                    (begin_equiv ?s1 ?s2)
                    (not (subocc_equiv ?s1 ?s2 ?s ?a))))))))
```

29.6 Грамматика описаний процесса для встроенных действий: временной разброс

Нижеследующие грамматические утверждения определяют описания процесса, установленные в KIF для встроенных действий: временной разброс.

```
< spread_axiom > ::= (forall (?s ?occ < variable >*)
                        < simple_spread_axiom >)
< partial_spread_axiom > ::= (forall (?s ?occ < variable >+)
                        < partial_spread_axiom >))
< spread_formula > ::= (implies < subocc_formula >
                        < simple_time_axiom >)
< simple_spread_axiom > ::= < spread_formula> |
                        (and < simple_spread_axiom > < simple_spread_axiom >+)
< partial_spread_formula > ::= (implies < subocc_formula >
                            < time_axiom >)
< partial_spread_axiom > ::= < partial_spread_formula> |
                        (and < partial_spread_axiom > < partial_spread_axiom >+)
```

30 Вариация неделимых вверх действий

Данный раздел характеризует все определения, обусловленные вариацией неделимых вверх действий.

30.1 Примитивная лексика вариации неделимых вверх действий

Лексика вариации неделимых вверх действий не требует примитивных соотношений. Критерий, используемый для классификации указанных действий: зависят или нет входные условия неделимых действий от входных условий супердействий для указанного действия.

30.2 Описываемые соотношения вариации неделимых вверх действий

В данном подразделе определены следующие соотношения:
- (state_filter ?a);
- (partial_state_filter ?a);
- (rigid_state_filter ?a);
- (time_filter ?a);
- (partial_time_filter ?a);
- (rigid_time_filter ?a);
- (state_nonfilter ?a);
- (partial_state_nonfilter ?a);
- (rigid_state_nonfilter ?a);
- (time_nonfilter ?a);
- (partial_time_nonfilter ?a);
- (rigid_time_nonfilter ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

30.3 Теории, обусловленные вариацией неделимых вверх действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

30.4 Дефинициональные расширения, обусловленные вариацией неделимых вверх действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

30.5 Определения вариации неделимых вверх действий

Для вариации неделимых вверх действий определены нижеследующие понятия.

30.5.1 state_filter

Некоторое неделимое комплексное действие ?a является фильтром состояния при условии: если и только если какие-либо и когда-либо допустимые события для действия ?a согласованы по состоянию, то все составляющие действия ?a могут действительно иметь место.

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

30.5.2 partial_state_filter

Некоторое неделимое комплексное действие ?a является частичным фильтром состояния тогда и только тогда, когда существуют допустимые события для действия ?a, согласованные по состоянию, в которых все составляющие действия для ?a могут действительно иметь место.

```
(forall (?a) (iff (partial_state_filter ?a)
(and (exists (?s1)
       (forall (?s2)
              (implies (and (subactivity ?a1 ?a)
                            (poss ?a ?s1)
                            (poss ?a ?s2)
                            (state_equiv ?s1 ?s2))
                       (poss_equiv ?a1 ?s1 ?s2))))
     (exists (?a2 ?s3 ?s4)
       (and (subactivity ?a2 ?a)
            (poss ?a ?s3)
            (poss ?a ?s4)
            (state_equiv ?s3 ?s4)
       (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.3 rigid_state_filter

Некоторое неделимое комплексное действие ?a является жестким фильтром состояния при условии: если и только если какие-либо и когда-либо допустимые события для действия ?a согласованы по состоянию, то существуют составляющие действия для ?a, которые не могут происходить (случаться).

```
(forall (?a) (iff (rigid_state_filter ?a)
(forall (?s1)
    (exists (?a1 ?s2)
           (and (state_equiv ?s1 ?s2)
                (poss ?a ?s1)
                (poss ?a ?s2)
                (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.5.4 time_filter

Некоторое неделимое комплексное действие ?a является фильтром по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласуются по времени начала отсчета, то все составляющие действия для ?a могут действительно происходить (случаться).

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

30.5.5 partial_time_filter

Некоторое неделимое комплексное действие ?a является частичным фильтром по времени тогда и только тогда, когда существуют допустимые события для ?a, согласованные по времени начала отсчета, в которых все составляющие действия для ?a могут действительно происходить (случаться).

```
(forall (?a) (iff (partial_time_filter ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (and (subactivity ?a1 ?a)
                              (poss ?a ?s1)
                              (poss ?a ?s2)
                              (begin_equiv ?s1 ?s2))
                         (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and (subactivity ?a2 ?a)
          (poss ?a ?s3)
          (poss ?a ?s4)
          (begin_equiv ?s3 ?s4)
          (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.6 rigid_time_filter

Некоторое неделимое комплексное действие ?a является жестким фильтром по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласуются по времени начала отсчета, то существуют составляющие действия для ?a, которые не могут происходить (случаться).

```
(forall (?a) (iff (rigid_time_filter ?a)
(forall (?s1)
        (exists (?a1 ?s2)
                (and (begin_equiv ?s1 ?s2)
                     (poss ?a ?s1)
                     (poss ?a ?s2)
                     (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.5.7 state_nonfilter

Некоторое неделимое комплексное действие ?a не является фильтром состояния при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то все события для составляющих действий ?a согласованы по допустимости.

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

30.5.8 partial_state_nonfilter

Некоторое неделимое комплексное действие ?a не является частичным фильтром по состоянию тогда и только тогда, когда существуют допустимые события для ?a, согласованные по состоянию, в которых все составляющие действия для ?a согласованы по допустимости.

```
(forall (?a) (iff (partial_state_nonfilter ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (and (subactivity ?a1 ?a)
                              (not (poss ?a ?s1))
                              (not (poss ?a ?s2))
                              (state_equiv ?s1 ?s2))
                         (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and (subactivity ?a2 ?a)
                (not (poss ?a ?s3))
                (not (poss ?a ?s4))
                (state_equiv ?s3 ?s4)
                (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.9 rigid_state_nonfilter

Некоторое неделимое комплексное действие ?a не является жестким фильтром по состоянию при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то существуют составляющие действия для ?a, не согласованные по допустимости.

```
(forall (?a) (iff (rigid_state_nonfilter ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and (state_equiv ?s1 ?s2)
            (not (poss ?a ?s1))
            (not (poss ?a ?s2))
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.5.10 time_nonfilter

Некоторое неделимое комплексное действие ?a не является фильтром по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласуются по времени начала отсчета, то все события для составляющих действий по ?a согласованы по допустимости.

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

30.5.11 partial_time_nonfilter

Некоторое неделимое комплексное действие ?a не является частичным фильтром по времени тогда и только тогда, когда существуют допустимые события для ?a, согласованные по времени начала отсчета, в которых все составляющие действия для ?a согласованы по допустимости.

```
(forall (?a) (iff (partial_time_nonfilter ?a)
(and (exists (?s1)
    (forall (?s2)
        (implies (and (subactivity ?a1 ?a)
                (not (poss ?a ?s1))
                (not (poss ?a ?s2))
                (begin_equiv ?s1 ?s2))
            (poss_equiv ?a1 ?s1 ?s2))))
  (exists (?a2 ?s3 ?s4)
    (and (subactivity ?a2 ?a)
        (not (poss ?a ?s3))
        (not (poss ?a ?s4))
        (begin_equiv ?s3 ?s4)
        (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.12 rigid_time_nonfilter

Некоторое неделимое комплексное действие ?a не является жестким фильтром по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласуются по времени начала отсчета, то существуют составляющие действия для ?a, не согласованные по допустимости.

```
(forall (?a) (iff (rigid_time_nonfilter ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and (begin_equiv ?s1 ?s2)
            (not (poss ?a ?s1))
            (not (poss ?a ?s2))
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.6 Грамматика описаний процесса для вариации неделимых вверх действий

Грамматика описаний технологического процесса для классов действий, описываемых для вариации эффекта затирания полезных данных, эквивалентна грамматике общего описания процесса, установленной ИСО 18629-11.

31 Вариация неделимых вниз действий

Данный раздел характеризует все определения, обусловленные вариацией неделимых вниз действий. Критерий, используемый для классификации указанных действий: зависят или нет входные условия неделимого комплексного действия от входных условий составляющих его действий.

31.1 Примитивная лексика вариации неделимых вниз действий

Лексика вариации неделимых вниз действий не требует примитивных соотношений.

31.2 Описываемые соотношения вариации неделимых вниз действий

В данном подразделе определены следующие соотношения:
- (state_ideal ?a);
- (partial_state_ideal ?a);
- (rigid_state_ideal ?a);
- (time_ideal ?a);
- (partial_time_ideal ?a);
- (rigid_time_ideal ?a);
- (state_nonideal ?a);
- (partial_state_nonideal ?a);
- (rigid_state_nonideal ?a);
- (time_nonideal ?a);
- (partial_time_nonideal ?a);
- (rigid_time_nonideal ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

31.3 Теории, обусловленные вариацией неделимых вниз действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

31.4 Дефинициональные расширения, обусловленные вариацией неделимых вниз действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

31.5 Определения вариации неделимых вниз действий

Для вариации неделимых вниз действий определены нижеследующие понятия.

31.5.1 state_ideal

Некоторое неделимое комплексное действие ?a является действием с идеальным состоянием при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то все супердействия для ?a согласованы по допустимости.

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

31.5.2 partial_state_ideal

Некоторое неделимое комплексное действие ?a является действием с частично идеальным состоянием тогда и только тогда, когда существуют допустимые события для ?a, согласованные по состоянию, в которых все супердействия для ?a согласованы по допустимости.

```
(forall (?a) (iff (partial_state_ideal ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (and (subactivity ?a ?a1)
                              (poss ?a ?s1)
                              (poss ?a ?s2)
                              (state_equiv ?s1 ?s2))
                         (poss_equiv ?a1 ?s1 ?s2))))
        (exists (?a2 ?s3 ?s4)
           (and (subactivity ?a ?a2)
                (poss ?a ?s3)
                (poss ?a ?s4)
                (state_equiv ?s3 ?s4)
                (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.3 rigid_state_ideal

Некоторое неделимое комплексное действие ?a является действием с жестким идеальным состоянием при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то существуют супердействия для ?a, не согласованные по допустимости.

```
(forall (?a) (iff (rigid_state_ideal ?a)
(forall (?s1)
        (exists (?a1 ?s2)
                (and (state_equiv ?s1 ?s2)
                     (subactivity ?a ?a1)
                     (poss ?a ?s1)
                     (poss ?a ?s2)
                     (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.5.4 time_ideal

Некоторое неделимое комплексное действие ?a является действием, идеальным по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по времени начала отсчета, то все супердействия для ?a согласованы по допустимости.

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

31.5.5 partial_time_ideal

Некоторое неделимое комплексное действие ?a является частично идеальным по времени тогда и только тогда, когда существуют допустимые события для ?a, согласованные по времени начала отсчета, в которых все супердействия для ?a согласованы по допустимости.

```
(forall (?a) (iff (partial_time_ideal ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (and (subactivity ?a ?a1)
                              (poss ?a ?s1)
                              (poss ?a ?s2)
                              (begin_equiv ?s1 ?s2))
                         (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and (subactivity ?a ?a2)
          (poss ?a ?s3)
          (poss ?a ?s4)
          (begin_equiv ?s3 ?s4)
          (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.6 rigid_time_ideal

Некоторое неделимое комплексное действие ?a является жестко идеальным по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то существуют супердействия для ?a, не согласованные по допустимости.

```
(forall (?a) (iff (rigid_time_ideal ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and (begin_equiv ?s1 ?s2)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.5.7 state_nonideal

Некоторое неделимое комплексное действие ?a является неидеальным по состоянию при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то все события для супердействий по ?a согласованы по допустимости.

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

31.5.8 partial_state_nonideal

Некоторое неделимое комплексное действие ?a является частично неидеальным по состоянию тогда и только тогда, когда существуют допустимые события для ?a, согласованные по состоянию, в которых все супердействия по ?a согласуются по допустимости.

```
(forall (?a) (iff (partial_state_nonideal ?a)
(and (exists (?s1)
    (forall (?s2)
        (implies (and (subactivity ?a ?a1)
                (not (poss ?a ?s1))
                (not (poss ?a ?s2))
                (state_equiv ?s1 ?s2))
            (poss_equiv ?a1 ?s1 ?s2))))
  (exists (?a2 ?s3 ?s4)
    (and (subactivity ?a ?a2)
        (not (poss ?a ?s3))
        (not (poss ?a ?s4))
        (state_equiv ?s3 ?s4)
        (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.9 rigid_state_nonideal

Некоторое неделимое комплексное действие ?a является жестко неидеальным по состоянию при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласованы по состоянию, то существуют супердействия по ?a, не согласованные по допустимости.

```
(forall (?a) (iff (rigid_state_nonideal ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and (state_equiv ?s1 ?s2)
          (not (poss ?a ?s1))
          (not (poss ?a ?s2))
          (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.5.10 time_nonideal

Некоторое неделимое комплексное действие ?a является неидеальным по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласуются по времени начала отсчета, то все события для супердействий по ?a согласуются по допустимости.

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

31.5.11 partial_time_nonideal

Некоторое неделимое комплексное действие ?a является частично неидеальным по времени тогда и только тогда, когда существуют допустимые события для ?a, согласованные по времени начала отсчета, в которых все супердействия по ?a согласованы по допустимости.

```
(forall (?a) (iff (partial_time_nonideal ?a)
(and (exists (?s1)
        (forall (?s2)
                (implies (and (subactivity ?a ?a1)
                              (not (poss ?a ?s1))
                              (not (poss ?a ?s2))
                              (begin_equiv ?s1 ?s2))
                         (poss_equiv ?a1 ?s1 ?s2))))
   (exists (?a2 ?s3 ?s4)
      (and (subactivity ?a ?a2)
        (not (poss ?a ?s3))
        (not (poss ?a ?s4))
        (begin_equiv ?s3 ?s4)
        (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.12 rigid_time_nonideal

Некоторое неделимое комплексное действие ?a является жестко неидеальным по времени при условии: если и только если какие-либо и когда-либо допустимые события для ?a согласуются по времени начала отсчета, то существуют супердействия для ?a, не согласованные по допустимости.

```
(forall (?a) (iff (rigid_time_nonideal ?a)
(forall (?s1)
        (exists (?a1 ?s2)
                (and (begin_equiv ?s1 ?s2)
                     (subactivity ?a ?a1)
                     (not (poss ?a ?s1))
                     (not (poss ?a ?s2))
                     (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.6 Грамматика описаний процесса для вариации неделимых вниз действий

Грамматика описаний технологического процесса для классов действий, описываемых в вариации неделимых вниз действий, эквивалентна грамматике общего описания технологического процесса в соответствии с ИСО 18629-11.

32 Входные условия совместных действий

Данный раздел характеризует все определения, обусловленные входными условиями совместных действий. Критерий, используемый для классификации указанных действий: сохраняются или нет входные условия заданного неделимого действия составом совместных действий.

32.1 Примитивная лексика входных условий совместных действий

Лексика входных условий совместных действий не требует примитивных соотношений.

32.2 Описываемые соотношения входных условий совместных действий

В данном подразделе определены следующие соотношения:

- (preserved_filter ?a ?a1 ?s);
- (noninterfering ?a);

- (unterfering ?a);
- (imperial ?a);
- (global_interfere ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

32.3 Теории, обусловленные входными условиями совместных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

32.4 Дефинициональные расширения, обусловленные входными условиями совместных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

32.5 Определения входных условий совместных действий

Для входных условий совместных действий определены нижеследующие понятия.

32.5.1 preserved_filter

Действия ?a и ?a1 являются элементами сохраненного фильтра тогда и только тогда, когда допустимые события какого-либо супердействия для ?a сохранены указанным составом совместных действий с ?a1.

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

32.5.2 noninterfering

Некоторое действие является непрепятствующим тогда и только тогда, когда каждое неделимое действие является элементом сохраненного фильтра с ?a.

```
(forall (?a ?s) (iff (noninterfering ?a ?s)
(forall (?a1)
    (implies (atomic ?a1)
        (preserved_filter ?a ?a1 ?s)))))
```

32.5.3 interfering

Некоторое действие является препятствующим тогда и только тогда, когда существует некоторое неделимое действие, не являющееся элементом некоторого сохраненного фильтра с ?a.

```
(forall (?a ?s) (iff (interfering ?a ?s)
(exists (?a1)
    (and (atomic ?a1)
        (not (preserved_filter ?a ?a1 ?s))))))
```

32.5.4 imperial

Некоторое действие является имперским тогда и только тогда, когда не существует неделимого действия, являющегося элементом сохраненного фильтра с ?a.

```
(forall (?a ?s) (iff (imperial ?a ?s)
(forall (?a1)
    (implies (atomic ?a1)
        (not (preserved_filter ?a ?a1 ?s))))))
```

32.5.5 global_interfere

Некоторое действие является глобально препятствующим тогда и только тогда, когда все события для ?a имеют одинаковые сохраненные фильтры для ?a.

```
(forall (?a ?s) (iff (global_interfere ?a)
(forall (?a1)
    (implies (atomic ?a1)
        (forall (?s1 ?s2)
            (iff (preserved_filter ?a ?a1 ?s1)
                (preserved_filter ?a ?a1 ?s2)))))))
```

32.6 Грамматика описаний процесса для входных условий совместных действий

Грамматика описаний технологического процесса для классов действий, описываемых для вариации входных условий совместных действий, эквивалентна грамматике общего описания технологического процесса в соответствии с ИСО 18629-11.

33 Эффекты совместных действий

Данный раздел характеризует все определения, обусловленные эффектами совместных действий. Критерий, используемый для классификации указанных действий: сохранены или нет эффекты неделимых действий составом совместных действий.

33.1 Примитивная лексика эффектов совместных действий

Лексика эффектов совместных действий не требует примитивных соотношений.

33.2 Описываемые соотношения эффектов совместных действий

В данном разделе определены следующие соотношения:
- (preserved_effect ?a ?a1 ?s);
- (nonclobbering ?a);
- (clobbering ?a);
- (meddling ?a);
- (global_clobber ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

33.3 Теории, обусловленные эффектами совместных действий

Для данной теории необходимы:
- occtree.th;
- psl_core.th.

33.4 Дефинициональные расширения, обусловленные эффектами совместных действий

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

33.5 Определения эффектов совместных действий

Для эффектов совместных действий определены нижеследующие понятия.

33.5.1 preserved_effect

Действия ?a и ?a1 сохраняют эффекты тогда и только тогда, когда указанные эффекты событий для какого-либо супердействия для ?a сохранены указанным составом совместных действий с ?a1.

```
(forall (?a ?a1 ?s) (iff ( preserved_effects ?a ?a1 ?s)
(forall (?a2 ?f)
    (implies (subactivity ?a ?a2)
        (iff (holds ?f (successor ?a2 ?s))
            (holds ?f (successor (conc ?a2 ?a1) ?s))))))
```

33.5.2 nonclobbering

Некоторое действие ?a не затирает полезные данные тогда и только тогда, когда любое неделимое действие сохраняет указанные эффекты для ?a.

```
(forall (?a ?s) (iff ( nonclobbering ?a ?s)
(forall (?a1)
    (implies (atomic ?a1)
        (preserved_effects ?a ?a1 ?s))))
```

33.5.3 clobbering

Некоторое действие затирает полезные данные тогда и только тогда, когда существует некоторое неделимое действие, не сохраняющее эффекты для ?a.

```
(forall (?a ?s) (iff ( clobbering ?a ?s)
(exists (?a1)
        (and (atomic ?a1)
                (not (preserved_effects ?a ?a1 ?s)))))
```

33.5.4 meddling

Некоторое действие является вмешивающимся тогда и только тогда, когда не существует некоторого неделимого действия, сохраняющего эффекты для ?a.

```
(forall (?a ?s) (iff ( meddling ?a ?s)
(forall (?a1)
        (implies (atomic ?a1)
                (not (preserved_effects ?a ?a1 ?s)))))
```

33.5.5 global_clobber

```
(defrelation global_clobber (?a) :=
(forall (?a1)
        (implies (atomic ?a1)
                (forall (?s1 ?s2)
                        (iff (preserved_effects ?a ?a1 ?s1)
                             (preserved_effects ?a ?a1 ?s2))))))
```

33.6 Грамматика описаний процесса для эффектов совместных действий

Грамматика описаний технологического процесса для классов действий, описываемых эффектами совместных действий, эквивалентна грамматике общего описания технологического процесса в соответствии с ИСО 18629-11.

34 Вариация препятствующих входных условий

Данный раздел характеризует все определения, обусловленные вариацией препятствующих входных условий. Критерий, используемый для классификации указанных действий: сохраняются или нет входные условия заданного неделимого действия составом совместных действий, если сохраняется состояние, предшествующее выполнению данных действий.

34.1 Примитивная лексика вариации препятствующих входных условий

Лексика вариации препятствующих входных условий не требует примитивных соотношений.

34.2 Описываемые соотношения вариации препятствующих входных условий

В данном подразделе определены следующие соотношения:

- (state_interfere ?a);
- (partial_interfere ?a);
- (unconditional_interfere ?a);
- (time_interfere ?a);
- (sometime_interfere ?a);
- (rigid_interfere ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

34.3 Теории, обусловленные вариацией препятствующих входных условий

Для данной теории необходимы:

- occtree.th;
- psl_core.th.

34.4 Дефинициональные расширения, обусловленные вариацией препятствующих входных условий

Для данных расширений необходимы:

- occ_precond.def;
- state_precond.def;
- precond.def.

34.5 Определения вариации препятствующих входных условий

Для вариации препятствующих входных условий определены нижеследующие понятия.

34.5.1 state_interfere

Некоторое действие ?a препятствует состоянию тогда и только тогда, когда какие-либо события, согласованные по состоянию, имеют те же сохраненные фильтры.

```
(forall (?a) (iff (state_interfere ?a)
(forall (?a1 ?s1 ?s2)
    (implies (and (atomic ?a1)
                  (state_equiv ?s1 ?s2))
             (iff (preserved_filter ?a1 ?a ?s1)
                  (preserved_filter ?a1 ?a ?s2)))))
```

34.5.2 partial_interfere

Некоторое действие ?a препятствует частично тогда и только тогда, когда существуют события, согласованные по состоянию и имеющие те же сохраненные фильтры.

```
(forall (?a) (iff (partial_interfere ?a)
(and (exists (?s1 ?a1)
       (forall (?s2)
              (implies (and (atomic ?a1)
                            (state_equiv ?s1 ?s2))
                       (iff (preserved_filter ?a1 ?a ?s1)
                            (preserved_filter ?a1 ?a ?s2)))))
     (exists (?a2 ?s3 ?s4)
       (and (atomic ?a2)
            (state_equiv ?s3 ?s4)
            (preserved_filter ?a2 ?a ?s3)
            (not (preserved_filter ?a2 ?a ?s4)))))))
```

34.5.3 unconditional_interfere

Некоторое действие препятствует безусловно тогда и только тогда, когда для каких-либо согласованных по состоянию событий существуют различные сохраненные фильтры.

```
(forall (?a) (iff (unconditional_interfere ?a)
(forall (?s1)
    (exists (?a1 ?s2)
          (and (atomic ?a1)
               (state_equiv ?s1 ?s2)
               (iff (preserved_filter ?a1 ?a ?s1)
                    (not (preserved_filter ?a1 ?a ?s2))))))))
```

34.5.4 time_interfere

Некоторое действие ?a препятствует по времени тогда и только тогда, когда какие-либо согласованные по времени начала отсчета события имеют те же сохраненные фильтры.

```
(forall (?a) (iff (time_interfere ?a)
(forall (?a1 ?s1 ?s2)
    (implies (and (atomic ?a1)
                  (begin_equiv ?s1 ?s2))
             (iff (preserved_filter ?a1 ?a ?s1)
                  (preserved_filter ?a1 ?a ?s2))))))
```

34.5.5 sometime_interfere

Некоторое действие ?a препятствует иногда тогда и только тогда, когда существуют согласованные по времени начала отсчета события, имеющие те же сохраненные фильтры.

```
(forall (?a) (iff (sometime_interfere ?a)
(and (exists (?s1 ?a1)
       (forall (?s2)
              (implies (and (atomic ?a1)
                            (begin_equiv ?s1 ?s2))
                       (iff (preserved_filter ?a1 ?a ?s1)
                            (preserved_filter ?a1 ?a ?s2)))))
```

```
(exists (?a2 ?s3 ?s4)
   (and (atomic ?a2)
        (begin_equiv ?s3 ?s4)
        (preserved_filter ?a2 ?a ?s3)
        (not (preserved_filter ?a2 ?a ?s4)))))
```

34.5.6 rigid_interfere

Некоторое действие препятствует жестко тогда и только тогда, когда для каких-либо согласованных по времени начала отсчета событий существуют другие сохраненные фильтры.

```
(forall (?a) (iff (rigid_interfere ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and (atomic ?a1)
             (begin_equiv ?s1 ?s2)
             (iff (preserved_filter ?a1 ?a ?s1)
                  (not (preserved_filter ?a1 ?a ?s2))))))))
```

34.6 Грамматика описаний процесса для вариации препятствующих входных условий

Грамматика описаний технологического процесса для класса действий, описываемых для вариации препятствующих входных условий, эквивалентна грамматике общего описания технологического процесса в соответствии с ИСО 18629-11.

35 Вариация эффектов затирания полезных данных

Данный раздел характеризует все определения, обусловленные вариацией эффектов затирания полезных данных. Критерий, используемый для классификации указанных действий: сохраняются или нет эффекты заданного неделимого действия составом совместных действий, когда состояние, предшествующее выполнению действия, сохраняется.

35.1 Примитивная лексика вариации эффектов затирания полезных данных

Лексика вариации эффектов затирания полезных данных не требует примитивных соотношений.

35.2 Описываемые соотношения вариации эффектов затирания полезных данных

В данном подразделе определены следующие соотношения:
- (state_clobber ?a);
- (partial_clobber ?a);
- (unconditional_clobber ?a);
- (time_clobber ?a);
- (sometime_clobber ?a);
- (rigid_clobber ?a).

Каждое понятие определяется неформальной семантикой и аксиомой KIF.

35.3 Теории, обусловленные вариацией эффектов затирания полезных данных

Для данных теорий необходимы:
- occtree.th;
- psl_core.th.

35.4 Дефинициональные расширения, обусловленные вариацией эффектов затирания полезных данных

Для данных расширений необходимы:
- occ_precond.def;
- state_precond.def;
- precond.def.

35.5 Определения вариации эффектов затирания полезных данных

Для вариации эффектов затирания полезных данных определены нижеследующие понятия.

35.5.1 state_clobber

Некоторое действие ?a затирает данные о состоянии тогда и только тогда, когда какие-либо согласованные по состоянию события сохраняют те же эффекты.

```
(forall (?a) (iff (state_clobber ?a)
(forall (?a1 ?s1 ?s2)
        (implies (and (atomic ?a1)
                      (state_equiv ?s1 ?s2))
                 (iff (preserved_effects ?a1 ?a ?s1)
                      (preserved_effects ?a1 ?a ?s2))))))
```

35.5.2 partial_state_clobber

Некоторое действие ?a частично затирает данные о состоянии тогда и только тогда, когда существуют согласованные по состоянию события, сохраняющие те же эффекты.

```
(forall (?a) (iff (partial_clobber ?a)
(and (exists (?s1 ?a1)
        (forall (?s2)
                (implies (and (atomic ?a1)
                              (state_equiv ?s1 ?s2))
                         (iff (preserved_effects ?a1 ?a ?s1)
                              (preserved_effects ?a1 ?a ?s2)))))
     (exists (?a2 ?s3 ?s4)
         (and (atomic ?a2)
              (state_equiv ?s3 ?s4)
              (preserved_effects ?a2 ?a ?s3)
              (not (preserved_effects ?a2 ?a ?s4)))))))
```

35.5.3 unconditional_clobber

Некоторое действие ?a затирает данные безусловно тогда и только тогда, когда существуют согласованные по состоянию события, не сохраняющие те же эффекты.

```
(forall (?a) (iff (unconditional_clobber ?a)
(forall (?s1)
        (exists (?a1 ?s2)
                (and (atomic ?a1)
                     (state_equiv ?s1 ?s2)
                     (iff (preserved_effects ?a1 ?a ?s1)
                          (not (preserved_effects ?a1 ?a ?s2)))))))))
```

35.5.4 time_clobber

Некоторое действие ?a затирает данные о времени тогда и только тогда, когда какие-либо согласованные по времени начала отсчета события сохраняют те же эффекты.

```
(forall (?a) (iff (time_clobber ?a)
(forall (?a1 ?s1 ?s2)
        (implies (and (atomic ?a1)
                      (begin_equiv ?s1 ?s2))
                 (iff (preserved_effects ?a1 ?a ?s1)
                      (preserved_effects ?a1 ?a ?s2))))))
```

35.5.5 sometime_clobber

Некоторое действие ?a иногда затирает данные о времени тогда и только тогда, когда существуют согласованные по времени начала отсчета события, сохраняющие те же эффекты.

```
(forall (?a) (iff (sometime_clobber ?a)
(and (exists (?s1 ?a1)
        (forall (?s2)
                (implies (and (atomic ?a1)
                              (begin_equiv ?s1 ?s2))
                         (iff (preserved_effects ?a1 ?a ?s1)
                              (preserved_effects ?a1 ?a ?s2)))))
     (exists (?a2 ?s3 ?s4)
         (and (atomic ?a2)
              (begin_equiv ?s3 ?s4)
              (preserved_effects ?a2 ?a ?s3)
              (not (preserved_effects ?a2 ?a ?s4)))))))
```

35.5.6 rigid_clobber

Некоторое действие ?a затирает данные жестко тогда и только тогда, когда существуют согласованные по времени начала отсчета события, не сохраняющие те же эффекты.

```
 (forall (?a) (iff (rigid_clobber ?a)
(forall (?s1)
        (exists (?a1 ?s2)
                (and (atomic ?a1)
                        (begin_equiv ?s1 ?s2)
                        (iff (preserved_effects ?a1 ?a ?s1)
                                (not (preserved_effects ?a1 ?a ?s2))))))))
```

35.6 Грамматика описаний процесса для вариации эффектов затирания полезных данных

Грамматика описаний технологического процесса для классов действий, описываемых для вариации эффектов затирания полезных данных, эквивалентна грамматике 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 42 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 или 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, описание комплексного действия по изготовлению изделия GT-350, состоящего из составляющих 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.

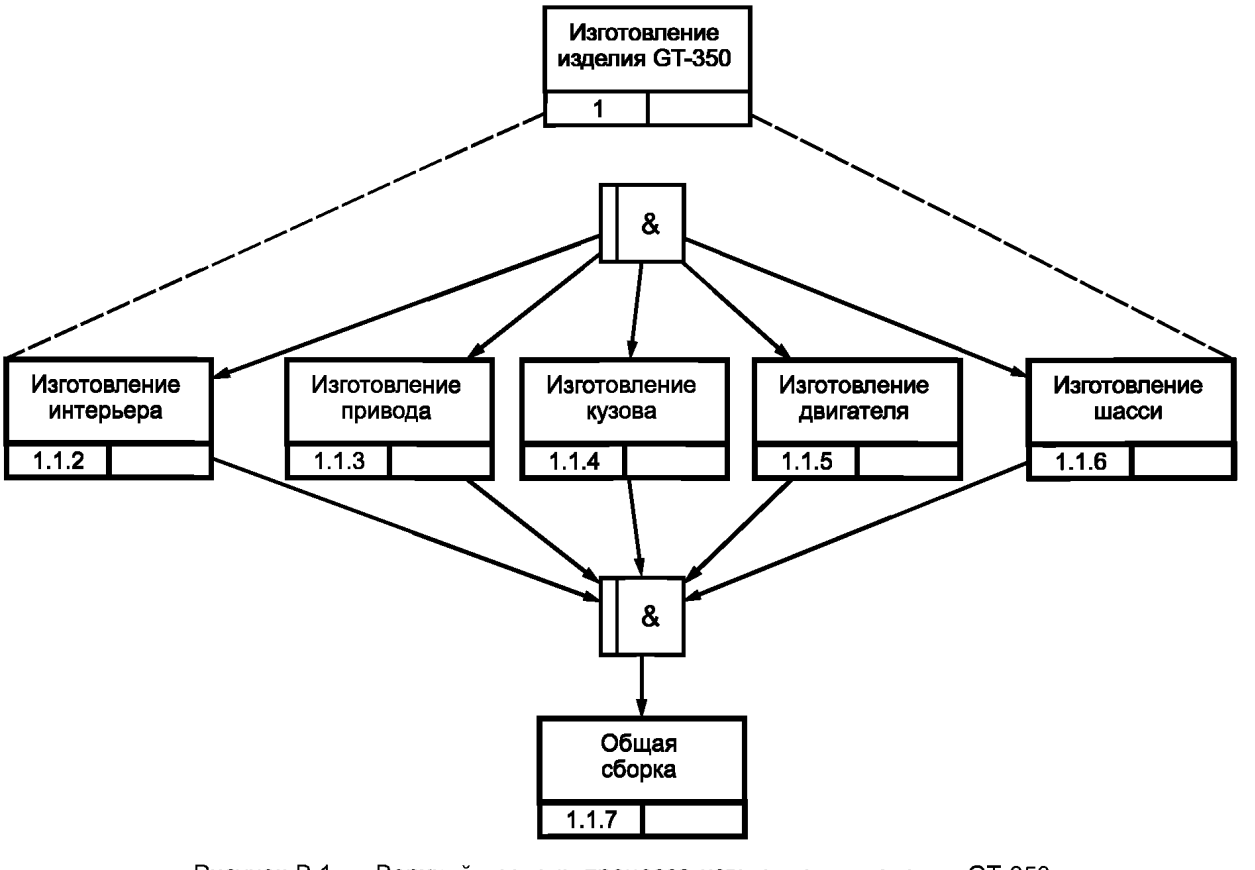

Рисунок В.1 — Верхний уровень процесса изготовления изделия GT-350

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)

(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 (?occ)
        (iff (occurrence_of ?occ make_engine)
          (= (beginof ?occ) 0700)))
(trigger make_engine)
(conditional make_engine)

(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.

Поддействие «make_engine» (изготовление двигателя) не выполняется, если двигатель уже существует. Поэтому комплексное действие «make_gt350» является условным.

Действие «make_gt350» выполняется в момент времени 0700. Поэтому данное действие является запускающим.

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.

В.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 В.2. Агрегат состоит из двигательного блока, жгутов и кабелей. Составляющие процессы детально рассмотрены в подразделах ниже. Двигатель изделия GT-350 собирается на сборочном стенде А004. Сборка одного двигателя требует 5 мин.

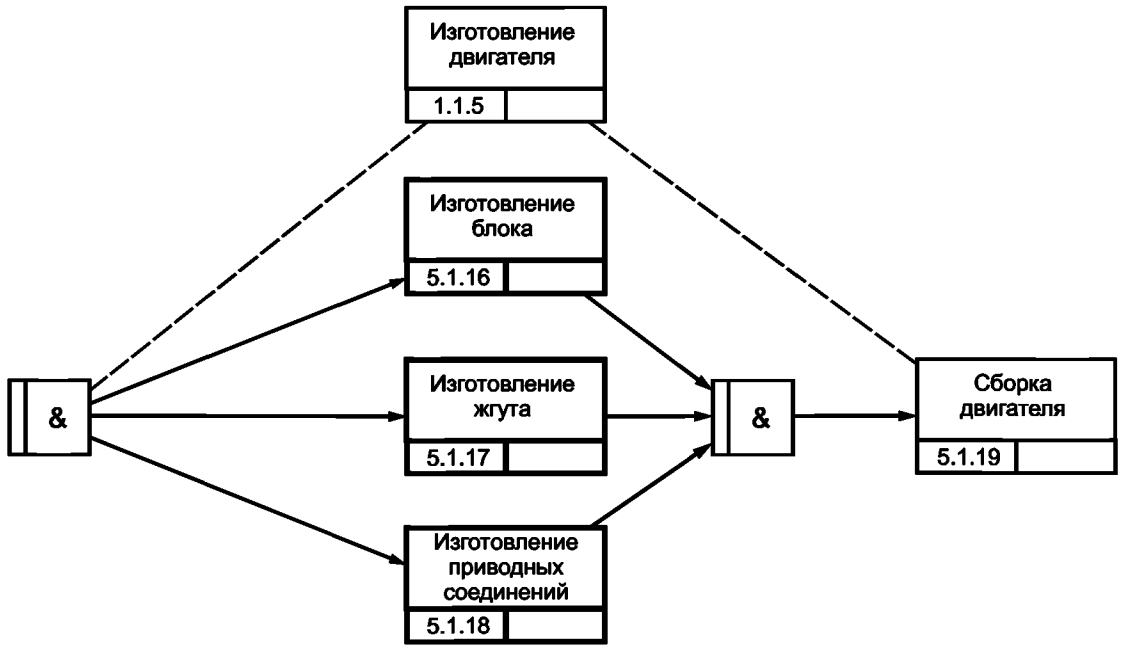

Рисунок В.2 — Процесс изготовления двигателя изделия GT-350

Ниже представлены некоторые действия и данные технологического процесса изготовления двигателя на языке PSL (внешнее ядро):

```
(subactivity make_block make_engine)
(subactivity make-harness make_engine)
(subactivity make-wires make_engine)
(subactivity assemble_engine make_engine)
(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)))))
```

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).

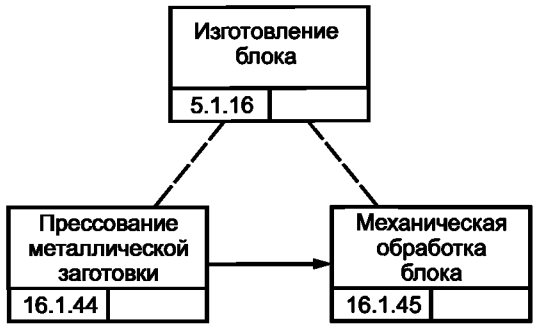

Рисунок В.3 — Процесс изготовления блока изделия GT-350

Представление некоторых действий и технологических данных на языке PSL (внешнее ядро):

```
(subactivity produce_molded_metal make_block)
(subactivity machine_block make_block)
(primitive machine_block)
(primitive produce_molded_metal)

(uniform 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)))))

(markov_precond machine_block)
(forall (?occ)
(implies (and   (occurrence_of ?occ machine_block)
                (legal ?occ))
         (prior molded ?occ)))

(markov_effects machine_block)
(implies (and   (occurrence_of ?occ machine_block)
                (legal ?occ))
         (holds finished ?occ)))

(markov_precond produce_molded_metal)
(implies (and (occurrence_of ?occ machine_block)
                (legal ?occ))
         (prior cast ?occ)))

(markov_effects produce_molded_metal)
(implies (and   (occurrence_of ?occ machine_block)
                (legal ?occ))
         (holds molded ?occ)))
```

Данное представление формализует технологический процесс, показанный на рисунке В.3.

Составляющее действие «produce_molded_metal» (прессование металлической заготовки) имеет входное условие, в соответствии с которым прессованию должно предшествовать литье. Таким образом, это есть действие с марковскими входными условиями (markov_precond).

Составляющее действие «produce_molded_metal» имеет эффект, заключающийся в том, что штамповка блока происходит после предварительного прессования металлической заготовки (допустимое событие). Это есть марковский эффект действия (markov_effect).

Составляющее действие «machine_block» (механическая обработка блока) имеет входное условие, в соответствии с которым до механической обработки блок штампуется (допустимое событие). Это есть действие с марковскими входными условиями.

Составляющее действие «machine_block» (механическая обработка блока) имеет эффект, в соответствии с которым блок зачищается после прессования металлической заготовки (допустимое событие). Это есть действие с марковским эффектом (markov_effect).

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.

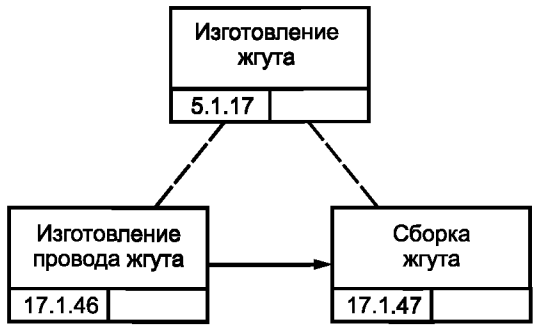

Рисунок В.4 — Процесс изготовления жгута изделия GT-350

Ниже дано представление некоторых действий и соответствующих технологических данных на языке PSL (внешнее ядро):

```
(subactivity make_harness_wire make_harness)
(subactivity assemble_harness make_harness)
(primitive assemble_harness)

(occurrence_constrained assemble_harness)
(forall (?occ)
(implies (and   (occurrence_of ?occ make_harness_wires)
                (legal ?occ))
         (legal (successor assemble_harness ?occ))))

(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, представленный на рисунке В.4.

Составляющее действие «assemble_harness» (сборка жгута) имеет входное условие, в соответствии с которым сборка следует за изготовлением проводов жгута (make_harness_wires), поэтому данное действие является событийно ограниченным (occurrence_constrained).



Рисунок В.5 — Процесс изготовления провода жгута

B.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 (см. рисунок 6).

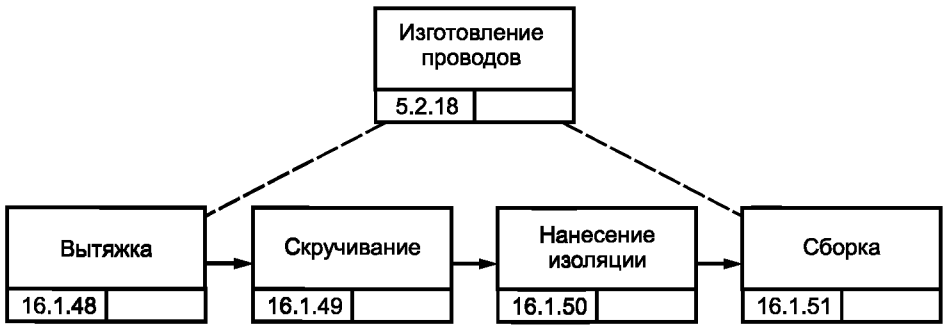

Рисунок B.6 — Процесс изготовления проводов изделия GT-350

Ниже дано представление некоторых действий и соответствующих технологических данных на языке PSL (внешнее ядро):

```
(subactivity extrude make_harness_wire)
(subactivity twist make_harness_wire)
(subactivity jacket make_harness_wire)
(primitive extrude)
(primitive twist)
(primitive jacket)
(unconstrained extrude)
(forall (?occ)
        (poss extrude ?occ))

(time_effects extrude)
(forall (?occ)
(implies (and (occurrence_of ?occ extrude)
              (= (beginof ?occ) 0700))
         (holds (temperature wire 100) ?occ)))
(unconstrained twist)
(forall (?occ)
    (poss twist ?occ))
(context_free twist)
(unconstrained jacket)
(forall (?occ)
    (poss jacket ?occ))

(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, представленный на рисунке B.5.

Составляющие действия «extrude» (вытяжка), «twist» (скручивание) и «jacket» (нанесение изоляции) всегда возможны. Следовательно, эти действия являются неограниченными.

Составляющее действие «twist» (скручивание) всегда имеет один и тот же эффект. Указанное действие является свободным в данном контексте (context_free).

Эффекты составляющего действия «extrude» (вытяжка) зависят от времени его выполнения (в момент времени 0700 провод имеет температуру 100 °C). Таким образом, это действие с эффектами, зависящими от времени (time_effect).

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		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. Дефинициональное расширение: расширение ресурсов

УДК 65.011:56.681.3 ОКС 25.040.40

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 *Т. Ф. Кузнецовой*

Сдано в набор 28.01.2014. Подписано в печать 01.04.2014. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 8,37. Уч.-изд. л. 6,70. Тираж 69 экз. Зак. 185

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

gost.e-ecolog.ru