**Теория атрибутных систем**

# Помеченные системы переходов

Помеченная система переходов – это кортеж (Sα, Lα, Tα), где

* Sα – непустое множество состояний;
* Lα – непустое множество меток;
* Tα ⊆ Sα × Lα × Sα – отношение переходов;
* Множества Sα и Lα не пересекаются.

# Системы с выходом/входом/выходом

Системы с выходом/входом/выходом (далее системы типа oio) – это вид помеченных систем переходов, в которых в состояниях хранятся выходы системы (информация, порождаемая переходом), а метки моделируют входы в систему (информацию, инициирующую переход).

Система типа oio – это кортеж (Sβ, Iα, Oα, Oβ, Tα), где

* Sβ – множество собственных состояний;
* Iα – непустое множество входов;
* Oα – непустое множество выходов;
* Oβ – множество константных выходов;
* Oα ∩ Oβ = ∅;
* (Sβ × Oα × Oβ, Iα, Tα) – помеченная система переходов;
* если ((sβ, oα, oβ), iα, (sβ′, oα′, oβ′)) ∈ Tα, то oβ = oβ′.

Так как постоянный выход не меняется, далее будем опускать oβ в переходах. Например, писать ((sβ, oα), iα, (sβ′, oα′)) ∈ Tα вместо ((sβ, oα, oβ), iα, (sβ′, oα′, oβ′)) ∈ Tα.

# Системы типа oio с ветвлением выходов

Системы типа oio с ветвлением выходов (далее системы типа oio+ob) – это вид систем типа oio, в которых при переходе порождается конечное (возможно пустое) мультимножество выходов (ветвей).

Пусть Oγ – множество всех конечных мультимножеств из элементов множества Oα. Пусть oγ ∪ oγ′ обозначает объединение мультимножеств oγ и oγ′. Пусть |oγ| обозначает количество элементов в мультимножестве oγ.

Система типа oio+ob – это кортеж (Sβ, Iα, Oα, Oβ, Bβ), где

* (Sβ, Iα, Oγ, Oβ, Tα) – система типа oio;
* Oα – непустое множество выходов;
* Bβ ⊆ (Sβ × Oα × Oβ) × Iα × (Sβ × Oγ × Oβ) – отношение ветвления;
* Если ((sβ, oγ1), iα, (sβ′, oγ2)) ∈ Tα, и ((sβ, oγ3), iα, (sβ′, oγ4)) ∈ Tα, то ((sβ, oγ1 ∪ oγ3), iα, (sβ′, oγ2 ∪ oγ4)) ∈ Tα;()
* Если ((sβ, oα), iα, (sβ′, oγ)) ∈ Bβ, то ((sβ, {oα}), iα, (sβ′, oγ)) ∈ Tα.

# Системы типа oio с упорядоченным ветвлением выходов

Системы типа oio с упорядоченным ветвлением выходов (далее системы типа oio+oob) – это вид систем типа oio, в которых при переходе порождается конечная (возможно пустая) последовательность выходов (ветвей).

Пусть Oδ – множество всех конечных последовательностей из элементов множества Oα. Пусть oδ ∪ oδ′ обозначает конкатенацию последовательностей oδ и oδ′. Пусть |oδ| обозначает количество элементов в последовательности oδ.

Система типа oio+oob – это кортеж (Sβ, Iα, Oα, Oβ, Bγ), где

* (Sβ, Iα, Oδ, Oβ, Tα) – система типа oio;
* Oα – непустое множество выходов;
* Bγ ⊆ (Sβ × Oα) × Iα × (Sβ × Oδ) – отношение упорядоченного ветвления;
* если ((sβ, oγ1), iα, (sβ′, oγ2)) ∈ Tα, и ((sβ, oγ3), iα, (sβ′, oγ4)) ∈ Tα, то ((sβ, oγ1 ∪ oγ3), iα, (sβ′, oγ2 ∪ oγ4)) ∈ Tα;
* если ((sβ, oα), iα, (sβ′, oδ)) ∈ Bγ, то ((sβ, {oα}), iα, (sβ′, oδ)) ∈ Tα.

# Системы с компонентным выходом/входом/выходом

Системы с компонентным выходом/входом/выходом (далее системы типа coio) – это вид систем типа oio, в которых выходы имеют составные части, называемые компонентами.

Система типа coio – это кортеж (Sβ, Iα, Cα, Cβ, Vα, Tα), где

* Cα – множество компонентов выхода;
* Cβ – множество компонентов константного выхода;
* Vα – множество значений;
* vβ ∉ Vα – неопределенное значение;
* Vγ = Vα ∪ {vβ} – расширенное множество значений;
* (Sβ, Iα, Oα, Oβ, Tα) – система типа oio;
* Oα – множество функций типа Cα → Vγ;
* Oβ – множество функций типа Cβ → Vγ.

Компоненты выхода определяют различные характеристики выхода.

Значение компонентов cα и cβ выходов oα и oβ определяются как oα(cα) и oβ(cβ), соответственно.

Пусть oα.cα и oβ.cβ обозначает oα(cα) и oβ(cβ), соответственно.

Пусть Sγ – множество всех систем типа coio.

# Системы типа coio с ветвлением выходов

Системы типа coio с ветвлением выходов (далее системы типа coio+ob) – это вид систем типа coio, в которых при переходе порождается конечное (возможно пустое) мультимножество выходов (ветвей).

Система типа coio+ob – это кортеж (Sβ, Iα, Cα, Cβ, Vα, Bβ), где

* Cα – множество компонентов выхода;
* Cβ – множество компонентов константного выхода;
* Vα – множество значений;
* vβ ∉ Vα – неопределенное значение;
* Vγ = Vα ∪ {vβ} – расширенное множество значений;
* (Sβ, Iα, Oα, Oβ, Bβ) – система типа oio+ob;
* Oα – множество функций типа Cα → Vγ;
* Oβ – множество функций типа Cβ → Vγ.

# Системы типа coio с упорядоченным ветвлением выходов

Системы типа coio с упорядоченным ветвлением выходов (далее системы типа coio+oob) – это вид систем типа coio, в которых при переходе порождается конечная (возможно пустая) последовательность выходов (ветвей).

Пусть Oγ – множество всех конечных последовательностей из элементов множества Oα. Пусть oγ ∪ oγ′ обозначает конкатенацию последовательностей oγ и oγ′. Пусть |oγ| обозначает количество элементов в последовательности oγ.

Система типа coio+oob – это кортеж (Sβ, Iα, Cα, Cβ, Vα, Bγ), где

* Cα – множество компонентов выхода;
* Cβ – множество компонентов константного выхода;
* Vα – множество значений;
* vβ ∉ Vα – неопределенное значение;
* Vγ = Vα ∪ {vβ} – расширенное множество значений;
* (Sβ, Iα, Oα, Oβ, Bγ) – система типа oio+oob;
* Oα – множество функций типа Cα → Vγ;
* Oβ – множество функций типа Cβ → Vγ.

# Системы типа coio с выходными значениями

Системы типа coio с выходными значениями (далее системы типа coio+ov) – это вид систем типа coio, переходы которых могут возвращать (порождать) значения.

Система типа coio+ov – это система типа coio, где

* value ∈ Cα – компонент выходного значения;
* oα.value – выходное значение.

Пусть sα имеет вид (sβ, oα).

Система sγ возвращает (порождает) значение в состоянии sα, если oα.value ≠ vβ.

Система sγ не возвращает (порождает) значения в состоянии sα, если oα.value = vβ.

Система sγ возвращает (порождает) значение vα в состоянии sα, если oα.value = vα.

# Системы типа coio с типизированными выходными значениями

Системы типа coio с типизированными выходными значениями (далее системы типа coio+tov) – это вид систем типа coio+ov, в которых выходные значения могут иметь тип.

Система типа coio+tov – это система типа coio+ov, где

* type ∈ Cα – компонент типа выходного значения;
* oα.type – тип выходного значения.

Пусть sα имеет вид (sβ, oα).

Система sγ возвращает (порождает) значение в состоянии sα, если oα.value ≠ vβ, или oα.type ≠ vβ.

Система sγ не возвращает (порождает) значения в состоянии sα, если oα.value = vβ, и oα.type = vβ.

Система sγ возвращает (порождает) безтиповое значение в состоянии sα, если oα.value ≠ vβ, и oα.type = vβ.

Система sγ возвращает (порождает) типизированное значение в состоянии sα, если oα.type ≠ vβ.

Система sγ возвращает (порождает) типизированное неопределенное значение в состоянии sα, если oα.value = vβ, и oα.type ≠ vβ.

Пусть tα ∈ Vα.

Система sγ возвращает (порождает) неопределенное значение типа tα в состоянии sα, если oα.value = vβ, и oα.type = tα.

Система sγ возвращает (порождает) значение vα типа tα в состоянии sα, если oα.value = vα, и oα.type = tα.

Система sγ возвращает (порождает) безтиповое значение vα в состоянии sα, если oα.value = vα, и oα.type = vβ.

# Системы типа coio с джампами

Системы типа coio с джампами (далее системы типа coio+j) – это вид систем типа coio, переходы которых могут инициироваться джампами и порождать джампы. Джампы – это исключительные ситуации, которые порождаются переходами, могут просачиваться через переходы и ловится переходами. Переходы с джампами – это абстракция механизма обработки исключений и механизма передачи управления (break, continue, goto, return и т. п.) в языках программирования.

Система типа coio+j – это система типа coio, где

* jvalue ∈ Cα – компонент джампа;
* oα.jvalue –джамп.

Пусть sα имеет вид (sβ, oα).

Система sγ возвращает (порождает) джамп в состоянии sα, если oα.jvalue ≠ vβ.

Система sγ не возвращает (порождает) джампа в состоянии sα, если oα.jvalue = vβ.

Система sγ возвращает (порождает) джамп vα в состоянии sα, если oα.jvalue = vα.

Пусть переход tα имеет вид ((sβ, oα), iα, (sβ′, oα′)).

Переход tα инициируется джампом, если oα.jvalue ≠ vβ.

Переход tα не инициируется джампом, если oα.jvalue = vβ.

Переход tα инициируется джампом vα, если oα.jvalue = vα.

Переход tα просачивает джамп, если oα.jvalue = oα′.jvalue ≠ vβ, и sβ = sβ′.

Переход tα просачивает джамп vα, если oα.jvalue = oα′.jvalue = vα, и sβ = sβ′.

Переход tα ловит джамп, если oα.jvalue ≠ vβ, и (oα.jvalue ≠ oα′.jvalue, или sβ ≠ sβ′).

Переход tα ловит джамп vα, если oα.jvalue = vα, и (oα.jvalue ≠ oα′.jvalue, или sβ ≠ sβ′).

# Системы типа coio с типизированными джампами

Системы типа coio с типизированными джампами (далее системы типа coio+tj) – это вид систем типа coio+j, в которых джампы могут иметь тип.

Система типа coio+tj – это система типа coio+j, где

* jtype ∈ Cα – компонент типа джампа;
* oα.jtype – тип джампа.

Пусть состояние sα имеет вид (sβ, oα).

Система sγ возвращает (порождает) джамп в состоянии sα, если oα.jvalue ≠ vβ, или oα.jtype ≠ vβ.

Система sγ не возвращает (порождает) джампа в состоянии sα, если oα.jvalue = vβ, или oα.jtype = vβ.

Система sγ возвращает (порождает) безтиповый джамп в состоянии sα, если oα.jvalue ≠ vβ, и oα.jtype = vβ.

Система sγ возвращает (порождает) типизированный джамп в состоянии sα, если oα.jtype ≠ vβ.

Система sγ возвращает (порождает) типизированный неопределенный джамп в состоянии sα, если oα.jvalue = vβ, и oα.jtype ≠ vβ.

Пусть tα ∈ Vα.

Система sγ возвращает (порождает) неопределенный джамп типа tα в состоянии sα, если oα.jvalue = vβ, и oα.jtype = tα.

Система sγ возвращает (порождает) джамп vα типа tα в состоянии sα, если oα.jvalue = vα ≠ vβ, и oα.jtype = tα.

Система sγ возвращает (порождает) безтиповый джамп vα в состоянии sα, если oα.jvalue = vα, и oα.jtype = vβ.

Пусть переход tα имеет вид ((sβ, oα), iα, (sβ′, oα′)).

Переход tα инициируется джампом, если oα.jvalue ≠ vβ, или oα.jtype ≠ vβ.

Переход tα не инициируется джампом, если oα.jvalue = vβ, и oα.jtype = vβ.

Переход tα инициируется безтиповым джампом, если oα.jvalue ≠ vβ, и oα.jtype = vβ.

Переход tα инициируется типизированным джампом, если oα.jvalue ≠ vβ, и oα.jtype ≠ vβ.

Переход tα инициируется типизированным неопределенным джампом, если oα.jvalue = vβ, и oα.jtype ≠ vβ.

Переход tα инициируется неопределенным джампом типа tα, если oα.jvalue = vβ, и oα.jtype = tα.

Переход tα инициируется джампом vα типа tα, если oα.jvalue = vα, и oα.jtype = tα.

Переход tα инициируется безтиповым джампом vα, если oα.jvalue = vα, и oα.jtype = vβ.

Переход tα просачивает джамп, если oα.jvalue = oα′.jvalue ≠ vβ, oα.jtype = oα′.jtype ≠ vβ, и sβ = sβ′.

Переход tα просачивает джамп vα, если oα.jvalue = oα′.jvalue = vα ≠ vβ, oα.jtype = oα′.jtype ≠ vβ, и sβ = sβ′.

Переход tα ловит джамп, если (oα.jvalue ≠ vβ, или oα.jtype ≠ vβ) и (oα.jvalue ≠ oα′.jvalue, oα.jtype ≠ oα′.jtype, или sβ ≠ sβ′).

Переход tα ловит джамп vα, если oα.jvalue = vα ≠ vβ, и (oα.jvalue ≠ oα′.jvalue, oα.jtype ≠ oα′.jtype, или sβ ≠ sβ′).

# Системы типа coio с контекстами

Системы типа coio с контекстами (далее системы типа coio+c) – это вид систем типа coio, выполнение переходов которых может зависеть от контекста и переходы которых могут возвращать (порождать) контексты для следующих переходов. Контексты делятся на глобальные (меняются редко), локальные (меняются часто) и промежуточные.

Система типа coio+c – это система типа coio, где

gcontext ∈ Cβ – компонент глобального контекста;

lcontext ∈ Cα – компонент локального контекста;

icontext ∈ Cα – компонент промежуточного контекста;

oβ.gcontext –глобальный контекст;

oα.lcontext –локальный контекст;

oα.icontext –промежуточный контекст.

Пусть переход tα имеет вид ((sβ, oα, oβ), iα, (sβ′, oα′, oβ′)).

Система sγ выполняет переход tα вне глобального контекста, если oβ.gcontext = vβ.

Система sγ выполняет переход tα в глобальном контексте, если oβ.gcontext ≠ vβ.

Система sγ выполняет переход tα в глобальном контексте vα, если oβ.gcontext = vα.

Система sγ выполняет переход tα вне локального контекста, если oα.lcontext = vβ.

Система sγ выполняет переход tα в локальном контексте, если oα.lcontext ≠ vβ.

Система sγ выполняет переход tα в локальном контексте vα, если oα.lcontext = vα.

Система sγ выполняет переход tα вне промежуточного контекста, если oα.icontext = vβ.

Система sγ выполняет переход tα в промежуточном контексте, если oα.icontext ≠ vβ.

Система sγ выполняет переход tα в промежуточном контексте vα, если oα.icontext = vα.

Пусть состояние sα имеет вид (sβ, oα, oβ).

Система sγ не возвращает (порождает) глобальный контекст в состоянии sα, если oβ.gcontext = vβ.

Система sγ возвращает (порождает) глобальный контекст в состоянии sα, если oβ.gcontext ≠ vβ.

Система sγ возвращает (порождает) глобальный контекст vα в состоянии sα, если oβ.gcontext = vα.

Система sγ не возвращает (порождает) локальный контекст в состоянии sα, если oα.lcontext = vβ.

Система sγ возвращает (порождает) локальный контекст в состоянии sα, если oα.lcontext ≠ vβ.

Система sγ возвращает (порождает) локальный контекст vα в состоянии sα, если oα.lcontext = vα.

Система sγ не возвращает (порождает) промежуточный контекст в состоянии sα, если oα.icontext = vβ.

Система sγ возвращает (порождает) промежуточный контекст в состоянии sα, если oα.icontext ≠ vβ.

Система sγ возвращает (порождает) промежуточный контекст vα в состоянии sα, если oα.icontext = vα.

# Системы типа coio+ov+j

Системы типа coio+ov+j являются комбинацией систем типов coio+ov и coio+ j.

Система типа coio+ov+j – это система типа coio, где

* value ∈ Cα – компонент выходного значения;
* jvalue ∈ Cα – компонент джампа;
* oα.value – выходное значение;
* oα.jvalue – джамп.

Определения, связанные с выходными значениями, модифицируются для таких систем.

Система sγ возвращает (порождает) значение в состоянии sα, если sγ как система типа coio+ov возвращает (порождает) значение в состоянии sα, и sγ как система типа coio+j не возвращает (порождает) джампа в состоянии sα.

Система sγ не возвращает (порождает) значения в состоянии sα, если sγ как система типа coio+ov не возвращает (порождает) значения в состоянии sα, или sγ как система типа coio+j возвращает (порождает) джамп в состоянии sα.

Система sγ возвращает (порождает) значение vα в состоянии sα, если sγ как система типа coio+ov возвращает (порождает) значение vα в состоянии sα, и sγ как система типа coio+j не возвращает (порождает) джампа в состоянии sα.

Системы типов coio+tov+j, coio+ov+tj и coio+tov+tj определяются аналогичным образом.

# Атрибутные системы

Атрибутные системы – это вид систем типа coio, в которых значения и компоненты определяются через такие онтологические сущности, как атом, атрибут и атрибутон.

Атрибутоны – это сущности, которые могут иметь атрибуты. Атрибуты – это сущности, которые могут иметь значения относительно атрибутонов. Атомы – это сущности, которые не имеют атрибутов.

Атрибутная система – это кортеж (Aα, Aβ, Cα, Cβ, Tα), где

* Aα – множество атомов;
* Aβ – множество атрибутонов;
* Aα ∩ Aβ = ∅;
* (Sβ, Iα, Cα, Cβ, Vα, Tα) – система типа coio;
* Sβ = Aβ × Vα → Vγ;
* Vα = Aα ∪ Aβ;
* Cα ⊆ Vα;
* Cβ ⊆ Vα;
* Iα = Vγ;
* concept ∈ Aα – спецификатор понятий.

Собственные состояния в атрибутных системах определяют содержимое атрибутонов (какие атрибуты имеют атрибутоны и значения этих атрибутов).

В качестве атрибута может выступать любое значение из Vα.

Пусть Aγ = Vα – множество атрибутов.

Атрибутон aβ имеет атрибут aγ в состоянии sβ, если sβ(aβ, aγ) ≠ vβ. Атрибутон aβ не имеет атрибута aγ в состоянии sβ, если sβ(aβ, aγ) = vβ. Атрибут aγ атрибутона aβ имеет значение vα в состоянии sβ, если sβ(aβ, aγ) = vα.

Такая онтологическая сущность как понятие определяется в атрибутных системах спецификатором понятий concept. В качестве понятия может выступать любое значение из Vα.

Пусть Cγ = Vα – множество понятий.

Содержимое понятия cγ определяется как множество атрибутонов и зависит от состояния: aтрибутон aβ принадлежит понятию cγ в состоянии sβ, если sβ(aβ, cγ) = concept.

Пусть sα(aβ, aγ1, aγ2, …, aγn) – сокращение для sα(…sα(sα(sα(aβ, aγ1), aγ2), …, aγn).

# Атрибутные системы с последовательностями

Атрибутные системы с последовательностями (далее атрибутные системы типа s) – это вид атрибутных систем, в которых определены атрибутоны специального вида, задающие последовательности значений.

Атрибутная система типа s – это атрибутная система, где

* Integer ⊂ Aα – множество целых чисел;
* seqLen, start ∈ Aα – спецификаторы длины и начального индекса последовательностей.

Атрибутон aβ является последовательностью в состоянии sβ, если sβ(aβ, seqLen) ∈ Integer, sβ(aβ, seqLen) ≥ 0, и sβ(aβ, start) ∈ Integer.

Значения sβ(aβ, seqLen) и sβ(aβ, start) атрибутов seqLen и start атрибутона aβ называются длиной и начальным индексом последовательности aβ в состоянии sβ.

Значение sβ(aβ, j), где sβ(aβ, start) ≤ j ≤ sβ(aβ, seqLen)+ sβ(aβ, start)-1, называется элементом последовательности aβ с индексом i в состоянии sβ.

Пусть Aδ(sβ) – множество атрибутонов, являющихся последовательностями в состоянии sβ.

Пусть Aδ(sβ, n) – множество атрибутонов, являющихся последовательностями длины n в состоянии sβ.

# Атрибутные системы с актуальными атрибутонами

Атрибутные системы с актуальными атрибутонами (далее атрибутные системы типа aa) – это вид атрибутных систем, в которых в каждом состоянии множество атрибутонов разделено на два непересекающихся подмножества – множество актуальных атрибутонов (атрибутонов, которые доступны в этом состоянии) и множество потенциальных атрибутонов (атрибутонов, которые не доступны в этом состоянии, но возможно будут доступны в следующих состояниях). Такие системы обеспечивают механизмы добавления новых атрибутонов (перевода их из множества потенциальных атрибутонов в множество актуальных атрибутонов) и удаления старых атрибутонов (перевода их из множества актуальных атрибутонов в множество потенциальных атрибутонов).

Атрибутная система типа aa – это атрибутная система, где

* actual ∈ Aα;
* actual ∈ Cβ – компонент спецификатора актуального атрибутона;
* oβ.actual ∈ Aβ – спецификатор актуального атрибутона;
* sβ(oβ.actual, oβ.actual) = actual для любого состояния (sβ, oα, oβ);
* если sβ(oβ.actual, aβ) ≠ actual, то sβ(aβ, vα) = vβ для любого vα ∈ Vα, т. е. aβ не имеет атрибутов в состоянии sβ.

Пусть состояние sγ имеет вид (sβ, oα, oβ).

Атрибутон aβ является актуальным в состоянии sγ, если sβ(oβ.actual, aβ) = actual.

Атрибутон aβ является потенциальным в состоянии sγ, если aβ не является актуальным в sγ. Потенциальные атрибутоны не имеют атрибутов.

# Атрибутные системы с ветвлением выходов

Атрибутные системы с ветвлением выходов (далее атрибутные системы типа ob) – это вид атрибутных систем, в которых при переходе порождается конечное (возможно пустое) мультимножество выходов (ветвей).

Атрибутная система типа ob – это кортеж (Aα, Aβ, Cα, Cβ, Oδ, Bβ), где

* Aα – множество атомов;
* Aβ – множество атрибутонов;
* Aα ∩ Aβ = ∅;
* (Sβ, Iα, Cα, Vα, Oδ, Bβ) – система типа coio+ob;
* Sβ = Aβ × Vα → Vγ;
* Vα = Aα ∪ Aβ;
* Cα ⊆ Vα;
* Cβ ⊆ Vα;
* Iα = Vγ;
* concept ∈ Aα – спецификатор понятий.

# Атрибутные системы с упорядоченным ветвлением выходов

Атрибутные системы с упорядоченным ветвлением выходов (далее атрибутные системы типа oob) – это вид атрибутных систем, в которых при переходе порождается конечная (возможно пустая) последовательность выходов (ветвей).

Атрибутная система типа oob – это кортеж (Aα, Aβ, Cα, Cβ, Oδ, Bγ), где

* Aα – множество атомов;
* Aβ – множество атрибутонов;
* Aα ∩ Aβ = ∅;
* (Sβ, Iα, Cα, Vα, Oδ, Bγ) – система типа coio+oob;
* Sβ = Aβ × Vα → Vγ;
* Vα = Aα ∪ Aβ;
* Cα ⊆ Vα;
* Cβ ⊆ Vα;
* Iα = Vγ;
* concept ∈ Aα – спецификатор понятий.

# Другие виды атрибутных систем

Аналогично системам типа coio можно определить атрибутные системы с выходными значениями (атрибутные системы типа ov), атрибутные системы с типизированными выходными значениями (атрибутные системы типа tov), атрибутные системы с джампами (атрибутные системы типа j), атрибутные системы с типизированными джампами (атрибутные системы типа tj) и атрибутные системы с контекстами (атрибутные системы типа c).