

13 Мая 2013

Инсерционное моделирование 2

(система инсерционного моделирования и когнитивные архитектуры)

Материалы к зачету

Инсерционное моделирование 1

основные понятия

Размеченные транзиционные системы

$$\langle S, A, T \rangle, T \subseteq S \times A \times S \quad s \xrightarrow{a} s'$$

Состояния
Действия
Переходы

Настроенные системы $S_0, S_\Delta, S_\perp \subseteq S$

начальные состояния

заключительные состояния

неопределенные состояния (можно доопределять)

Недетерминизм

Трассовая эквивалентность

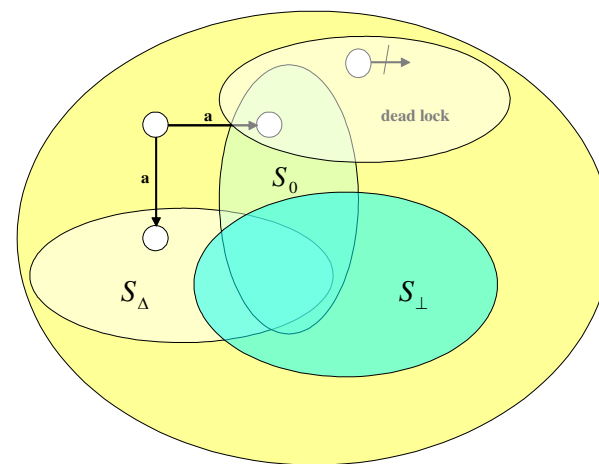
История: $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \dots \xrightarrow{a_n} \dots$

Трасса: $a_1 a_2 \dots a_n \dots$

Трасса для атрибутивной системы:

$$\varphi(s_1) \xrightarrow{a_1} \varphi(s_2) \xrightarrow{a_2} \dots \xrightarrow{a_n} \dots$$

$L_\Delta^0(S)$ трассы из S_0 в S_Δ



$$S \sim_T S' \Leftrightarrow L(S_\Delta^0) = L((S')_\Delta^0)$$

Лекция 2 Сети Петри

$$N = \langle S, T, W \rangle$$

S – места

T – переходы

$$W : (S \times T) \cup (T \times S) \rightarrow \text{Nat}$$

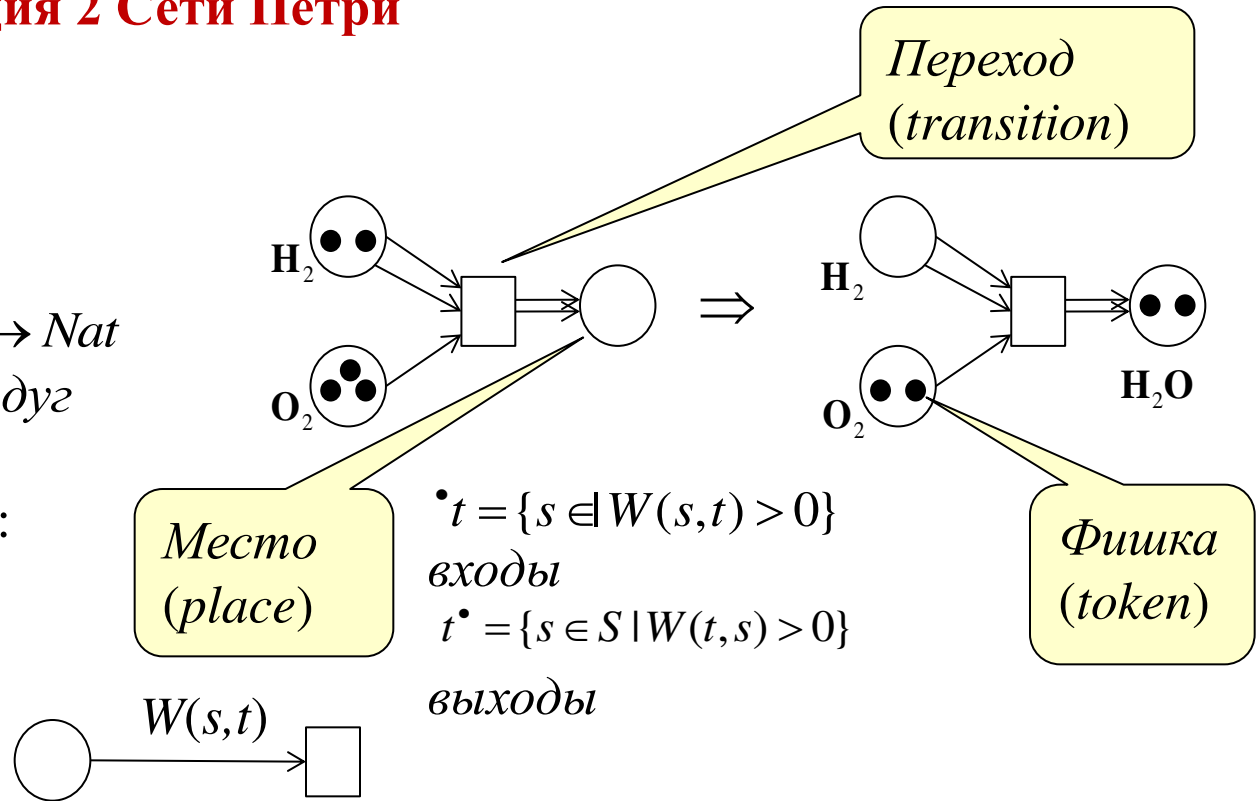
мультимножество дуг

разметка (marking):

$$M : S \rightarrow \text{Nat}$$

Размеченная сеть:

$$\langle S, T, W, M_0 \rangle$$



Транзиционная система ассоциированная с сетью Петри
(граф достижимости)

$$M \xrightarrow{t} M' \Leftrightarrow M'(s) = M(s) - W(s,t) + W(t,s)$$

$$\forall (s \in \bullet t) (M(s) \geq W(s,t))$$

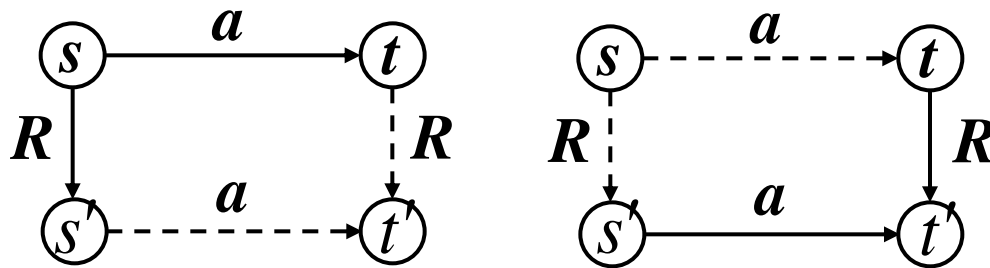
Лекция 2 Бисимуляционная эквивалентность

s и s' бисимуляционно эквивалентны
 $s \sim_B s' \Leftrightarrow \exists(\text{бисимуляция } R)((s, s') \in R)$

Отношение бисимуляции (bisimulation): $R \subseteq S^2$

1. $(s, s') \in R \Rightarrow (s \in S_{\Delta} \Leftrightarrow s' \in S_{\Delta}, s \in S_{\perp} \Leftrightarrow s' \in S_{\perp})$
2. $(s, s') \in R \wedge s \xrightarrow{a} t \Rightarrow \exists t'((t, t') \in R \wedge s' \xrightarrow{a} t')$
3. $(s, s') \in R \wedge s' \xrightarrow{a} t' \Rightarrow \exists t((t, t') \in R \wedge s \xrightarrow{a} t)$

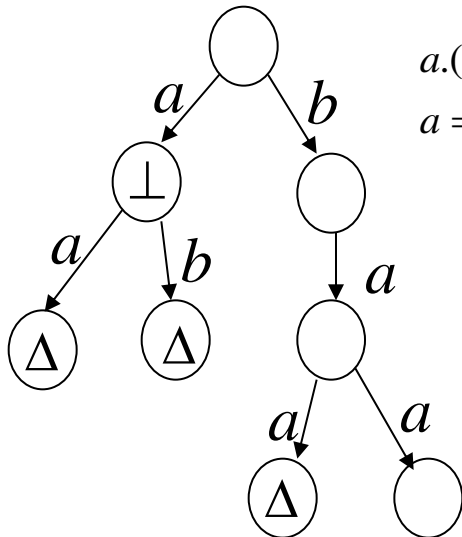
Для детерминированных систем бисимуляционная эквивалентность состояний совпадает с трассовой.



bisimulation => bisimilarity

Лекция 3: Алгебра поведений (процессов)

- Два сорта: $\langle U, A \rangle$
 - U – поведения
 - A – действия
- Сигнатура:
 - префиксинг $a.u$, $a \in A, u \in U$
 - недетерминированный выбор $u + v$, $u, v \in U$
 - константы $\Delta, 0, \perp$
 - отношение аппроксимации \sqsubseteq
- Аксиомы:
 - аксиомы для недетерминированного выбора
 - 0 есть нейтральный элемент недетерминированного выбора
 - \sqsubseteq есть отношение частичного порядка с наименьшим элементом \perp
 - Обе операции монотонны и непрерывны (сохраняют наименьшие верхние грани)

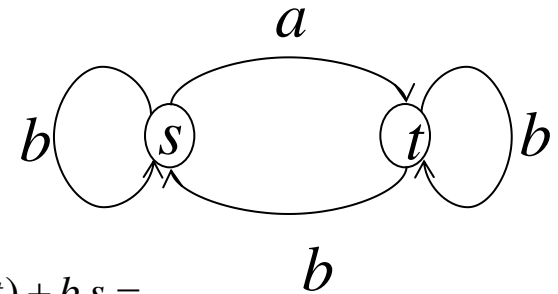


$$a.(a + b + \perp) + b.a.(a + a.0),$$

$$a = a.\Delta$$

$$s = a.t + b.s$$

$$t = b.s + b.t$$



$$s = a.t + b.s = a.(b.s + b.t) + b.s =$$

$$a.(b.s + b.t) + b.(a.t + b.s) =$$

$$a.(b.(a.t + b.s) + b.t) + b.(a.t + b.s) = \dots$$

Лекция 4: Поведение транзитивных систем

Поведение транзитивной системы

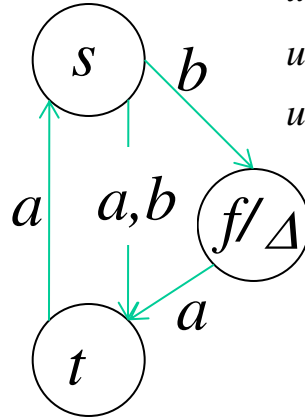
$$u_s = \sum_{s \xrightarrow{a} t} a.u_t + \varepsilon_s$$

$$s \notin S_\Delta \cup S_\perp \Rightarrow \varepsilon_s = 0$$

$$s \in S_\Delta \setminus S_\perp \Rightarrow \varepsilon_s = \Delta$$

$$s \in S_\perp \setminus S_\Delta \Rightarrow \varepsilon_s = \perp$$

$$s \in S_\Delta \cap S_\perp \Rightarrow \varepsilon_s = \Delta + \perp$$



$$u_s = a.u_t + b.u_f$$

$$u_t = a_s$$

$$u_f = a.u_t + \Delta$$

$$u \sim_B v \Leftrightarrow u = v$$

$$s \sim_B s' \Leftrightarrow \text{beh}(s) = \text{beh}(s')$$

$$x_i = f_i(x_1, x_2, \dots, x_m), i = 1, 2, \dots, m$$

$$x_i = \bigsqcup_{n=0}^{\infty} x_i^{(n)}, \quad (2)$$

$$x_i^{(0)} = \perp,$$

$$x_i^{(n+1)} = f_i(x_1^{(n)}, x_2^{(n)}, \dots, x_m^{(n)})$$

Транзитивная система, определяемая поведением

Множество $U \subset F(A)$ называется транзитивно замкнутым, если

$$a.u + v \in U \Rightarrow u \in U$$

Множество U есть транзитивная система:

$$a.u + v \xrightarrow{a} u$$

$$U_\Delta = \{u \mid u = u + \Delta\}$$

$$U_\perp = \{u \mid u = u + \perp\}$$

Последовательная композиция

$$uv = \sum_{u \xrightarrow{a} u'} a.(u'v) + \sum_{u=u+\varepsilon} \varepsilon v$$

$$0v = 0, \Delta v = v, \perp v = \perp$$

$$a \mapsto a.\Delta = (a; \Delta) = a\Delta$$

$$\frac{u \xrightarrow{a} u'}{u \parallel v \xrightarrow{a} u' \parallel v} \quad \frac{v \xrightarrow{b} v'}{u \parallel v \xrightarrow{b} u \parallel v'}$$

$$\frac{u \xrightarrow{a} u'}{u \parallel (v + \Delta) \xrightarrow{a} u' \parallel (v + \Delta)} \quad \frac{v \xrightarrow{b} v'}{(u + \Delta) \parallel v \xrightarrow{b} (u + \Delta) \parallel v'}$$

$$(u + \Delta) \parallel (v + \Delta) = (u + \Delta) \parallel (v + \Delta) + \Delta$$

$$(u + \perp) \parallel v = (u + \perp) \parallel v + \perp$$

$$u \parallel (v + \perp) = u \parallel (v + \perp) + \perp$$

Общая параллельная композиция

$$u \parallel v = \sum_{\substack{u \xrightarrow{a} u' \\ v \xrightarrow{b} v'}} (a \times b).(u' \parallel v') + \sum_{u \xrightarrow{a} u'} a.(u' \parallel v) + \sum_{v \xrightarrow{b} v'} b.(u \parallel v') + (\varepsilon_u \parallel \varepsilon_v)$$

$$\varepsilon \parallel \varepsilon' = \varepsilon' \parallel \varepsilon$$

$$\varepsilon \parallel \Delta = \varepsilon$$

$$\varepsilon \parallel \perp = \perp$$

$$0 \parallel \Delta = 0 \parallel 0 = 0$$

Среда

Агент E вместе с функцией погружения:

$$\langle E, C, A, Ins \rangle$$

E – транзитивно замкнутое
множество состояний среды

C – действия среды

A – действия агентов погружаемых в среду

Ins – непрерывная функция погружения

$$Ins : E \times F(A) \rightarrow E \quad Ins(e, u) = e[u] = e[u]_E$$

$$e[u, v] = (e[u])[v]$$

$$e[u_1, u_2, \dots],$$

$$e'[e[u_1, u_2, \dots]_E, \dots]_{E'}$$

Все с точностью до
бисимуляционной эквивалентности

Модель императивной программы

Императивная среда

$$s \xrightarrow{\alpha} s, \alpha(s) = 1$$

$$s \xrightarrow{y} y(s)$$

Отношение переходов

$$\frac{E \xrightarrow{a} E', u \xrightarrow{a} u'}{E[u] \xrightarrow{a} E'[u']}$$

$$E[u + v] = E[u] + E[v]$$

Недетерминированные программы

$$E[u, v] = E[u; v]$$

Последовательные программы

$$E[u, v] = E[u \parallel v]$$

Параллельные программы

Логика Хора-Флойда

$$\alpha \rightarrow [P]\beta$$

$$\alpha \rightarrow \langle P \rangle \beta$$

$$\alpha \rightarrow \langle P \rangle \beta \Leftrightarrow$$

$$\alpha \rightarrow \langle P \rangle 1 \wedge$$

$$\alpha \rightarrow [P]\beta$$

$$\frac{\alpha \rightarrow [P]\gamma, \gamma \rightarrow [Q]\beta}{\alpha \rightarrow [PQ]\beta}$$

$$\alpha \rightarrow [PQ]\beta$$

$$\frac{\alpha \wedge \gamma \rightarrow [P]\beta, \alpha \wedge \bar{\gamma} \rightarrow [Q]\beta}{\alpha \rightarrow [\text{if}(\gamma, P, Q)]\beta}$$

$$\alpha \rightarrow [\text{if}(\gamma, P, Q)]\beta$$

$$\frac{\alpha \rightarrow \delta, \delta \wedge \gamma \rightarrow [P]\delta, \delta \wedge \bar{\gamma} \rightarrow \beta}{\alpha \rightarrow [\text{while}(\gamma, P)]\beta}$$

$$\alpha \rightarrow [\text{while}(\gamma, P)]\beta$$

$$\alpha \rightarrow \beta(t_1, t_2, \dots)$$

$$\frac{\alpha \rightarrow \beta(t_1, t_2, \dots)}{\alpha \rightarrow [(x_1 := t_1, x_2 := t_2, \dots)]\beta(x_1, x_2, \dots)}$$

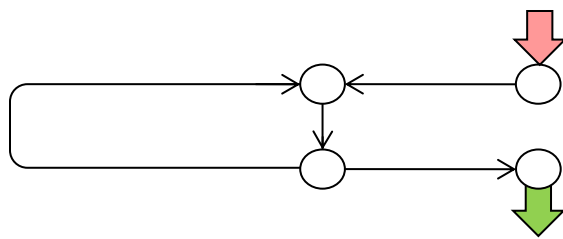
Частичная корректность программ

(если предусловие и программа останавливается, то постусловие)

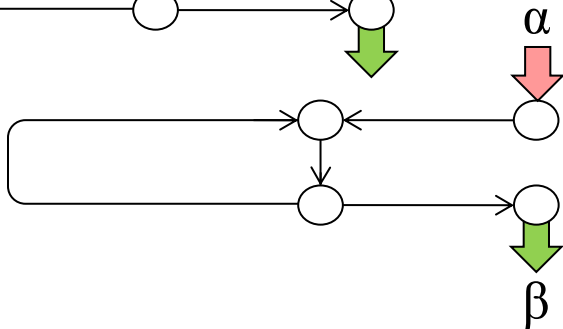
Полная корректность программ

(если предусловие, то программа останавливается и постусловие)

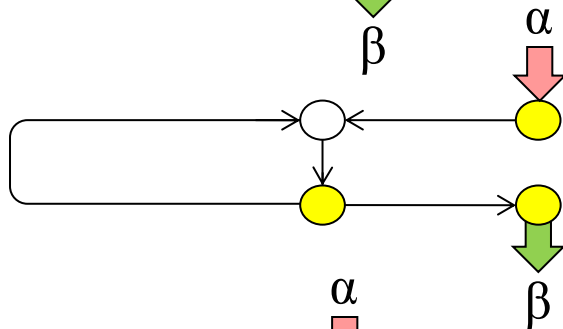
Основные этапы метода Флойда



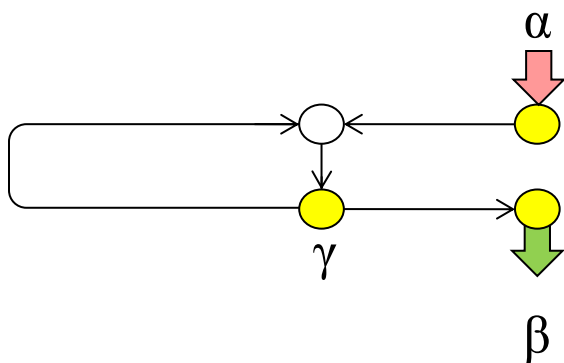
1. Представление программы в виде транзичионной системы



2. Спецификация программы (пред- и постусловия)



3. Выделение контрольных точек



4. Аннотирование программы (утверждения и предположения в контрольных точках)

Лекция 9 Инсерционная модель Метода Флойда

$$\varphi[p] \xrightarrow{\text{ask } u} (\varphi \wedge [p]u)[p], \text{Sat}(\varphi \wedge [p]u)$$

$$\varphi[p] \xrightarrow{x:=y} \varphi[p * (x := y)]$$

$$\varphi[p] \xrightarrow{\text{go to L}} \varphi[p, \text{model.L}]$$

$$\varphi[p] \xrightarrow{\text{assumption } \psi} (\varphi \wedge [p]\psi)[p]$$

$$\varphi[p] \xrightarrow{\text{assertion } \psi} \psi[\text{empty}], \neg \text{Sat}(\neg(\varphi \rightarrow [p]\psi)) \quad (|= \varphi \rightarrow [p]\psi)$$

$$\varphi[p] \xrightarrow{\text{assertion } \psi} 0, \text{Sat}(\neg(\varphi \rightarrow [p]\psi))$$

Лекция 9 Инсерционная модель Метода Флойда

$$\varphi[p] \xrightarrow{\text{ask } u} (\varphi \wedge [p]u)[p], \text{Sat}(\varphi \wedge [p]u)$$

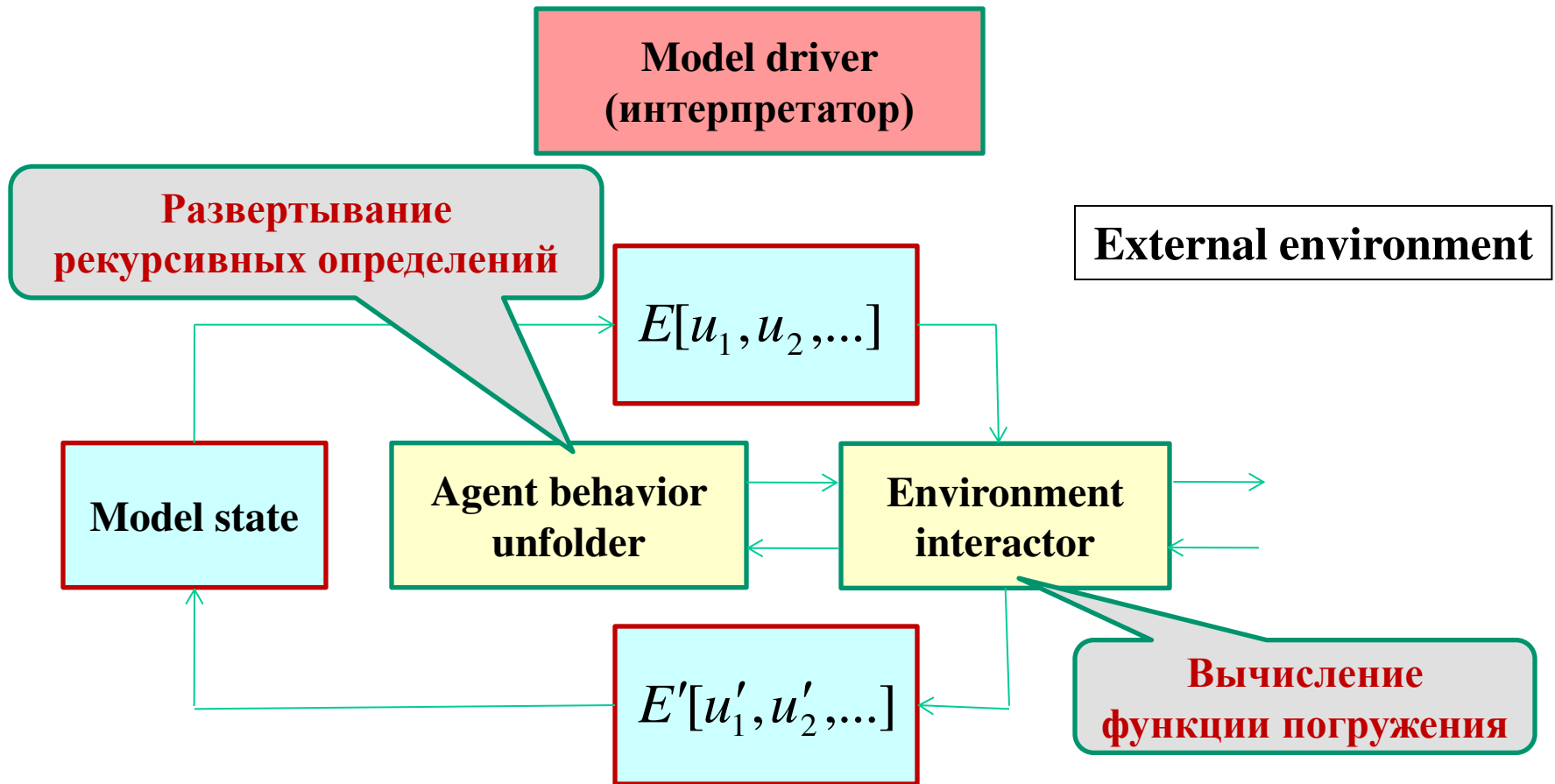
$$\varphi[p] \xrightarrow{x:=y} \varphi[p * (x := y)]$$

$$\varphi[p] \xrightarrow{\text{go to L}} \varphi[p, \text{model.L}]$$

$$\varphi[p] \xrightarrow{\text{assumption } \psi} (\varphi \wedge [p]\psi)[p]$$

$$\varphi[p] \xrightarrow{\text{assertion } \psi} \psi[\text{empty}], \neg \text{Sat}(\neg(\varphi \rightarrow [p]\psi)) \quad (\models \varphi \rightarrow [p]\psi)$$

$$\varphi[p] \xrightarrow{\text{assertion } \psi} 0, \text{Sat}(\neg(\varphi \rightarrow [p]\psi))$$



Лекция 11 Графические модели (SDL, MSC)

Лекция 12 Примеры инсерционных машин

Лекция 13 Когнитивные архитектуры

Лекция 14 VRS

Описание среды

Типы:**Типы данных:**

простые: int, real, Bool, enumerated (имена, значения),
symbolic (свободные термы), agent behaviors (уравнения в алгебре
поведений)

списки: list (m) of τ (simple)

функциональные: $(\tau_1, \tau_2, \dots) \rightarrow \tau$

массивы (рассматриваются как функциональные типы с
ограничениями)

Типы агентов:

Определяется набором имен и типизированных атрибутов

Атрибуты среды и атрибуты агентов рассматриваются как
неинтерпретированные функциональные символы.

Атрибутные выражения:

атрибуты среды (с параметрами) или выражения вида
<тип агента> <имя>.<атрибут>[(<параметры>)]

Initial states: задаются формулами или конкретными значениями
атрибутов, а также состояниями агентов, погруженных в среду.

Базовые протоколы

Алгебраическое представление:

Forall $x(\alpha(x) \rightarrow \langle P(x) \rangle \beta(x))$

Предусловия:

Формула 1-го порядка с литералами:

Постусловия:

Формула 1-го порядка, как в предусловии

Присваивания $x:=y$, рассматриваемые как утверждения вида Next $x=y$

Операторы обновления списков add to.

Конкретные модели

Состояния – состояния памяти (все атрибуты имеют конкретные значения)

+ состояния агентов: $s[m_1 : u_1, m_2 : u_2, \dots]$

Действия – базовые протоколы

$$B = \forall x(\alpha(r, x) \rightarrow \langle P(r, x) \rangle \beta(r, x))$$

$$s \xrightarrow{B} s' \Leftrightarrow \exists x(\alpha(s(r), x) \neq 0 \wedge s' = \beta(r, x)(s))$$

Символьные модели

Состояния – формулы, определяющие свойства среды и агентов

Действия – базовые протоколы

$$s \xrightarrow{B} s' \Leftrightarrow$$

$$\exists x(s \wedge \alpha(x) \neq 0),$$

$$s' = \exists x \text{ pt}(s \wedge \alpha(x), \beta(x))$$

Проблема выполнимости:

Применимость базового протокола;

Выполнимость нового состояния;

Условия целостности (невыполнимость отрицания);

Целевое условие (достижимость);

Логические теории, используемые в VRS

Линейные неравенства для вещественных атрибутов

Линейные неравенства для целочисленных атрибутов

Теория свободных термов (символьные данные)

Перечислимые типы (в том числе Boolean)

Смешанная теория

$\text{int} < \text{real} < \text{symb}$

$\text{Bool}, \text{enum}\{a, b, \dots\} < \text{symb}$

Теория с функциональными атрибутами

(неинтерпретированные функциональные символы)

Списки (очереди)

Алгебра поведений

Теория линейных целочисленных неравенств разрешима:

Алгоритм Прессбургера

Теория нелинейных вещественных неравенств разрешима:

Алгоритм Тарского

Теория нелинейных целочисленных неравенств **неразрешима**:

Результат Матиасевича (10-я проблема Гильберта)

Теория свободных термов разрешима:

Алгоритм унификации (метод резолюций Робинсона)

Выполнимость пропозициональных функций

SAT проблема

DPLL: Davis-Putnam-Logemann-Loveland

алгоритм

$$f(x_1, x_2, \dots) = x_1 \wedge f(1, x_2, \dots) \vee \overline{x_1} \wedge f(0, x_2, \dots)$$

Продолжение развертки – разложение Шеннона.

Стратегии выбора переменной

для разложения – современные алгоритмы решения

SAT проблемы

Model checking

BDD, темпоральная логика (LTL,...)

Модель определяется базовыми протоколами с булевскими атрибутами

Символьная модель: состояния системы – булевские формулы

SMT

Бескванторные формулы

Формулы с кванторами

Инсерционное моделирование 2

1. Последовательная композиция и итерация
2. Параметрические системы уравнений в алгебре поведений
3. Нелинейные уравнения в алгебре поведений
4. Расширенная алгебра поведений
5. Многоосновная расширенная алгебра поведений
6. Классификация функций погружения
7. Параллельное погружение
8. Мобильность
9. Аддитивные, коммутативные и приоритетные погружения
10. Конечно-разложимые среды
11. Атрибутные среды
12. Локальные свойства
13. Переходы конкретных и символьных сред
14. Дискретные и непрерывные нейроны
15. Дискретные автоматные сети
16. Нейронные сети
17. Абстракции для атрибутных сред
18. Абстракции для систем
19. ICAR, IMS и APS
20. Среда императивного программирования
21. Многоуровневые императивные среды

22. Денотационная семантика функционального программирования
23. Операционная семантика функционального программирования
24. Многоуровневая семантика функционального программирования
25. Стратегии вычислений в функциональном программировании
26. Параллельное функциональное программирование
27. Алгебраическое программирование (операционная семантика)
28. Денотационная семантика
29. Теорема Черча-Россера
30. Свойства систем переписывания
31. Расширения алгебраического программирования.
32. Логическое программирование, денотационная семантика
33. Логическое программирование, операционная семантика
34. Алгоритмы унификации.
34. Стратегии переписывания.

Задачи

Задачи к лекции 2

1. Показать, что, если $u = aub$, то $u = au$
2. Решить уравнение $u = aubuc + \Delta$
3. Построить алгоритм решения нелинейных уравнений (обобщение алгоритма синтеза стековых автоматов), если это возможно.

Решение: система линейных уравнений.

Упражнение к лекции 3

Написать остальные два уравнения для сети из автоматов

Задачи

1. Построить поведение непрерывного нейрона
2. Построить среду для дискретных нейронов
3. Построить среду для непрерывных нейронов (аттракторы)

Задача к лекции 4:

перенести алгоритмы обучения непрерывных сетей на дискретные.

Задачи к лекции 6

1. Построить семантику вызова по имени
2. Исследовать выполнение операторов и проверку условий в многоуровневых средах (использование и изменение атрибутов принадлежащих различным уровням)

Задачи к лекции 8

1. Доказать $s^n(0) + s^m(0) \xrightarrow{*} s^{n+m}(0)$
2. Построить правила для умножения $*$ и доказать

$$s^n(0) * s^m(0) \xrightarrow{*} s^{n*m}(0)$$