7 Февраля 2012

Инсерционное моделирование 2

Лекция 1

Обзор и введение

Размеченные транзиционные системы

$$< S, A, T >, \quad T \subseteq S \times A \times S$$

$$s \xrightarrow{a} s'$$

Абстракция:
отвлекаемся от внутренних состояний, рассматриваем то, что можно наблюдать.
Самая полезная абстракция в компьютерной науке.

***A* – множество меток**
символы,
события,
действия

Непрерывные динамические системы:
действия – длительность перехода
+информация о состояниях (проекция,…)

Автоматы:
входные символы + …, вход/выход

Программы:
операторы, условия, условные операторы

Исчисления:
состояния – формулы,
действия – правила вывода

Взаимодействующие распределенные системы:
передача (прием) сообщений

Базы данных
запросы, ответы

Поведение и эквивалентность

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

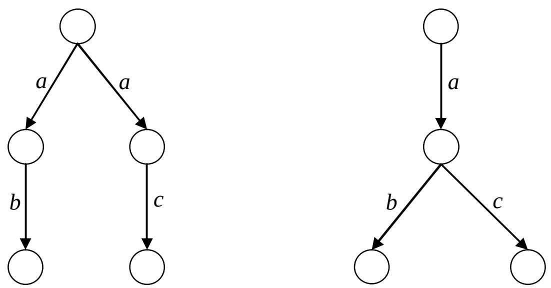
Трасса: $a_1 a_2 \ldots a_n \ldots$

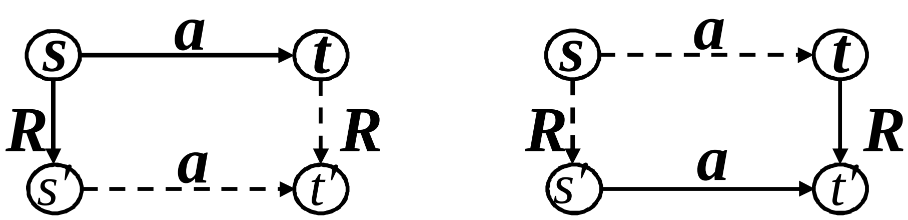
$L^0_\Delta(S)$ **трассы из** S_0 **в** S_Δ

Трассовая эквивалентность

$$\boxed{S \sim_T S' \Leftrightarrow L(S^0_\Delta) = L((S')^0_\Delta)}$$

Трассовая эквивалентность слишком слаба:





bisimulation => bisimila

Бисимуляционная эквивалентность

Бесконечные деревья

Алгебра поведений (процессов)

- **Сигнатура:**

  префиксинг *a.u*, недетерминированный выбор *u + v*, константы Δ, 0, $\perp$

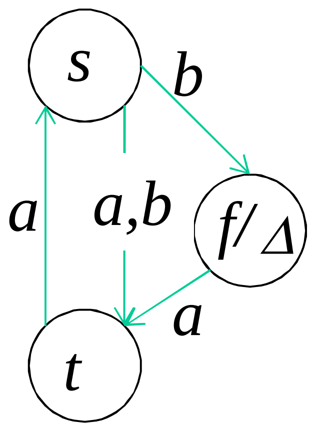
  отношение аппроксимации $\sqsubseteq$

- **Нормальна форма:** $u = (\sum_{i \in I} a_i.u_i) + \varepsilon_u$
- **Теорема о неподвижной точке и теорема анализа:** $u_s = \sum_{s \xrightarrow{a} t} a.u_t + \varepsilon_s$
- **Отождествление состояний с поведениями:**

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

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

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



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

$$u_t = a_s$$

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

- **Состояния бисимуляционно эквивалентны <=> их поведения совпадают**

Обогащенная алгебра поведений

Последовательная композиция

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

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

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

Функциональные рекурсивные определения (поведений!)
Сравнить с функциональным программированием

Параллельная композиция

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

$$\varepsilon \parallel \varepsilon' = \varepsilon' \parallel \varepsilon, \quad \varepsilon \parallel \Delta = \varepsilon, \quad \varepsilon \parallel \perp = \perp, \quad 0 \parallel \Delta = 0 \parallel 0 = 0$$

Функция погружения

Агенты: *размеченные или атрибутные транзиционные системы*
Среды: *атрибутные транзиционные системы наделенные функцией погружения*
Композиция: *непрерывная* ***функция погружения*** *характеризующая поведение среды с погруженными в нее агентами (insertion function)*
Инсерционная модель: *среда с погруженными в нее агентами*

$$\mathrm{Ins}: E \times F(A) \to E \qquad \mathrm{Ins}(e,u) = e[u] \qquad e[u]_E$$

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

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

Все с точностью до бисимуляционной или трассовой эквивалентности

Применения инсерционного моделирования:

Построение операционной семантики языков последовательного и параллельного программирования и моделей распределенных вычислений

Функция погружения

$$\frac{P \xrightarrow{a} P',\ s \xrightarrow{a} s'}{s[P] \xrightarrow{\delta} s'[P']}$$

$$s[P,Q] = s[P;Q]$$

$$s[P+Q] = s[P] + s[Q]$$

Переходы среды

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

$$u(s) \Rightarrow s \xrightarrow{u} s$$

u – условие y – присваивание

Переходы программы

$$if\,(u,P,Q) \xrightarrow{u} P$$

$$if\,(u,P,Q) \xrightarrow{\neg u} Q$$

$$while(u,P) \xrightarrow{u} (P; while(u,P))$$

$$while(u,P) \xrightarrow{\neg u} \Delta$$

$$x := y \xrightarrow{x:=y} \Delta,$$

$$\frac{P \xrightarrow{a} \Delta}{(P;Q) \xrightarrow{a} Q}$$

$$\frac{P \xrightarrow{a} P',\ P' \neq \Delta}{(P;Q) \xrightarrow{a} (P';Q)}$$

$$if\,(u,P,Q) = u.P + (\neg u).Q$$

$$while(u,P) = u.(P; while(u,P)) + (\neg u).\Delta$$

$$f_i(x_1, x_2, \ldots) = E_i(f, x),$$
$$i = 1, \ldots, n, x = (x_1, x_2, \ldots)$$
$$f_i \in \Phi, x_j \in V, E_i(f, x) \in T_{\Omega \cup \Phi}(C, V)$$

$$\frac{F = G[z := f_i(t_1, t_2, \ldots)], f_i(x_1, x_2, \ldots) = E_i \in R}{R[F] \to R[simpl(G[z := E_i[x_1 := t_1, x_2 := t_2, \ldots]])]}$$

$$R[F] \to R[simpl(F)]$$

Синтаксис

$T_\Omega(C,V)$

$T_\Omega(C)$

$$rs(x_1, x_2, \ldots)(s_1 = t_1, s_2 = t_2, \ldots)$$

Операционная семантика

$$t \to t' \qquad t \leftrightarrow t'$$

$$\frac{t = G[x := s], s = s_i[x_1 := u_1, x_2 := u_2, \ldots]}{t \to G[x := t_i[x_1 := u_1, x_2 := u_2, \ldots]]}$$

$1)\ t \xrightarrow{*} t'$ — ***Семантика переписывания***

$2)\ t \xleftrightarrow{*} t'$ — ***Эквациональная семантика***

Расширения алгебраического программирования

- Условные правила
- Несвободная алгебра
- **Упрощения после переписывания (APS)**
- Ассоциативно коммутативное переписывание
- Типы (ML)

Операционная семантика языка логического программирования

Алгоритм перечисления ответов на запрос представленный в виде конъюнкции атомарных формул (литералов)

$$(Q,\sigma) \to (Q',\sigma')$$

Логическая программа

$p(t_1,t_2,\ldots):-$ аксиомы

$p(t_1,t_2,\ldots):-P_1 \wedge P_2 \wedge \ldots$ правила

$t_i \in T_\Omega(A,V),\ p \in \Pi$

P_i атомарные формулы

$Q(x)$ запрос

Унификация: $s\sigma = t\sigma$ **,**
наиболее общий унификатор

$$\frac{Q = q \wedge R,\ p:-S \in_r P,\ \tau = mgu(p,q\sigma)}{(Q,\sigma) \to (R \wedge S,\sigma\tau)}$$

$$Var(p) \cap Var(q) = \varnothing$$

Переименование переменных

$$(Q,\varepsilon) \xrightarrow{*} (1,\sigma)$$

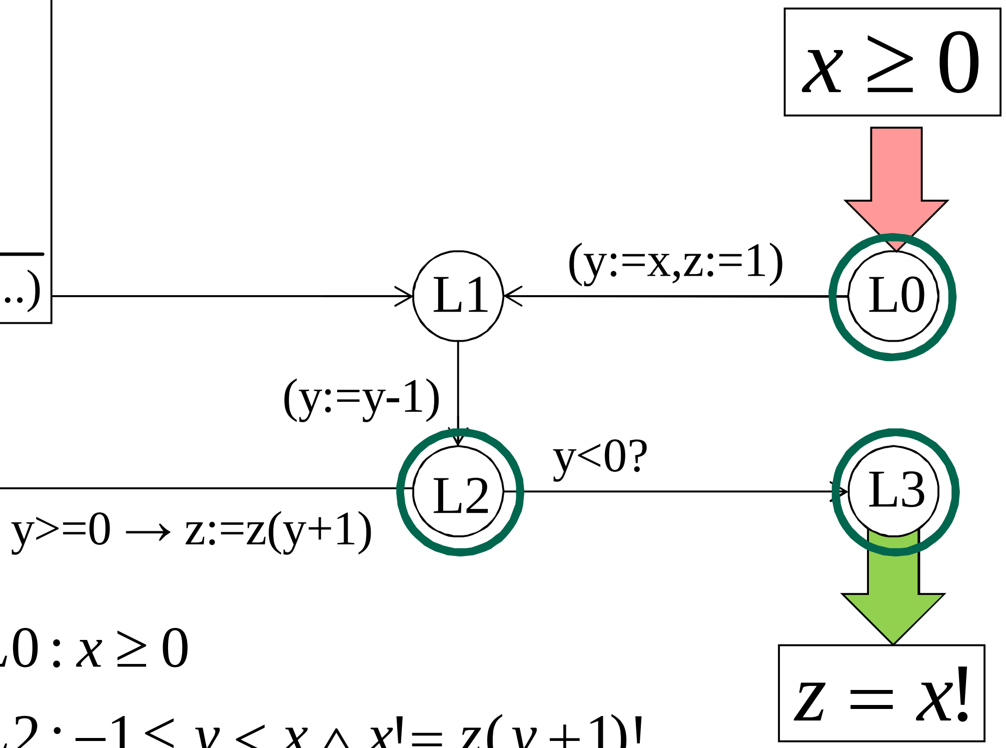
Исчисление Хоара и метод Флойда доказательства корректности императивных программ

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

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

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

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



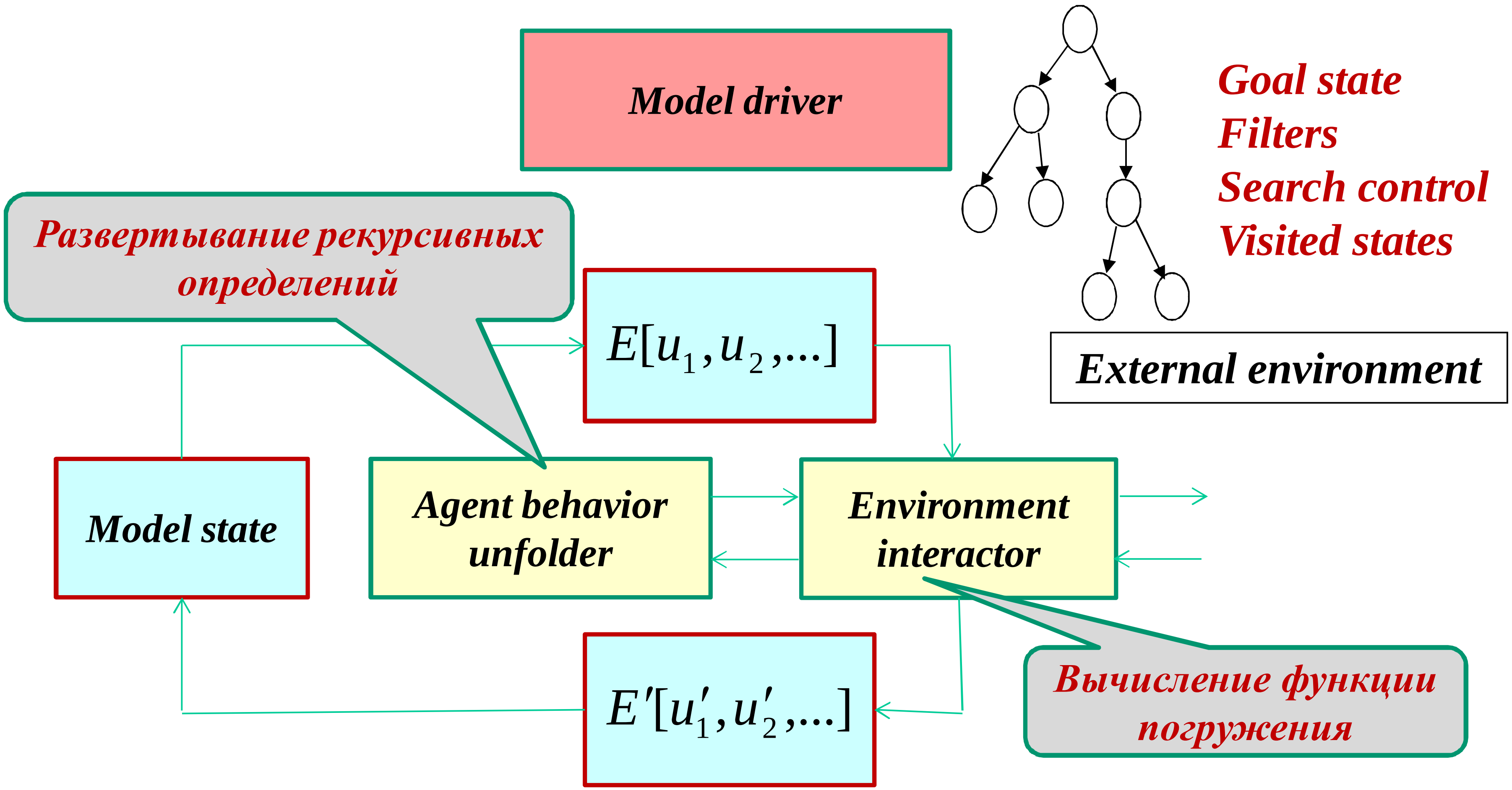
$$L0 : x \geq 0$$

$$L2 : -1 \leq y < x \wedge x! = z(y+1)!$$

$$L3 : z = x!$$

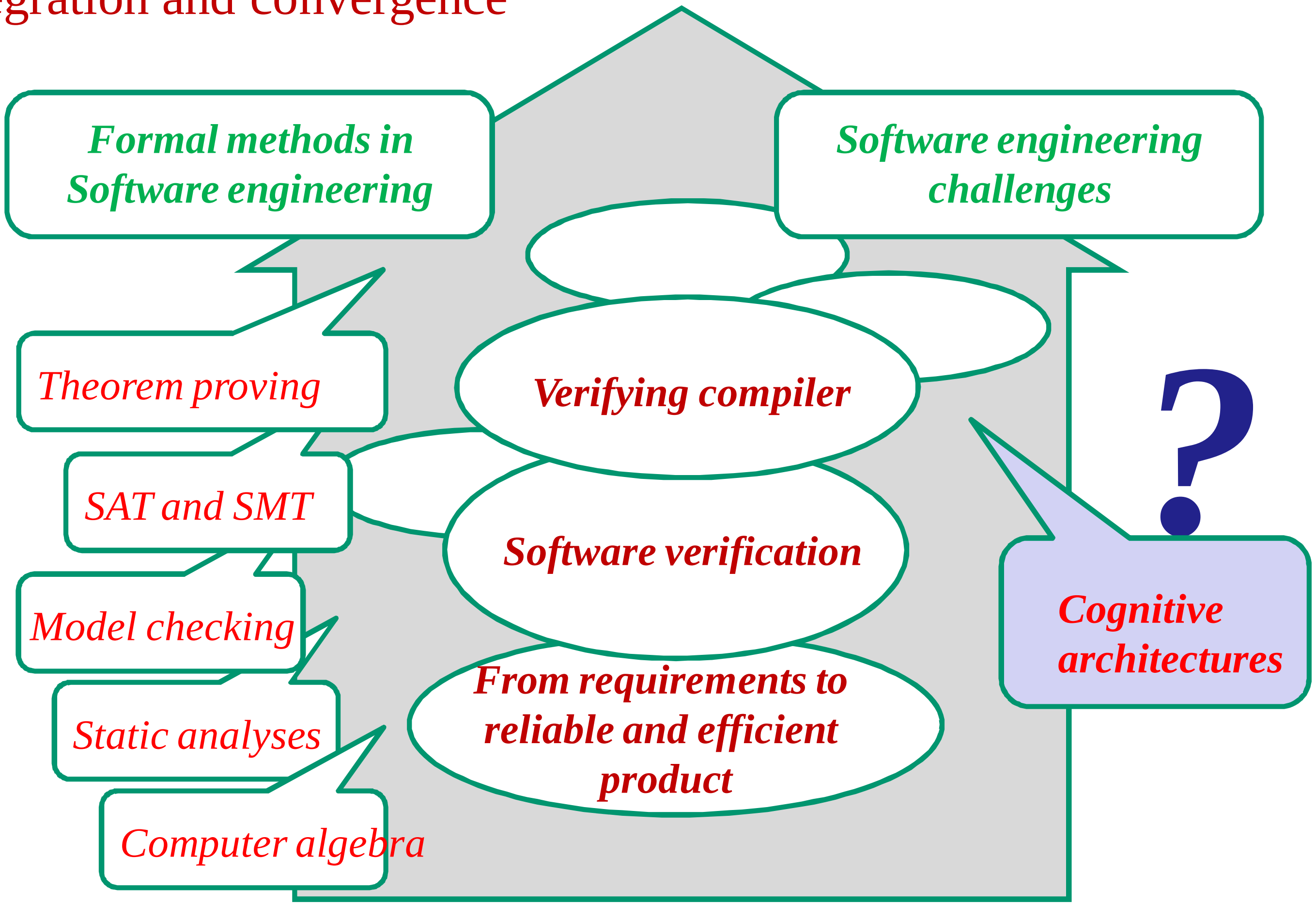
Аналитическая инсерционная машина

(верификация, прогнозирование, принятие решений)



http://bicasociety.org/

Integration and convergence



Лекция 13: APS/IMS

Лекция 15: Верификационная машина

ИНСЕРЦИОННОЕ МОДЕЛИРОВАНИЕ 2

Дальнейшие приложения инсерционного моделирования:

Верификация

- требований,
- спецификаций,
- распределенных систем

Семантика алгебр и исчислений взаимодействующих процессов:

- CCS
- π-calculus
- Исчисление мобильных амбиентов

Сети Петри

Графические языки и их семантика

- MSC
- SDL
- UCM (объединение концепций сетей Петри и мобильных амбиентов)

Язык базовых протоколов

Символьное моделирование