

# Синтез функциональных программ при помощи метода дедуктивных таблиц.

Подготовил: Фастовец Н.Н.

Научный руководитель: Корухова Ю.С.

# Содержание

- Автоматический синтез программ
- Дедуктивные таблицы
- Свойства дедуктивных таблиц
- Дедуктивные правила
- Пример синтеза
- вспомогательные таблицы
- Пример ввода вспомогательных таблиц
- Заключение

# Автоматический синтез программ

Предпосылки :

- Увеличение сложности ПО
- Увеличение требований к надежности ПО

Основные направления:

- Дедуктивный синтез
- Индуктивный синтез
- Трансформационный синтез

# Дедуктивные таблицы (1)

Один из методов дедуктивного синтеза.

Спецификация задается в виде формулы логики предикатов первого порядка

$$\forall x \exists y Q[x,y]$$

где  $x$  – входная переменная,

$y$  – выходная переменная,

$Q$  – логическая формула, устанавливающая связь между входными и выходными переменными.

# Дедуктивные таблицы (2)

## Структура дедуктивной таблицы

Утверждения	Цели	Выходной терм		
		f1(x)	...	fn(x)
A1		t1		tn
...		...		...
Ak		t1'		tn'
	G1	s1		sn
	...	...		...
	Gm	s1'		sn'

$\text{If}(A_1 \vee \dots \vee A_n) \text{ then } (G_1 \wedge \dots \wedge G_m)$

# Дедуктивные таблицы (3)

Терм  $t$ , не содержащий свободных (на связанных с квантором) переменных, удовлетворяет строке таблицы

A		s
---	--	---

или

	G	s
--	---	---

если для некоторой подстановки  $\lambda$ :

- Утверждение  $A\lambda$  не содержит свободных переменных и является ложным ( $G\lambda$  – истинным)
- Если строка имеет выходной терм  $s$ , то  $s\lambda$  не содержит свободных переменных и равен  $t\lambda$ .

# Дедуктивные таблицы (4)

Доказательство в дедуктивной таблице проводится до получения финальной строки

	true	t1		tn
--	------	----	--	----

**ИЛИ**

false		t1		tn
-------	--	----	--	----

# Свойства дедуктивных таблиц (1)

- Двойственность

строка

	$G$	$t1$
--	-----	------

эквивалентна строке

$\neg G$		$t1$
----------	--	------



# Свойства дедуктивных таблиц (2)

- Переименование свободных переменных

$\pi$  – перестановка

строка

	$G$	$t$
--	-----	-----

эквивалентна строке

	$G\pi$	$t\pi$
--	--------	--------

# Свойства дедуктивных таблиц

## (3)

- Добавление примера

$\lambda$  – подстановка

Таблица

	G	t
--	---	---

ЭКВИВАЛЕНТНА

	G	t
	$G\lambda$	$t\lambda$

# Свойства дедуктивных таблиц (4)

- Тождественные преобразования

Эквивалентность таблиц не нарушается при применении в строках тождественных преобразований на основе тождеств алгебры логики:

$$A \wedge A = A; \quad A \wedge \text{false} = \text{false};$$

$$A \vee \text{true} = \text{true}; \quad \neg \neg A = A;$$

# Свойства дедуктивных таблиц (5)

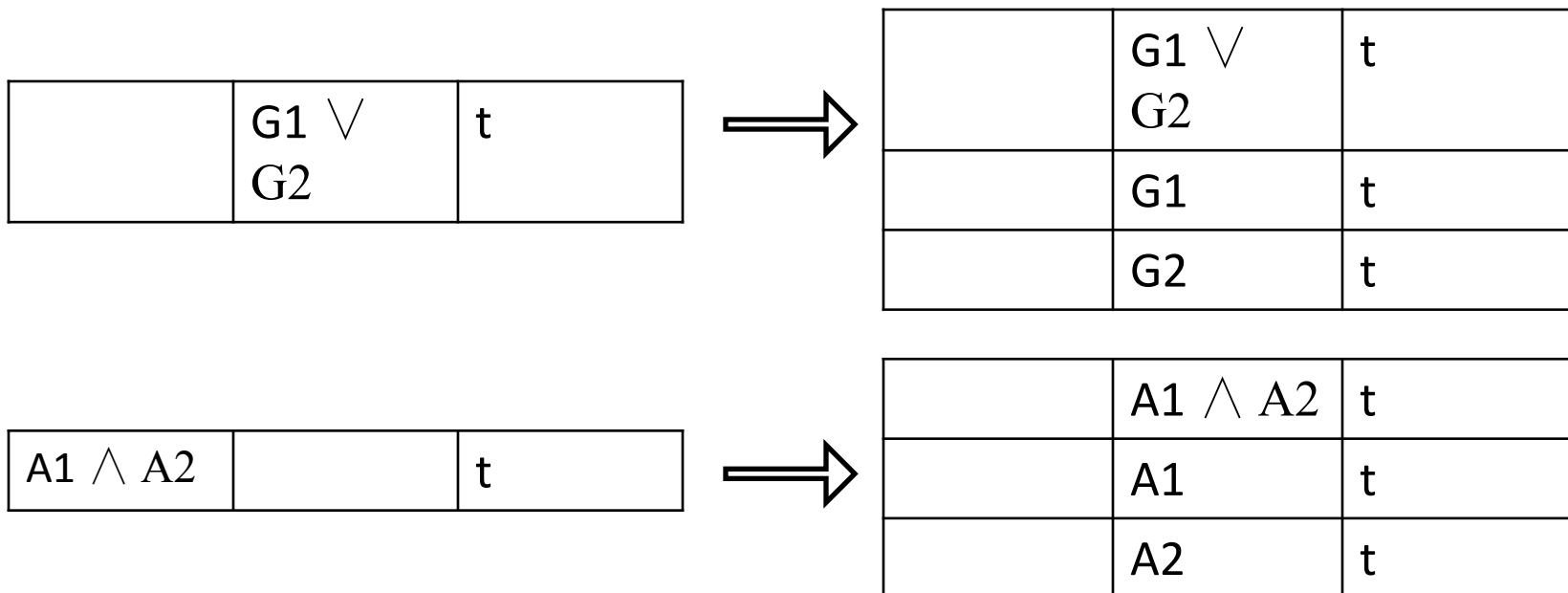
- Добавление и удаление тождественно истинного утверждения и тождественно ложной цели
- Добавление и удаление свободной переменной

# Дедуктивные правила (1)

- Вычислимые термы – термы, не содержащие свободных переменных, которые можно получить на основе уже определенных констант, функций и предикатов.
- Таблицы, для которых совпадают множества удовлетворяющих им вычисляемых термов термов совпадают, называются подобными.

# Дедуктивные правила (2)

- Правило OR- и AND-разделения



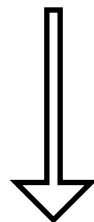
$G1, G2, A1, A2$  – логические выражения,  
 $t$  – выходной терм.

# Дедуктивные правила (3)

- Правило резолюции

i		G1[P]	t
---	--	-------	---

k		G2[P']	s
---	--	--------	---



r		G1λ [true] ∧ G2λ [false]	If Pλ then sλ else tλ
---	--	--------------------------	-----------------------

G1,G2 – логические выражения,

P, P' – некоторые подвыражения G1 и G2,

λ – подстановка, для которой Pλ = P'λ,

s и t – выходные термы.

# Дедуктивные правила (4)

- Правило индукции.

$wf$ -отношение – отношение, исключаящее бесконечно убывающие цепочки.

Спецификация:

	$Q[a,z]$	$z$
--	----------	-----

где  $a$  – входной параметр,  $z$  – выходная переменная,  $Q$  – логическая формула.

Гипотеза индукции:

$\text{If } x <_{wf} a \text{ then } Q[x, f(x)]$		
--	--	--

где  $<_{wf}$  –  $wf$ -отношение в рассматриваемой теории

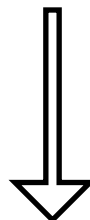


# Дедуктивные правила (5)

- Правило замены эквивалентных термов

i		$G1[L=R]$	t
---	--	-----------	---

k		$G2\langle T \rangle$	s
---	--	-----------------------	---



r		$G1\lambda [false] \wedge G2\lambda\langle R\lambda \rangle$	If $[L=R]\lambda$ then $s\lambda$ else $t\lambda$
---	--	--	---

где  $G1, G2$  – логические выражения,

$[L=R]$  – подвыражение  $G1$ ,

$T$  – некоторый терм, входящий в  $G2$ , такой что для некоторой подстановки  $\lambda: T\lambda = L\lambda$ ,

$s$  и  $t$  – выходные термы.

# Пример синтеза функциональной программы

(1)

Спецификация:

$z = \text{fact}(n)$  is

if  $n = 0$  then  $z = 1$  else  $z = n * \text{fact}(n-1)$

В таблице

1		$(n=0) \wedge (z=1) \vee \neg (n=0) \wedge (z=n * \text{fact}(n-1))$	$z$
---	--	--	-----

согласно тождеству

if  $A$  then  $B$  else  $C \equiv A \wedge B \vee \neg A \wedge C$

# Пример синтеза функциональной программы (2)

## 1. OR-разделение

1		$(n=0) \wedge (z=1) \vee \neg (n=0) \wedge (z=n*\text{fact}(n-1))$	z
2		$(n=0) \wedge (z=1)$	z
3		$\neg (n=0) \wedge (z=n*\text{fact}(n-1))$	z

## 2. Резолюция. Общее подвыражение . (n=0).

4		$(z1=1) \wedge (z2=n*\text{fact}(n-1))$	If n = 0 then z1 else z2
---	--	---	--------------------------

## 3. Добавление примера $\{z1 \leftarrow 1, z2 \leftarrow n*\text{fact}(n-1)\}$

4		true	If n = 0 then 1 else $n*\text{fact}(n-1)$
---	--	------	---

# Вспомогательные таблицы (1)

Пусть в таблице присутствуют строки

i		$G[t[a],x]$	$r[a,x]$
...	...	...	...
k		$F[G[t'[a],x']]$	$r'[a,x']$

где  $G, F$  – логические выражения,  
 $a$  – входной параметр,  $t[a], t'[a]$  – термы  
над входным параметром,  $x, x'$  –  
выходные переменная,  $r[a,x], r'[a,x']$  –  
выходные термы.  $F$  содержит реплику  $G$ .

# Вспомогательные таблицы (2)

Если  $t'[a] <_{wf} t[a]$ , можно в выражении  $G$  заменить все вхождения  $t[a]$  на произвольную константу и построить обобщенную цель  $G'[c,x]$ , которая станет исходной целью для вспомогательной таблицы (соответственно – спецификацией для вспомогательных функций).

1a		$G'[c,x]$	x
----	--	-----------	---

# Вспомогательные таблицы (3)

- Исходная цель

1a		$G'[c,x]$	x
----	--	-----------	---

- Гипотеза индукции

2a	if $y <_{wf} c$ then $G'(y, f_{new}(y))$		x
----	--	--	---

- Воспроизведение шагов доказательства

$(k+1)a$		$F'[G'[t''[c], x'']]$	$r'''[c, x'']$
----------	--	-----------------------	----------------

- Резолюция с гипотезой индукции

$(k+2)a$		$F'[true] \wedge (t''[c] <_{wf} c)$	$r'''[c, f_{new}(t''[c])]$
----------	--	-------------------------------------	----------------------------

# Вспомогательные таблицы (4)

## Лемма

$k+1$	$G_1(y, f_{\text{new}}(y))$		
-------	-----------------------------	--	--

имеющаяся строка (i)

i		$G[t[a],x]$	$r[a,x]$
---	--	-------------	----------

их резолюция завершает  
доказательство

$k+2$		true	$r''[a, f_{\text{new}}(t[a])]$
-------	--	------	--------------------------------

# Пример вывода вспомогательной таблицы (1)

Для функции сортировки списка чисел задается спецификация вида

$$z = \text{sort}(a) \text{ is } \text{perm}(a, z) \wedge \text{ord}(z)$$

$\text{perm}(x, y)$  – предикат, равный true, если его аргументы-списки есть перемтановки друг-друга.

$\text{ord}(x)$  – предикат, равный true, если его аргумент-список упорядочен.



# Пример вывода вспомогательной таблицы (2)

- В ходе доказательства получается

20		$\text{perm}(\text{tail}(b), w^x) \wedge$ $\text{perm}(\text{tail}(b), x2^w) \wedge$ $\text{perm}(\text{tail}(b), x3^x) \wedge$ $[\text{if iselem}(f, x) \text{ then head}(b) \leq f] \wedge$ $[\text{if iselem}(h, w) \text{ then } h \leq \text{head}(b)]$	$\text{if } b = \text{null} \text{ then null}$ $\text{else}$ $\text{sort}(w)^\wedge$ $\text{head}(b) * \text{sort}(x)$
----	--	--	--

- Далее, строка

30		$\text{perm}(\text{tail}(\text{tail}(b)), w^y) \wedge \text{perm}(\text{tail}(\text{tail}(b)), v2^w) \wedge$ $\text{perm}(\text{tail}(\text{tail}(b)), x3^y) \wedge$ $[\text{if } f = \text{head}(\text{tail}(b)) \text{ then head}(b) \leq f] \wedge$ $[\text{if iselem}(f, y) \text{ then head}(b) \leq f] \wedge$ $[\text{if iselem}(h, w) \text{ then } h \leq \text{head}(b)] \wedge \neg \text{tail}(b) = \text{null}$	$t_{(30)}$
----	--	--	------------

- Исходная цель вспомогательной таблицы

#	Assertion	Goal	f1	f2
1 a		$\text{perm}(c, w^x) \wedge \text{perm}(c, x2^w) \wedge$ $\text{perm}(c, x3^x) \wedge$ <b>[if iselem(f, x) then head(b) ≤ f] ∧</b> <b>[if iselem(h, w) then h ≤ head(b)]</b>	w	x

- Лемма

3 1		$\text{perm}(y, \text{les}(y)^{\text{gre}(y)}) \wedge \text{perm}(y, x2^{\text{les}(y)}) \wedge$ $\text{perm}(y, x3^{\text{gre}(y)}) \wedge$ <b>[if iselem(f, gre(y)) then head(b) ≤ f] ∧</b> <b>[if iselem(h, les(y)) then h ≤ head(b)]</b>		
--------	--	---	--	--

Конец

# ИСТОЧНИКИ

- Источники
- Kreitz, C. Program Synthesis Chapter III.2.5 of *Automated Deduction – A Basis for Applications*, pp. 105–134, Kluwer, 1998.
- Ayari, A., Basin D.: A High-Order Interpretation Of Deductive Tableau
- J. Symbolic Computation (2001) 11, 1-32
- Manna Z., Waldinger R.: Fundamentals of Deductive Program Synthesis
- IEEE Transactions on Software Engineering, vol. 12, No. 8, August 1992
- Traugott J. Deductive Synthesis of Sorting Programs
- Journal of Symbolic Computation, vol.7, 1989.