Доказательство теорем в системе Isabelle / HOL. Интенсивный курс.

- Данный курс подготовлен на основе курса формальных методов за авторством Тобиаса Нипкова (Tobias Nipkow), профессора Мюнхенского Технологического Университета.
- Note: this material is based on the original course developed by Prof. Tobias Nipkow from TUM University. For more details see http://isabelle.in.tum.de/coursematerial/PSV2009-1/

Примечание*

Переведено и озвучено:

- Якимов И.А.
  - ivan.yakimov.research@yandex.ru
- Кузнецов А.С.
  - askuznetsov@sfu-kras.ru

Слайды, отмеченные звездочкой*, добавлены переводчиками, так же добавлены некоторые примечания.

На момент перевода и чтения курса в СФУ оригинальный курс отрыт и доступен публично http://isabelle.in.tum.de/coursematerial/PSV2009-1/

Симплификация:
Доказательство через упрощение

Доказательство теорем в системе Isabelle / HOL. Интенсивный курс.

- Данный курс подготовлен на основе курса формальных методов за авторством Тобиаса Нипкова (Tobias Nipkow), профессора Мюнхенского Технологического Университета.
- Note: this material is based on the original course developed by Prof. Tobias Nipkow from TUM University. For more details see http://isabelle.in.tum.de/coursematerial/PSV2009-1/

Примечание*

Переведено и озвучено:

- лектор — Кузнецов А.С.
  - askuznetsov@sfu-kras.ru
- ассистент — Якимов И.А.
  - ivan.yakimov.research@yandex.ru

Слайды, отмеченные звездочкой*, добавлены переводчиками, также добавлены некоторые примечания.

На момент перевода и чтения курса в СФУ оригинальный курс отрыт и доступен публично http://isabelle.in.tum.de/coursematerial/PSV2009-1/

Основы переписывания термов
Рерайтинг

Переписывание термов означает …

- Использование уравнений ***l* = *r*** слева направо
  - столько, сколько возможно
- Терминология:
  - уравнение ⇝ правило переписывания, шаблон

Пример:

- Уравнение:

$$
\begin{aligned}
0 + n &= n && (1)\\
(Suc\ m) + n &= Suc\ (m + n) && (2)\\
(Suc\ m \leq Suc\ n) &= (m \leq n) && (3)\\
(0 \leq m) &= True && (4)
\end{aligned}
$$

- Переписывание:

$$
\begin{aligned}
0 + Suc\ 0 &\leq Suc\ 0 + x && \stackrel{(1)}{=}\\
Suc\ 0 &\leq Suc\ 0 + x && \stackrel{(2)}{=}\\
Suc\ 0 &\leq Suc\ (0 + x) && \stackrel{(3)}{=}\\
0 &\leq 0 + x && \stackrel{(4)}{=}\\
& True &&
\end{aligned}
$$

Более формально

Подстановка это отображение переменных на термы

- Уравнение **$l = r$** применимо к терму **$t[s]$** если существует подстановка **σ** такая, что **$\sigma(l) = s$**
- Результат: **$t[\sigma(r)]$**
- Прим.: **$t[s] = t[\sigma(r)]$** // $s = \sigma(l) = \sigma(r)$; $t[s] = t[\sigma(l)] = t[\sigma(r)]$;

Пример:

Уравнение:	**$0+n = n$**	// **l:** $0+n$;	**r:** n;
Терм:	**$a+(0+(b+c))$**	// **s:** $0+(b+c)$;	**t:** $a+(s)$;
Подстановка:	**$\sigma = \{n \mapsto b+c\}$**	// **$\sigma(l)$:** $0+(b+c)$;	**$\sigma(r)$:** $(b+c)$;
Результат:	**$a+(b+c)$**	// **$t[\sigma(r)]$:** $a+(b+c)$;	

Условное переписывание

Переписывание может иметь условие:

$$[[P_1 \ldots P_n]] \Longrightarrow l = r$$

применимо к терму t[s] с подстановкой σ если:

- $\sigma(l) = s$ и
- $\sigma(P_1), \ldots, \sigma(P_n)$ доказуемы (также переписыванием)

Интерлюдия — переменные в Isabelle

Схематические переменные

Три вида переменных:

- Связанные: $\forall$ x. x = x
- Свободные: x = x
- Схематические: ?x = ?x

Схематические переменные:

- Логически:
  – свободные = схематические
- Операционно:
  – свободные переменные фиксированы
  – схематические переменные инстанциируются подстановками

От x к ?x

Формулируем **лемму** со свободными переменными:

lemma *app_Nil2[simp]: xs @ [] = xs*

…

done

После доказательства: Isabelle подменяет xs на ?xs (внутренне):

?xs @ [] = ?xs

Теперь применимо с любыми значениями для ?xs

Пример: переписываем

rev(a @ []) = rev a

используя app_Nil2 с $\sigma = \{ ?xs \rightarrow a \}$

Переписывание термов в Isabelle

Базовые техники упрощения

Цель: 1. [|P_1; …; P_m|] ==> C

apply (simp add: eq_1 … eq_n)

Упрощаем **P_1 … P_m** и **C** используя

- Леммы с атрибутом **simp**
- Правила из **primrec**, **fun** и **datatype**
- Дополнительные леммы **eq_1 … eq_n**
- Посылки **P_1 … P_m**

simp можно настраивать, удаляя или добавляя правила:

(simp … del: …)

`auto VS simp`

- **auto** действует на все цели сразу
- **simp** действует на первую цель
- **auto** использует **simp** и не только

Терминация

Симплификация может уйти в бесконечный цикл.
Isabelle использует правила упрощения (почти) вслепую слева направо.

Пример: f(x) = g(x), g(x) = f(x), …

$$[|P_1...P_n|] ==> l = r$$

применимо как правило упрощения только

если l «больше» чем r и каждое P_i

n < m ==> (n < Suc m) = True   Хорошо

Suc n < m ==> (n < m) = True   Плохо

Переписывание и definition

- Определения не имеют атрибута simp.
  - они должны быть использованы явно:
  - (simp add: f_def)

Расширение рерайтинга

Локальные предположения

- Упрощение A → B:
  - упростить A до A'
  - упростить B используя A'

Разделение случаев с simp

$$P\ (\text{if } A \text{ then } s \text{ else } t)$$

$$=$$

$$(A \rightarrow P(s)) \wedge (\sim A \rightarrow P(t))$$

Автоматически

$$P\ (\text{case } e \text{ of } 0 \Rightarrow a \mid Suc\ n \Rightarrow b)$$

$$=$$

$$(e = 0 \rightarrow P(a)) \wedge (\forall n.\ e = Suc\ n \rightarrow P(b))$$

Вручную: (simp split: nat.split)

Аналогично для любого типа данных t: t.split

Упорядоченный рерайтинг

Проблема: ?x + ?y = ?y + ?x не терминирует

Решение: перестановочные правила упрощения используются только если *запись* терма становится лексикографически меньше.

Пример: b + a ~> a + b, но не a + b ~> b + a

Для типов nat, int и т. д.:

- леммы add_ac сортирует любую сумму (+)
- леммы times_ac сортируют любое произведение (*)

Пример: (simp: add_ac) вернет:

(b + c) + a ~> … ~> a + (b + c)

Препроцессинг

правила упрощения (рекурсивно) подвергаются препроцессингу:

$$
\begin{array}{rcl}
\neg A & \mapsto & A = False \\
A \longrightarrow B & \mapsto & A \Longrightarrow B \\
A \wedge B & \mapsto & A, B \\
\forall x. A(x) & \mapsto & A(?x) \\
A & \mapsto & A = True
\end{array}
$$

Пример:

$$
(p \longrightarrow q \wedge \neg r) \wedge s \quad \mapsto \quad \left\{ \begin{array}{r} p \Longrightarrow q = True \\ p \Longrightarrow r = False \\ s = True \end{array} \right\}
$$

Трейсинг

lemma …

using [[simp_trace]]

…

oops