

並列代入に対する代入補題の自動証明

坂口和彦

筑波大学 情報学群 情報科学類

2015/3/4 PPL2015

- ▶ 単純型付き λ 計算と System F の強正規化定理を Coq で証明

Theorem (強正規化定理)

t が型の付く項ならば、 t から始まる簡約列は有限長

¹ $u[x_0 := t_0] \dots [x_n := t_n]$ とは異なることに注意

- ▶ 単純型付き λ 計算と System F の強正規化定理を Coq で証明

Theorem (強正規化定理)

t が型の付く項ならば、 t から始まる簡約列は有限長

- ▶ 項中の複数の変数を同時に置き換える代入を**並列代入**と呼ぶ:

$$\begin{aligned} & u[x_0, \dots, x_n := t_0, \dots, t_n] \\ = & u \text{ 中の変数 } x_0, \dots, x_n \text{ を } t_0, \dots, t_n \text{ で置き換えた項}^1 \end{aligned}$$

¹ $u[x_0 := t_0] \dots [x_n := t_n]$ とは異なることに注意

- ▶ 単純型付き λ 計算と System F の強正規化定理を Coq で証明

Theorem (強正規化定理)

t が型の付く項ならば、 t から始まる簡約列は有限長

- ▶ 項中の複数の変数を同時に置き換える代入を**並列代入**と呼ぶ:

$$\begin{aligned} & u[x_0, \dots, x_n := t_0, \dots, t_n] \\ &= u \text{ 中の変数 } x_0, \dots, x_n \text{ を } t_0, \dots, t_n \text{ で置き換えた項}^1 \end{aligned}$$

- ▶ 強正規化定理の証明には並列代入が必要

¹ $u[x_0 := t_0] \dots [x_n := t_n]$ とは異なることに注意

- ▶ 単純型付き λ 計算と System F の強正規化定理を Coq で証明

Theorem (強正規化定理)

t が型の付く項ならば、 t から始まる簡約列は有限長

- ▶ 項中の複数の変数を同時に置き換える代入を**並列代入**と呼ぶ:

$$\begin{aligned} & u[x_0, \dots, x_n := t_0, \dots, t_n] \\ &= u \text{ 中の変数 } x_0, \dots, x_n \text{ を } t_0, \dots, t_n \text{ で置き換えた項}^1 \end{aligned}$$

- ▶ 強正規化定理の証明には並列代入が必要
- ▶ 並列代入に関する基本的な性質をなるべく楽に証明したい

¹ $u[x_0 := t_0] \dots [x_n := t_n]$ とは異なることに注意

名前による表現の問題点

- ▶ 変数の「名前」によって束縛と被束縛を対応付ける項の定義を **名前による表現**と呼ぶ
- ▶ 名前による表現は形式的証明に向かない

名前による表現の問題点

- ▶ 変数の「名前」によって束縛と被束縛を対応付ける項の定義を**名前による表現**と呼ぶ
- ▶ 名前による表現は形式的証明に向かない

$$(\lambda y. y x)[x := y]$$

名前による表現の問題点

- ▶ 変数の「名前」によって束縛と被束縛を対応付ける項の定義を**名前による表現**と呼ぶ
- ▶ 名前による表現は形式的証明に向かない
- ▶ 束縛変数の名前の付け替えが必要

$$\begin{aligned}(\lambda y. y x)[x := y] &=_{\alpha} (\lambda z. z x)[x := y] \\ &= \lambda z. z x[x := y] \\ &= \lambda z. z y\end{aligned}$$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= x (\in \mathbb{N}) \mid (t t) \mid (\lambda t)$$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= x (\in \mathbb{N}) \mid (t t) \mid \underline{(\lambda t)}$$

束縛の位置には変数を書かない

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例: $\lambda x. \lambda y. (\lambda z. x y) x$

$\lambda \quad \lambda \quad (\lambda \quad)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例: $\lambda x. \lambda y. (\lambda z. x y) x$

$\lambda \quad \lambda \quad (\lambda \quad)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例: $\lambda x. \lambda y. (\lambda z. x y) x$

$\lambda \quad \lambda \quad (\lambda \quad \quad)$

De Bruijn 表現 [de Bruijn, 1972]

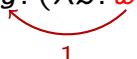
形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例:

$$\lambda x. \lambda y. (\lambda z. x y) x$$


$$\lambda \quad \lambda \quad (\lambda \quad)$$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

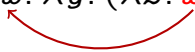
本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例:

$\lambda x. \lambda y. (\lambda z. x y) x$


 $\lambda \quad \lambda \quad (\lambda \quad 2 \quad)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例: $\lambda x. \lambda y. (\lambda z. x y) x$

$\lambda \quad \lambda \quad (\lambda \quad 2 \quad)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例: $\lambda x. \lambda y. (\lambda z. x y) x$

$\lambda \quad \lambda \quad (\lambda \quad 2 \quad)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例:

$\lambda x. \lambda y. (\lambda z. x y) x$

1

$\lambda \quad \lambda \quad (\lambda \quad 2 \quad 1)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例: $\lambda x. \lambda y. (\lambda z. x y) x$

$\lambda \quad \lambda \quad (\lambda \quad 2 \quad 1)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid (\lambda t)$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例:

$$\lambda x. \lambda y. (\lambda z. x y) x$$

$\lambda \quad \lambda \quad (\lambda \quad 2 \quad 1)$

De Bruijn 表現 [de Bruijn, 1972]

形式的証明に向けた λ 計算の定義

本研究では λ 項の定義として de Bruijn 表現を用いる:

$$t ::= \underline{x (\in \mathbb{N})} \mid (t t) \mid \underline{(\lambda t)}$$

いくつ外側の λ に対応するかを書く 束縛の位置には変数を書かない

例:

$$\begin{array}{ccccccc} \lambda x. \lambda y. (\lambda z. x y) & x \\ \uparrow & & \downarrow \\ & & 1 \\ \lambda & \lambda & (\lambda & 2 & 1) & 1 \end{array}$$

De Bruijn 表現での代入の定義


主に 2 つの定義方法が知られている:

定義	証明向き	実装向き
代入 (1)	○	
代入 (2)		○
並列代入 (1)		
並列代入 (2)	○	○

De Bruijn 表現での代入の定義

主に 2 つの定義方法が知られている:

定義	証明向き	実装向き
代入 (1)	○	
代入 (2)		○
並列代入 (1)		
並列代入 (2)	○	○

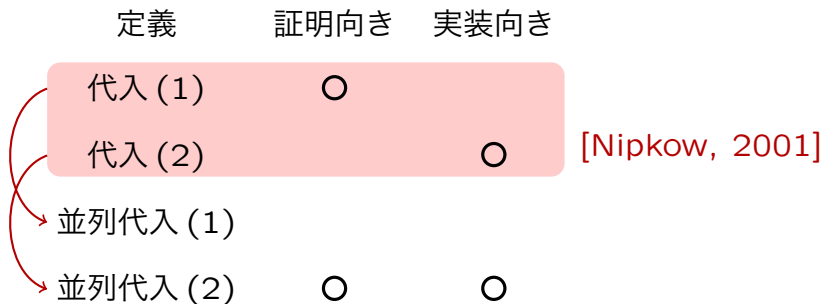


自然な拡張

De Bruijn 表現での代入の定義

主に 2 つの定義方法が知られている:

定義	証明向き	実装向き	
代入 (1)	○		[Nipkow, 2001]
代入 (2)		○	
並列代入 (1)			
並列代入 (2)	○	○	



自然な拡張

De Bruijn 表現での代入の定義

主に 2 つの定義方法が知られている:

定義	証明向き	実装向き	
代入 (1)	○		[Nipkow, 2001]
代入 (2)		○	
並列代入 (1)			本研究
並列代入 (2)	○	○	

自然な拡張

研究の概要

定義	証明向き	実装向き
代入 (1)	○	
代入 (2)		○
並列代入 (1)		
並列代入 (2)	○	○

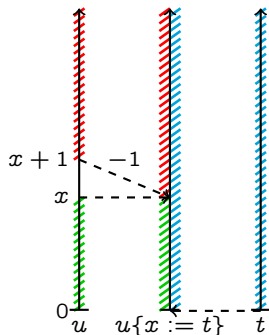
- ▶ 並列代入の場合には (2) の定義が証明向き
- ▶ 並列代入 (2) に関する代数的性質の (半) 自動証明
- ▶ 応用
 - ▶ Church-Rosser の定理
 - ▶ 強正規化定理

代入 (1) の定義

$$y\{x := t\} = \begin{cases} y - 1 & (x < y) \\ t & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)\{x := t\} = u\{x := t\} v\{x := t\}$$

$$(\lambda u)\{x := t\} = \lambda u\{x + 1 := t\uparrow^1\}$$

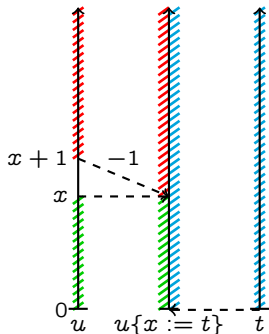


代入 (1) の定義

$$y\{x := t\} = \begin{cases} y - 1 & (x < y) \\ t & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)\{x := t\} = u\{x := t\} v\{x := t\}$$

$$(\lambda u)\{x := t\} = \lambda u\{x + 1 := t\}$$



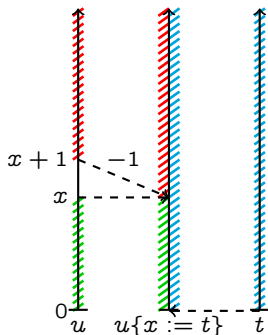
- ▶ λu の自由変数は u の自由変数として見ると 1 つ大きい

代入 (1) の定義

$$y\{x := t\} = \begin{cases} y - 1 & (x < y) \\ t & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)\{x := t\} = u\{x := t\} v\{x := t\}$$

$$(\lambda u)\{x := t\} = \lambda u\{x + 1 := t^1\}$$



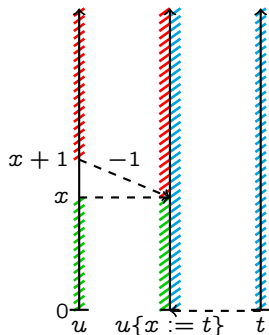
- ▶ λu の自由変数は u の自由変数として見ると 1 つ大きい
 - ▶ 代入位置 x を 1 増やす

代入 (1) の定義

$$y\{x := t\} = \begin{cases} y - 1 & (x < y) \\ t & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)\{x := t\} = u\{x := t\} v\{x := t\}$$

$$(\lambda u)\{x := t\} = \lambda u\{x + 1 := t^{\uparrow 1}\}$$



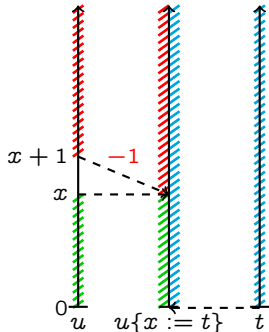
- ▶ λu の自由変数は u の自由変数として見ると 1 つ大きい
 - ▶ 代入位置 x を 1 増やす
 - ▶ $t^{\uparrow d}$ は t 中の全ての自由変数に d を足す項 (シフト)

代入 (1) の定義

$$y\{x := t\} = \begin{cases} y - 1 & (x < y) \\ t & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)\{x := t\} = u\{x := t\} v\{x := t\}$$

$$(\lambda u)\{x := t\} = \lambda u\{x + 1 := t^{\uparrow 1}\}$$



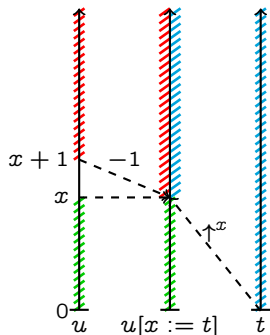
- ▶ λu の自由変数は u の自由変数として見ると 1 つ大きい
 - ▶ 代入位置 x を 1 増やす
 - ▶ $t^{\uparrow d}$ は t 中の全ての自由変数に d を足す項 (シフト)
- ▶ 項 $u\{x := t\}$ に元々 u にあった自由変数 x は残らない
 - ▶ x より大きい自由変数 y から 1 を引く

代入 (2) の定義

$$y[x := t] = \begin{cases} y - 1 & (x < y) \\ t \uparrow^x & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)[x := t] = u[x := t] v[x := t]$$

$$(\lambda u)[x := t] = \lambda u[x + 1 := t]$$

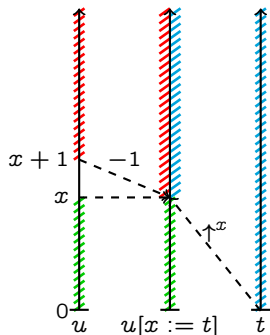


代入 (2) の定義

$$y[x := t] = \begin{cases} y - 1 & (x < y) \\ t \uparrow^x & (x = y) \\ y & (x > y) \end{cases}$$

$$(u v)[x := t] = u[x := t] v[x := t]$$

$$(\lambda u)[x := t] = \lambda u[x + 1 := t]$$



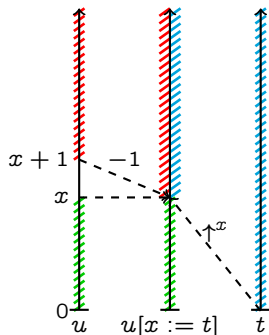
- ▶ (1) の定義では代入位置と t に対するシフトの量が同じだけ増える

代入 (2) の定義

$$y[x := t] = \begin{cases} y - 1 & (x < y) \\ t \uparrow^x & (x = y) \\ y & (x > y) \end{cases}$$

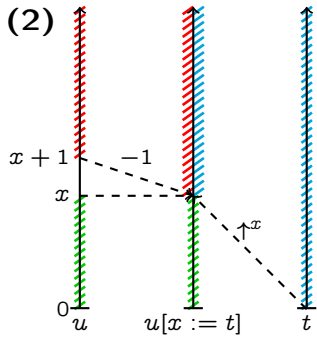
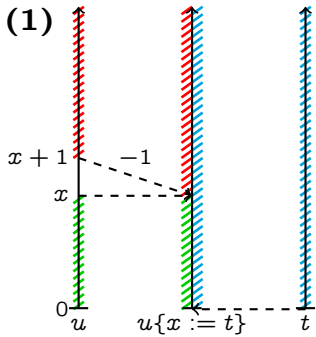
$$(u v)[x := t] = u[x := t] v[x := t]$$

$$(\lambda u)[x := t] = \lambda u[x + 1 := t]$$

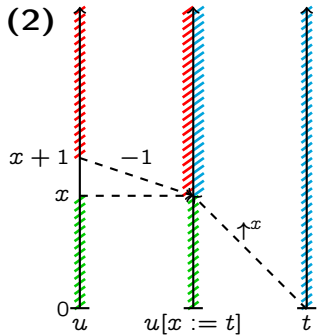
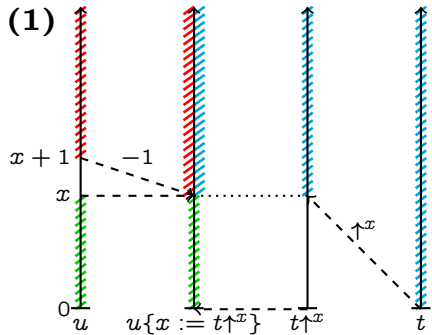


- ▶ (1) の定義では代入位置と t に対するシフトの量が同じだけ増える
- ▶ 変数を置き換える時点で代入位置の分だけシフトしても同じ
 - ▶ $t \underbrace{\uparrow^1 \dots \uparrow^1}_d = t \uparrow^d$

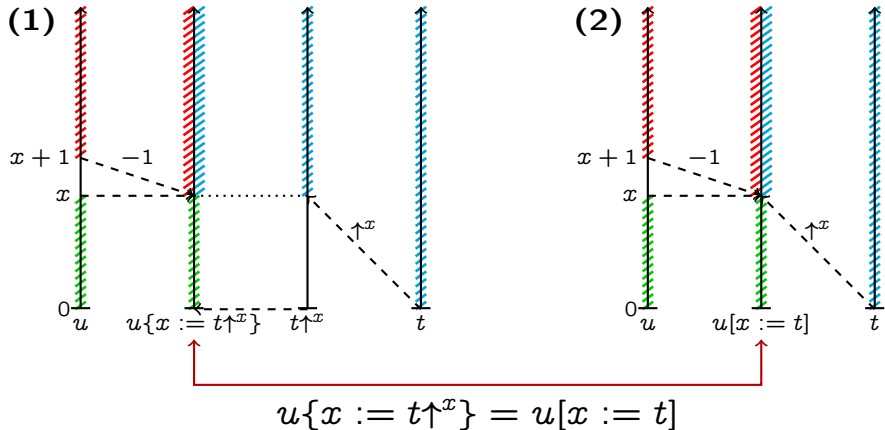
2つの代入の関係



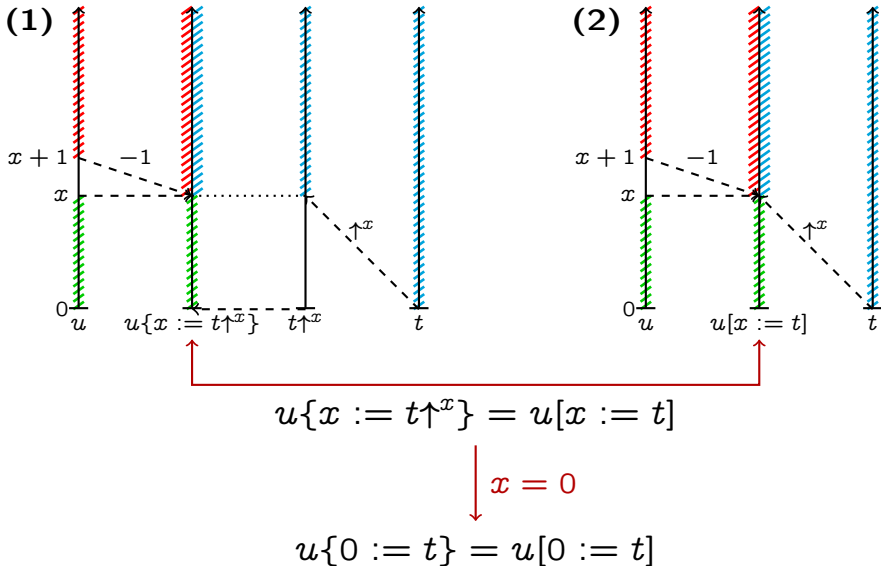
2つの代入の関係



2つの代入の関係



2つの代入の関係



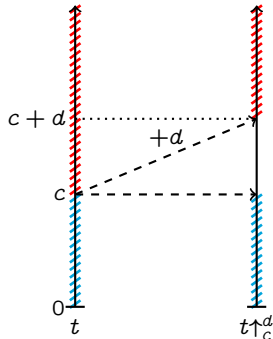
シフトの定義

$$x \uparrow_c^d = \begin{cases} x + d & (c \leq x) \\ x & (x < c) \end{cases}$$

$$(t u) \uparrow_c^d = t \uparrow_c^d u \uparrow_c^d$$

$$(\lambda t) \uparrow_c^d = \lambda t \uparrow_{c+1}^d$$

$$t \uparrow^d = t \uparrow_0^d$$




シフトの定義

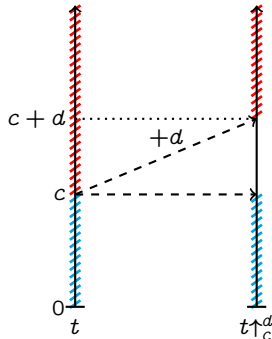
$$x \uparrow_c^d = \begin{cases} x + d & (c \leq x) \\ x & (x < c) \end{cases}$$

$$(tu) \uparrow_c^d = t \uparrow_c^d u \uparrow_c^d$$

$$(\lambda t) \uparrow_c^d = \lambda t \uparrow_{c+1}^d$$

$$\underline{t \uparrow^d} = t \uparrow_0^d$$


t 中の全ての自由変数に d を足した項



シフトの定義

$$x \uparrow_c^d = \begin{cases} x + d & (c \leq x) \\ x & (x < c) \end{cases}$$

$$(tu) \uparrow_c^d = t \uparrow_c^d u \uparrow_c^d$$

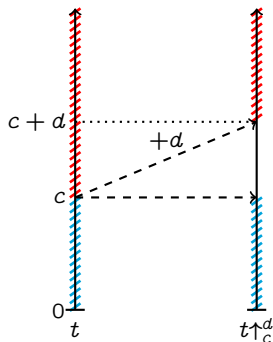
$$(\lambda t) \uparrow_c^d = \lambda t \uparrow_{c+1}^d$$

$$\underline{t \uparrow_c^d} = t \uparrow_0^d$$

t 中の **全て** の自由変数に d を足した項

↓ 一般化

$t \uparrow_c^d$: t 中の c 以上の自由変数に d を足した項



シフトの定義

$$x \uparrow_c^d = \begin{cases} x + d & (c \leq x) \\ x & (x < c) \end{cases}$$

$$(tu) \uparrow_c^d = t \uparrow_c^d u \uparrow_c^d$$

$$(\lambda t) \uparrow_c^d = \lambda t \uparrow_{c+1}^d$$

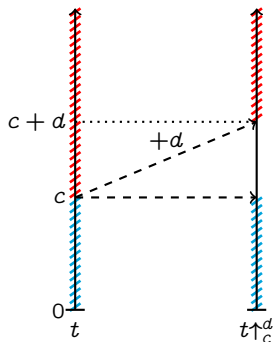
$$\underline{t \uparrow^d} = t \uparrow_0^d$$

t 中の **全て** の自由変数に d を足した項

↓ 一般化

↑ $c = 0$

$t \uparrow_c^d$: t 中の c 以上の自由変数に d を足した項



シフトの定義

$$x \uparrow_c^d = \begin{cases} x + d & (c \leq x) \\ x & (x < c) \end{cases}$$

$$(tu) \uparrow_c^d = t \uparrow_c^d u \uparrow_c^d$$

$$(\lambda t) \uparrow_c^d = \lambda t \uparrow_{c+1}^d$$

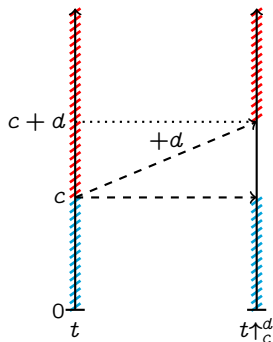
$$\underline{t \uparrow^d} = t \uparrow_0^d$$

t 中の全ての自由変数に d を足した項

一般化

$c = 0$

$t \uparrow_c^d$: t 中の c 以上の自由変数に d を足した項



シフトの定義

$$x \uparrow_c^d = \begin{cases} x + d & (c \leq x) \\ x & (x < c) \end{cases}$$

$$(tu) \uparrow_c^d = t \uparrow_c^d u \uparrow_c^d$$

$$(\lambda t) \uparrow_c^d = \lambda t \uparrow_{c+1}^d$$

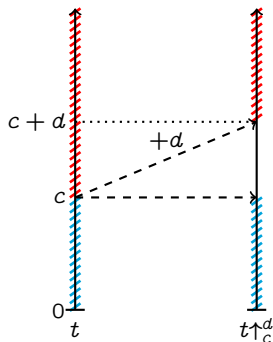
$$\underline{t \uparrow^d} = t \uparrow_0^d$$

t 中の全ての自由変数に d を足した項

一般化

$c = 0$

$t \uparrow_c^d$: t 中の c 以上の自由変数に d を足した項



(1) の定義:

(1) の定義:

- ▶ Church-Rosser の定理を示すには以下の補題があれば十分

$$c \leq c' \Rightarrow t \uparrow_c^1 \uparrow_{c'+1}^1 = t \uparrow_{c'}^1 \uparrow_c^1$$

$$x \leq c \Rightarrow t\{x := u\} \uparrow_c^1 = t \uparrow_{c+1}^1 \{x := u \uparrow_c^1\}$$

$$c \leq x \Rightarrow t\{x := u\} \uparrow_c^1 = t \uparrow_c^1 \{x + 1 := u \uparrow_c^1\}$$

$$t \uparrow_x^1 \{x := u\} = t$$

$$x \leq y \Rightarrow t\{y + 1 := v \uparrow_x^1\} \{x := u\} \{y := v\} = t\{x := u\} \{y := v\}$$

(1) の定義:

- ▶ Church-Rosser の定理を示すには以下の補題があれば十分
- ▶ 証明全体を通してシフトの右上の数は 1 以外に出現しない
- ▶ 証明が簡潔 (多くの補題の形式的証明は 1 行で済んでいる)

$$c \leq c' \Rightarrow t \uparrow_c^1 \uparrow_{c'+1}^1 = t \uparrow_{c'}^1 \uparrow_c^1$$

$$x \leq c \Rightarrow t\{x := u\} \uparrow_c^1 = t \uparrow_{c+1}^1 \{x := u \uparrow_c^1\}$$

$$c \leq x \Rightarrow t\{x := u\} \uparrow_c^1 = t \uparrow_c^1 \{x + 1 := u \uparrow_c^1\}$$

$$t \uparrow_x^1 \{x := u\} = t$$

$$x \leq y \Rightarrow t\{y + 1 := v \uparrow_x^1\} \{x := u\} \{y := v\} = t\{x := u\} \{y := v\}$$

シフトと代入の代数的性質 [Nipkow, 2001]

(1) の定義:

- ▶ Church-Rosser の定理を示すには以下の補題があれば十分
- ▶ 証明全体を通してシフトの右上の数は 1 以外に出現しない
- ▶ 証明が簡潔 (多くの補題の形式的証明は 1 行で済んでいる)

$$c \leq c' \Rightarrow t \uparrow_c^1 \uparrow_{c'+1}^1 = t \uparrow_{c'}^1 \uparrow_c^1$$

$$x \leq c \Rightarrow t\{x := u\} \uparrow_c^1 = t \uparrow_{c+1}^1 \{x := u \uparrow_c^1\}$$

$$c \leq x \Rightarrow t\{x := u\} \uparrow_c^1 = t \uparrow_c^1 \{x + 1 := u \uparrow_c^1\}$$

$$t \uparrow_x^1 \{x := u\} = t$$

$$x \leq y \Rightarrow t\{y + 1 := v \uparrow_x^1\} \{x := u\} \{y := v\} = t\{x := u\} \{y := v\}$$

(2) の定義:

- ▶ 直接証明に使うのには向かない
- ▶ $u[x := t] = u\{x := t \uparrow^x\}$ で書き換えて証明すると良い

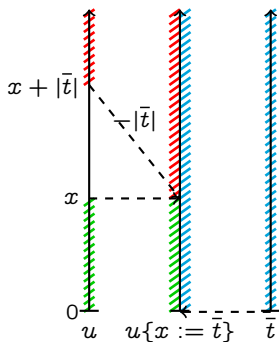
並列代入 (1)

$$y\{x := \bar{t}\} = \begin{cases} y - |\bar{t}| & (x + |\bar{t}| \leq y) \\ \bar{t}_{y-x} & (x \leq y < x + |\bar{t}|) \\ y & (y < x) \end{cases}$$

$$(u v)\{x := \bar{t}\} = (u\{x := \bar{t}\}) (v\{x := \bar{t}\})$$

$$(\lambda u)\{x := \bar{t}\} = \lambda (u\{x + 1 := \bar{t}\uparrow^1\})$$

$$\bar{t}\uparrow_c^d = [t\uparrow_c^d \mid t \leftarrow \bar{t}]$$



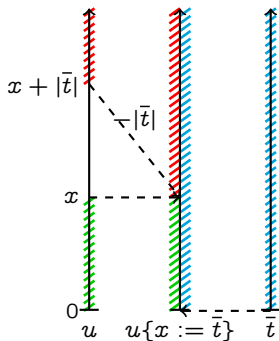
並列代入 (1)

$$y\{x := \bar{t}\} = \begin{cases} y - |\bar{t}| & (x + |\bar{t}| \leq y) \\ \bar{t}_{y-x} & (x \leq y < x + |\bar{t}|) \\ y & (y < x) \end{cases}$$

$$(u v)\{x := \bar{t}\} = (u\{x := \bar{t}\}) (v\{x := \bar{t}\})$$

$$(\lambda u)\{x := \bar{t}\} = \lambda (u\{x + 1 := \bar{t}^{\uparrow 1}\})$$

$$\bar{t}^{\uparrow c} = [t^{\uparrow c} \mid t \leftarrow \bar{t}]$$



$$\blacktriangleright u\{x := \bar{t}\} \approx u[x, \dots, x + |\bar{t}| - 1 := \bar{t}_0, \dots, \bar{t}_{|\bar{t}|-1}]$$

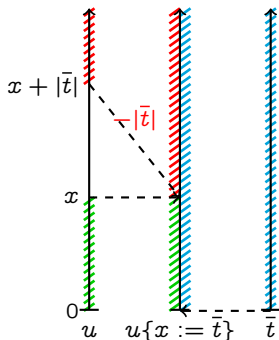
並列代入 (1)

$$y\{x := \bar{t}\} = \begin{cases} y - |\bar{t}| & (x + |\bar{t}| \leq y) \\ \bar{t}_{y-x} & (x \leq y < x + |\bar{t}|) \\ y & (y < x) \end{cases}$$

$$(uv)\{x := \bar{t}\} = (u\{x := \bar{t}\})(v\{x := \bar{t}\})$$

$$(\lambda u)\{x := \bar{t}\} = \lambda(u\{x + 1 := \bar{t}^{\uparrow 1}\})$$

$$\bar{t}^{\uparrow c} = [t^{\uparrow c} \mid t \leftarrow \bar{t}]$$



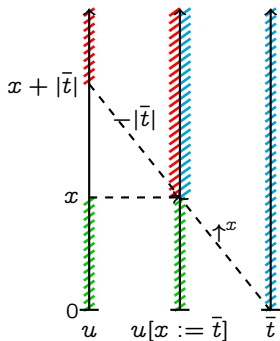
- ▶ $u\{x := \bar{t}\} \approx u[x, \dots, x + |\bar{t}| - 1 := \bar{t}_0, \dots, \bar{t}_{|\bar{t}|-1}]$
- ▶ 代入の範囲が変数 $|\bar{t}|$ 個分なので -1 ではなく $-|\bar{t}|$

並列代入 (2)

$$y[x := \bar{t}] = \begin{cases} y - |\bar{t}| & (x + |\bar{t}| \leq y) \\ \bar{t}_{y-x} \uparrow^x & (x \leq y < x + |\bar{t}|) \\ y & (y < x) \end{cases}$$

$$(u v)[x := \bar{t}] = (u[x := \bar{t}]) (v[x := \bar{t}])$$

$$(\lambda u)[x := \bar{t}] = \lambda (u[x + 1 := \bar{t}])$$

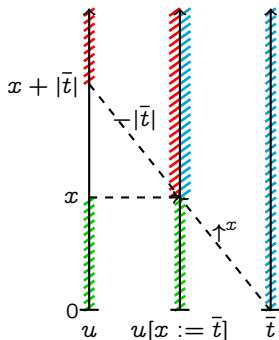


並列代入 (2)

$$y[x := \bar{t}] = \begin{cases} y - |\bar{t}| & (x + |\bar{t}| \leq y) \\ \bar{t}_{y-x} \uparrow^x & (x \leq y < x + |\bar{t}|) \\ y & (y < x) \end{cases}$$

$$(u v)[x := \bar{t}] = (u[x := \bar{t}]) (v[x := \bar{t}])$$

$$(\lambda u)[x := \bar{t}] = \lambda (u[x + 1 := \bar{t}])$$



Coq 上で使う定義:

$$y[x := \bar{t}] = \begin{cases} \text{nth}(y - x - |\bar{t}|, \bar{t}, y - x) \uparrow^x & (x \leq y) \\ y & (y < x) \end{cases}$$

$$(u v)[x := \bar{t}] = (u[x := \bar{t}]) (v[x := \bar{t}])$$

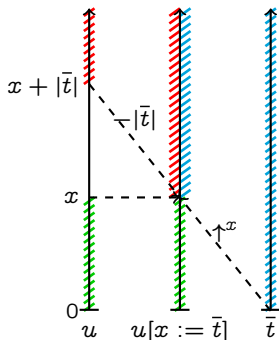
$$(\lambda u)[x := \bar{t}] = \lambda (u[x + 1 := \bar{t}])$$

並列代入 (2)

$$y[x := \bar{t}] = \begin{cases} y - |\bar{t}| & (x + |\bar{t}| \leq y) \\ \bar{t}_{y-x} \uparrow^x & (x \leq y < x + |\bar{t}|) \\ y & (y < x) \end{cases}$$

$$(u v)[x := \bar{t}] = (u[x := \bar{t}]) (v[x := \bar{t}])$$

$$(\lambda u)[x := \bar{t}] = \lambda (u[x + 1 := \bar{t}])$$



Coq 上で使う定義:

$$y[x := \bar{t}] = \begin{cases} \text{nth}(y - x - |\bar{t}|, \bar{t}, y - x) \uparrow^x & (x \leq y) \\ y & (y < x) \end{cases}$$

$$(u v)[x := \bar{t}] = (u[x := \bar{t}]) (v[x := \bar{t}])$$

$$(\lambda u)[x := \bar{t}] = \lambda (u[x + 1 := \bar{t}])$$

並列代入 (1) の代入補題

$$\begin{aligned}x \leq y &\Rightarrow t\{y + |\bar{u}| := \bar{v} \uparrow_x^{|\bar{u}|}\}\{x := \bar{u}\{y := \bar{v}\}\} \\ &= t\{x := \bar{u}\}\{y := \bar{v}\}\end{aligned}$$

並列代入 (1) の代入補題

$$\begin{aligned}x \leq y &\Rightarrow t\{y + |\bar{u}| := \bar{v} \uparrow_x^{|\bar{u}|}\}\{x := \bar{u}\{y := \bar{v}\}\} \\ &= t\{x := \bar{u}\}\{y := \bar{v}\}\end{aligned}$$

- ▶ $d = 1$ の形に限らないシフトが必要

並列代入 (1) の代入補題

$$\begin{aligned}x \leq y &\Rightarrow t\{y + |\bar{u}| := \bar{v} \uparrow_x^{|\bar{u}|}\}\{x := \bar{u}\{y := \bar{v}\}\} \\ &= t\{x := \bar{u}\}\{y := \bar{v}\}\end{aligned}$$

- ▶ $d = 1$ の形に限らないシフトが必要
- ▶ 代入補題の証明に必要な各種補題を一般化する必要がある

並列代入 (1) の代入補題

$$\begin{aligned}x \leq y &\Rightarrow t\{y + |\bar{u}| := \bar{v} \uparrow_x^{|\bar{u}|}\}\{x := \bar{u}\{y := \bar{v}\}\} \\ &= t\{x := \bar{u}\}\{y := \bar{v}\}\end{aligned}$$

- ▶ $d = 1$ の形に限らないシフトが必要
- ▶ 代入補題の証明に必要な各種補題を一般化する必要がある
- ▶ Nipkow の方法のように簡単な証明にならない

シフトと並列代入 (2) に関する代数的性質

$$t \uparrow_n^0 = t \quad (1)$$

$$c \leq c' \leq c + d \Rightarrow t \uparrow_c^d \uparrow_{c'}^{d'} = t \uparrow_c^{d'+d} \quad (2)$$

$$c' \leq c \Rightarrow t \uparrow_c^d \uparrow_{c'}^{d'} = t \uparrow_{c'}^{d'} \uparrow_{d'+c}^d \quad (3)$$

$$c \leq n \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_c^d [d + n := \bar{u}] \quad (4)$$

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d] \quad (5)$$

$$c \leq n \wedge |\bar{u}| + n \leq d + c \Rightarrow$$

$$t \uparrow_c^d [n := \bar{u}] = t \uparrow_c^{d-|\bar{u}|} \quad (6)$$

$$m \leq n \Rightarrow t[m := \bar{u}][n := \bar{v}] = t[|\bar{u}| + n := \bar{v}]$$

$$[m := \bar{u}[n - m := \bar{v}]] \quad (7)$$

$$t[|\bar{v}| + n := \bar{u}][n := \bar{v}] = t[n := \bar{v} \uparrow \bar{u}] \quad (8)$$

$$t[n := []] = t \quad (9)$$

where

$$\bar{t} \uparrow_c^d = [t \uparrow_c^d \mid t \leftarrow \bar{t}]$$

$$\bar{t}[n := \bar{u}] = [t[n := \bar{u}] \mid t \leftarrow \bar{t}]$$

シフトと並列代入 (2) に関する代数的性質

$$t \uparrow_n^0 = t \quad (1)$$

$$c \leq c' \leq c + d \Rightarrow t \uparrow_c^d \uparrow_{c'}^{d'} = t \uparrow_c^{d'+d} \quad (2)$$

$$c' \leq c \Rightarrow t \uparrow_c^d \uparrow_{c'}^{d'} = t \uparrow_{c'}^{d'} \uparrow_{d'+c}^d \quad (3)$$

$$c \leq n \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_c^d [d + n := \bar{u}] \quad (4)$$

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d] \quad (5)$$

$$c \leq n \wedge |\bar{u}| + n \leq d + c \Rightarrow$$

$$t \uparrow_c^d [n := \bar{u}] = t \uparrow_c^{d-|\bar{u}|} \quad (6)$$

$$m \leq n \Rightarrow t[m := \bar{u}][n := \bar{v}] = t[|\bar{u}| + n := \bar{v}]$$

$$[m := \bar{u}[n - m := \bar{v}]] \quad (7)$$

$$t[|\bar{v}| + n := \bar{u}][n := \bar{v}] = t[n := \bar{v} \uparrow \bar{u}] \quad (8)$$

$$t[n := []] = t \quad (9)$$

where

$$\bar{t} \uparrow_c^d = [t \uparrow_c^d \mid t \leftarrow \bar{t}]$$

$$\bar{t}[n := \bar{u}] = [t[n := \bar{u}] \mid t \leftarrow \bar{t}]$$

補題 (5)

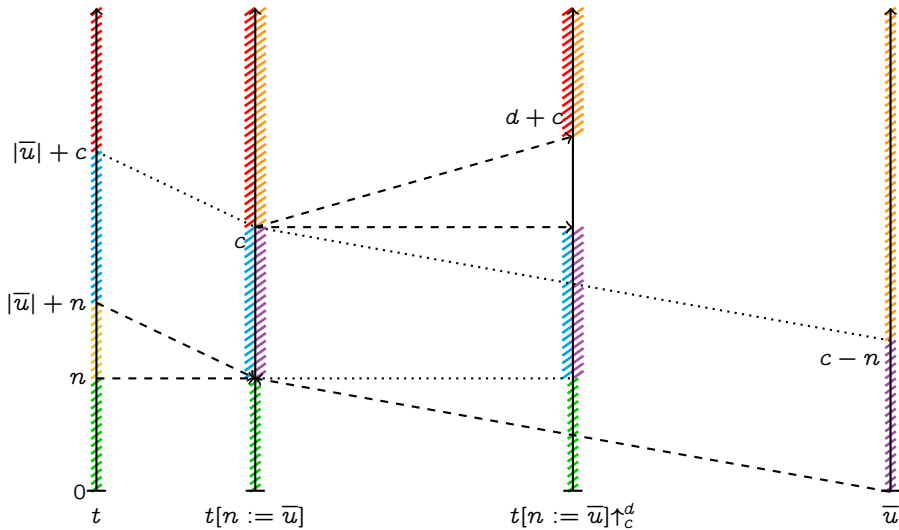
$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

補題 (5)

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

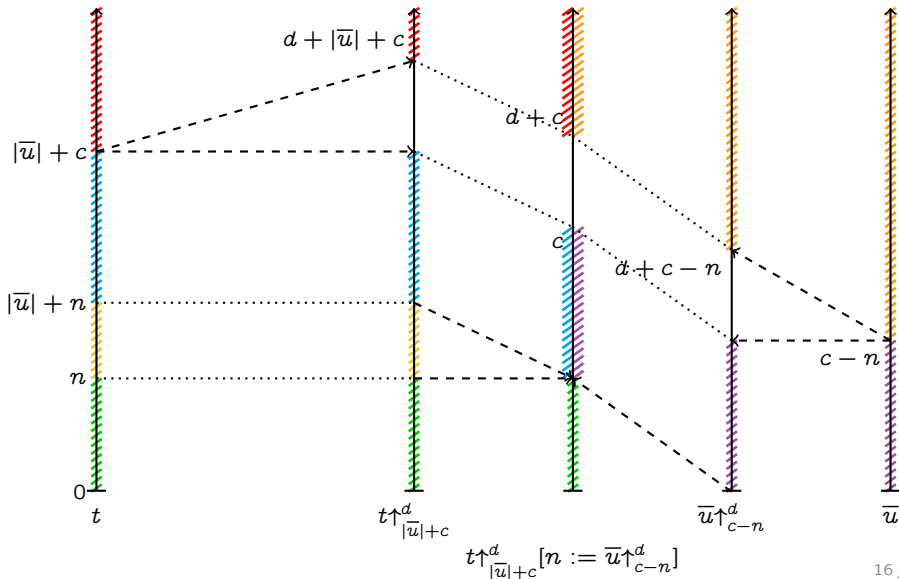
補題 (5)

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$



補題 (5)

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$



補題 (5)

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$



証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{n+c'-n}^d]$$

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{n+c'-n}^d]$$

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{n+c'-n}^d]$$

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{c'}^d]$$

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{c'}^d]$$

項 t についての帰納法

1. 変数の場合
2. 関数適用の場合
3. λ 抽象の場合

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{c'}^d]$$

項 t についての帰納法

1. 変数の場合

2. 関数適用の場合

3. λ 抽象の場合

} 等式のソルバで証明

証明の流れ

$$n \leq c \Rightarrow t[n := \bar{u}] \uparrow_c^d = t \uparrow_{|\bar{u}|+c}^d [n := \bar{u} \uparrow_{c-n}^d]$$

仮定の除去

$$\forall m. n \leq m \Rightarrow P[n, m]$$

$$\Leftrightarrow \forall m'. P[n, n + m']$$

$$t[n := \bar{u}] \uparrow_{n+c'}^d = t \uparrow_{|\bar{u}|+(n+c')}^d [n := \bar{u} \uparrow_{c'}^d]$$

項 t についての帰納法

1. 変数の場合
 2. 関数適用の場合
 3. λ 抽象の場合
- 場合分けを展開して証明
- 等式のソルバで証明

2. 関数適用の場合

$$\begin{aligned} t[n := \bar{u}] \uparrow_{n+c}^d u[n := \bar{u}] \uparrow_{n+c}^d &= \\ t \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d] u \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d] \end{aligned}$$

3. λ抽象の場合

$$\lambda t[n + 1 := \bar{u}] \uparrow_{n+c+1}^d = \lambda t \uparrow_{|\bar{u}|+(n+c)+1}^d [n + 1 := \bar{u} \uparrow_c^d]$$

2. 関数適用の場合

$$\begin{aligned} t[n := \bar{u}] \uparrow_{n+c}^d u[n := \bar{u}] \uparrow_{n+c}^d &= \\ t \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d] u \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d] \end{aligned}$$

3. λ抽象の場合

$$\lambda t[n + 1 := \bar{u}] \uparrow_{n+c+1}^d = \lambda t \uparrow_{|\bar{u}|+(n+c)+1}^d [n + 1 := \bar{u} \uparrow_c^d]$$

- ▶ **+1** の付け替えと帰納法の仮定での書き換えだけで示せる

証明 - 関数適用とλ抽象の場合

2. 関数適用の場合

$$\begin{aligned} t[n := \bar{u}] \uparrow_{n+c}^d u[n := \bar{u}] \uparrow_{n+c}^d &= \\ t \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d] u \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d] \end{aligned}$$

3. λ抽象の場合

$$\lambda t[n + 1 := \bar{u}] \uparrow_{n+c+1}^d = \lambda t \uparrow_{|\bar{u}|+(n+c)+1}^d [n + 1 := \bar{u} \uparrow_c^d]$$

- ▶ +1 の付け替えと帰納法の仮定での書き換えだけで示せる
- ▶ 以下の仮定を追加して Coq の等式に関するソルバ **congruence** を用いて証明

$$\text{addSn} : \forall m n. (m + 1) + n = (m + n) + 1$$

$$\text{addnS} : \forall m n. m + (n + 1) = (m + n) + 1$$

- ▶ Congruence closure アルゴリズムに基く決定手続き
 - ▶ $x_1 = x'_1 \wedge \dots \wedge x_n = x'_n \Rightarrow y = y'$
 - ▶ 証明に使うのは等式での書き換えのみ
 - ▶ 定義を展開しないと証明できない問題は解けない

- ▶ Congruence closure アルゴリズムに基づく決定手続き
 - ▶ $x_1 = x'_1 \wedge \dots \wedge x_n = x'_n \Rightarrow y = y'$
 - ▶ 証明に使うのは等式での書き換えのみ
 - ▶ 定義を展開しないと証明できない問題は解けない
- ▶ 仮定が量化された等式であってもある程度扱える

- ▶ Congruence closure アルゴリズムに基づく決定手続き
 - ▶ $x_1 = x'_1 \wedge \cdots \wedge x_n = x'_n \Rightarrow y = y'$
 - ▶ 証明に使うのは等式での書き換えのみ
 - ▶ 定義を展開しないと証明できない問題は解けない
- ▶ 仮定が量化された等式であってもある程度扱える
- ▶ 応用
 - ▶ 定義を量化された等式の形で仮定に追加すると定義の展開が可能
 - ▶ 帰納法と congruence だけで自然数の加算や乗算の可換性と結合性が示せる

証明 - 変数の場合

$$1. x[n := \bar{u}] \uparrow_{n+c}^d = x \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d]$$

- ▶ 場合分けを全て展開してプレスバーガー算術に帰着させて解く
- ▶ 自動証明で解けない部分は手で証明する

証明 - 変数の場合

$$1. x[n := \bar{u}] \uparrow_{n+c}^d = x \uparrow_{|\bar{u}|+(n+c)}^d [n := \bar{u} \uparrow_c^d]$$

- ▶ 場合分けを全て展開してプレスバーガー算術に帰着させて解く
- ▶ 自動証明で解けない部分は手で証明する

場合分けの結果として得られる命題:

- ▶ $|\bar{t}| + (n + c) \leq x \wedge n \leq x + d \wedge n \leq x \Rightarrow \text{nth}(x - n - |\bar{t}|, \bar{t}, x - n) \uparrow_{n+c}^d = \text{nth}(x + d - n - |\bar{t} \uparrow_c^d|, \bar{t} \uparrow_c^d, x + d - n) \uparrow^n$
- ▶ $x < |\bar{t}| + (n + c) \wedge n \leq x \wedge n \leq x \Rightarrow \text{nth}(x - n - |\bar{t}|, \bar{t}, x - n) \uparrow_{n+c}^d = \text{nth}(x - n - |\bar{t} \uparrow_c^d|, \bar{t} \uparrow_c^d, x - n) \uparrow^n$
- ▶ $|\bar{t}| + (n + c) \leq x \wedge x + d < n \wedge n \leq x \Rightarrow \text{nth}(x - n - |\bar{t}|, \bar{t}, x - n) \uparrow_{n+c}^d = x + d$
- ▶ $x < |\bar{t}| + (n + c) \wedge x < n \wedge n \leq x \Rightarrow \text{nth}(x - n - |\bar{t}|, \bar{t}, x - n) \uparrow_{n+c}^d = x$
- ▶ ...

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け
- ▶ 減算を多く含む問題に対して非常に長い時間がかかる

Notation $\text{minn } x \ y := (x - (x - y))$.

Lemma $\text{minnC } x \ y : \text{minn } x \ (y - 0) = \text{minn } y \ (x - 0)$.

Proof. Time lia.

Restart. Time omega.

Qed.

Lemma $\text{minnA } x \ y \ z : \text{minn } (x - 0) \ (\text{minn } (y - 0) \ z) =$
 $\text{minn } (\text{minn } (x - 0) \ (y - 0)) \ z$.

Proof. Time lia.

Restart. Time omega.

Qed.

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け
- ▶ 減算を多く含む問題に対して非常に長い時間がかかる

Notation $\text{minn } x \ y := (x - (x - y))$.

Lemma $\text{minnC } x \ y : \text{minn } x \ (y - 0) = \text{minn } y \ (x - 0)$.

Proof. Time lia. $\rightarrow 0$ 秒

Restart. Time omega.

Qed.

Lemma $\text{minnA } x \ y \ z : \text{minn } (x - 0) \ (\text{minn } (y - 0) \ z) =$
 $\text{minn } (\text{minn } (x - 0) \ (y - 0)) \ z$.

Proof. Time lia.

Restart. Time omega.

Qed.

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け
- ▶ 減算を多く含む問題に対して非常に長い時間がかかる

Notation $\text{minn } x \ y := (x - (x - y))$.

Lemma $\text{minnC } x \ y : \text{minn } x \ (y - 0) = \text{minn } y \ (x - 0)$.

Proof. Time lia. → 0 秒

Restart. Time omega. → 13 秒

Qed.

Lemma $\text{minnA } x \ y \ z : \text{minn } (x - 0) \ (\text{minn } (y - 0) \ z) =$
 $\text{minn } (\text{minn } (x - 0) \ (y - 0)) \ z$.

Proof. Time lia.

Restart. Time omega.

Qed.

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け
- ▶ 減算を多く含む問題に対して非常に長い時間がかかる

Notation $\text{minn } x \ y := (x - (x - y))$.

Lemma $\text{minnC } x \ y : \text{minn } x \ (y - 0) = \text{minn } y \ (x - 0)$.

Proof. Time lia. → 0 秒

Restart. Time omega. → 13 秒

Qed.

Lemma $\text{minnA } x \ y \ z : \text{minn } (x - 0) \ (\text{minn } (y - 0) \ z) =$
 $\text{minn } (\text{minn } (x - 0) \ (y - 0)) \ z$.

Proof. Time lia. → 5 秒

Restart. Time omega.

Qed.

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け
- ▶ 減算を多く含む問題に対して非常に長い時間がかかる

Notation $\text{minn } x \ y := (x - (x - y))$.

Lemma $\text{minnC } x \ y : \text{minn } x \ (y - 0) = \text{minn } y \ (x - 0)$.

Proof. Time lia. → 0 秒

Restart. Time omega. → 13 秒

Qed.

Lemma $\text{minnA } x \ y \ z : \text{minn } (x - 0) \ (\text{minn } (y - 0) \ z) =$
 $\text{minn } (\text{minn } (x - 0) \ (y - 0)) \ z$.

Proof. Time lia. → 5 秒

Restart. Time omega. → timeout

Qed.

omega タクティクと lia タクティク

- ▶ 量化を含まない範囲のプレスバーガー算術のソルバ
- ▶ 自然数の減算 $x \dot{-} y$ については $x \leq y$ と $y \leq x$ で場合分け
- ▶ 減算を多く含む問題に対して非常に長い時間がかかる

Notation $\text{minn } x \ y := (x - (x - y))$.

Lemma $\text{minnC } x \ y : \text{minn } x \ (y - 0) = \text{minn } y \ (x - 0)$.

Proof. Time lia. → 0 秒

Restart. Time omega. → 13 秒

Qed.

Lemma $\text{minnA } x \ y \ z : \text{minn } (x - 0) \ (\text{minn } (y - 0) \ z) =$
 $\text{minn } (\text{minn } (x - 0) \ (y - 0)) \ z$.

Proof. Time lia. → 5 秒

Restart. Time omega. → timeout

Qed.

- ▶ 論理式中の減算の数を減らすことが重要

1. 仮定の除去と算術式を簡単にする操作を交互に適用

場合分けの後の命題に対する自動証明

1. 仮定の除去と算術式を簡単にする操作を交互に適用
 - ▶ 算術式を簡単にする操作: 減算や不等号の両辺に共通する部分式を除去

場合分けの後の命題に対する自動証明

1. 仮定の除去と算術式を簡単にする操作を交互に適用
 - ▶ 算術式を簡単にする操作: 減算や不等号の両辺に共通する部分式を除去
2. ゴールが算術式の等式になるまで `f_equal` タクティクを適用

$$f(x_1, \dots, x_n) = f(y_1, \dots, y_n)$$
$$\hookrightarrow x_1 = y_1, \dots, x_n = y_n$$

場合分けの後の命題に対する自動証明

1. 仮定の除去と算術式を簡単にする操作を交互に適用
 - ▶ 算術式を簡単にする操作: 減算や不等号の両辺に共通する部分式を除去
2. ゴールが算術式の等式になるまで `f_equal` タクティクを適用

$$f(x_1, \dots, x_n) = f(y_1, \dots, y_n)$$
$$\hookrightarrow x_1 = y_1, \dots, x_n = y_n$$

3. $n \div (n \div m)$ と $n + (m \div n)$ の形の式について場合分け

場合分けの後の命題に対する自動証明

1. 仮定の除去と算術式を簡単にする操作を交互に適用
 - ▶ 算術式を簡単にする操作: 減算や不等号の両辺に共通する部分式を除去
2. ゴールが算術式の等式になるまで `f_equal` タクティクを適用

$$f(x_1, \dots, x_n) = f(y_1, \dots, y_n)$$
$$\hookrightarrow x_1 = y_1, \dots, x_n = y_n$$

3. $n \div (n \div m)$ と $n + (m \div n)$ の形の式について場合分け
4. 仮定の除去

場合分けの後の命題に対する自動証明

1. 仮定の除去と算術式を簡単にする操作を交互に適用
 - ▶ 算術式を簡単にする操作: 減算や不等号の両辺に共通する部分式を除去
2. ゴールが算術式の等式になるまで `f_equal` タクティクを適用

$$f(x_1, \dots, x_n) = f(y_1, \dots, y_n)$$
$$\hookrightarrow x_1 = y_1, \dots, x_n = y_n$$

3. $n \div (n \div m)$ と $n + (m \div n)$ の形の式について場合分け
4. 仮定の除去
5. `lia` タクティクを実行

証明の長さ

Table: Coq 上での証明の行数²

補題	命題の行数	証明の行数	合計
(1)	1	1	2
(2)	2	1	3
(3)	2	1	3
(4)	2	4	6
(5)	4	6	10
(6)	3	4	7
(7)	4	6	10
(8)	2	4	6
(9)	1	1	2
合計	21	28	49

²1 行の長さは 80 文字を上限とした

De Bruijn 代数 [Schäfer et al., 2015]

- ▶ De Bruijn 表現の項と並列代入について健全かつ完全な代数的構造
- ▶ 決定可能

Autosubst [Schäfer and Tebbi, 2014]

- ▶ De Bruijn 代数を応用して Coq 上に自動証明の仕組みを構築
- ▶ 項の定義から代入の定義を自動生成
- ▶ 非常に複雑な Ltac スクリプトでの自動化

- ▶ シフトと並列代入の代数的性質に対する (半) 自動証明
 - ▶ Coq が元々持っているタクティクを上手く利用している
- ▶ 応用
 - ▶ Church-Rosser の定理
 - ▶ 単純型付き λ 計算と System F の強正規化定理
強正規化定理の証明でも同様の自動証明の仕組みを利用
- ▶ <https://github.com/pi8027/lambda-calculus>



de Bruijn, N. G. (1972).

Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem.

Indagationes Mathematicae, 75(5):381–392.



Nipkow, T. (2001).

More Church-Rosser proofs.

Journal of Automated Reasoning, 26(1):51–66.



Schäfer, S., Smolka, G., and Tebbi, T. (2015).

Completeness and decidability of de Bruijn substitution algebra in Coq.

In Proceedings of the 2015 Conference on Certified Programs and Proofs, CPP '15, pages 67–73. ACM.

 Schäfer, S. and Tebbi, T. (2014).

Autosubst: Automation for de Bruijn syntax and substitution in Coq.