Explicit Substitutions in the Reduction of Lambda Terms

Auteurs : Gopalan Nadathur & Xiaochu Qi

Nathan Gerday
Alexis Baudin

Octobre 2019

Groupe de recherche : présentation d'article

Sommaire

Introduction

Présentation de l'article

- Écrit dans le cadre de la conférence *Principles and Practice of Declarative Programming (PPDP)* en 2003
- Décomposé en 3 grandes parties
    1. Présentation du mécanisme utilisé pour la substitution
    2. Présentation de 3 algorithmes avec différentes variations
    3. Comparaison des performances

Objectifs et Motivations

- La substitution dans le lambda calcul est complexe et très coûteuse si implémentée naïvement
- Ils cherchent donc a utiliser la notation par suspension pour effectuer les substitutions que nous présenterons en détails
- Enfin ils écrivent et présentent des algorithmes à l'aide du langage SML et détaillent les principes qui les régissent

Présentation de la notation de suspension

Représentation de de Bruijn des variables liées

- Objectif : éliminer les noms de variables liées, pour :
  - travail modulo α–conversion
  - meilleure efficacité de la β–réduction

Représentation de de Bruijn des variables liées

- Objectif : éliminer les noms de variables liées, pour :
  - travail modulo α–conversion
  - meilleure efficacité de la β–réduction
- Méthode : on remplace chaque variable par la profondeur de son abstraction

Représentation de de Bruijn des variables liées

- Objectif : éliminer les noms de variables liées, pour :
  - travail modulo α–conversion
  - meilleure efficacité de la β–réduction
- Méthode : on remplace chaque variable par la profondeur de son abstraction
- Exemple :

$$\lambda x.((\lambda y.xy)t)$$

Représentation de de Bruijn des variables liées

- Objectif : éliminer les noms de variables liées, pour :
  - travail modulo α–conversion
  - meilleure efficacité de la β–réduction
- Méthode : on remplace chaque variable par la profondeur de son abstraction
- Exemple :

$$\lambda x.((\lambda y.xy)t)$$
$$\equiv$$
$$(\lambda(\lambda(\#2\ \#1))t))$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t)$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \; \rightarrow \lambda x.(xy)[t/y]$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \ \rightarrow \lambda x.(xy)[t/y] \ \rightarrow \lambda x.xt$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \ \rightarrow \lambda x.(xy)[t/y] \ \rightarrow \lambda x.xt$$

$$\overset{\texttt{de Bruijn}}{\equiv}$$

$$(\lambda(\lambda(\#2\ \#1))t))$$

Définitions

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \ \to \lambda x.(xy)[t/y] \ \to \lambda x.xt$$

$$\overset{\texttt{de Bruijn}}{\equiv}$$

$$(\lambda(\lambda(\#2\ \mathbf{\#1}))\mathbf{t}))$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \ \rightarrow \lambda x.(xy)[t/y] \ \rightarrow \lambda x.xt$$

$$\overset{\texttt{de Bruijn}}{\equiv}$$

$$(\lambda(\lambda(\#2\ \mathbf{\#1}))\mathbf{t})) \quad \rightarrow \quad (\lambda(\#2\ t))$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \;\to \lambda x.(xy)[t/y] \;\to \lambda x.xt$$

$$\overset{\texttt{de Bruijn}}{\equiv}$$

$$(\lambda(\lambda(\#2\ \mathbf{\#1}))\mathbf{t})) \quad \to \quad (\lambda(\color{red}{\#2}\ t))$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \ \rightarrow \lambda x.(xy)[t/y] \ \rightarrow \lambda x.xt$$

$$\overset{\texttt{de Bruijn}}{\equiv}$$

$$(\lambda(\lambda(\#2\ \mathbf{\#1}))\mathbf{t})) \quad \rightarrow \quad (\lambda(\#2\ t)) \quad \rightarrow (\lambda(\#1\ t))$$

Exemple de substitution

$$\lambda x.((\lambda y.xy)t) \;\rightarrow \lambda x.(xy)[t/y] \;\rightarrow \lambda x.xt$$

$$\overset{\texttt{de Bruijn}}{\equiv}$$

$$(\lambda(\lambda(\#2\ \mathbf{\#1}))\mathbf{t})) \quad \rightarrow \quad (\lambda(\#2\ t)) \quad \rightarrow (\lambda(\#1\ t))$$

$\Rightarrow$ Besoin de rigueur sur l'indexation des variables

$\Rightarrow$ Alourdissement de quelques notations.

La notation de suspension

$[\![t, ol, nl, e]\!]$:

- t : terme avant substitution

Présentation de la notation de suspension
Définitions

La notation de suspension

$[\![t, ol, nl, e]\!]$:

- t : terme avant substitution
- ol : nombre de variables de t liées à l'extérieur

Définitions

La notation de suspension

$[\![t, ol, nl, e]\!]$:

- t : terme avant substitution
- ol : nombre de variables de t liées à l'extérieur
- nl : référence pour le décalage des variables liées

Définitions

La notation de suspension

$[\![t, ol, nl, e]\!]$:

- t : terme avant substitution
- ol : nombre de variables de t liées à l'extérieur
- nl : référence pour le décalage des variables liées
- e : environnement qui encode les substitutions "suspendues"

  Liste de "termes d'environnement" :

  $\boxed{e[i] \textit{ substitution de } \#i}$, i^{eme} *variable liée à l'extérieur*

Exemple

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ @1 :: (t, 1) :: @0 :: nil]\!]$$

- $\#1$: $@\_\_$
- $\#2$: $(t, \_\_)$
- $\#3$: $@\_\_$

Présentation de la notation de suspension
Définitions

Exemple

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ \mathtt{@}1 :: (t, 1) :: \mathtt{@}0 :: nil]\!]$$

- $\#1$: $\mathtt{@}\_\_ \rightarrow$ décalage $\rightarrow \#1$
- $\#2$: $(t, \_\_)$
- $\#3$: $\mathtt{@}\_\_$

Présentation de la notation de suspension
Définitions

Exemple

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ \texttt{@}1 :: (t, 1) :: \texttt{@}0 :: nil]\!]$$

- $\#1$: $\texttt{@}\_\_ \rightarrow$ décalage $\rightarrow \#1$
- $\#2$: $(t, \_\_) \rightarrow$ substitution $+$ décalage dans $t \rightarrow [\![t, 0, 1, nil]\!]$
- $\#3$: $\texttt{@}\_\_$

Présentation de la notation de suspension
Définitions

Exemple

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ \texttt{@}1 :: (t, 1) :: \texttt{@}0 :: nil]\!]$$

- $\#1$: $\texttt{@}\_\_ \rightarrow$ décalage $\rightarrow \#1$
- $\#2$: $(t, \_\_) \rightarrow$ substitution + décalage dans $t \rightarrow [\![t, 0, 1, nil]\!]$
- $\#3$: $\texttt{@}\_\_ \rightarrow$ décalage $\rightarrow \#2$

Exemple

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ @1 :: (t, 1) :: @0 :: nil]\!]$$

- $\#1$: $@\_\_ \rightarrow$ décalage $\rightarrow \#1$
- $\#2$: $(t, \_\_) \rightarrow$ substitution + décalage dans $t \rightarrow [\![t, 0, 1, nil]\!]$
- $\#3$: $@\_\_ \rightarrow$ décalage $\rightarrow \#2$

$$\Downarrow$$
$$((\#1\ [\![t, 0, 1, nil]\!])\ \#2)$$

Règles de réécriture

Génération des substitutions

$$(\beta_s) \quad ((\lambda t_1) t_2) \quad \rightarrow \quad [\![t_1, 1, 0, (t_2, 0) :: nil]\!]$$

Première substitution

Règles de réécriture

Génération des substitutions

$$(\beta_s) \quad ((\lambda t_1)t_2) \quad \rightarrow \quad [\![t_1, 1, 0, (t_2, 0) :: nil]\!]$$

Première substitution

$$(\beta'_s) \quad ((\lambda[\![t_1, ol+1, nl+1, @nl :: e]\!])t_2 \quad \rightarrow \quad [\![t_1, ol+1, nl, (t_2, nl) :: e]\!]$$

Ajout d'une substitution,

sur la première variable libre extérieure

Règles de réécriture

Cas de base : suspension sans substitution

$$(r_1) \quad [\![c, ol, nl, e]\!] \quad \rightarrow \quad c$$

si c est une constante

Règles de réécriture

Cas de base : suspension sans substitution

$$(r_1) \quad [\![c, ol, nl, e]\!] \quad \rightarrow \quad c$$

si c est une constante

$$(r_2) \quad [\![x, ol, nl, e]\!] \quad \rightarrow \quad x$$

si x est une variable libre

Règles de réécriture

Cas de base : suspension sans substitution

$$(r_1) \quad [\![c, ol, nl, e]\!] \quad \rightarrow \quad c$$

si c est une constante

$$(r_2) \quad [\![x, ol, nl, e]\!] \quad \rightarrow \quad x$$

si x est une variable libre

$$(r_9) \quad [\![t, 0, 0, nil]\!] \quad \rightarrow \quad t$$

Pas de substitution suspendue

Règles de réécriture

Gestion des variables liées

Règles de réécriture

Gestion des variables liées

Renommage : réindexation des variables de de Bruijn

$(r_3) \quad [\![\#i, ol, nl, e]\!] \quad \rightarrow \quad \#j$

si $i > ol$ *(avec* $j = i - ol + nl$*)*

Variable pas dans l'environnement

Règles de réécriture

Gestion des variables liées

Renommage : réindexation des variables de de Bruijn

$(r_3)\quad [\![\#i, ol, nl, e]\!] \quad \rightarrow \quad \#j$

$\quad\quad$ *si* $i > ol$ *(avec* $j = i - ol + nl$*)*

$\quad\quad$ *Variable pas dans l'environnement*

$(r_4)\quad [\![\#i, ol, nl, e]\!] \quad \rightarrow \quad \#j$

$\quad\quad$ *si* $i \leq ol$ *et* $e[i] = @l$ *(avec* $j = nl - l$*)*

$\quad\quad$ *Variable dans l'environnement mais non substituée*

Règles de réécriture

Gestion des variables liées

Renommage : réindexation des variables de de Bruijn

$$(r_3) \quad [\![\#i, ol, nl, e]\!] \quad \rightarrow \quad \#j$$

si $i > ol$ (avec $j = i - ol + nl$)

Variable pas dans l'environnement

$$(r_4) \quad [\![\#i, ol, nl, e]\!] \quad \rightarrow \quad \#j$$

si $i \leq ol$ et $e[i] = @l$ (avec $j = nl - l$)

Variable dans l'environnement mais non substituée

Application de la substitution

$$(r_5) \quad [\![\#i, ol, nl, e]\!] \quad \rightarrow \quad [\![t, 0, j, nil]\!]$$

si $i \leq ol$ et $e[i] = (t, l)$ (avec $j = nl - l$)

Variable dans l'environnement et substituée

Règles de réécriture

Propagation des substitutions

$$(r_6) \quad [\![(t_1\ t_2), ol, nl, e]\!] \quad \rightarrow \quad ([\![t_1, ol, nl, e]\!]\ [\![t_2, ol, nl, e]\!])$$

Propagation de la suspension

Règles de réécriture

Propagation des substitutions

$$(r_6) \quad [\![(t_1\ t_2), ol, nl, e]\!] \quad \rightarrow \quad ([\![t_1, ol, nl, e]\!]\ [\![t_2, ol, nl, e]\!])$$

Propagation de la suspension

$$(r_7) \quad [\![(\lambda t), ol, nl, e]\!] \quad \rightarrow \quad (\lambda [\![t, ol+1, nl+1, @nl :: e]\!])$$

Passage d'une variable liée à l'extérieur

Présentation de la notation de suspension
Règles de réécriture

Propagation des substitutions

$(r_6)\quad [\![(t_1\ t_2), ol, nl, e]\!] \quad\rightarrow\quad ([\![t_1, ol, nl, e]\!]\ [\![t_2, ol, nl, e]\!])$

Propagation de la suspension

$(r_7)\quad [\![(\lambda t), ol, nl, e]\!] \quad\rightarrow\quad (\lambda [\![t, ol+1, nl+1, @nl :: e]\!])$

Passage d'une variable liée à l'extérieur

$(r_8)\quad [\![[\![t, ol, nl, e]\!], 0, nl', nil]\!] \quad\rightarrow\quad [\![t, ol, nl+nl', e]\!]$

Suspension dans une suspension sans substitution : décalage

Utilisation implicite de la notation de suspension

Utilisation implicite de la notation de suspension

- On utilise les suspensions pour propager les substitutions sur les termes

Utilisation implicite de la notation de suspension

- On utilise les suspensions pour propager les substitutions sur les termes
- Cependant, les termes eux-mêmes ne peuvent pas contenir de suspensions.

Utilisation implicite de la notation de suspension

- On utilise les suspensions pour propager les substitutions sur les termes
- Cependant, les termes eux-mêmes ne peuvent pas contenir de suspensions.
- Ces dernières ne seront appliquées qu'implicitement au travers des paramètres des appels récursifs à la méthode de normalisation du terme

```
fun head_norm1
 (term as ref(bv(i)),ol,nl,env,whnf) =
   ...
    head_norm1(t,ol',nl+nl'-l,e',whnf)
```

Utilisation implicite de la notation de suspension

Pas de suspension dans le terme

```
type rawterm = const of string
  | bv of int
  | ptr of (rawterm ref)
  | app of (rawterm ref * rawterm ref)
  | lam of (rawterm ref)
```

Utilisation implicite de la notation de suspension

Pas de suspension dans le terme

```
type rawterm = const of string
  | bv of int
  | ptr of (rawterm ref)
  | app of (rawterm ref * rawterm ref)
  | lam of (rawterm ref)
```

Mais uniquement dans les closures

```
type eitem = dum of int
  | bndg of clos * int
and clos = cl of term * int * int * (eitem list)
```

Utilisation implicite de la notation de suspension
Intérêts et défauts

Intérêts

- Cette utilisation nous permet de commencer à utiliser les règles permettant la réduction en utilisant les suspensions
- Il n'est pas nécessaire de modifier les structure des termes pour gérer l'utilisation des suspensions

Utilisation implicite de la notation de suspension
Intérêts et défauts

Intérêts

- Cette utilisation nous permet de commencer à utiliser les règles permettant la réduction en utilisant les suspensions
- Il n'est pas nécessaire de modifier les structure des termes pour gérer l'utilisation des suspensions

Défauts

- Les termes ne pouvant pas contenir de suspensions, il faut passer par de nombreuses étapes supplémentaires afin d'utiliser les règles de substitution

Utilisation explicite de la notation de suspension

Utilisation explicite de la notation de suspension

Exemple rappel

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ @1 :: (t, 1) :: @0 :: nil]\!]$$

Utilisation explicite de la notation de suspension

Exemple rappel

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ @1 :: (t, 1) :: @0 :: nil]\!]$$

Idée : la suspension est un terme

```
type term = ...
    | susp of term * int * int * (eitem list)
```

Utilisation explicite de la notation de suspension

Exemple rappel

$$[\![((\#1\ \#2)\ \#3),\ ol = 3,\ nl = 2,\ @1 :: (t, 1) :: @0 :: nil]\!]$$

Idée : la suspension est un terme

```
type term = ...
    | susp of term * int * int * (eitem list)

and eitem = dum of int
    | bndg of  term * int

type env = eitem list
```

Utilisation explicite de la notation de suspension

Intérêt de la méthode

- Utiliser les règles de réécriture directement

Utilisation explicite de la notation de suspension

Intérêt de la méthode

- Utiliser les règles de réécriture directement
- Évaluer la suspension seulement quand c'est nécessaire

Utilisation explicite de la notation de suspension

Intérêt de la méthode

- Utiliser les règles de réécriture directement
- Évaluer la suspension seulement quand c'est nécessaire

Principal inconvénient : coût de la construction

$$[\![(t_1\ \ t_2), ol, nl, e]\!] \rightarrow ([\![t_1, ol, nl, e]\!]\ [\![t_2, ol, nl, e]\!])$$

Combinaison de l’utilisation implicite et explicite

Combinaison de l'utilisation implicite et explicite

Objectifs

- Ne pas avoir à évaluer toutes les substitutions $\rightarrow$ seulement celles nécessaires au calcul de la forme normale de tête

Combinaison de l'utilisation implicite et explicite

Objectifs

- Ne pas avoir à évaluer toutes les substitutions $\rightarrow$ seulement celles nécessaires au calcul de la forme normale de tête
- Ne créer des structures de suspensions que quand on ne les évalue pas

Combinaison de l'utilisation implicite et explicite

Objectifs

- Ne pas avoir à évaluer toutes les substitutions $\rightarrow$ seulement celles nécessaires au calcul de la forme normale de tête
- Ne créer des structures de suspensions que quand on ne les évalue pas

Procédure

- *Implicite :* trouver la forme normale de tête

Combinaison de l'utilisation implicite et explicite

Objectifs

- Ne pas avoir à évaluer toutes les substitutions $\rightarrow$ seulement celles nécessaires au calcul de la forme normale de tête
- Ne créer des structures de suspensions que quand on ne les évalue pas

Procédure

- *Implicite :* trouver la forme normale de tête
- *Explicite :* laisser les termes sous forme de suspension si pas besoin de les évaluer

Combinaison de l'utilisation implicite et explicite

Nombre de termes créés pendant la procédure

	Suspension implicite	Suspension explicite	Combinaison des deux
[typeinf]	20 834 989	11 044 078	4 508 664
[compiler]	4 565 938	777 803	331 973
[church]	227 271	214 334	148 970
[hilbert]	220 358	27 263	11 932

Combinaison de l'utilisation implicite et explicite

Nombre de termes créés pendant la procédure

	Suspension implicite	Suspension explicite	Combinaison des deux
[typeinf]	20 834 989	11 044 078	4 508 664
[compiler]	4 565 938	777 803	331 973
[church]	227 271	214 334	148 970
[hilbert]	220 358	27 263	11 932

$\Rightarrow$ Utilisation mémoire bien meilleure