Sur l'espace des termes et des machines

TRE M1 Informatique – Supervisé par Damiano Mazza

Sambo Boris Eng

Introduction

Introduction

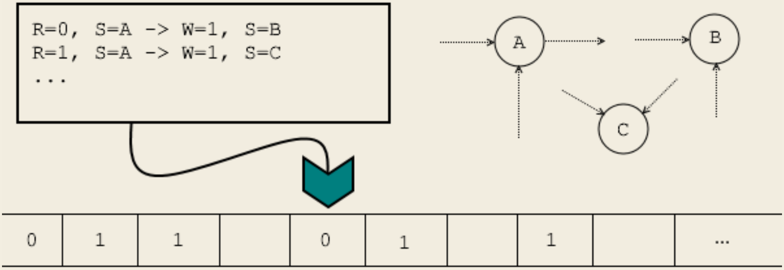
Le concept de calcul : machines

Une première approximation : **la Machine de Turing** [Turing 1936]

Introduction

Le concept de calcul : machines

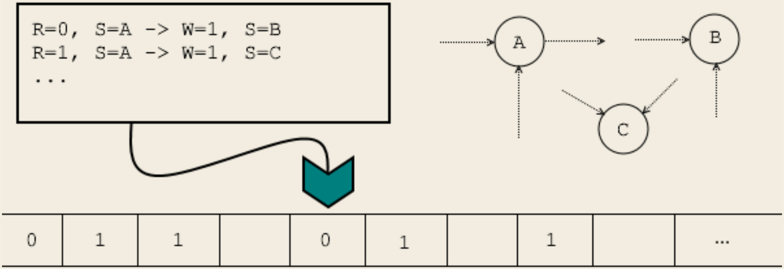
Une première approximation : **la Machine de Turing** [Turing 1936]



Introduction

Le concept de calcul : machines

Une première approximation : **la Machine de Turing** [Turing 1936]



Modèle dynamique procédural

Introduction

Le concept de calcul : termes

Un autre modèle majeur : λ-**calcul** [Church 1932]

Introduction

Le concept de calcul : termes

Un autre modèle majeur : λ-**calcul** [Church 1932]

Abstrait	**Concret/Syntaxique**
$f : n \mapsto n+1$	$\lambda n.\lambda s.\lambda z.s\,(n\,s\,z)$

Introduction

Le concept de calcul : termes

Un autre modèle majeur : λ-**calcul** [Church 1932]

Abstrait	**Concret/Syntaxique**
$f : n \mapsto n + 1$	$\lambda n.\lambda s.\lambda z.s\,(n\,s\,z)$

Statique. $M, N := x \mid \lambda x.M \mid MN$

Introduction

Le concept de calcul : termes

Un autre modèle majeur : λ-**calcul** [Church 1932]

Abstrait	Concret/Syntaxique
$f : n \mapsto n+1$	$\lambda n.\lambda s.\lambda z.s\,(n\,s\,z)$

Statique. $M, N := x \mid \lambda x.M \mid MN$

Dynamique. $(\lambda x.M)N \longrightarrow_\beta M\{x := N\}$

Introduction

Le concept de calcul : termes

Un autre modèle majeur : λ-**calcul** [Church 1932]

Abstrait	**Concret/Syntaxique**
$f : n \mapsto n + 1$	$\lambda n.\lambda s.\lambda z.s\,(n\,s\,z)$

Statique. $M, N := x \mid \lambda x.M \mid MN$

Dynamique. $(\lambda x.M)N \longrightarrow_{\beta} M\{x := N\}$

Modèle dynamique algébrique

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :
  - Usage de la géométrie de l'interaction [Schöpp]

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :
  - Usage de la géométrie de l'interaction [Schöpp]
  - Une "bonne" mesure de l'espace

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :
  - Usage de la géométrie de l'interaction [Schöpp]
  - Une "bonne" mesure de l'espace
- **Correspondance dynamique/statique** :

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :
  - Usage de la géométrie de l'interaction [Schöpp]
  - Une "bonne" mesure de l'espace
- **Correspondance dynamique/statique** :
  - Une machine peut être simulée par des circuits [Cook-Levin]

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :
  - Usage de la géométrie de l'interaction [Schöpp]
  - Une "bonne" mesure de l'espace
- **Correspondance dynamique/statique** :
  - Une machine peut être simulée par des circuits [Cook-Levin]
  - Et pour le λ-calcul ?

Introduction

Motivations

Thèse de Church-Turing : même puissance. Efficacité différente ?

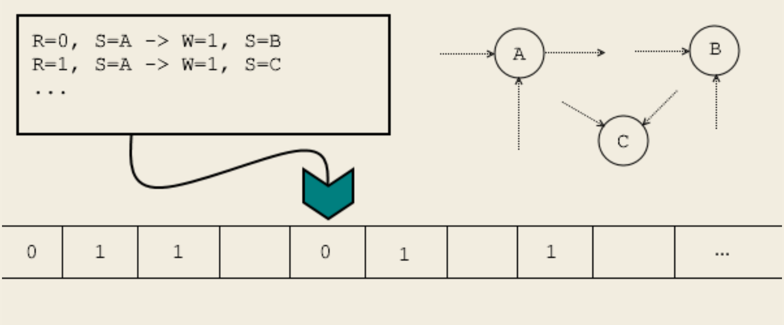
- **Thèse d'invariance** : efficacité similaire.
  - Temps machines $\cong$ temps termes [Accattoli, Dal Lago]
  - Et pour l'espace ?
- **Usage d'outils logiques** :
  - Usage de la géométrie de l'interaction [Schöpp]
  - Une "bonne" mesure de l'espace
- **Correspondance dynamique/statique** :
  - Une machine peut être simulée par des circuits [Cook-Levin]
  - Et pour le λ-calcul ?

Rapprocher termes et machines.

Complexité

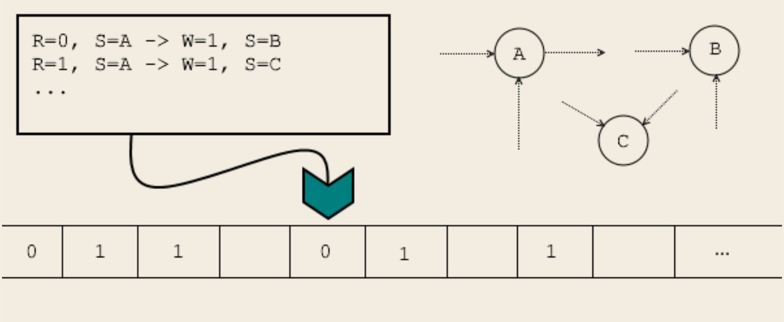
Complexité

des machines



Complexité

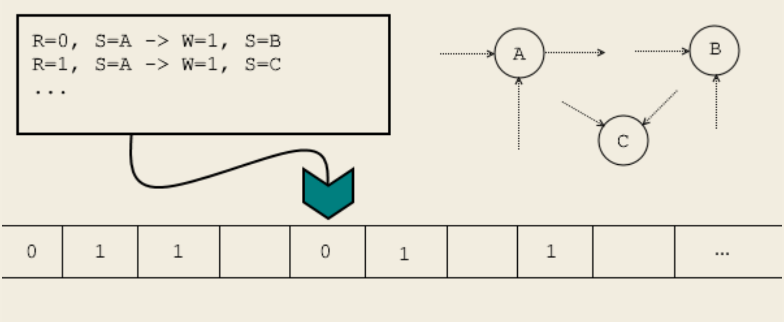
des machines



Décision. Entrée binaire $\longleftrightarrow$ état YES ou NO (terminaison)

Complexité

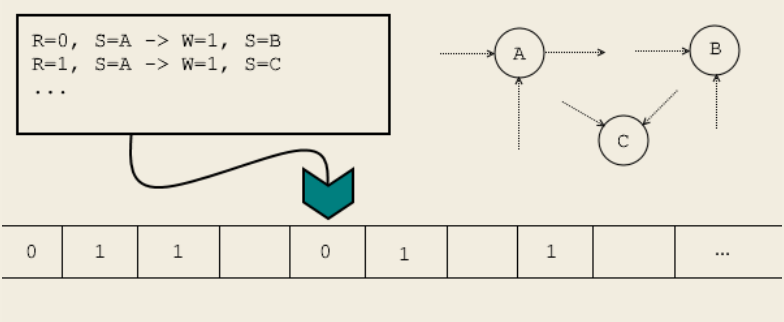
des machines



Décision. Entrée binaire $\longleftrightarrow$ état YES ou NO (terminaison)

Complexité

des machines

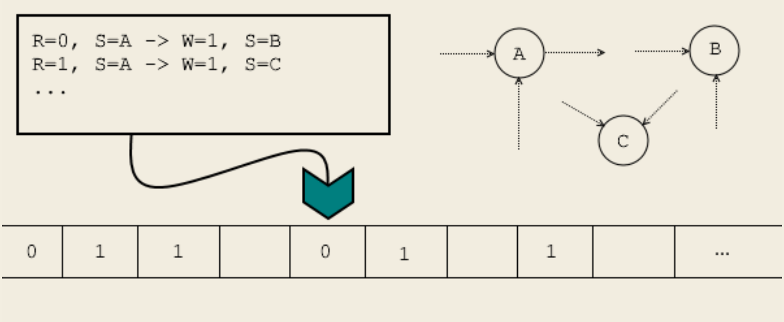


Décision. Entrée binaire $\longleftrightarrow$ état YES ou NO (terminaison)

- **Temps** : nombre de déplacement du pointeur

Complexité

des machines

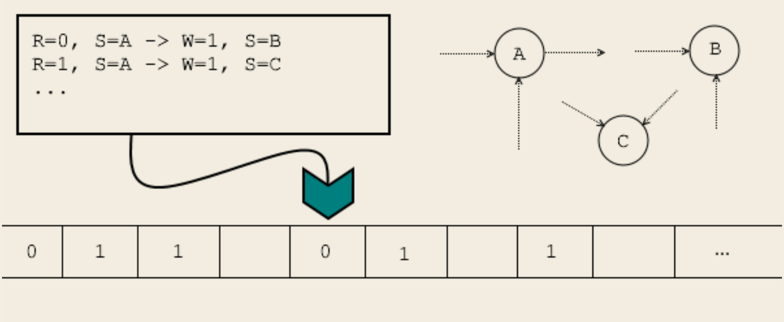


Décision. Entrée binaire $\longleftrightarrow$ état YES ou `NO` (terminaison)

- **Temps** : nombre de déplacement du pointeur
- **Espace** : nombre de cases parcourues

Complexité

des machines



Décision. Entrée binaire $\longleftrightarrow$ état YES ou `NO` (terminaison)

- **Temps** : nombre de déplacement du pointeur
- **Espace** : nombre de cases parcourues

Le modèle standard de complexité

Complexité

raisonnable des termes

Encodage nécessaire dans les termes noté $\underline{0}$, $\underline{1}$, $\underline{w_1 \ldots w_n}$

Complexité

raisonnable des termes

Encodage nécessaire dans les termes noté $\underline{0}$, $\underline{1}$, $\underline{w_1 \ldots w_n}$

Décision : $M\,\underline{w} \longrightarrow^* \underline{b}$

Complexité

raisonnable des termes

Encodage nécessaire dans les termes noté $\underline{0}$, $\underline{1}$, $\underline{w_1 \ldots w_n}$

Décision : $M\ \underline{w} \longrightarrow^* \underline{b}$

Definition

Un **modèle de coût raisonnable** est une mesure de complexité avec

- écart **polynomial** en temps
- écart **constant** en espace

par rapport aux machines de Turing

Complexité

raisonnable des termes

Encodage nécessaire dans les termes noté $\underline{0}$, $\underline{1}$, $\underline{w_1 \dots w_n}$

Décision : $M\ \underline{w} \longrightarrow^* \underline{b}$

Definition

Un **modèle de coût raisonnable** est une mesure de complexité avec

- écart **polynomial** en temps
- écart **constant** en espace

par rapport aux machines de Turing

- **Temps** : résolu par Accattoli et Dal Lago (réduction de tête)

Complexité

raisonnable des termes

Encodage nécessaire dans les termes noté $\underline{0}$, $\underline{1}$, $\underline{w_1 \dots w_n}$

Décision : $M\ \underline{w} \longrightarrow^* \underline{b}$

Definition

Un **modèle de coût raisonnable** est une mesure de complexité avec

- écart **polynomial** en temps
- écart **constant** en espace

par rapport aux machines de Turing

- **Temps** : résolu par Accattoli et Dal Lago (réduction de tête)
- **Espace** : nous utilisons la géométrie de l'interaction

Complexité

Solution pour l’espace des termes

λ-calcul (trop compliqué)

Complexité

Solution pour l’espace des termes

λ-calcul (trop compliqué)

$\downarrow$

λ-calcul fini (simple)

Complexité

Solution pour l'espace des termes

λ-calcul (trop compliqué)

$\downarrow$

λ-calcul fini (simple)

$\downarrow$

Usage de la géométrie de l'interaction

Approximations du calcul

Approximations polyadiques

Un calcul fini

Une idée de la logique linéaire [Girard]

Definition

On peut approximer M par une construction $\langle M, \ldots, M \rangle$ [Mazza]

Approximations polyadiques

Un calcul fini

Une idée de la logique linéaire [Girard]

Definition

On peut approximer M par une construction $\langle M, \ldots, M \rangle$ [Mazza]

Lambda-calcul polyadique affine [Mazza].

$$t, u := x \mid \lambda a.t \mid tu \mid \langle t_1, \ldots, t_n \rangle \mid t[\,\langle x_1, \ldots, x_n \rangle := u]$$

Approximations polyadiques

Un calcul fini

Une idée de la logique linéaire [Girard]

Definition

On peut approximer M par une construction $\langle M, \ldots, M \rangle$ [Mazza]

Lambda-calcul polyadique affine [Mazza].

$$t, u := x \mid \lambda a.t \mid tu \mid \langle t_1, \ldots, t_n \rangle \mid t[\, \langle x_1, \ldots, x_n \rangle := u]$$

Approximation polyadique.

- $t \sqsubset M$

Approximations polyadiques

Un calcul fini

Une idée de la logique linéaire [Girard]

Definition

On peut approximer M par une construction $\langle M, \ldots, M \rangle$ [Mazza]

Lambda-calcul polyadique affine [Mazza].

$$t, u := x \mid \lambda a.t \mid tu \mid \langle t_1, \ldots, t_n \rangle \mid t[\,\langle x_1, \ldots, x_n \rangle := u]$$

Approximation polyadique.

- $t \sqsubset M$
- $\langle x_0, x_1, x_2 \rangle \sqsubset x$

Système de type intersection non-idempotent

Propriétés quantitatives

On associe un type aux termes $M : A$ pour forcer une cohérence subjective.

Système de type intersection non-idempotent

Propriétés quantitatives

On associe un type aux termes $M : A$ pour forcer une cohérence subjective.

- **Terminaison** : typage $\Longleftrightarrow$ terminaison

Système de type intersection non-idempotent

Propriétés quantitatives

On associe un type aux termes $M : A$ pour forcer une cohérence subjective.

- **Terminaison** : typage $\Longleftrightarrow$ terminaison
- **Traçage d'occurrence** : informations quantitatives

Système de type intersection non-idempotent

Propriétés quantitatives

On associe un type aux termes $M : A$ pour forcer une cohérence subjective.

- **Terminaison** : typage $\Longleftrightarrow$ terminaison
- **Traçage d'occurrence** : informations quantitatives

Terme	Occurrences	Type
M	3	$\langle A_1, A_2, A_3 \rangle$

Système de type intersection non-idempotent

Propriétés quantitatives

On associe un type aux termes $M : A$ pour forcer une cohérence subjective.

- **Terminaison** : typage $\Longleftrightarrow$ terminaison
- **Traçage d'occurrence** : informations quantitatives

Terme	Occurrences	Type
M	3	$\langle A_1, A_2, A_3 \rangle$

- **Traçage d'usage** : un type par usage concret (polymorphisme)

$$x : \langle A \multimap B, \quad A \rangle$$

Système de type intersection non-idempotent

Exemple

Dérivation de type = preuve de typabilité

$$\dfrac{\dfrac{}{x : A \multimap B \vdash x : A \multimap B} \qquad \dfrac{}{x : A \vdash x : A}}{\dfrac{x : \langle A \multimap B, A \rangle \vdash xx : B}{\vdash \lambda x.xx : \langle A \multimap B, A \rangle \multimap B}}$$

Système de type intersection non-idempotent

Exemple

Dérivation de type = preuve de typabilité

$$\dfrac{\dfrac{}{x : A \multimap B \vdash x : A \multimap B} \qquad \dfrac{}{x : A \vdash x : A}}{\dfrac{x : \langle A \multimap B, A \rangle \vdash xx : B}{\vdash \lambda x.xx : \langle A \multimap B, A \rangle \multimap B}}$$

Theorem

Ces arbres induisent des termes polyadiques affines [Mazza]

Système de type intersection non-idempotent

Exemple

Dérivation de type = preuve de typabilité

$$\dfrac{\dfrac{}{x : A \multimap B \vdash x : A \multimap B} \qquad \dfrac{}{x : A \vdash x : A}}{\dfrac{x : \langle A \multimap B, A\rangle \vdash xx : B}{\vdash \lambda x.xx : \langle A \multimap B, A\rangle \multimap B}}$$

Theorem

Ces arbres induisent des termes polyadiques affines [Mazza]

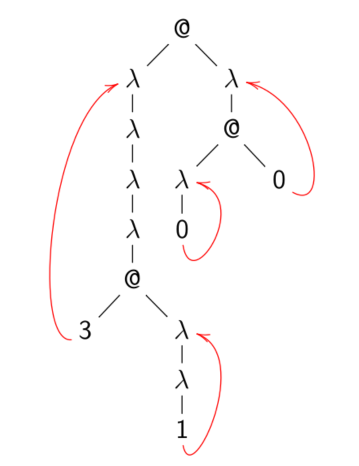
Terme polyadique affine induit : $\lambda a.(x_0 x_1)[\,\langle x_0, x_1\rangle := \langle a, a\rangle\,]$

Résultats

Géométrie de l'interaction [Girard]

pour borner l'espace

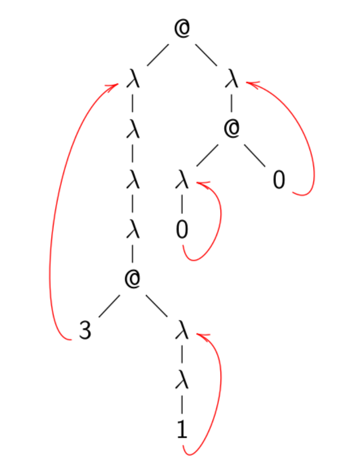
Procédure sur les termes finis



Géométrie de l'interaction [Girard]

pour borner l'espace

Procédure sur les termes finis

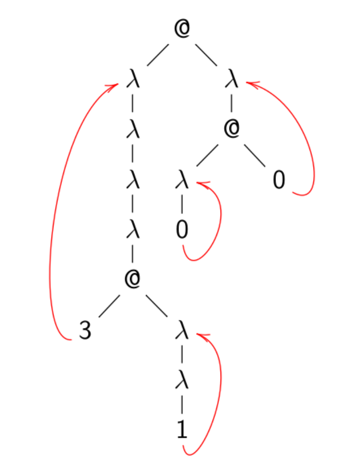


Évaluation ⊆ chemins du terme

Géométrie de l'interaction [Girard]

pour borner l'espace

Procédure sur les termes finis



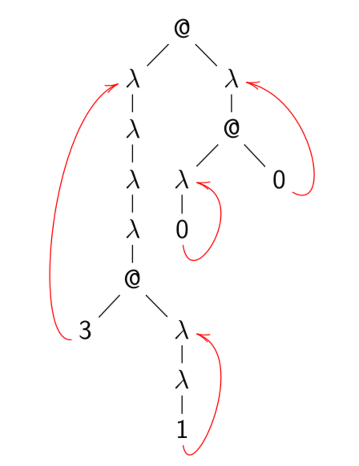
Évaluation ⊆ chemins du terme

- On parcourt statiquement le terme

Géométrie de l'interaction [Girard]

pour borner l'espace

Procédure sur les termes finis



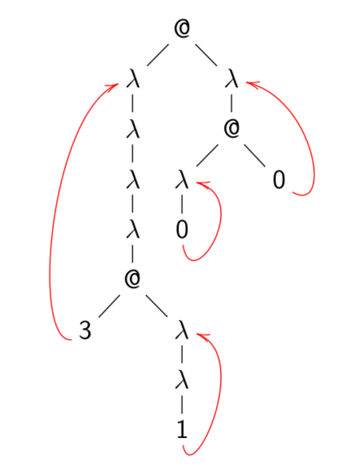
Évaluation ⊆ chemins du terme

- On parcourt statiquement le terme
- On se souvient des directions dans une pile

Géométrie de l'interaction [Girard]

pour borner l'espace

Procédure sur les termes finis



Évaluation $\subseteq$ chemins du terme

- On parcourt statiquement le terme
- On se souvient des directions dans une pile
- On a des informations sur le résultat de la réduction

Borne spatiale

pour le lambda-calcul

Theorem

Si l'on a

$$\dfrac{\begin{matrix}\vdots \\ \delta\end{matrix}}{\vdash M : String \multimap Bool}$$

alors M s'exécute en espace $O(depth(\delta) + log|\delta|)$

Borne spatiale

pour le lambda-calcul

Theorem

Si l'on a

$$\dfrac{\begin{array}{c}\vdots \\ \delta\end{array}}{\vdash M : String \multimap Bool}$$

alors M s'exécute en espace $O(depth(\delta) + log|\delta|)$

Preuve. voir tableau.

Borne spatiale

pour le lambda-calcul

Theorem

Si l'on a

$$\dfrac{\begin{matrix}\vdots \\ \delta\end{matrix}}{\vdash M : String \multimap Bool}$$

alors M s'exécute en espace $O(depth(\delta) + log|\delta|)$

Preuve. voir tableau.

- Espace de de M = profondeur de $\delta \longrightarrow \lambda\textbf{SPACE}(f)$

Borne spatiale

pour le lambda-calcul

Theorem

Si l'on a

$$\dfrac{\begin{matrix}\vdots \\ \delta\end{matrix}}{\vdash M : String \multimap Bool}$$

alors M s'exécute en espace $O(depth(\delta) + log|\delta|)$

Preuve. voir tableau.

- Espace de de M = profondeur de $\delta \longrightarrow \lambda\mathbf{SPACE}(f)$
- $\lambda\mathbf{SPACE}(f) \subseteq \mathbf{SPACE}(O(f))$: corollaire.

Borne spatiale

pour le lambda-calcul

Theorem

Si l'on a

$$\dfrac{\begin{array}{c}\vdots\\ \delta\end{array}}{\vdash M : \mathit{String} \multimap \mathit{Bool}}$$

alors M s'exécute en espace $O(\mathit{depth}(\delta) + \mathit{log}|\delta|)$

Preuve. voir tableau.

- Espace de de M = profondeur de $\delta \longrightarrow \lambda\mathbf{SPACE}(f)$
- $\lambda\mathbf{SPACE}(f) \subseteq \mathbf{SPACE}(O(f))$: corollaire.
- $\mathbf{SPACE}(f) \subseteq \lambda\mathbf{SPACE}(O(f))$: pas abordé.

Dynamique et statique

Machines et circuits

- **Borodin** : $\mathbf{DEPTH}(f) \subseteq \mathbf{SPACE}(O(f))$

Dynamique et statique

Machines et circuits

- **Borodin** : $\mathbf{DEPTH}(f) \subseteq \mathbf{SPACE}(O(f))$
- **Cook-Levin** : une machine induit une famille de circuits uniforme (générée efficacement par un programme)

Dynamique et statique

Machines et circuits

- **Borodin** : $\mathbf{DEPTH}(f) \subseteq \mathbf{SPACE}(O(f))$
- **Cook-Levin** : une machine induit une famille de circuits uniforme (générée efficacement par un programme)

Theorem

La famille de termes polyadiques affines induites dans le théorème de borne spatiale est ***DLOGTIME****-uniforme*

Dynamique et statique

Machines et circuits

- **Borodin** : $\mathbf{DEPTH}(f) \subseteq \mathbf{SPACE}(O(f))$
- **Cook-Levin** : une machine induit une famille de circuits uniforme (générée efficacement par un programme)

Theorem

La famille de termes polyadiques affines induites dans le théorème de borne spatiale est ***DLOGTIME****-uniforme*

$$\frac{\text{Machine de Turing}}{\text{Circuit booléen}} \cong \frac{\lambda\text{-calcul}}{\lambda\text{-calcul polyadique affine}}$$

Conclusion

- Création de la classe $\lambda\mathbf{SPACE}(f)$ fondé sur la profondeur des types intersections

Conclusion

- Création de la classe $\lambda\mathbf{SPACE}(f)$ fondé sur la profondeur des types intersections
- Reste à prouver $\mathbf{SPACE}(f) \subseteq \lambda\mathbf{SPACE}(O(f))$. Il faudrait coder une machine de Turing par un terme typé par une dérivation intersection de profondeur efficace [voir Accattoli, Dal Lago]

Conclusion

- Création de la classe $\lambda\mathbf{SPACE}(f)$ fondé sur la profondeur des types intersections
- Reste à prouver $\mathbf{SPACE}(f) \subseteq \lambda\mathbf{SPACE}(O(f))$. Il faudrait coder une machine de Turing par un terme typé par une dérivation intersection de profondeur efficace [voir Accattoli, Dal Lago]
- Transfert de résultats et d'outils entre le λ-calcul et les machines [Mazza, Cook-Levin]

Conclusion

- Création de la classe λ**SPACE**(f) fondé sur la profondeur des types intersections
- Reste à prouver **SPACE**$(f) \subseteq \lambda$**SPACE**$(O(f))$. Il faudrait coder une machine de Turing par un terme typé par une dérivation intersection de profondeur efficace [voir Accattoli, Dal Lago]
- Transfert de résultats et d'outils entre le λ-calcul et les machines [Mazza, Cook-Levin]
- Passage du mécanique à l'algébrique

Conclusion

- Création de la classe λ**SPACE**(f) fondé sur la profondeur des types intersections
- Reste à prouver **SPACE**$(f) \subseteq \lambda$**SPACE**$(O(f))$. Il faudrait coder une machine de Turing par un terme typé par une dérivation intersection de profondeur efficace [voir Accattoli, Dal Lago]
- Transfert de résultats et d'outils entre le λ-calcul et les machines [Mazza, Cook-Levin]
- Passage du mécanique à l'algébrique

Uniformité

Il existe une machine qui génère des circuits

$$\cong$$

Un λ-terme est approximé par des termes affines