Terminaison de fonctions

David Delahaye

Faculté des Sciences
David.Delahaye@lirmm.fr

Master Informatique M1 2024-2025

Terminaison de fonctions

Problème de l'arrêt

- Problème de décision qui détermine, à partir d'une description d'un programme, et d'une entrée, si le programme s'arrête avec cette entrée ou non.
- Alan Turing a montré en 1936 que le problème de l'arrêt est indécidable, c'est-à-dire qu'il n'existe pas de programme correct, complet et terminant répondant à ce problème.
- Ce problème est généralisé par le théorème de Rice : toute propriété sémantique non triviale d'un programme est indécidable.

Terminaison de fonctions

Comment montrer la terminaison d'une fonction ?

- Une technique consiste à transformer la fonction en système de réécriture du premier ordre et à démontrer que le système obtenu est terminant (on dit aussi fortement normalisable).
- Il existe une grande communauté de l'Informatique théorique qui travaille sur la terminaison des systèmes de réécriture.
- Les techniques utilisées passent par la notion d'ordre de terminaison.
- Problème difficile.
  $f(g(x), y) \rightarrow f(y, y)$ terminant ?
  $f(g(a), g(a)) \rightarrow f(g(a), g(a)) \rightarrow f(g(a), g(a)) \rightarrow \ldots$.

Terminaison de fonctions

Comment montrer la terminaison d'une fonction ?

- Une technique consiste à transformer la fonction en système de réécriture du premier ordre et à démontrer que le système obtenu est terminant (on dit aussi fortement normalisable).
- Il existe une grande communauté de l'Informatique théorique qui travaille sur la terminaison des systèmes de réécriture.
- Les techniques utilisées passent par la notion d'ordre de terminaison.
- Problème difficile.
  $f(g(x), y) \to f(y, y)$ terminant ?
  $f(g(a), g(a)) \to f(g(a), g(a)) \to f(g(a), g(a)) \to \ldots$.

Terminaison de fonctions

Comment montrer la terminaison d'une fonction ?

- Une technique consiste à transformer la fonction en système de réécriture du premier ordre et à démontrer que le système obtenu est terminant (on dit aussi fortement normalisable).
- Il existe une grande communauté de l'Informatique théorique qui travaille sur la terminaison des systèmes de réécriture.
- Les techniques utilisées passent par la notion d'ordre de terminaison.
- Problème difficile.
  $f(g(x), y) \rightarrow f(y, y)$ terminant ?
  $f(g(a), g(a)) \rightarrow f(g(a), g(a)) \rightarrow f(g(a), g(a)) \rightarrow \ldots$.

Termes du premier ordre

Définitions préliminaires

- $\mathcal{V} \equiv$ ensemble de variables d'individu x, y, etc.
- $\mathcal{S}_\mathcal{F} \equiv$ ensemble de symboles de fonctions f, g, etc.
- Arité (nombre d'arguments) $m : \mathcal{S}_\mathcal{F} \to \mathbb{N}$:
  - Exemple : pour $f(x, y)$ avec $f \in \mathcal{S}_\mathcal{F}$, $m(f) = 2$.

Termes du premier ordre

- Plus petit ensemble $\mathcal{T}$ t.q. :
  - Si $x \in \mathcal{V}$ alors $x \in \mathcal{T}$.
  - Si $f \in \mathcal{S}_\mathcal{F}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $f(t_1, \ldots, t_n) \in \mathcal{T}$.
- Les constantes sont des fonctions d'arité 0.
- Si $t \in \mathcal{T}$, $FV(t)$ est l'ensemble des variables de t.

Substitution

Définition

- Une substitution σ est une application de $\mathcal{V}$ vers $\mathcal{T}$.
- Elle se définit par récurrence structurelle comme suit :
  - Si $x \in \mathcal{V}$ alors $\sigma(x) = \sigma(x)$.
  - Si $f \in \mathcal{S}_{\mathcal{F}}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n))$.

Exemple

- Soit la substitution σ telle que $\sigma(x) = a$ et $\sigma(y) = f(b)$, où a et b sont des constantes, et f un symbole de fonction unaire ;
- $\sigma(f(x, y)) = f(a, f(b))$.

Position et substitution à une position donnée

Position

- Une position est un élément de $(\mathbb{N} - \{0\})^*$;
- Étant donné un terme t, le terme $t|_p$ désigne le terme à la position p et se définit par récurrence structurelle sur les positions :
  - Si $p = \epsilon$, $t|_\epsilon = t$.
  - Si $p = i \cdot p'$ et $t = f(t_1, \ldots, t_n)$, où l'on a $i \leq n$ et où p' est une position, alors $t|_{i \cdot p'} = t_i|_{p'}$.
- Exemples : si $t = f(x, g(y, z))$, $t|_\epsilon = f(x, g(y, z))$, $t|_1 = x$, $t|_2 = g(y, z)$, $t|_{21} = y$, $t|_{22} = z$.

Substitution à une position donnée

- La notation $t[u]_p$ désigne la substitution de u au terme $t|_p$ dans t.
- Exemple : si $t = f(x, g(y, z))$, $t[h(a)]_{21} = f(x, g(h(a), z))$.

Systèmes de réécriture

Règles de réécriture et systèmes de réécriture

- Une règle de réécriture est une paire $l \to r$ t.q. $FV(r) \subseteq FV(l)$ et l n'est pas une variable ($l \notin \mathcal{V}$).
- Un système de réécriture est un ensemble de règles de réécriture.

Réécriture de termes

- Un terme s se réécrit en t dans un système $\mathcal{R}$, noté $s \to_{\mathcal{R}} t$, si et seulement s'il existe une position p dans t t.q. $s \to_{\mathcal{R},p} t$ est dérivé à partir des règles suivantes :

$$\frac{l \to r \in \mathcal{R}}{\sigma(l) \to_{\mathcal{R},\epsilon} \sigma(r)}\ \text{tête} \qquad \frac{l \to_{\mathcal{R},q} r \text{ et } p \neq \epsilon}{u[l]_p \to_{\mathcal{R},p\cdot q} u[r]_p}\ \text{cont}$$

Systèmes de réécriture

Réécriture de termes (exemple)

- On considère le système $\mathcal{R} = \{f(x, y) \to g(x, x)\}$.
- Comment se réécrit le terme $g(x, g(f(a, b), y))$?
- On a : $g(x, g(f(a, b), y)) \to_{\mathcal{R},21} g(x, g(g(a, a), y))$.

$$\cfrac{\cfrac{\cfrac{f(x, y) \to g(x, x) \in \mathcal{R}}{f(a, b) \to_{\mathcal{R},\epsilon} g(a, a)}\ \text{tête}}{g(f(a, b), y) \to_{\mathcal{R},1} g(g(a, a), y)}\ \text{cont}}{g(x, g(f(a, b), y)) \to_{\mathcal{R},21} g(x, g(g(a, a), y))}\ \text{cont}$$

Systèmes de réécriture

Réécriture de termes (exemple)

- On considère le système $\mathcal{R} = \{f(x,y) \rightarrow g(x,x)\}$.
- Comment se réécrit le terme $g(x, g(f(a,b), y))$?
- On a : $g(x, g(f(a,b), y)) \rightarrow_{\mathcal{R},21} g(x, g(g(a,a), y))$.

$$\cfrac{\cfrac{\cfrac{f(x,y) \rightarrow g(x,x) \in \mathcal{R}}{f(a,b) \rightarrow_{\mathcal{R},\epsilon} g(a,a)}\text{tête}}{g(f(a,b),y) \rightarrow_{\mathcal{R},1} g(g(a,a),y)}\text{cont}}{g(x,g(f(a,b),y)) \rightarrow_{\mathcal{R},21} g(x,g(g(a,a),y))}\text{cont}$$

Systèmes de réécriture

Notions de fermetures

- Un terme s se n-réécrit en t, noté $s \rightarrow^n_{\mathcal{R}} t$, si et seulement si $s = s_0 \rightarrow_{\mathcal{R}} s1 \rightarrow_{\mathcal{R}} \ldots \rightarrow_{\mathcal{R}} s_n = t$.
- $s \rightarrow^+_{\mathcal{R}} t$ si et seulement s'il existe $n > 0$ t.q. $s \rightarrow^n_{\mathcal{R}} t$.
- $\rightarrow^=_{\mathcal{R}}$ est la fermeture réflexive de $\rightarrow_{\mathcal{R}}$.
- $\rightarrow^*_{\mathcal{R}}$ est la fermeture réflexive transitive de $\rightarrow_{\mathcal{R}}$.
- $\leftrightarrow_{\mathcal{R}}$ est la fermeture symétrique de $\rightarrow_{\mathcal{R}}$.
- $\leftrightarrow^*_{\mathcal{R}}$ est la fermeture réflexive, symétrique et transitive de $\rightarrow_{\mathcal{R}}$.
- Le terme t est réducible si et seulement s'il existe s t.q. $t \rightarrow^*_{\mathcal{R}} s$.
- Le terme t est en forme normale si et seulement s'il n'est pas réducible.
- Le terme s est une forme normale de t si et seulement si $t \rightarrow^*_{\mathcal{R}} s$ et s est en forme normale.

Systèmes de réécriture

Notions de terminaison

- L'élément s est faiblement normalisable dans $\mathcal{R}$ si et seulement si s possède au moins une forme normale.
- L'élément s est fortement normalisable dans $\mathcal{R}$ si et seulement s'il n'existe aucune suite infinie $s = s_0 \rightarrow_{\mathcal{R}} s1 \rightarrow_{\mathcal{R}} \ldots$
- Le système $\mathcal{R}$ est faiblement normalisable si et seulement si tout élément est faiblement normalisable dans $\mathcal{R}$.
- Le système $\mathcal{R}$ termine ou est fortement normalisable ou nœthérien ou bien fondé si et seulement si tout élément est fortement normalisable dans $\mathcal{R}$.

Systèmes de réécriture

Techniques pour montrer la terminaison

- Par ordres de réduction (interprétation, ordres polynômiaux).
- Par paires de dépendance.
- Par ordre produit.
- Par ordre lexicographique.
- Par ordre multi-ensemble.
- Par ordres de simplification (ordres récursifs sur les chemins).
- Par ajournement.
- Par projection.

Techniques pour montrer la terminaison

Terminaison par ordres de réduction

- Une relation d'ordre strict est une relation binaire antiréflexive (ou irréflexive), antisymétrique et transitive.
- Un ordre strict $\succ$ sur $\mathcal{T}$ est un ordre de réduction si et seulement si :
  - Chaque symbole $f \in \mathcal{S}_{\mathcal{F}}$ est monotone par rapport à $\succ$.
  - $\succ$ est stable par substitution.
  - $\succ$ est bien fondé.
- Théorème : Un système de réécriture $\mathcal{R}$ termine si et seulement s'il existe un ordre de réduction $\succ$ t.q. $l \succ r$ pour toute règle $l \to r \in \mathcal{R}$.

Techniques pour montrer la terminaison

Interprétation comme ordre de réduction

- On se donne une interprétation I et $\succ_I$ un ordre strict bien fondé sur le domaine D_I de I.
- Définition : la relation $\succ$ sur $\mathcal{T}$ est donnée par : $s \succ t$ si et seulement si $[\![s]\!]^I_\rho \succ_I [\![t]\!]^I_\rho$ pour toute assignation ρ.
- Corollaire : La relation $\succ$ est stable par substitution.
- Théorème : Si pour tout $f \in \mathcal{S}_\mathcal{F}$, l'interprétation $I(f)$ est monotone par rapport à $\succ_I$, alors $\succ$ est un ordre de réduction.

Techniques pour montrer la terminaison

Interprétation comme ordre de réduction

- Exemple : $f(f(x)) \rightarrow f(x)$.
- Interprétation : $D_I = \mathbb{N}$, $I(f)(x) = x + 1$.
- Ordre strict bien fondé sur D_I : $>$ sur $\mathbb{N}$.
- Monotonie de $I(f)$: pour $x, y \in \mathbb{N}$, si $x > y$ alors $I(f)(x) = x + 1 > I(f)(y) = y + 1$.
- Donc : $\succ$ est un ordre de réduction.
- On doit montrer que : $f(f(x)) \succ f(x)$, c'est-à-dire
  $[\![f(f(x))]\!]^I_\rho > [\![f(x)]\!]^I_\rho$.
  $[\![f(f(x))]\!]^I_\rho = I(f)(I(f)(\rho(x))) = \rho(x) + 1 + 1 = \rho(x) + 2$ (1)
  $[\![f(x)]\!]^I_\rho = I(f)(\rho(x)) = \rho(x) + 1$ (2)
  (1) $>$ (2).

Techniques pour montrer la terminaison

Ordres polynomiaux comme ordres de réduction

- Un $\mathbb{N}$-polynôme n-aire P est un polynôme à n arguments $X_1, \ldots, X_n$ ayant des coefficients dans $\mathbb{N}$.
- Une interprétation polynomiale I est définie par :
  - Son domaine D_I est un sous-ensemble de $\mathbb{N}^*$.
  - Pour chaque symbole de fonction $f \in \mathcal{S}_\mathcal{F}$ d'arité n, il existe un $\mathbb{N}$-polynôme n-aire P_f t.q. $I(f)(a_1, \ldots, a_n) = P_f(a_1, \ldots, a_n)$.
- Un $\mathbb{N}$-polynôme n-aire P est complètement monotone si et seulement si toute variable X_i, $i = 1, \ldots, n$, de P apparaît au moins une fois dans la définition de P avec un coefficient non nul.
- Théorème : Soit I une interprétation polynomiale. Si chaque $I(f)$ est spécifiée par un polynôme P_f complètement monotone, alors l'ordre $\succ$ associé à $\succ_I$ est un ordre de réduction.

Techniques pour montrer la terminaison

Ordres polynomiaux comme ordres de réduction

- Exemple (symboles o, s, p) :

$$\left\{\begin{array}{l} p(o,x) \rightarrow x \\ p(s(x),y) \rightarrow s(p(x,y)) \end{array}\right.$$

- Interprétation polynomiale : $D_I = \mathbb{N}^*$.
  $I(o) = 1$, $I(s)(x) = x + 1$, $I(p)(x,y) = 2x + y$.
- Ordre strict bien fondé sur D_I : $>$ sur $\mathbb{N}^*$.
- Tous les polynômes sont complètement monotones, donc $\succ$ associé à $\succ_I$ est un ordre de réduction.

Techniques pour montrer la terminaison

Ordres polynomiaux comme ordres de réduction

- On doit démontrer $l \succ r$ pour toute règle $l \rightarrow r$ du système.
  - $[\![p(o,x)]\!]^I_\rho = 2 \times 1 + \rho(x) = \rho(x) + 2$ (1)
    $[\![x]\!]^I_\rho = \rho(x)$ (2)
    (1) $>$ (2).
  - $[\![p(s(x),y)]\!]^I_\rho = 2 \times (\rho(x) + 1) + \rho(y) = 2 \times \rho(x) + \rho(y) + 2$ (1)
    $[\![s(p(x,y))]\!]^I_\rho = 2 \times \rho(x) + \rho(y) + 1$ (2)
    (1) $>$ (2).

Exercices

Ordres polynomiaux

1. $$f(x, g(y, z)) \to g(f(x, y), f(x, z))$$

2. $$\begin{cases} g(x, g(y, z)) \to g(g(x, y), z) \\ g(g(x, y), z) \to g(y, y) \end{cases}$$

3. $$\begin{cases} g(g(x, y), z) \to g(x, g(y, z)) \\ g(f(x), f(y)) \to f(g(x, y)) \\ g(f(x), g(y, z)) \to g(g(x, y), z) \end{cases}$$