Élimination des quantificateurs sur les corps réels clos en Coq

Cyril Cohen et Assia Mahboubi

INRIA Saclay – Île-de-France
LIX École Polytechnique
INRIA Microsoft Research Joint Centre
cohen@crans.org

3 février 2012

This work has been partially funded by the FORMATH project, nr. 243847, of the FET program within the 7th Framework program of the European Commission.

Élimination des quantificateurs

Étant donné une formule du premier ordre :

$$\forall x, \left(x > 0 \Rightarrow \exists y, (y^2 \leq x \wedge y^5 - y + 3x = 0)\right)$$

Trouver une formule *équivalente* et *sans quantificateurs*

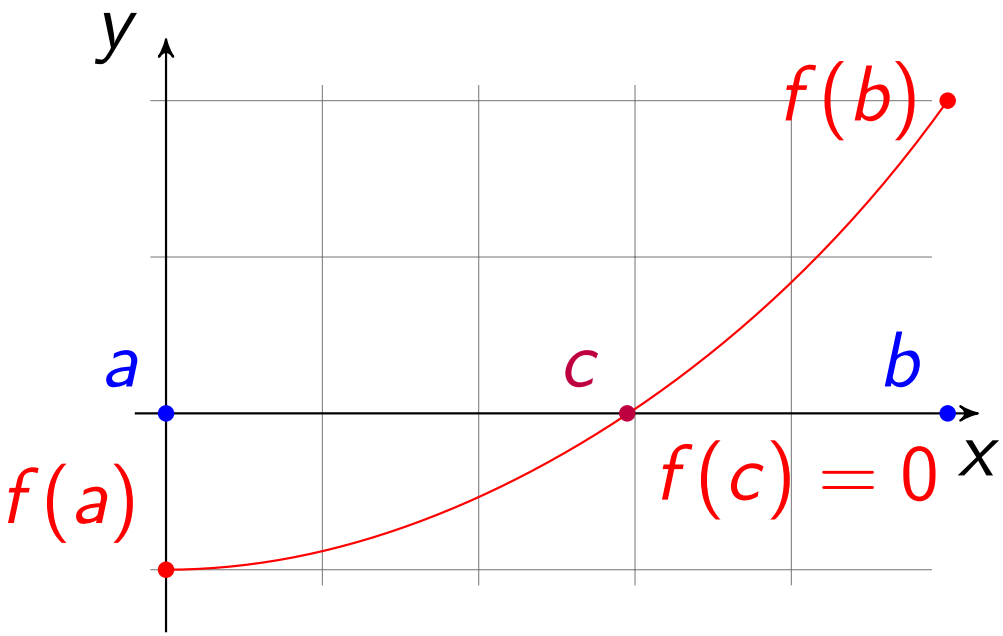
Formule du premier ordre sur le langage des corps réels clos

Les termes : $\bar{x}, \ldots, \bar{0}, \bar{1}, (a \bar{+} b), (\bar{-} a), (a \bar{*} b)$ et $a^{\bar{-}1}$.
Les formules :

- $a \bar{=} b$, $a \bar{<} b$, $\bar{\neg}\phi$, $\phi \bar{\vee} \psi$, $\phi \bar{\wedge} \psi$ et $\phi \bar{\Rightarrow} \psi$
- $\bar{\forall} \bar{x}, \phi(x)$ et $\bar{\exists} \bar{x}, \phi(x)$

Corps réel clos

Corps + ordonné + propriété des valeurs intermédiaires pour les polynômes.



Interprétation

Étant donné R, on interprète termes et formules dans R et dans un environnement ρ :

$$
\begin{aligned}
[\bar{x}]_{R,(\bar{x}\mapsto 1)} &= 1 \\
[a \bar{+} b]_{R,\rho} &= [a]_{R,\rho} + [b]_{R,\rho} \\
R, \rho \models (a \bar{<} b) &= [a]_{R,\rho} < [b]_{R,\rho} \\
R, \rho \models \bar{\exists} \bar{x}, \phi(x) &= \exists a,\ (R, (\rho \cup \bar{x} \mapsto a) \models \phi(x))
\end{aligned}
$$

Deux niveaux de symboles

- L'algorithme opère sur les termes et formules abstrait formés à partir de l'alphabet $\Sigma = \bar{x}, \bar{y}, \bar{0}, \bar{1}, \bar{+}, \bar{-}, \bar{*}, \bar{\cdot}^{-1}, \bar{=}, \bar{<}, \bar{\neg}, \bar{\vee}, \bar{\wedge}, \bar{\Rightarrow}, \bar{\exists}, \bar{\forall}$.
- Sa preuve de correction utilise des symboles "Coq" :
  $x, y, 0, 1, +, -, *, \cdot^{-1}, =, <, \neg, \vee, \wedge, \Rightarrow, \exists, \forall$.

Élimination des quantificateurs

On veut montrer que pour toute formule ϕ,
il existe une formule ψ telle que :

- $\forall R, \forall \rho, \quad (R, \rho \models \phi) \Leftrightarrow (R, \rho \models \psi)$
- ψ est sans quantificateurs,
  i.e. ne contient pas d'occurrence de $\bar{\exists}$ ni de $\bar{\forall}$

Réduction à l'élimination d'un seul quantificateur

En fait il suffit de trouver un algorithme pour éliminer le quantificateur de formules du type :

$$\bar{\exists}\bar{x}, P(\bar{x}) \bar{=} \bar{0} \bar{\wedge} \bar{\bigwedge} Q_i(\bar{x}) \bar{>} \bar{0}$$

Où le P et les Q_i sont des termes n'utilisant que des symboles de Σ sauf $\cdot^{-1}$: ce sont des *polynômes formels*

Côté mathématiques constructives

$$\exists x, P(x) = 0 \wedge \bigwedge Q_i(x) > 0$$

$$\text{avec } P,\ Q_i \in R[X]$$

- trouver les racines de P : OK (procédure d'isolation des racines)
- tester les signes des Q_i : OK

Comptage du nombre de racines

Pour trouver une formule équivalente à

$$\bar{\exists}\bar{x}, P(\bar{x}) \bar{=} \bar{0} \bar{\wedge} \bar{\bigwedge} Q_i(\bar{x}) \bar{>} \bar{0}$$

On va fabriquer une fonction `count_gt0` qui

- prend deux polynômes, trouve le nombre de racines de P qui rendent tous les Q_i positifs et renvoie `true` si ce nombre est strictement positif, et `false` sinon.
- n'utilise que les symboles Coq suivants : $0, 1, +, -, *, =, <, \neg, \vee, \wedge, \Rightarrow$.

Algorithme de comptage

Preuve en deux temps

1. On construit `count_gt0` et on prouve que

$$\texttt{count\_gt0}\ p\ q \Leftrightarrow \exists x, (p(x) = 0) \wedge \bigwedge (q_i(x) > 0)$$

2. On trouve un algorithme `CountGt0` tel que :

$$\begin{aligned}
&\forall R, \forall \rho, \\
&\qquad R, \rho \models \texttt{CountGt0}\ P\ Q \\
\Leftrightarrow &\qquad \texttt{count_gt0}\ [P]_{R,\rho}\ [Q]_{R,\rho} \\
\Leftrightarrow &\qquad \exists x, ([P]_{R,\rho}(x) = 0) \wedge \bigwedge ([Q_i]_{R,\rho}(x) > 0) \\
\Leftrightarrow &\qquad R, \rho \models \left(\bar{\exists} \bar{x}, (P(\bar{x}) \bar{=} \bar{0}) \bar{\wedge} \bar{\bigwedge} (Q_i(\bar{x}) \bar{>} \bar{0})\right)
\end{aligned}$$

Plan de la preuve

1. Théorie mathématique sur le comptage
   - Comptage du nombre de racines
   - Tarski query
   - Indice de Cauchy
   - Variations de signes de la sequence des pseudo-restes
2. Algorithme d'élimination des quantificateurs
   - Construction de l'algorithme
   - Preuve de correction

Plan de la preuve

1. Théorie mathématique sur le comptage
   - Comptage du nombre de racines
   - Tarski query
   - Indice de Cauchy
   - Variations de signes de la sequence des pseudo-restes N'utilise que les opérations d'anneaux
2. Algorithme d'élimination des quantificateurs
   - Construction de l'algorithme
   - Preuve de correction

Théorie mathématiques sur les polynômes

- Support pour interpréter les manipulations de l'algorithme (partie 2)
- On pourrait faire des mathématiques classiques mais Coq est constructif
  $\Rightarrow$ mathématiques constructives
- La complexité des preuves (vues comme algorithmes) n'a pas d'importance

Détails d'un élément de la preuve

On détaille le lien entre les "Tarski queries" et le comptage des racines.

Tarski query

Definition :

$$\mathrm{TQ}(P, Q) = \sum_{x \in \mathrm{roots}(P)} \mathrm{sign}\left(Q(x)\right)$$

Contraintes

On a :

$$\mathrm{TQ}(P, Q) = \sum_{x \in \mathrm{roots}(P)} \mathrm{sign}\left(Q(x)\right)$$

On veut savoir si :

$$\exists x, P(x) = 0 \wedge \bigwedge Q_i(x) > 0$$

Contraintes

On a :

$$\mathrm{TQ}(P, Q) = \sum_{x \in \mathrm{roots}(P)} \mathrm{sign}\left(Q(x)\right)$$

On veut savoir si :

$$\exists x, P(x) = 0 \wedge \bigwedge Q_i(x) > 0$$

c'est-à-dire si

$$\left(\sum_{x \in \mathrm{roots}(P)} [\forall i, Q_i(x) > 0] \right) > 0$$

avec $[\mathrm{true}] = 1$ and $[\mathrm{false}] = 0$

Pour un seul Q_i

$$\exists x, P(x) = 0 \wedge Q(x) > 0$$

On veut :

$$\sum_{x \in \mathrm{roots}(P)} [Q(x) > 0]$$

Pour un seul Q_i

$$\exists x, P(x) = 0 \wedge Q(x) > 0$$

On veut :

$$\sum_{x \in \mathrm{roots}(P)} [\mathrm{sign}\,(Q(x)) = 1]$$

Pour un seul Q_i

$$\exists x, P(x) = 0 \wedge Q(x) > 0$$

On veut :

$$\mathrm{C}^1(P, Q) := \sum_{x \in \mathrm{roots}(P)} [\mathrm{sign}\,(Q(x)) = 1]$$

Pour un seul Q_i

$$\exists x, P(x) = 0 \wedge Q(x) > 0$$

On veut :

$$\mathrm{C}^{\varepsilon}(P, Q) := \sum_{x \in \mathrm{roots}(P)} [\mathrm{sign}\,(Q(x)) = \varepsilon]$$

avec $\varepsilon \in \{1, -1, 0\}$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}(P, Q) = \sum_{x \in \mathrm{roots}(P)} \mathrm{sign}\,(Q(x))$$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}(P, Q) \;=\; \sum_{x \in \mathrm{roots}(P)} \mathrm{sign}\,(Q(x))$$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) \;=\; \sum_{x \in z} \mathrm{sign}\,(Q(x))$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) = \sum_{\substack{x \in z \\ Q(x)>0}} \mathrm{sign}\,(Q(x)) + \sum_{\substack{x \in z \\ Q(x)<0}} \mathrm{sign}\,(Q(x))$$

$$\text{avec } z = \mathrm{roots}(P)$$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) = \sum_{\substack{x \in z \\ Q(x)>0}} 1 + \sum_{\substack{x \in z \\ Q(x)<0}} -1$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) = \sum_{\substack{x \in z \\ Q(x)>0}} 1 - \sum_{\substack{x \in z \\ Q(x)<0}} 1$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) = \sum_{x \in z} [Q(x) > 0] - \sum_{x \in z} [Q(x) < 0]$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) = \sum_{x \in z} [\mathrm{sign}\,(Q(x)) = 1] - \sum_{x \in z} [\mathrm{sign}\,(Q(x)) = -1]$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$\mathrm{TQ}_z(Q) \;=\; \mathrm{C}^1_z(Q) \;-\; \mathrm{C}^{-1}_z(Q)$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$\begin{aligned} \mathrm{TQ}_z(Q) &= \mathrm{C}^1_z(Q) - \mathrm{C}^{-1}_z(Q) \\ \mathrm{TQ}_z(Q^2) &= \mathrm{C}^1_z(Q) + \mathrm{C}^{-1}_z(Q) \end{aligned}$$

avec $z = \mathrm{roots}(P)$

Relation entre TQ et C^{ε}

$$
\begin{array}{rcl}
\mathrm{TQ}_z(Q) & = & \mathrm{C}^1_z(Q) - \mathrm{C}^{-1}_z(Q) \\
\mathrm{TQ}_z(Q^2) & = & \mathrm{C}^1_z(Q) + \mathrm{C}^{-1}_z(Q) \\
\mathrm{TQ}_z(1) & = & \mathrm{C}^1_z(Q) + \mathrm{C}^{-1}_z(Q) + \mathrm{C}^0_z(Q)
\end{array}
$$

$$\text{avec } z = \mathrm{roots}(P)$$

Équation matricielle

$$\begin{pmatrix} \mathrm{TQ}_z(Q) \\ \mathrm{TQ}_z(Q^2) \\ \mathrm{TQ}_z(1) \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{C}_z^1(Q) \\ \mathrm{C}_z^{-1}(Q) \\ \mathrm{C}_z^0(Q) \end{pmatrix}$$

avec $z = \mathrm{roots}(P)$

Équation matricielle

$$\begin{pmatrix} \mathrm{TQ}_z(Q) \\ \mathrm{TQ}_z(Q^2) \\ \mathrm{TQ}_z(1) \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{C}_z^1(Q) \\ \mathrm{C}_z^{-1}(Q) \\ \mathrm{C}_z^0(Q) \end{pmatrix}$$

avec $z = \mathrm{roots}(P)$

$$\begin{vmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix} = 2$$

Équation matricielle

$$\begin{pmatrix} \mathrm{TQ}_z(Q) \\ \mathrm{TQ}_z(Q^2) \\ \mathrm{TQ}_z(1) \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{C}_z^1(Q) \\ \mathrm{C}_z^{-1}(Q) \\ \mathrm{C}_z^0(Q) \end{pmatrix}$$

avec $z = \mathrm{roots}(P)$

$$\begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \text{ est inversible}$$

Pour plusieurs Q_i

On generalise C^{ε} de la même manière :

$$\mathrm{C}^{\varepsilon_1,\ldots,\varepsilon_n}(P, Q_1, \ldots, Q_n) = \sum_{x \in \mathrm{roots}(P)} [\forall i, \mathrm{sign}\,(Q_i(x)) = \varepsilon_i]$$

Système matriciel

$$\begin{pmatrix} \mathrm{TQ}_z(Q_1Q_2) \\ \mathrm{TQ}_z(Q_1^2Q_2) \\ \mathrm{TQ}_z(Q_2) \\ \mathrm{TQ}_z(Q_1Q_2^2) \\ \mathrm{TQ}_z(Q_1^2Q_2^2) \\ \mathrm{TQ}_z(Q_2^2) \\ \mathrm{TQ}_z(Q_1) \\ \mathrm{TQ}_z(Q_2^2) \\ \mathrm{TQ}_z(1) \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{C}_z^{1,1}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,1}(Q_1, Q_2) \\ \mathrm{C}_z^{0,1}(Q_1, Q_2) \\ \mathrm{C}_z^{1,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{0,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{1,0}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,0}(Q_1, Q_2) \\ \mathrm{C}_z^{0,0}(Q_1, Q_2) \end{pmatrix}$$

avec $z = \mathrm{roots}(P)$

Système matriciel

Exemple avec 2 polynômes Q_i

$$\begin{pmatrix} \mathrm{TQ}_z(Q_1Q_2) \\ \mathrm{TQ}_z(Q_1^2Q_2) \\ \mathrm{TQ}_z(Q_2) \\ \mathrm{TQ}_z(Q_1Q_2^2) \\ \mathrm{TQ}_z(Q_1^2Q_2^2) \\ \mathrm{TQ}_z(Q_2^2) \\ \mathrm{TQ}_z(Q_1) \\ \mathrm{TQ}_z(Q_2^2) \\ \mathrm{TQ}_z(1) \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}^{\otimes 2} \begin{pmatrix} \mathrm{C}_z^{1,1}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,1}(Q_1, Q_2) \\ \mathrm{C}_z^{0,1}(Q_1, Q_2) \\ \mathrm{C}_z^{1,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{0,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{1,0}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,0}(Q_1, Q_2) \\ \mathrm{C}_z^{0,0}(Q_1, Q_2) \end{pmatrix}$$

avec $z = \mathrm{roots}(P)$

Système matriciel

Exemple avec 2 polynômes Q_i

$$\begin{pmatrix} \mathrm{TQ}_z(Q_1Q_2) \\ \mathrm{TQ}_z(Q_1^2Q_2) \\ \mathrm{TQ}_z(Q_2) \\ \mathrm{TQ}_z(Q_1Q_2^2) \\ \mathrm{TQ}_z(Q_1^2Q_2^2) \\ \mathrm{TQ}_z(Q_2^2) \\ \mathrm{TQ}_z(Q_1) \\ \mathrm{TQ}_z(Q_2^2) \\ \mathrm{TQ}_z(1) \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}^{\otimes 2} \begin{pmatrix} \mathrm{C}_z^{1,1}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,1}(Q_1, Q_2) \\ \mathrm{C}_z^{0,1}(Q_1, Q_2) \\ \mathrm{C}_z^{1,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{0,-1}(Q_1, Q_2) \\ \mathrm{C}_z^{1,0}(Q_1, Q_2) \\ \mathrm{C}_z^{-1,0}(Q_1, Q_2) \\ \mathrm{C}_z^{0,0}(Q_1, Q_2) \end{pmatrix}$$

avec $z = \mathrm{roots}(P)$

Observations

- Partie difficile à expliquer à un humain
- Pour que la preuve passe, il faut un ordonnancement cohérent des contraintes σ_i et des ε_i
- Pour montrer l'équation matricielle précédente il est nécessaire que l'hypothèse de récurrence soit valable pour tout z et non seulement sur les racines de P.

Plan de la preuve

1. Théorie mathématique sur le comptage
   - Comptage du nombre de racines
   - Tarski query
   - Indice de Cauchy
   - Variations de signes de la sequence des pseudo-restes
2. Algorithme d'élimination des quantificateurs
   - Construction de l'algorithme
   - Preuve de correction

Algorithme d'élimination des quantificateurs

But : trouver la contre-partie formelle `CountGt0` de `count_gt0`

- Trouver une méthode systématique pour trouver la contre-partie formelle d'un programme n'utilisant que des symboles de la théorie des anneaux, de la récursion et le `if`, `then`, `else`
- Prouver la correction de cette transformation

Cas triviaux

Les contreparties formelles des symboles $0, 1, +, -, *, \ldots$ sur R sont respectivement $\bar{0}, \bar{1}, \bar{+}, \bar{-}, \bar{*}, \ldots$

Cas simples

Pour les fonctions n'utilisant pas de conditionnelles, la traduction est directe.
Par exemple pour la contre-partie de la multiplication des polynômes :

```
Fixpoint MulPoly (p q : polyF) :=
  if p is a :: p'
  then AddPoly (map (Mul a) q)
               (Const 0 :: (MulPoly p' q))
  else [::].
```

On a trivialement :

$$\forall R, \forall \rho,\ [\texttt{MulPoly}\ P\ Q]_{R,\rho} = [P]_{R,\rho} * [Q]_{R,\rho}$$

Cas compliqués

Comment traduire la fonction coefficient de tête `lcoef` ?

Dans le code de `LCoef`, on ne peut pas faire de test à zéro. Pour cela il faudrait connaître les valeurs que l'environnement ρ donne aux paramètres.

Solution

```
Fixpoint LCoef (p : polyF)
 (K : term -> formula) : formula :=
  match p with
  | [::] => K 0
  | a :: q => ((q == 0) /\ (K a))
          \/ ((q != 0) /\ (LCoef q K))
  end.
```

K est appelé “continuation”

Correction

Si la continuation K s'interprète en k, alors
`(LCoef _ K)` s'interprète en `(k (lcoef _))`

$$\forall K, \forall k, \quad (\forall a, \forall \rho,\ [K(a)]_\rho = k([a]_\rho)) \Rightarrow\ \forall P, \forall \rho,\ [\texttt{LCoef}\ P\ K]_\rho = k(\texttt{lcoef}\ [P]_\rho).$$

Cas du comptage

La méthode précédente se généralise et les trois cas vus ici permettent de traiter l'ensemble du code de `count_gt0`.

On peut trouver une contre-partie `Count` à `count` telle que

$$\forall K, \forall k, \quad (\forall n \in \mathbb{N}, \forall \rho,\ [K(n)]_\rho = k(n)) \Rightarrow \forall P, \forall \rho,\ [\texttt{Count}\ P\ Q\ K]_\rho = k(\texttt{count}\ [P]_\rho\ [Q]_\rho).$$

Cas du comptage positif

Puis en utilisant $K(n) = (\bar{0} \bar{<} n)$ et $k(n) = (0 < n)$
On a :

$$\forall P, \forall \rho,$$
$$[\texttt{Count}\ P\ Q\ (n \mapsto \bar{0} \bar{<} n)]_\rho = (0 < (\texttt{count}\ [P]_\rho\ [Q]_\rho)).$$

Cas du comptage positif

Puis en utilisant $K(n) = (\bar{0} \bar{<} n)$ et $k(n) = (0 < n)$
On a :

$$\forall P, \forall \rho,$$
$$[\texttt{Count}\ P\ Q\ (n \mapsto \bar{0} \bar{<} n)]_\rho = (\texttt{count\_gt0}\ [P]_\rho\ [Q]_\rho).$$

Cas du comptage positif

Puis en utilisant $K(n) = (\bar{0} \bar{<} n)$ et $k(n) = (0 < n)$
On a :

$$\forall P, \forall \rho,$$
$$[\texttt{CountGt0}\ P\ Q]_\rho = (\texttt{count\_gt0}\ [P]_\rho\ [Q]_\rho).$$

avec $\texttt{CountGt0}\ P\ Q := \texttt{Count}\ P\ Q\ (n \mapsto \bar{0} \bar{<} n)$

Observations

- En utilisant cette méthode de programmation par continuation, on a évité la définition de structures annexes et simplifié l'algorithme.
- La preuve est naturelle à mener : l'interprétation commute avec toutes les opérations que l'on utilise.

Conclusion

- L'algorithme est formalisé en Coq
- Une preuve qui s'appuie sur des concepts mathématiques variés (arithmétique polynomiale, analyse polynomiale réelle, calcul matriciel)
- Deux niveaux de programmation :
  1. Niveau mathématique : jamais exécuté
  2. Niveau algorithmique : exécutable (mais pas efficace dans le cas présent)

Travaux futurs

- Prouver l'algorithme de décomposition algébrique cylindrique (CAD) en réutilisant les mêmes outils.
- L'utiliser pour résoudre des problèmes au "niveau mathématiques"

Fin

Merci de votre attention.

Des questions ?