Typing the higher-order μ-calculus

CAMILLE BONNIN ENCADRÉE PAR M. MME. CINZIA DI GIUSTO ET M. ETIENNE LOZES

CONTEXTE

Le μ-calcul

- Logique introduite par Scott et de Bakker en 1969.
- Etendue par Kozen en 1983 (forme actuelle).

- Sert à faire de la vérification de programmes.

- Permet d'exprimer d'autres logiques temporelles comme LTL ou CTL.

Opérateur de point fixe μ

- Le nom de μ-calcul vient de l'opérateur de point fixe μ.
- Point fixe de g : x tel que x = g(x)
- Théorème de point fixe de Knaster-Tarski : « Soit f une fonction **croissante** d'un treillis complet dans un treillis complet, alors f a un plus petit point fixe »
  - Ne fonctionne qu'avec les fonctions **croissantes**, toutes les formules du μ-calcul n'ont pas forcement de sens.
    - Il faut trier ces formules.

Ordre supérieur

- Variables et formules peuvent être :
  - prédicat $(\bullet)$
  - $\bullet \to \bullet$ (fonction)
  - $(\bullet \to \bullet) \to \bullet$
  - $\bullet \to (\bullet \to \bullet)$
  - $(\bullet \to \bullet) \to (\bullet \to \bullet)$
  - . . .
- $\Rightarrow$ Toutes les variables et formules n'ont pas le même type.
  - Trier les formules en fonctions des compatibilités entre les types de leurs variables via un système de typage.

Objectif du TER

- Trier les formules via le typage.
- Γ = ensemble des triplets (variable libre, type, variance)
- On cherche Γ et le type de la formule (inférence de types).
- Ici on a déjà Δ = ensemble des couples (variable libre, type).
  - Pour trouver Γ, on va compléter Δ avec les variances (inférence de variances).

Le μ-calcul d'ordre supérieur - opérateurs -

- Syntaxe du μ-calcul d'ordre supérieur :

$\Phi, \Psi ::= \top \mid \Phi \wedge \Psi \mid \neg\Phi \mid \langle a \rangle \Phi \mid X \mid \mu X : \tau \,.\, \Phi \mid \lambda X^{v} : \tau \,.\, \Phi \mid \Phi\Psi$ avec :

  - $\top$ ("top") : constante, n'importe quel état ;
  - $\Phi \wedge \Psi$ ("conjonction") : représente le « et » logique ;
  - $\neg\Phi$ ("négation") : négation logique ;
  - $\langle a \rangle \Phi$ ("diamant") : « possible », vraie si après une action a, Φ devient vraie ;
  - $\mu X : \tau \,.\, \Phi$ ("plus petit point fixe") : plus petit point fixe (el tel que el = Φ(el)) ;
  - $\lambda X^{v} : \tau \,.\, \Phi$ ("lambda abstraction") : représente les fonctions ;
  - $\Phi\Psi$ ("application") : application de fonction.

Le μ-calcul d'ordre supérieur - exemples de formules -

$$\langle a \rangle (Y \wedge \neg X)$$

- « Le prédicat qui indique qu'après l'action a, la conjonction des prédicats Y et de la négation de X est vraie. »

$$\mu X : \bullet \,.\, X \wedge Y$$

- « Le plus petit point fixe de f : X → X ^ Y, le plus petit X tel que X = f(X). »
  - Théorème du point fixe : $\mu X : \bullet \,.\, X \wedge Y$ n'est défini que si $\lambda X : \bullet \,.\, X \wedge Y$ est croissante

  => variances

Variances
- variance d'une fonction / variable -

- Etend la notion de monotonie

- **Définition 5** ($\sqcap$-additivité et $\sqcup$-additivité)**.**
  Soient $(A, \sqcap, \sqcup)$ et $(B, \sqcap, \sqcup)$ deux treillis.
  Une fonction $f : A \to B$ est $\sqcap$-additive (resp $\sqcup$-additive) si $\forall x, y \in A^2$, $f(x \sqcap y) = f(x) \sqcap f(y)$ (resp $f(x \sqcup y) = f(x) \sqcup f(y)$).

- Variance d'une variable :
  - formule = fonction de plusieurs variables
  - variance d'une variable = variance de la fonction où on a fixé les autres variables

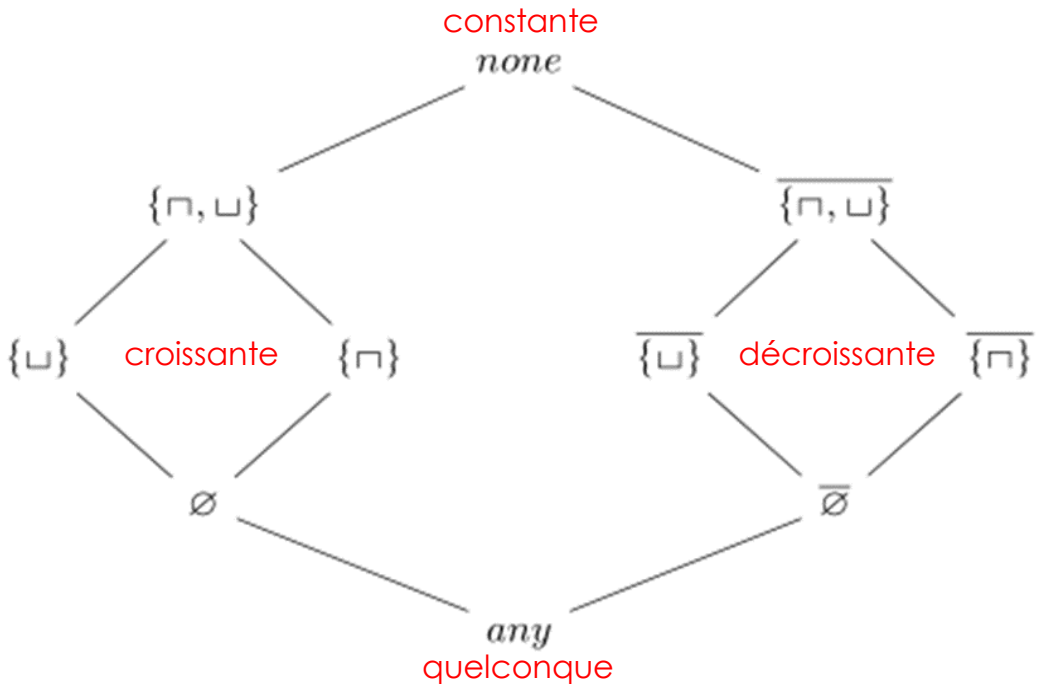

FIGURE 1 – Treillis des variances.

Variances
- exemple -

- Dans $\langle a \rangle (Y \wedge \neg X)$, Y est croissante et X est décroissante.
  - $\mu Y : \bullet \,.\, \langle a \rangle (Y \wedge \neg X)$ est bien défini mais $\mu X : \bullet \,.\, \langle a \rangle (Y \wedge \neg X)$ n'est pas défini.

- La variance de Y est $\{\sqcup\}$:

$$
\begin{aligned}
f(Y_1 \sqcup Y_2) &= \langle a \rangle ((Y_1 \sqcup Y_2) \wedge \neg X) \\
&= \langle a \rangle ((Y_1 \wedge \neg X) \sqcup (Y_2 \wedge \neg X)) \\
&= (\langle a \rangle (Y_1 \wedge \neg X)) \sqcup (\langle a \rangle (Y_2 \wedge \neg X)) \\
&= f(Y_1) \sqcup f(Y_2)
\end{aligned}
$$

($\langle a \rangle (A \sqcup B) = (\langle a \rangle A) \sqcup (\langle a \rangle B)$)

CONTRIBUTION

Inférence de variances - problème -

- On utilise les notation suivantes :
  - Δ : ensemble de couples (variable libre, type) (environnement de typage sans variances)
  - Γ : ensemble de triplets (variable libre, variance, type) (environnement de typage)
  - τ : type
  - Φ : formule

- **Problème** :
  - On connaît Δ et on a une formule Φ, on cherche à compléter Δ avec les variances pour obtenir Γ.
  - On cherche aussi le type de Φ.

Inférence de variances - solution proposée -

- Solution : on définit une fonction récursive *type*(Φ, Δ) = (Γ, τ) où τ est le type de Φ.
  - On définit *type*(.) à l'aide de règles d'inférence.

- Avec la formule Φ suivante : $\langle a \rangle (Y \wedge \neg X)$, on aura :
  - $\Delta = \{\{Y : \bullet\}, \{X : \bullet\}\}$;
  - $\Gamma = \{\{Y^{\{\sqcup\}} : \bullet\}, \{X^{\overline{\{\sqcap\}}} : \bullet\}\}$;
  - $\tau = \bullet$.

Inférence de variances
- règles d'inférence de variances -

Définition 18 (Règles de typage)**.** Les règles d'inférence de variances du μ-calcul d'ordre supérieur sont les suivantes :

$$\frac{}{type(\Delta, \top) = (\emptyset, \bullet)}(\text{T-TOP}) \qquad \frac{type(\Delta, \Phi) = (\Gamma_1, \bullet) \qquad type(\Delta, \Psi) = (\Gamma_2, \bullet)}{type(\Delta, \Phi \wedge \Psi) = (\Gamma_1 \wedge \Gamma_2, \bullet)}(\text{T-ET})$$

$$\frac{type(\Delta, \Phi) = (\Gamma, \tau)}{type(\Delta, \neg\Phi) = (\overline{\{\sqcap, \sqcup\}} \circ \Gamma, \tau)}(\text{T-NEG}) \qquad \frac{type(\Delta, \Phi) = (\Gamma, \bullet)}{type(\Delta, \langle a \rangle \Phi) = (\{\sqcup\} \circ \Gamma, \bullet)}(\text{T-DIAMANT})$$

$$\frac{(X : \tau) \in \Delta}{type(\Delta, X) = (X^{\{\sqcap, \sqcup\}} : \tau, \tau)}(\text{T-VAR})$$

$$\frac{type(\Delta \cup \{X : \tau\}, \Phi) = (\Gamma, \sigma) \qquad \sigma = \tau \qquad \Gamma = \Gamma' \cup \{X^v : \sigma\} \qquad v \succeq \varnothing}{type(\Delta, \mu X : \tau \,.\, \Phi) = (\Gamma', \tau)}(\text{T-MU})$$

$$\frac{type(\Delta \cup \{X : \sigma\}, \Phi) = (\Gamma, \tau) \qquad \Gamma = \Gamma' \cup \{X^{v'} : \sigma'\} \qquad v \preccurlyeq v'}{type(\Delta, \lambda X^v : \sigma \,.\, \Phi) = (\Gamma', \sigma'^v \to \tau)}(\text{T-LAMBDA})$$

$$\frac{type(\Delta, \Phi) = (\Gamma_1, \sigma^v \to \tau) \qquad type(\Delta, \Psi) = (\Gamma_2, \sigma)}{type(\Delta, \Phi\Psi) = (\Gamma_1 \wedge v \circ \Gamma_2, \tau)}(\text{T-APP})$$

Inférence de variances - règle du μ -

$$\frac{type(\Delta \cup \{X : \tau\}, \Phi) = (\Gamma, \sigma) \qquad \sigma = \tau \qquad \Gamma = \Gamma' \cup \{X^{v} : \sigma\} \qquad v \geq \varnothing}{type(\Delta, \mu X : \tau . \Phi) = (\Gamma', \tau)} (\text{T-MU})$$

- Type du résultat = type de X = type de Φ. (X = Φ(X))
- On rajoute X dans les environnements de typage pour typer X et Φ mais on l'enlève du résultat car X est liée.
- La variance de X (= variance de f : X → Φ(X)) doit être croissante pour respecter le théorème du point fixe.

Inférence de variances - implémentation -

- On a réalisé une implémentation de la fonction *type*(.) en OCaml (langage appris pour l'occasion).
  - utilisation des listes associatives ;
  - utilisation de la récurrence ;
  - code structuré en plusieurs modules (variance, variance_syntaxe, mu-calcul, mu-calcul_syntaxe) ;
  - détection et localisation des échecs de typage ;
  - résultats de tous les tests obtenus immédiatement.

Inférence de variances - résultats de l'implémentation -

- Extrait des résultats des tests (28 formules testées) :
  - cas de bases ;
  - quelques formules plus compliquées.
- Tests de cas positifs et négatifs.

f	Δ	Γ	τ
$(\mu F : (\bullet^{\overline{\varnothing}} \to \bullet) \,.\, \lambda X^{\overline{\varnothing}} : \bullet \,.$ $\langle a \rangle (Y \wedge (F \neg (FX)))) \; [b]Y$	$\{\{Y : \bullet\}\}$	$\{\{Y^{any} : \bullet\}\}$	$\bullet$
$(\lambda X^{\varnothing} : \bullet \,.\, X) \wedge X$	$\{\{X : \bullet\}\}$	Erreur dans $\wedge$: $\lambda X^{\varnothing} : \bullet \,.\, X$ n'a pas le type $\bullet$!	
$\mu X : \bullet \,.\, ((\lambda Y^{\overline{\varnothing}} : \bullet \,.\, \neg Y) X))$	$\emptyset$	Erreur dans μ : X a la variance $\overline{\varnothing}$ qui n'est pas $\geqslant \varnothing$!	
$(\mu F : (\bullet^{\overline{\varnothing}} \to \bullet) \,.$ $\lambda X^{\overline{\varnothing}} : \bullet \,.\, F(\neg (FX))$	$\emptyset$	$\emptyset$	$(\bullet^{\overline{\varnothing}} \to \bullet)$
$\mu X : \bullet \,.\, [a]X$	$\emptyset$	$\emptyset$	$\bullet$

Conclusion
- résumé -

- TER basé un article de Lange, Lozes et Guzmán de 2014, « Model-checking process equivalences ».
- But : résoudre un sous problème de l'inférence de types, l'inférence de variances.
- Solution : une fonction récursive, *type*(.) définie par des règles d'inférence.
- Inconvénient de la solution : la présence d'indications de variances dans les types des variables présentes dans Δ.

Conclusion
- améliorations possibles -

- Améliorations possibles de la solution proposée:
  - supprimer ces d'indications de variances ;
  - supprimer Δ et résoudre le problème d'inférence de types ;
  - rajouter l'aspect polyadique (permet la manipulations de tupples d'états).