Regional Logic[1] et Dynamic Frames[2]

Romain Bardou

12 Juin 2008

[1] Anindya Banerjee, David A. Naumann, et Stan Rosenberg
[2] Ioannis T. Kassios

Alias de pointeurs

Pointeurs x et y.

```
*x = 1;
*y = 2;
assert *x == *y;
```

$x = y$?

Invariants (1/2)

Exemples :

- $x \neq 0$
  (pas de division par 0)
- $c =$ longueur de t
  (pas d'accès hors du tableau)
- s est un arbre de recherche
  (recherche d'un seul côté)
- s est arbre équilibré
  (recherche efficace)
- ...

Invariants (2/2)

Alias de pointeurs : invariants rompus accidentellement ?

```
*x = 1;
assert invariant(y);
```

Modularité

- Rafinement
- Abstraction
- Variables de spécification
- Preuves modulaires : contexte restreint

Problème : *aliasing abstrait*

Solutions abordées

Problème : raisonner sur des ensembles de pointeurs.

Solutions *dynamiques* :

- *cadres* dynamiques,
- logique de *régions*.

Variables de région qui évoluent.

Définition

Cadre dynamique : variable de spécification contenant un ensemble de locations allouées

Prédicats :

- $\Xi(f)$: les variables de f sont *préservées*
- $\Delta(f)$: seules les variables de f sont *modifiées*
- $\Lambda(f)$: *swinging pivots requirement* : f n'est agrandi que par des locations fraîches
- $disjoint(f, g)$

Encadrement (framing)

"Préserver les variable de f préserve v" :

$$f \ \textbf{frames} \ v \quad \stackrel{\text{def}}{=} \quad \Xi(f) \Rightarrow v' = v$$

"x et y sont indépendantes" :

$$f \ \textbf{frames} \ x \ \wedge \ g \ \textbf{frames} \ y \ \wedge \ \mathit{disjoint}(f, g)$$

Propriété

$$f \ \textbf{frames} \ x \ \wedge \ g \ \textbf{frames} \ y \ \wedge \ \mathit{disjoint}(f, g) \ \wedge \ \Delta(f) \implies y' = y$$

Auto-encadrement

Pour préserver $disjoint(f, g)$:

$$\left.\begin{array}{r} f \ \textbf{frames} \ f \\ g \ \textbf{frames} \ g \\ disjoint(f, g) \\ \Delta(f) \\ \Lambda(f) \end{array}\right\} \Longrightarrow disjoint(f, g)$$

Affectation abstraite

Soit x une variable de spécification telle que :

$$f \textbf{ frames } (f, x, y, z, ...)$$

Définition

$$x := E \quad \stackrel{\text{def}}{=} \quad \begin{cases} \Delta(f) \\ \Lambda(f) \\ x' = E \\ y' = y \\ z' = z \\ ... \end{cases}$$

Exemple (1/3)

Composant *List* (interface) :

$$\begin{array}{l} \textbf{spec var } L \in \mathbb{Z}^* \\ \textbf{spec var } rep \\ rep \textbf{ frames } (rep, L) \end{array}$$

Implémentation :

$$\begin{array}{l} \textbf{type } list = (Nil \mid Cons \textbf{ of } \mathbb{Z} \times list) \textbf{ ref} \\ \textbf{let } nth = ... \textbf{ and } length = ... \textbf{ and } last = ... \\ \\ \textbf{var } head : list \\ L = \lambda i.\ nth\ i\ head \\ rep = \{\mathrm{nth}\ i\ head \mid 0 \leq i < \mathrm{length}\ head\} \end{array}$$

Exemple (2/3)

Insertion au début (interface) :

$$\textbf{method } insert(x) \textbf{ ensures } L := [x]^\frown L$$

L est encadrée par *rep* : cette affectation ne modifie que L et *rep*.

Implémentation :

$$insert(x) = head \leftarrow Cons(x, head)$$

Vérifie bien la spécification, en particulier $\Delta(rep)$.

Exemple (3/3)

Concaténation au début (interface) :

$$\begin{array}{l}\textbf{method } \mathit{concat}(l) \\ \quad \textbf{requires } \mathit{disjoint}(\mathit{rep}, l.\mathit{rep}) \\ \quad \textbf{ensures } \left\{\begin{array}{l} L' = l^\frown L \\ \Delta(\mathit{rep} \cup l.\mathit{rep}) \\ \mathit{rep}' = \mathit{rep} \cup l.\mathit{rep} \end{array}\right.\end{array}$$

Le composant courant “avale” le *rep* de *l*.

Implémentation :

$$\begin{array}{l}\mathit{concat}(l) = \\ \quad \text{last } l \leftarrow !\mathit{head}; \\ \quad \mathit{head} \leftarrow !l\end{array}$$

Régions (1/2)

Région : ensemble fini de références

Variables de région :

- Variables *ghosts* de type **rgn**
  - En écriture seule dans le code
  - Utilisable dans les spécifications
- Utilisables dans les effets

Régions (2/2)

Construction :

- Région vide **emp**
- Singleton $\langle E \rangle$
- Toutes les références allouées **all** (variable)

Opérations :

- Union $R_1 \cup R_2$
- Intersection $R_1 \cap R_2$
- Différence $R_1 - R_2$

Prédicats :

- Inclusion $R_1 \subseteq R_2$
- Disjoint $R_1 \# R_2$

Encadrement

"Si θ, alors θ' ne dépend que des références lues par les effets $\bar{\epsilon}$" :

$$\theta \vdash \bar{\epsilon} \ \mathbf{frm} \ \theta'$$

Séparateur

"Ce que $\bar{\epsilon}_r$ lit, $\bar{\epsilon}_w$ ne peut pas modifier" :

$$\bar{\epsilon}_r \star \bar{\epsilon}_w$$

Proche de l'opérateur $*$ de la logique de séparation

Règle d'encadrement

$$\frac{\vdash \{\theta\}\ C\ \{\theta'\}[\bar{\epsilon}_C] \qquad \theta \vdash \bar{\epsilon}_\psi\ \textbf{frm}\ \psi \qquad \theta \Rightarrow \bar{\epsilon}_\psi \star \bar{\epsilon}_C}{\vdash \{\theta \wedge \psi\}\ C\ \{\theta' \wedge \psi\}[\bar{\epsilon}_C]}$$

Exemple (1/5)

```
type node = {
    item : int ;
    left : node ;
    right : node } ref

let setLeftZero x =
    var y : node in y := x.left ; y.item := 0
```

Exemple (2/5)

$$\theta \stackrel{\text{def}}{=} x \neq \textbf{null} \wedge x.\mathit{left} \in r_1 \wedge x.\mathit{right} \in r_2 \wedge r_1 \# r_2 \wedge \mathit{closed}$$

$$\mathit{closed} \stackrel{\text{def}}{=} r_1.\mathit{left} \subseteq r_1 \wedge r_1.\mathit{right} \subseteq r_1 \wedge r_2.\mathit{left} \subseteq r_2 \wedge r_2.\mathit{right} \subseteq r_2$$

$$\psi \stackrel{\text{def}}{=} \forall x : \mathit{node} \in r_2,\ x.\mathit{item} > 0$$

Spécification :

$$\{\theta \wedge \psi\}\ \text{setLeftZero}\ \{\psi\}[\textbf{wr}\ r_1.\mathit{item}]$$

Exemple (3/5)

$$\{x \neq \mathbf{null}\}\ y := x.\mathit{left}\ \{y = x.\mathit{left}\}[\mathbf{wr}\ \ y]$$

Encadrement de θ :

$$\vdash x, r_1, r_2, \langle x\rangle.\mathit{left}, r_1.\mathit{left}, r_2.\mathit{left}, \langle x\rangle.\mathit{right}, r_1.\mathit{right}, r_2.\mathit{right}\ \mathbf{frm}\ \theta$$

Encadrement de ψ :

$$\vdash r_2, r_2.\mathit{item}\ \mathbf{frm}\ \psi$$

Règle d'encadrement avec $\theta \wedge \psi$:

$$\{\theta \wedge \psi\}\ y := x.\mathit{left}\ \{y = x.\mathit{left} \wedge \theta \wedge \psi\}[\mathbf{wr}\ \ y]$$

Exemple (4/5)

$$\{x \neq \textbf{null}\}\ y.item := 0\ \{y.item = 0\}[\textbf{wr}\ \langle y\rangle.item]$$

L'effet dépend de la valeur courante de y.

Rappel :

$$\vdash r_2, r_2.item\ \textbf{frm}\ \psi$$

Donc pour encadrer avec ψ il faut d'abord montrer $\langle y\rangle\#r_2$.

Exemple (5/5)

Pour combiner la séquence $y := x.\mathit{left}; y.\mathit{item} := 0$ il faut encore quelques :

- Encadrements
- Affaiblissements des pré- et post-conditions

En particulier, on ne peut combiner directement l'effet **wr** $\langle y \rangle.\mathit{item}$: il faut l'affaiblir en $r_1.\mathit{item}$.

Conclusion

Bonus :

- Expressif (pas de limitation statique)
- Générique (permet d'encoder des systèmes existants)
- Simple (pour les cadres dynamiques du moins)

Malus :

- Verbeux
- Bas niveau

Références

Cadres dynamiques

I. T. Kassios.
Dynamic frames : Support for framing, dependencies and sharing without restrictions (FM'06)

Logique de régions

A. Banerjee, D. A. Naumann, et S. Rosenberg.
Regional logic for local reasoning about global invariants (ECOOP'08)

Les cadres dynamiques aiment le café

J. Smans, B. Jacobs, F. Piessens, et W. Schulte.
An automatic verifier for Java-like programs based on dynamic frames (FASE'08)