



Formalisation d'une analyse de région pour Framac/WP

Jérémy Damour^{1,2} Allan Blanchard¹ Loïc Correnson¹ Frédéric Loulergue²

¹CEA List ²Université d'Orléans



Table des matières



1. Problématique
2. Cartes de régions
3. Analyse de région
4. Correction



1. Problématique

Problématique

```
void f(short x, int y) {  
    x = 1;  
    short *p = (short*)&y;  
    p[0] = 2;  
    p[1] = 3;  
}
```

■ Non supporté par *Frama-C* / *WP*.

Problématique

```
void f(short x, int y) {  
    x = 1;  
    short *p = (short*)&y;  
    p[0] = 2;  
    p[1] = 3;  
}
```

- Non supporté par *Frama-C / WP*.
- Nécessite une analyse d'**alias** et de **forme**.

Problématique

```
void f(short x, int y) {  
    x = 1;  
    short *p = (short*)&y;  
    p[0] = 2;  
    p[1] = 3;  
}
```

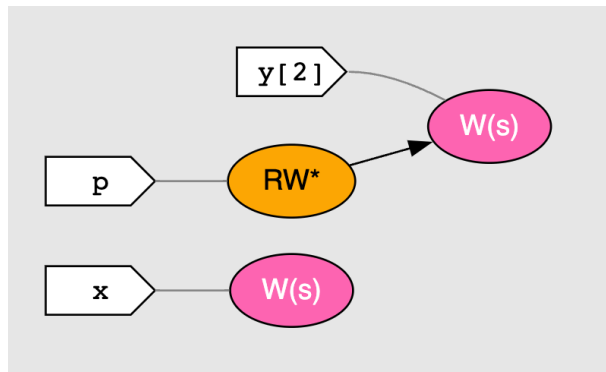
- Non supporté par *Frama-C / WP*.
- Nécessite une analyse d'**alias** et de **forme**.
- Projet ANR CoMeMoV.



2. Cartes de régions

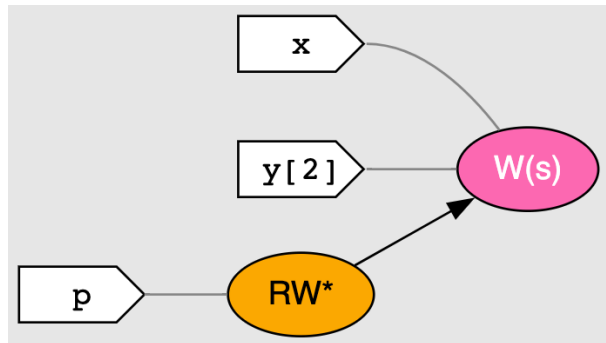
Carte de régions (pointeur)

```
void f(short x, int y) {  
    x = 1;  
    short *p = (short*)&y;  
    p[0] = 2;  
    p[1] = 3;  
}
```



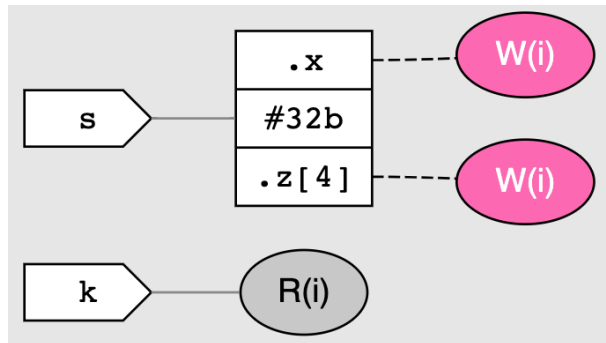
Carte de régions (pointeur alias)

```
void g(short x, int y) {  
    short *p = &x;  
    *p = 1;  
    p = (short *)&y;  
    p[0] = 0;  
    p[1] = 1;  
}
```



Carte de régions (agrégats)

```
void f(int k) {  
    struct {  
        int x, y, z[4];  
    } s;  
    s.x = 1;  
    s.z[k] = 0;  
}
```



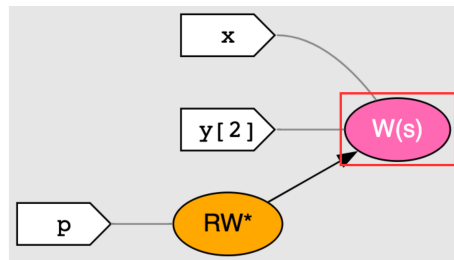
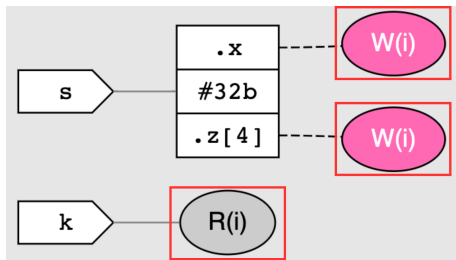
Qu'est-ce qu'une carte de régions?

- Un **graphe**, acyclique en l'absence de pointeur.
- Contient des **étiquettes** qui sont les **variables du programme**.

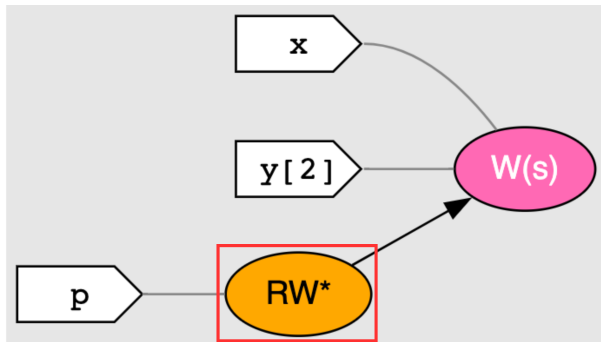
Qu'est-ce qu'une carte de régions?

- Un **graphe**, acyclique en l'absence de pointeur.
- Contient des **étiquettes** qui sont les **variables du programme**.
- Une région est caractérisée par :
 - Une **taille**.
 - Les informations sur le type des **accès en mémoire** (**lecture** et/ou **écriture**).
 - Un **contenu**.

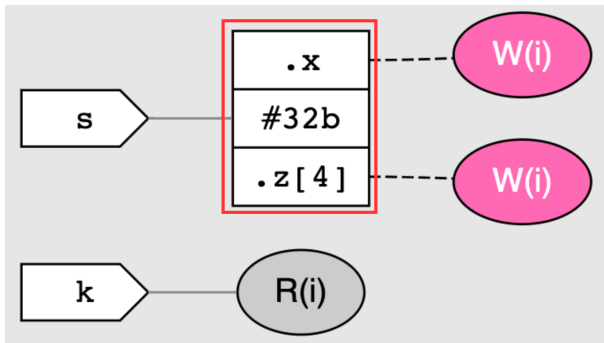
Région primitive



Région pointeur



Région agrégat





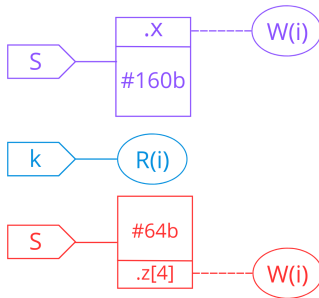
3. Analyse de région

Que fait une analyse de région?

- Construit une **carte de régions**.
- **Cartographie** les accès en mémoire.
- Analyse les **expressions**, **l-values** et **affectations** d'un programme C.
- C'est une **analyse** d'**alias** (*alias analysis*) et de **forme** (*shape analysis*).

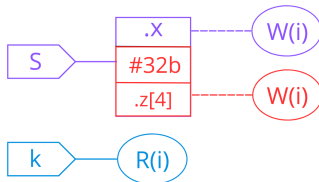
Fusion de régions (agrégat)

```
void f(int k) {  
    struct {  
        int x, y, z[4];  
    } s;  
    s.x = 1;  
    s.z[k] = 0;  
}
```



Fusion de régions (agrégat)

```
void f(int k) {  
    struct {  
        int x, y, z[4];  
    } s;  
    s.x = 1;  
    s.z[k] = 0;  
}
```



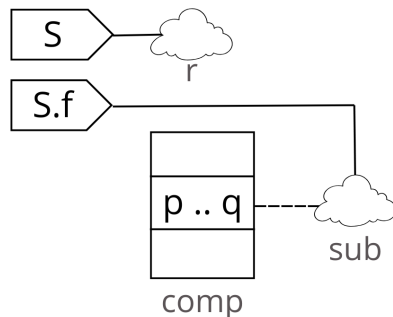
Implémentation Coq / Rocq



- Nécessite une **relation d'équivalence** entre les régions.
- Utilisation d'un **UnionFind**.
- **Monade** d'état et d'erreur.

Analyse d'un accès champs

```
Fixpoint analyse_expr (e : expression) := ...
end with analyse_lval (l : lval) :=
  match l with
  | L_struct s f =>
    let* r := analyse_lval s in
    let*  $\tau$  := typeof s in
    let* (p, q) := offset_of  $\tau$  f in
    let* (comp, sub) := new_range  $\tau$  p q in
    merge r comp;;
    ret sub
  ...
end.
```





4. Correction

Correction de l'analyse (l-value)

Théorème

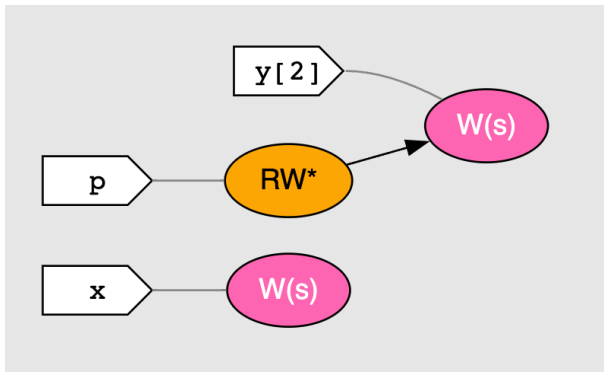
$$G \vdash l : r \rightarrow \text{analyse_lval } l' \ G = \text{Return } r' \ G' \rightarrow G' \vdash l : r \wedge G' \vdash l' : r'$$

- $G \vdash l : r$: La l-value l est localisée dans la région r .

Empreinte mémoire

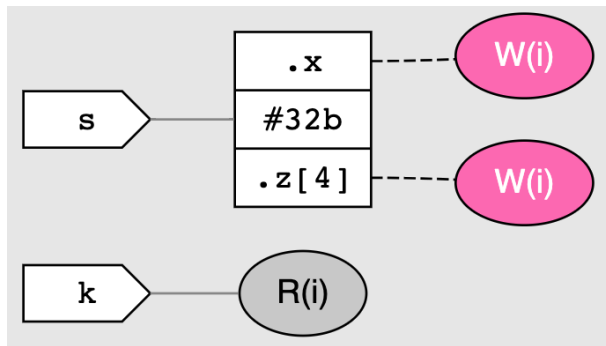
Définition

$$G \vdash a \in r$$



Théorème

$$G \vdash r_1 \stackrel{*}{\mapsto} r_2 \vee G \vdash r_2 \stackrel{*}{\mapsto} r_1 \vee G \vdash r_1 \# r_2$$



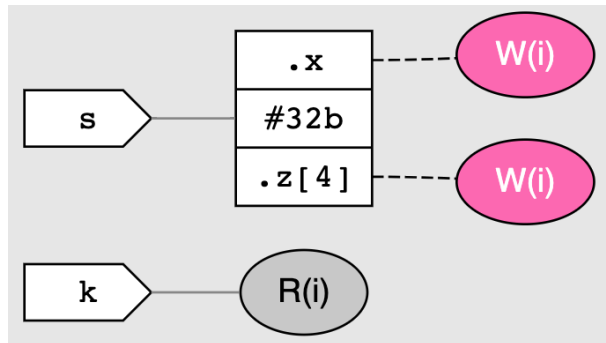
Représentation (l-value)

Théorème

$$G \vdash l : r \rightarrow \llbracket l \rrbracket_m = \lfloor a \rfloor \rightarrow G \vdash m \rightarrow G \vdash a \in r$$

- $G \vdash a \in r$: L'adresse a est dans la région r .
- $G \vdash l : r$: La l-value l est localisée dans la région r .
- $\llbracket l \rrbracket_m = \lfloor a \rfloor$: La l-value l s'évalue dans la mémoire m en une adresse a .
- $G \vdash m$: La mémoire m est cohérente avec la carte de régions G .

```
void f(int k) {  
    struct {  
        int x, y, z[4];  
    } s;  
    s.x = 1;  
    s.z[k] = 0;  
}
```



Développement Coq / Rocq



Formalisation en *Coq / Rocq*

- Plus de **6000 lignes**.
- Utilisation d'une **monade d'état et d'erreur**.
- Preuve avec des **tripplets** de Hoare.
- Définitions de nouvelles **notations** et **tactiques**.

Travaux futurs

- Formalisation de la fonction de **fusion des régions**.
- Formalisation des **gardes**.
- Ajout des **appels de fonctions**.



**Merci
de votre attention !**

