

Génération automatique de tests

Laura Lowenthal

Quotient au Val d'Ajol - février 2008



Le but

Mais qu'est-ce que tu entends par génération automatique ?

Nouvelle option pour mocac

```
mocac -test exemple.mlm → exemple.mli  
                           exemple.ml  
                           exemple_test.ml
```

Entrée

- Ensemble de déclarations de types
 - Juste les types *variants* privés
 - Définitions des constructeurs, leur type, leurs relations
- Nom du module, options, etc.

Sortie

Code de test, à exécuter avec le module `Gentest` de Moca

Exemple

sequence.mlm

```
(* The type of polymorphic sequences of elements.  
   Repetition is allowed. Order is relevant. *)
```

```
type 'a t = private  
  | Empty  
  | Element of 'a  
  | Concat of 'a t * 'a t  
begin  
  associative  
  neutral (Empty)  
end  
  
;;
```



Exemple

sequence_test.ml

```
open Sequence;;
open Gentest;;

testing "Sequence (automatic)";;

testi 0 (let x = element 21 in
         concat (x, empty) = x)
...
testi 5
(let x = element 5 in
 let y = concat (element 60, element 47) in
 let z = empty in
 concat (concat (x, y), z) =
         concat (x, concat (y, z)))
```



La procédure générale

Comment fait-on ?

```
testi 5
(let x = element 5 in
let y = concat (element 60, element 47) in
let z = empty in
concat (concat (x, y), z) =
        concat (x, concat (y, z)))
```



La procédure générale

Comment fait-on ?

```
testi 5
(let x = element 5 in
 let y = concat (element 60, element 47) in
 let z = empty in
 concat (concat (x, y), z) =
         concat (x, concat (y, z)))
```

Plusieurs éléments à générer

- Des équations



La procédure générale

Comment fait-on ?

```
testi 5
(let x = element 5 in
let y = concat (element 60, element 47) in
let z = empty in
concat (concat (x, y), z) =
        concat (x, concat (y, z)))
```

Plusieurs éléments à générer

- Des équations
- Des substitutions



La procédure générale

Comment fait-on ?

```
testi 5
(let x = element 5 in
 let y = concat (element 60, element 47) in
 let z = empty in
 concat (concat (x, y), z) =
         concat (x, concat (y, z)))
```

Plusieurs éléments à générer

- Des équations
- Des substitutions
- Des valeurs



Génération d'équations

Pas varyadiques : `eqnrel.ml`

`eqnset_of_rels` : `generator` \rightarrow `relations` \rightarrow `EqnSet`

puisque concat est associative :

$$\text{concat} (\text{concat} (x, y), z) = \text{concat} (x, \text{concat} (y, z))$$


Génération d'équations

Pas varyadiques : eqnrel.ml

```
eqnset_of_rels : generator → relations → EqnSet
```

puisque concat est associative :

```
concat (concat (x, y), z) =
                concat (x, concat (y, z))
```

Varyadiques : genr_varyadic_equations.ml

```
testing_eqns_of_rels : gen → rels → eqn list
```

puisque concat est associative :

```
concat [concat [x; y]; z] = concat [x; y; z]
concat [x; concat [y; z]; concat [w; x1]] =
                concat [x; y; z; w; x1]
concat [concat [x; y; z]; w] =
                concat [x; y; z; w]
```

Génération d'équations

Pas varyadiques : eqnrel.ml

```
eqnset_of_rels : generator → relations → EqnSet
```

puisque concat est associative :

```
concat (concat (x, y), z) =
                concat (x, concat (y, z))
```

Varyadiques : genr_varyadic_equations.ml

```
testing_eqns_of_rels : gen → rels → eqn list
```

puisque concat est associative : **équations ad-hoc !**

```
concat [concat [x; y]; z] = concat [x; y; z]
concat [x; concat [y; z]; concat [w; x1]] =
                concat [x; y; z; w; x1]
concat [concat [x; y; z]; w] =
                concat [x; y; z; w]
```

Génération de substitutions

Du général au particulier

Une fois qu'on a une équation

- La structure est très simple : variables, application
- Les variables ne sont pas typées



Génération de substitutions

Du général au particulier

Une fois qu'on a une équation

- La structure est très simple : variables, application
- Les variables ne sont pas typées

Il faut

- Détecter les variables des équations **et leurs types**
- Générer le nombre nécessaire de valeurs de chaque type
- Construire une ou plusieurs associations entre les variables et les valeurs

Génération de substitutions

Du général au particulier

Une fois qu'on a une équation

- La structure est très simple : variables, application
- Les variables ne sont pas typées

Il faut

- Détecter les variables des équations **et leurs types**
→ `typed_vars.ml`
- Générer le nombre nécessaire de valeurs de chaque type
- Construire une ou plusieurs associations entre les variables et les valeurs

Génération de substitutions

Du général au particulier

Une fois qu'on a une équation

- La structure est très simple : variables, application
- Les variables ne sont pas typées

Il faut

- Détecter les variables des équations **et leurs types**
→ `typed_vars.ml`
- Générer le nombre nécessaire de valeurs de chaque type
→ `genr_values.ml`
- Construire une ou plusieurs associations entre les variables et les valeurs

Génération de substitutions

Du général au particulier

Une fois qu'on a une équation

- La structure est très simple : variables, application
- Les variables ne sont pas typées

Il faut

- Détecter les variables des équations **et leurs types**
→ `typed_vars.ml`
- Générer le nombre nécessaire de valeurs de chaque type
→ `genr_values.ml`
- Construire une ou plusieurs associations entre les variables et les valeurs
→ `genr_substitutions.ml`

Génération de valeurs

`genr_values.ml`

- Génération de valeurs d'un type déterminé
- Arguments
 - Le type
 - Le nombre
 - La profondeur maximale
 - Aléatoires ou pas
- Parcours récursif de la structure du type
- Substitution de variables par des instances



À faire

- À améliorer
 - S'il n'y a pas assez des valeurs pour générer toutes les substitutions, répéter
 - Reconnaître le type de quelques constructeurs connus (`[]`, `::`)
 - Équations varyadiques aléatoires au lieu de fixes
 - Paramétrer et/ou faire aléatoire la taille des listes

À faire

- À améliorer
 - S'il n'y a pas assez des valeurs pour générer toutes les substitutions, répéter
 - Reconnaître le type de quelques constructeurs connus (`[]`, `::`)
 - Équations varyadiques aléatoires au lieu de fixes
 - Paramétrer et/ou faire aléatoire la taille des listes
- À rajouter ?
 - Indications de l'utilisateur
 - Diségalités

C'est tout !

- Des questions ?
- Envie de voir du code ?