

# TD numéro 2

## Modules, Makefile, Tests

Programmation avancée, ENSIIE

Semestre 2, 2010–11

### Exercice 1

On souhaite développer un navigateur web.

1. Proposer un découpage en modules, en donnant les dépendances entre modules. Esquisser l'interface de chacun des modules.
2. Écrire le fichier `Makefile` correspondant.

### Exercice 2

On rappelle la spécification algébrique d'un dictionnaire :

```
type ('k,'v) dict
```

```
creer : int -> dict
```

```
inserer : dict -> 'k -> 'v -> dict
```

```
rechercher : dict -> 'k -> ('v | ERR)
```

```
supprimer : dict -> 'k -> (dict | ERR)
```

```
rechercher(creer(i), k) = ERR
```

```
rechercher(inserer(d,k,v),k) = v
```

```
rechercher(inserer(d,k,v),k') = rechercher(d,k')  $k \neq k'$ 
```

```
rechercher(supprimer(d,k),k) = ERR
```

```
rechercher(supprimer(d,k),k') = rechercher(d,k')  $k \neq k'$ 
```

On suppose qu'on ne rajoute pas de clef préexistante dans un dictionnaire.

1. Quelles sont les trois façons de construire un dictionnaire ? En déduire toutes les formes de dictionnaire utilisant au plus trois de ces méthodes.
2. Étant donné un dictionnaire quelconque `d`, quels tests faut-il lui faire subir pour vérifier que la spécification algébrique est respectée ?
3. Écrire un algorithme permettant de générer les tests jusqu'à une certaine profondeur. On pourra utiliser des valeurs aléatoires pour les clefs et les valeurs.

Pour information, dans le langage de programmation Haskell, la bibliothèque `QuickCheck` permet de générer automatique ce type de tests. `QuickCheck` a été porté dans un certain nombre d'autres langages.