

Programmation 1

L3 RI

Table des matières

1	CamL	2
1.1	Introduction à CamL	2
1.2	CamL et orienté objet	2
1.3	Théorie des catégories	2
1.4	Références	2
1.5	Les exceptions	2
1.6	Programmation d'ordre supérieure	2
2	Scala - OOP et FP	3

1 CamL

1.1 Introduction à CamL

Robin Milner (ML : meta-language). Typage = démontrer. P.L Curien crée CAM (categorical abstract machine) \Rightarrow CAML.

Inférence de type : résolution de l'équation aux domaines (résoudre une équation de types).

<fun> place-holder.

1.2 CamL et orienté objet

Liste : constructeurs, extracteurs, observateurs, combineurs.

car (hd) : Content Address Register.

cdr (tl) : Content Decrement Register.

API (Application Programming Interface) fait le lien entre concret et abstrait. Types abstraits \mapsto module.

1.3 Théorie des catégories

Catégories : Set, Group, Ring, Field, Vector. \neq ensemble (cf paradoxe B. Russell).

Objet terminal $T : A \rightarrow \exists! T$.

Objet initial $I : I \rightarrow \exists! A$.

Somme et produit : TO DO schémas. Unique à un iso. près.

1.4 Références

Assigner un nom à une boîte, pas à une valeur. Modifier boîte \rightarrow impureté, effet de bord.

Structures modifiables en CamL : type $t = \{a : \text{int ref}\}$.

1.5 Les exceptions

Changement de thread. Un déroutement peut être matériel, système (kernel panic) ou programme.

CamL : try TrucQuiPeutRaiser with |telleException -> tel traitement.

1.6 Programmation d'ordre supérieure

Appeler une fonction avec ses paramètres et un futur : une fonction qui va s'appliquer au résultat. On peut alors prendre un futur exceptionnel ou faire du pipeline.

On peut empiler des fonctions dans le futur (ex factorielle).

2 Scala - OOP et FP

peut-être les grands principes et/ou conseils pour faire de l'OOP efficacement ?