Programmation 1

L3 RI

Table des matières

1 CamL

1.1 Introduction à CamL

Robin Milner (ML : meta-language). Typer = démontrer. P.L Curien crée CAM (categorical abstract machine) ⇒ 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, combinateurs.

car (hd) : Content Adress Register.

cdr (tl) : Content Decrement Register.

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

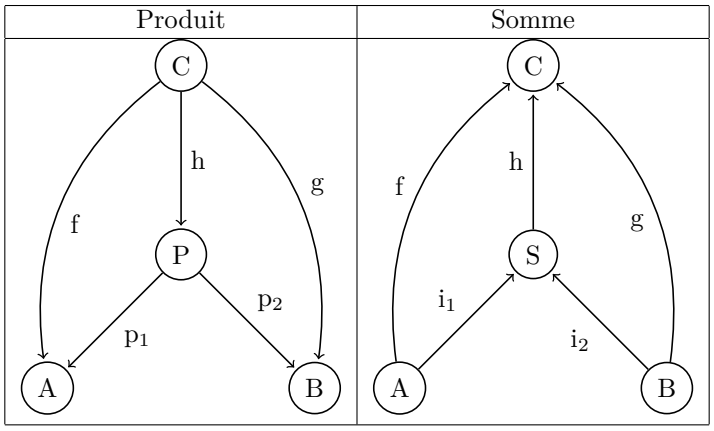
1.3 Théorie des catégories

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

Objet terminal T : $A \to ^{\exists !} T$.

Objet initial I : $I \to ^{\exists !} A$.

Somme et produit : Unique à un iso. près.



1.4 Références

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

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

1.5 Les exceptions

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

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

1.6 Programmation d'ordre supérieure

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

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

2 Scala - OOP et FP

2.1 OOP

Classes, classes abstraites, constructeurs, champs, méthodes
Instances de classe
Héritage, composition, agrégation

2.2 FP

Fonctions, composition de fonctions
Pas de variables globales
Utilisation de "Pattern-matching"