Rapport de projet

Calcul formel Ocaml

Luxon JEAN-PIERRE & Raphael AUVERT

Sommaire

I Choix de conception

Dans le cadre de ce projet, il a été décidé de garder le type *basic_expr* dans la mesure où il traitait de manière satisfaisante les expressions. Cependant, le programme final n'effectue pas les opérations sur ce type, car *basic_expr* ne permet pas au moment du filtrage par motif de donner l'information sur la nature de l'opération mathématique. Par ailleurs, la structure de ce type est trop permissive. En effet, il est possible d'avoir des fonctions trigonométriques d'arité quelconque, mais aussi d'appeler des fonctions avec des paramètres invalides (exemple : *sqrt(-2)*).

Il a donc été décidé de définir un type qui sera le type « de travail » du programme (*gen_math_expr* qui est le type générique, mais instancié comme étant un *math_expr*). La récupération de la formule ce fait de manière habituelle, mais il sera convertit en *math_expr* définit ci-joint :

```
type ('n,'op) gen_math_expr =
  | Pi                                                    (* Pi : 3.14...   *)
  | Exp1                                                  (* e : exp(1)     *)
  | Val of 'n                                             (* Constant value *)
  | Var of string                                         (* Variable       *)
  | Unop of 'op * ('n,'op) gen_math_expr                  (* '+','-' unaire *)
  | Binop of 'op *
      ('n,'op) gen_math_expr *
      ('n,'op) gen_math_expr                              (* '+','-','*'    *)
  | Frac of ('n,'op) gen_math_expr * ('n,'op) gen_math_expr (* Fraction       *)
  | Pow of ('n,'op) gen_math_expr * ('n,'op) gen_math_expr  (* Power          *)
  | Sqrt of ('n,'op) gen_math_expr                        (* Square root    *)
  | Expo of ('n,'op) gen_math_expr                        (* Exponential    *)
  | Log of ('n,'op) gen_math_expr                         (* Logarithm      *)
  | Cos of ('n,'op) gen_math_expr                         (* Cosine         *)
  | Sin of ('n,'op) gen_math_expr                         (* Sine           *)
  | Tan of ('n,'op) gen_math_expr                         (* Tangent        *)
  | Acos of ('n,'op) gen_math_expr                        (* Secant         *)
  | Asin of ('n,'op) gen_math_expr                        (* Cosecant       *)
  | Atan of ('n,'op) gen_math_expr                        (* Cotangent      *)
;;


(* The Mathematical expression that will be used in the program *)
type math_expr = (Num.num,char) gen_math_expr;;
```

L'ensemble des opérations du programme se font à travers ce type.

II Difficutés

Les plus grandes difficultés rencontrés lors de la réalisation du programme concerne la simplification d'expressions ainsi que la dérivation et l'integration. En effet, dans le cas de la simplification, certains cas de simplification étaait délicats à traiter, notamment les opération de type *x op x op ... x op x*, x étant une expression quelconque et op une opération parmi('*+', '-', '*', '/*').

En ce qui concerne la dérivation, le fait de devoir vérifier que la variable filtrée correspond bien à la variable en laquelle la dérivation devait ce faire s'avérait très délicat pour les formules complexes.

Enfin, on a dû se contenter de faire une intégration basique pour les fonctions.

III Fonctionnalités

simpl() : La fonction fonctionne pour les cas très simples, et même pour certains cas complexes. Cependant, pour certains fractions et certaines sommes, il est posibble que cela ne se fasse pas correctement.

eval() : Fonctionne sans problème connu.

subst() : La substition fonctionne correctement ;.Aucun bug n'a été trouvé.

solve() : La résolution fonctionne pour les équations du permier et second degré. Il ne sait pas gérer les les degrés quelconques, bien que cela ait été prévu à la base.

derive() : La dérivation est satisfaisante si on ne met pas plusieurs variables différentes. En effet, pour les cas compliqués, la fonction peut renvoyé des résultats faux.

integ() : Seuls les cas de base fourni dans le formailre sont traités. Il était prévu de généraliser l'intégration, en incluant notamment l'intégratio par partie, mais le temps ne nous le permettait pas.

plot () : L'affichage est tès basique mais fonctionne bien.