

```
1 (*
2 Si consideri la Sintassi Astratta sotto, per il Lambda Calcolo:
3   Lambda ::= Var | Const | [$] Var Lambda | [@] Lambda Lambda
4 dove usiamo $ per l'astrazione e @ per l'applicazione.
5 Si chiede di fornire in Ocaml strutture, tipi, operazioni per la
6 costruzione, l'accesso e la presentazione degli Abstract Syntax
7 Tree dei termini del Linguaggio del Lambda Calcolo.
8 In particolare, le operazioni richieste devono includere:
9 mkConst, mkVar, mkAbs, mkApp,
10 isConst, isVar, isAbs, isApp
11 getId, getTem, getFun, getArg
12 printAST (per la presentazione degli AST)
13 printCRT (per la presentazione nella sintassi concreta, vedi Lezioni,
14   degli AST)
15 *)
16
17 (*
18 Questo modulo e' il corrispettivo del modulo header della versione
19 fornita in C, vedi modulo Last.h
20 *)
21
22 (* Definizione del tipo last: iniziale nome minuscola *)
23 (* Tipi Algebrici: iniziale nome costruttore Maiuscola *)
24
25 type last = Var of string
26           | Const of string
27           | Abs of string * last
28           | App of last * last
29           ;;
30
31 (* segnatura operazioni in OCaml solo moduli di tipi astratti *)
32
```