```
;; calc :: Expr -> number
;; evaluates arithmetical expressions
;; with conditionals and identifiers
(define (calc expr)
  (match expr
    [(num n) n]
    [(add l r) (+ (calc l) (calc r))]
    [(sub l r) (- (calc l) (calc r))]
    [(if0 c t f) (if  (zero? (calc c)) (calc t) (calc f))]
    [(with x e b) (calc (subst b x (num (calc e))))]
    [(id x) (error 'calc "Open expression (free occurrence of ~a)" x)]))
```