

```
;; Further reduces a Value to a numV or closureV
;; and caches the reduced value
;; strict :: Value -> Value [without exprV]
(define (strict v)
  (match v
    [(exprV expr env cache)
     (if (not (unbox cache))
         (let ([val (strict (eval expr env))])
           (set-box! cache val)
           val)
         (unbox cache))])
    [_ v]))
```