

HASKELL-declarations

decl → typedekl | fundecl

typedekl → var₁, ..., var_n :: type, $n \geq 1$

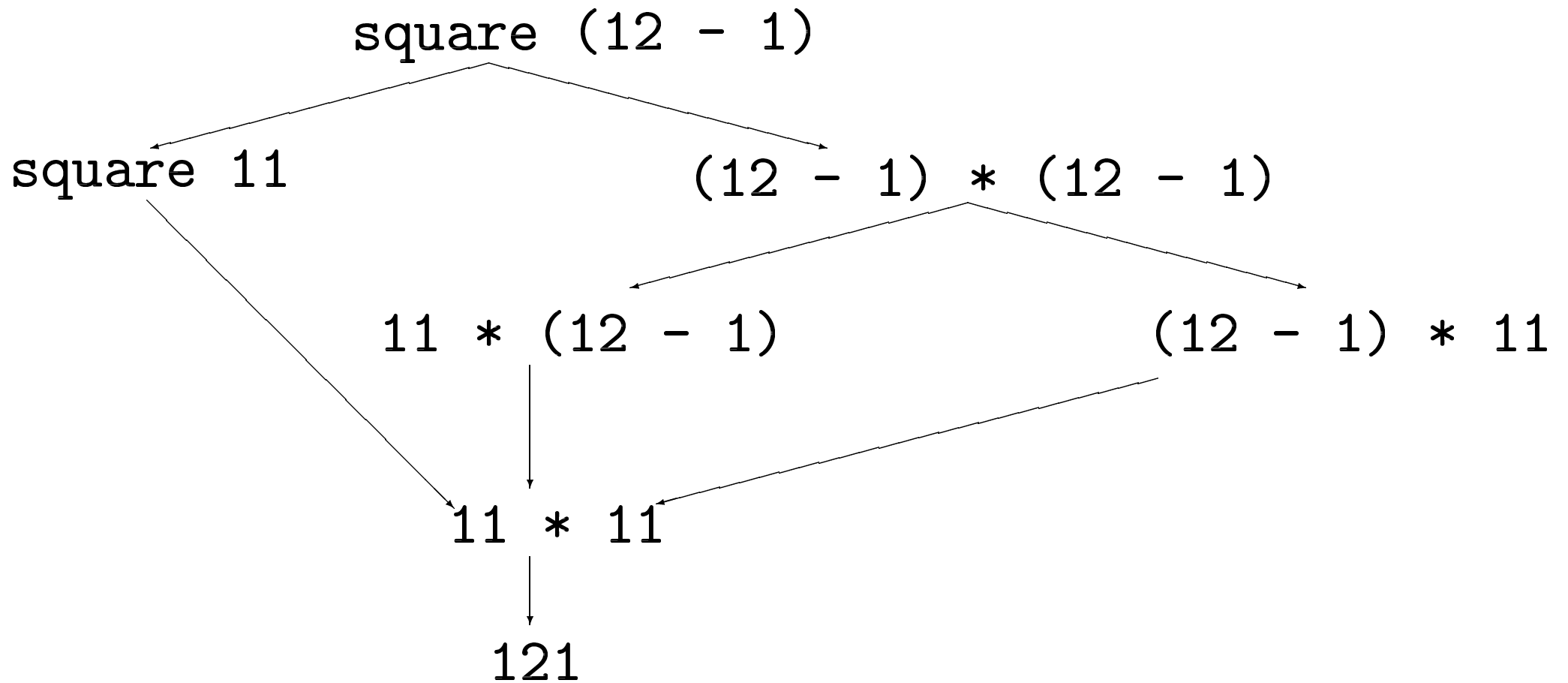
var → string starting with lower case symbol

fundecl → funlhs rhs

funlhs → var pat

rhs → = exp

Evaluation in HASKELL



Function Declarations with Pattern Matching

```
len :: [data] -> Int
len []           = 0
len (x : xs)    = 1 + len xs
```

```
fac :: Int -> Int
fac 0          = 1
fac (x+1)      = (x+1) * fac x
```

```
second :: [Int] -> Int
second []           = 0
second (x : [])    = 0
second (x : y : xs) = y
```

```
half :: Int -> Int
half 0          = 0
half 1          = 0
half (x+2)      = 1 + half x
```