FP

Lez 5

Lists
&
tagged values

Recap: PM su liste

- Sia `f : 'a list → 'b.`
  - Il PM tipico è:

```
let f xs =
 match xs with
    |[] -> e1
    | y::ys -> e2
```

dove `y,ys` tipicamente occorrono in `e2`

- Si possono fare PM più articolati

PM su liste 2

```
let rec ordered xs =
 match xs with
 |[] -> true
 |[x] -> true
 |x::(y::ys) -> x <= y &&
                ordered (y :: ys)
```

- Nota: `[x]` è sy-sugar per `x :: []`

PM su liste 3

- Anche PM non esaustivo da funzioni parziali:

```
let rec last xs =
  match xs with
   | [y] → y
   | _ :: ys → last ys

last : 'a list → 'a
```

- Vedremo tra pochissimo come gestire la parzialità

Mini F#

Il nostro linguaggio, per adesso

- `p ::= c | id |(p,p)| [] | p :: p`
- `e ::= c | id | fun p -> e | e e |(e,e)|()`
  `| let p = e| let p = e in e`
  `| let rec id = e in e | e :: e | []`
  `| match e with p1 → e1 … pn → en`
- `t ::= b | α | t → t | α list | t * t |unit`
- `b ::= int | float | string | bool …`