

## Lemmi di inversione (tipi)

Un lemma di inversione permette di stabilire il tipo delle sottoespressioni di un'espressione ben tipata.

Per esempio, se abbiamo le seguenti regole:

$$\text{true} : \text{Bool} \quad \text{false} : \text{Bool} \quad \frac{\Gamma \vdash c : \text{Bool}, \Gamma \vdash e_1 : T, \Gamma \vdash e_2 : T}{\Gamma \vdash \text{if } c \text{ then } e_1 \text{ else } e_2 : T},$$

seguono immediatamente le inverse:

- $\Gamma \vdash \text{true} : T \implies T = \text{Bool}$
- $\Gamma \vdash \text{false} : T \implies T = \text{Bool}$
- $\Gamma \vdash \text{if } c \text{ then } e_1 \text{ else } e_2 : T \implies \Gamma \vdash c : \text{Bool} \wedge \Gamma \vdash e_1 : T \wedge \Gamma \vdash e_2 : T$

Sulla base dei lemmi di inversione possiamo costruire un algoritmo di type checking:

```
let rec typeof = function
| True -> Bool
| False -> Bool
| If c t e ->
    let tc = typeof c in
    let tt = typeof t in
    let te = typeof e in
    if tc = Bool && tt = te then tt
    else failwith "type error"
```