

## EIDI 2 Cheatsheet

github contributors: [https://github.com/zepatrik/eidi2\\_cheatsheet](https://github.com/zepatrik/eidi2_cheatsheet)  
licence CC BY-SA

20. Februar 2018

## 1 Logik

$$\neg(A \vee B) \equiv \neg A \wedge B$$
$$A \vee (B \wedge A) \equiv A \wedge (B \vee A) \equiv A$$
$$A \implies B \equiv \neg A \vee B$$

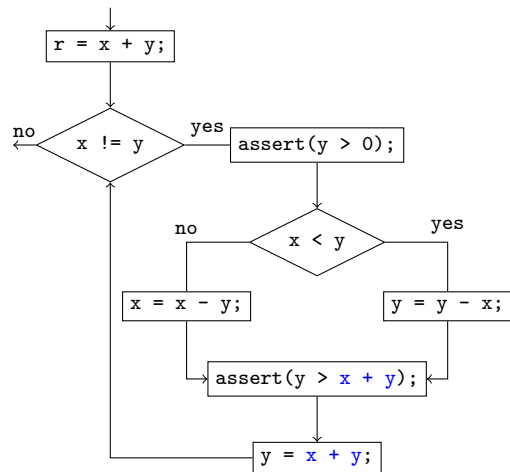
## 2 Verifikation

### 2.1 WP

$$\text{WP}[x = \text{read}();](B) \equiv \forall x. B$$
$$I \Rightarrow \text{WP}[b](B_0, B_1) \equiv I \Rightarrow (((\neg b) \Rightarrow B_0) \wedge (b \Rightarrow B_1))$$

### 2.2 Terminierung

1. vor jedem Schleifendurchlauf  $r > 0$
2.  $r$  wird bei jedem Durchlauf kleiner



## 3 ocaml

### 3.1 Funktoren

```
module type A = sig
  type t
  val f : t -> t
end

module B: A = struct
  type t = int
  let f x = x + 1
end

module Ext(X: A) = struct
  include X
  let g x = f (f x)
end

module C = Ext (B)
```