FPV Week 1 - FPV Summary for the week 1

Funktionale Programmierung (Technische Universität München)

$6. x = 1 \implies x \leq 3 \wedge y > 0 \equiv x \neq 1 \vee x \leq 3 \wedge y > 0. \quad FALSE$

$7. x < 8 \wedge y = x \implies y \neq 12 \equiv x \geq 8 \vee y \neq x \vee y \neq 12. \quad TRUE$

$8. x = 1 \vee y = 1 \implies x > 0 \equiv (x \neq 1 \wedge y \neq 1) \vee x > 0. \quad FALSE$

$9. x \neq 5 \implies false \equiv x = 5 \vee false. \quad FALSE$

$10. true \implies x \neq y \equiv false \vee x \neq y. \quad FALSE$

$11. false \implies x = 1 \equiv true \vee x = 1. \quad TRUE$

$12. x \geq 1 \implies 2x + 3 = 5 \equiv x < 1 \vee 2x + 3 = 5. \quad FALSE$

$13. A \wedge x = y \implies A \equiv \neg A \vee x \neq y \vee A. \quad TRUE$

$14. B \implies A \vee B \equiv \neg B \vee A \vee B. \quad TRUE$

$15. A \implies (B \implies A) \equiv \neg A \vee (B \implies A) \equiv \neg A \vee \neg B \vee A. \quad TRUE$

$16. (A \implies B) \implies A \equiv (\neg A \vee B) \implies A \equiv (A \wedge \neg B) \vee A \equiv A. \quad FALSE$

Summary:

1. The logical equivalence of $A \implies B \equiv \neg A \vee B$ is essential to find the truth value of an implication.

2. According to the characteristics of logical equivalence, $false \implies A \equiv true$ and $A \implies true \equiv true$.

3 Assertion

1. Program state: $i = 0 \wedge \forall n$

True assertions: $i \geq 0, \quad true, \quad , i = 0$

2. Program state: $i \leq 10 \wedge \forall n \wedge x = in$

True assertions: $0 \leq i \leq 10, \quad n = 1 \implies x = i$

3. Program state: $i = 10 \wedge \forall n \wedge x = 10n$

True assertions: $i \geq 0, \quad i = 10, \quad i > 0, \quad x = 10n, \quad x = in \wedge i = 10$

Summary:

1. In each point of a program the assertion ***assert(true)*** always holds. Analogously, the assertion ***assert(false)*** never holds.

2. Considering the variant variables is essential to tell whether the assertion inside the loop holds. (More contents in the coming weeks)

4 The Strong and the Weak

Recap of the results of C:

$i \geq 0, \quad i = 10, \quad i > 0, \quad x \neq n, \quad x = 10n, \quad x = in \wedge i = 10$

A rearrangement of the holding facts in the order from the weakest to the strongest is:

$i \geq 0 \quad \ll \quad i > 9 \quad \ll \quad i = 10 \quad \ll \quad x = in \wedge i = 10$

$x \neq n \quad \ll \quad x = 10 \quad \ll \quad x = in \wedge i = 10$

($\ll$ is no formal notation, but serves here as "weaker than" infix)

As discussed in the previous exercise, ***assert(true)*** always holds and ***assert(false)*** never holds. Hence true is always the strongest assertion while false is always the weakest assertion.

Open question: What is a formal/informal definition for a stronger/weaker assertion?

5 Strongest Post-conditions

We define $B = SP[A, s]$ as the strongest post-condition of assertion A at statement s, where B is the strongest assertion after statement s given the previous assertion A.

("SP[A, s]" is no formal notation, but is introduced with respect to Weakest Precondition in the next weeks)

1.$SP[true, x = 5] :\equiv x = 5$

2.$SP[x = 0 \wedge y = 2, x = 18] :\equiv x = 18 \wedge y = 2$

3.$SP[x \geq 7 \wedge y = -2, x = y + x] :\equiv x \geq 5 \wedge y = -2$

4.$SP[x = 3i \wedge i \geq 0, i = i + 1] :\equiv x = 3(i - 1) \wedge i \geq 1$

5.$SP[y = 2 + x, x = read()] :\equiv true$

6.$SP[i = i + 1, i = 0] :\equiv false$

7.$SP_{yes}[n \geq 2 \wedge x = 12, n > 3] :\equiv n \geq 2 \wedge x = 12 \wedge n \leq 3 \equiv 2 \leq n \leq 3 \wedge x = 12$

$SP_{no}[n \geq 2 \wedge x = 12, n > 3] :\equiv n \geq 2 \wedge x = 12 \wedge n > 3 \equiv n > 3 \wedge x = 12$

8.$SP_{yes}[x = 2i \wedge 0 > i, i == 0] :\equiv x = 2i \wedge 0 > i$

$SP_{no}[x = 2i \wedge 0 > i, i == 0] :\equiv false$

9.$SP_{decomposition}[i = 0 \wedge x > 0, i \neq 0 \wedge y = 1] :\equiv (i = 0 \wedge x > 0) \vee (i \neq 0 \wedge y = 1)$