

Aufgabe 1 – White Box Tests Pfadüberdeckung

Betrachten Sie die Methode `suche`, die im total geordneten Feld `alphabet` das Element `zeichen` findet bzw. die Aussage `-1` über das Fehlen dieses Elementes liefert:

```
public static int suche(final char zeichen, final char[] alphabet) {
    int erstes = 0;
    int letztes = alphabet.length - 1;

    while (erstes <= letztes) {
        final int mitte = erstes + ((letztes - erstes) / 2);
        if (alphabet[mitte] < zeichen) {
            erstes = mitte + 1;        // Rechts weitersuchen
        } else if (alphabet[mitte] > zeichen) {
            letztes = mitte - 1;       // Links weitersuchen
        } else {
            return mitte;              // Zeichen gefunden
        }
    }
    return -1;                        // Zeichen nicht gefunden
}
```

Betrachten Sie neben der in der Vorlesung vermittelten Anweisungs- und Zweigüberdeckung die folgenden Definitionen:

Definition 1: Sei S ein Vorkommen einer Anweisung oder Bedingung. Dann ist

$\text{def}(S) := \{x \mid S \text{ greift schreibend auf Variable } x \text{ zu}\}$

$\text{use}(S) := \{x \mid S \text{ greift lesend auf Variable } x \text{ zu}\}$

Beispiel:

$S := [X = Y + 1;] \text{ def}(S) = \{X\}, \text{ use}(S) = \{Y\}$

$S := \text{read}(X); \text{ def}(S) = \emptyset, \text{ use}(S) = \{X\}$

$S := \text{if } (X > 0) \text{ def}(S) = \emptyset, \text{ use}(S) = \{X\}$

Definition 2: Falls $v \in \text{def}(S) \cap \text{use}(S')$ und $v \notin \text{def}(S'')$ für jede Anweisung S'' , die auf einem Pfad von S und S' durchlaufen wird, so bildet $[v, S, S']$ eine **def-use-Kette (du-Kette)**.

Beispiel: du-Ketten für Variable `letztes`:

```
[letztes, letztes = alphabet.length - 1, while (erstes <= letztes)]
[letztes, letztes = alphabet.length - 1, final int mitte = erstes + ((letztes - erstes) / 2)]
[letztes, letztes = mitte - 1, while (erstes <= letztes)]
[letztes, letztes = mitte - 1, final int mitte = erstes + ((letztes - erstes) / 2)]
```

Aufgaben

- Bestimmen Sie konkrete Testfälle zur vollständigen Anweisungsüberdeckung und erläutern Sie, warum eine vollständige Anweisungsüberdeckung erreicht wird.
- Bestimmen Sie konkrete Testfälle zur vollständigen Zweigüberdeckung und erläutern Sie, warum eine vollständige Zweigüberdeckung erreicht wird.
- Bestimmen Sie alle du-Ketten in `suche`.

- d) Bestimmen Sie konkrete Testfälle, so dass jede du-Kette mindestens einmal durchlaufen wird („du-Ketten-Überdeckung“) und erläutern Sie, warum eine vollständige du-Ketten-Überdeckung erreicht wird.

Hinweis: Weitere Varianten für datenorientierte Tests basierend auf dem du-Prinzip sind

- all defs-Tests (defs steht für define)
- all p-uses-Tests (p-use steht für predicative use)
- all c-uses-Tests (c-use steht für computational use)
- all c-uses/some p-uses-Tests
- all p-uses/some c-uses-Tests
- und weitere.