

LU01d - Trace Table

Trace Tables sind ein nützliches Werkzeug, um den Ablauf eines Programms oder eines Algorithmus Schritt für Schritt zu verfolgen. Sie helfen dabei, die Funktionsweise eines Programms besser zu verstehen, Fehler zu identifizieren und die Korrektheit eines Algorithmus zu überprüfen.

Was ist ein Trace Table?

Ein Trace Table, auch Verfolgungstabelle genannt, ist eine Tabelle, die die Änderung von Variablenwerten und den Kontrollfluss eines Programms zeigt. Jeder Schritt des Programms wird in einer neuen Zeile der Tabelle festgehalten, sodass die Auswirkungen jeder Anweisung auf den Programmzustand nachvollziehbar werden.

Aufbau eines Trace Tables

Ein typischer Trace Table besteht aus mehreren Spalten und Zeilen:

- **Spalten:** Jede Spalte repräsentiert eine Variable oder einen wichtigen Zustand des Programms (z.B. Schleifenindex, Ergebnisvariablen).
- **Zeilen:** Jede Zeile entspricht einem Ausführungsschritt des Programms.

In den Zellen der Tabelle wird der Wert der jeweiligen Variable oder des Zustands nach Ausführung der entsprechenden Anweisung eingetragen.

Beispiel eines Trace Tables

Nehmen wir als Beispiel einen einfachen Algorithmus zur Berechnung der Summe einer Liste von Zahlen.

```
zahlen = [1, 2, 3]
summe = 0
for zahl in zahlen:
    summe += zahl
```

Um diesen Algorithmus mit einem Trace Table zu verfolgen, könnten wir folgende Tabelle erstellen:

Schritt	zahl	summe
Initial	-	0
1	1	1
2	2	3
3	3	6

Schrittweise Analyse des Beispiels

- **Initial:** Bevor der Algorithmus startet, hat die Variable `summe` den Wert 0, und die Variable `zahl` ist noch nicht gesetzt.
- **Schritt 1:** In der ersten Schleifeniteration nimmt `zahl` den Wert 1 an und `summe` wird auf 1 erhöht.
- **Schritt 2:** In der zweiten Schleifeniteration nimmt `zahl` den Wert 2 an und `summe` wird auf 3 erhöht.
- **Schritt 3:** In der dritten und letzten Schleifeniteration nimmt `zahl` den Wert 3 an und `summe` wird auf 6 erhöht.

Der Trace Table zeigt klar, wie die Variable `summe` im Laufe des Programms geändert wird und wie die Schleifenvariable `zahl` nacheinander die Werte aus der Liste `zahlen` annimmt.

Anwendungsfälle von Trace Tables

Trace Tables sind besonders nützlich in folgenden Situationen:

- **Fehlerbehebung:** Um herauszufinden, warum ein Programm nicht wie erwartet funktioniert, kann man den Programmablauf Schritt für Schritt nachvollziehen.
- **Algorithmus-Analyse:** Sie helfen beim Verständnis und bei der Analyse von Algorithmen, insbesondere bei komplexen rekursiven Funktionen.
- **Lehre und Lernen:** Sie sind ein hervorragendes Werkzeug für das Lehren und Lernen von Programmabläufen, insbesondere für Anfänger.

Schlussfolgerung

Trace Tables sind ein einfaches, aber mächtiges Werkzeug, um den Ablauf eines Programms nachzuvollziehen. Sie bieten eine klare und strukturierte Methode, um Variablenänderungen und den Kontrollfluss in einem Programm zu verfolgen. Dies macht sie besonders wertvoll für das Debugging und das Verständnis von Algorithmen.

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu01/tracetable?rev=1722955027>

Last update: **2024/08/06 16:37**

