ulm university universität
uulm

Institut für Numerische Mathematik

Dr. Andreas F. Borchert

26. April 2019
Blatt 1

Parallele Programmierung mit C++ (SS 2019)

Abgabe bis zum 3. Mai 2019, 14:00 Uhr

Lernziele:

- Modellierung einfacher Prozesse mit CSP
- Implementierung und Benutzung einer Klasse in C++

Hinweise:

Melden Sie sich für diese Veranstaltung in SLC an. Ohne diese Anmeldung können Sie keine Lösungen elektronisch einreichen. Sobald Sie sich angemeldet haben, wird es bis zu einer Stunde brauchen, bis Sie anschließend Lösungen einreichen können.

Aufgabe 1: Getränkeautomat

Modellieren Sie mit Hilfe von CSP einen Getränkeautomat, der Tee zum Preis von 1 Euro und Kaffee zum Preis von 1,50 Euro anbietet. Zu unterstützende Münzen sind die für 50 Cent und 1 Euro, die in einer beliebigen Reihenfolge verwendet werden dürfen. Der Automat sollte beim Kaffee in der Lage sein, beim Bezahlen mit zwei einzelnen Euromünzen 50 Cent wieder herauszugeben. Die Rückgabe des Wechselgeldes erfolgt dabei vor der Ausgabe des Kaffee. Dabei sollte

$$\alpha Automat = \{muenze50, muenze100, rueckgabe\_muenze50, tee, kaffee\}$$

gelten.

So könnte ein möglicher Verlauf aussehen:

```
thales$ trace -p automat.csp
Alphabet: {kaffee, muenze100, muenze50, rueckgabe_muenze50, tee}
Acceptable: {muenze100, muenze50}
muenze50
Acceptable: {muenze100, muenze50}
```

```
muenze100
Acceptable: {kaffee}
kaffee
Acceptable: {muenze100, muenze50}
muenze100
Acceptable: {muenze100, muenze50, tee}
muenze100
Acceptable: {rueckgabe_muenze50}
rueckgabe_muenze50
Acceptable: {kaffee}
kaffee
Acceptable: {muenze100, muenze50}
OK
thales$
```

Ergänzen Sie den Automaten mit einer Fehlfunktion, bei der vier 50-Cent-Münzen in einer Reihe akzeptiert werden, dann aber der Automat seine Funktion einstellt:

```
thales$ echo muenze50 muenze50 muenze50 muenze50 | trace -p fehlfunktion.csp
Alphabet: {kaffee, muenze100, muenze50, rueckgabe_muenze50, tee}
Acceptable: {muenze100, muenze50}
Acceptable: {muenze100, muenze50}
Acceptable: {muenze50, tee}
Acceptable: {kaffee, muenze50}
Acceptable: {}
OK
thales$
```

Testen Sie Ihre Lösung interaktiv mit unserer CSP-Implementierung und reichen Sie Ihre Lösungen auf der Theon ein mit folgendem Kommando:

```
submit pp 1 automat.csp fehlfunktion.csp
```

Hierbei sollte *automat.csp* der funktionerende Automat sein und *fehlfunktion.csp* derjenige mit Fehlfunktion.

Viel Erfolg!