

## Blatt 4

### Test Case Generation

#### Aufgabe 1: Constraint Solver

Entwickle einen Solver, der durch Bi-Partitionierung eine Lösung für eine Menge von Constraints ermittelt.

- zu definierende Datentypen: `Interval`, `Bool3`
- Variablen werden durchnummeriert  $(0, \dots, n)$
- eine Belegung der Variablen wird in einem Intervall-Vektor (`V`) abgelegt
- jede Constraint ist in einer Constraint Function „hartcodiert“, die folgende Signatur hat:  
`Bool3 constr(const vector<Interval> &iv)`
- die Constraints werden in einer Liste von Constraint Function Pointern gehalten

Bi-Partitionierungsalgorithmus:

wähle Startbelegung `V0`

```
while (mind. eine Constr. Fkt. wertet zu  $\top$  aus && nicht unlösbar) {  
  foreach ( $c_i \in$  Constr. Fkt.) {  
    if ( $c_i(V) == \text{true}$ )  $\Rightarrow c_i$  erfüllt  
    else if ( $c_i(V) == \text{false}$ )  $\Rightarrow$  unlösbar(V)  
    else  $\Rightarrow c_i(V) == \top$   
  } if (alle Constr. Fkt. erfüllt)  $\Rightarrow$  break, V Solution  
  suche größte Kante in V, teile entlang dieser Kante in V1, V2  
  wiederhole für V1, V2  
}
```

## Aufgabe 2: Lösung ermitteln

Ermittle mit dem implementierten Constraint Solver eine Lösung, die die folgenden Constraints erfüllt:

**c1**  $x_1^2 + x_2^2 < 1$

**c2**  $x_1 < x_2$

**c3**  $0,5 < x_2$

Schreibe hierzu drei entsprechende Constraint Functions.

**Abgabe: Bis Mittwoch, 09.01.2008, per Mail.**