

Aufgabenblatt 10

- Christian Rebischke (432108)
- Sajedah Majdi (493981)

Aufgabe 2

Welche STL-Algorithmen können eingesetzt werden, um Programmieraufwand zu sparen?

Wir können den STL-Algorithmus `accumulate` verwenden, um über alle Bestandteile des Vektors zu iterieren und diesen je nach gewählten `binary operator` (`binop`) zu verändern.

Würde sich Ihre Wahl des verwendeten Algorithmus ändern, wenn der Container modifiziert werden darf? Wenn ja wie?

Wenn ich den Container modifizieren dürfte, würde ich vermutlich statt `accumulate` in einigen Fällen `transform` benutzen, um beispielsweise den `abs()` Operator über alle Vektor Bestandteile auszuführen, was bei allen Normen nötig ist.

Identifizieren Sie die Gemeinsamkeiten der verschiedenen Normberechnungen.

Wie bereits erwähnt, lässt sich `accumulate` über alle Normberechnungen verwenden. Der nicht gemeinsame Teil, wäre dann der jeweilige `binary operator` und ein `unop` Operator für das `sqrt()` in der Hilbertnorm.

Entwerfen Sie ein interface um diese Gemeinsamkeiten auszunutzen.

Das von uns verwendete Interface trägt den Namen `genericNorm` und dient uns als Wrapper und Grundlage für alle anderen Normen. Je nach Norm wählen wir dann den richtigen `binary operator` und benutzen einen `unop` Operator oder einfach die Identität um das letzte Ergebnis einfach durchzureichen, falls ein `unop` nicht nötig ist.

Unsere `genericNorm`, welche als Grundlage für alle anderen Normen dient.

```
template<typename T>
T genericNorm(
    std::vector<T> &vec,
    std::function<T(T, T)> binop,
    std::function<T(T)> unop
) {
    return unop(std::accumulate(vec.begin(), vec.end(), 0, binop));
}
```

Welche Art von Austauschbarkeit würden Sie für die sich unterscheidenden Teile wählen und warum?

Da wir in unserem Quellcode einen komplizierteren Testfall haben und bereits einen Vektor vom Typ `double` als Testvektor nehmen, handelt es sich wohl um statischen Polymorphismus, obwohl unsere Normen sicherlich auch dynamisch verwendet werden könnten, wenn man die `main()` Funktion dementsprechend anpasst.