Formale Systeme, Automaten, Prozesse *Tim Luther, 410886*
Übungsblatt 2 *Til Mohr, 405959*
Tutorium 11 *Simon Michau, 406133*

Aufgabe H7 Der Aufruf der Methode `check` gibt für einen regulären Ausdruck (`RA`) R zurück, ob dieser nur eine endliche Anzahl an Wörtern beschreibt. Folgender Algorithmus kann dies mit einer Worst-Case Laufzeit von $O(n)$ entscheiden, wobei n die Anzahl der Teilausdrücke, die R aufbauen, ist:

```
bool check(RA R) {
        if(R ∈ Σ) return true;
        if(R is A + B || R is AB) return check(A) && check(B);
        if(R is A* || R is A+) return isEmpty(A);
}

bool isEmpty(RA R) {
        if(R is ϵ) return true;
        if(R ∈ Σ \ {ϵ}) return false;
        if(R is A + B || R is AB) return isEmpty(A) && isEmpty(B);
}
```

A und B sind hier reguläre Ausdrücke, die hier R darstellen können.

Aufgabe H8

a) Die Äquivalenzklassen sind $[a]_{\equiv_L} = \{a,b\}^*$ und $[c]_{\equiv_L} = \{a,b\}^*c\{a,b,c\}^*$. Erstere beschreibt alle Worte (keines aus L), bei denen man nur genau ein c hinten anhängen kann, sodass sie immer noch in der Sprache L sind. Die zweite Äquivalenzklasse beschreibt alle Worte (auch alle aus L), die mindestens ein c haben, wodurch man nichts an das Wort anhängen kann, sodass es in der Sprache ist. Beide Äquivalenzklassen vereinigt bilden Σ^*:

$$[a]_{\equiv_L} \cup [c]_{\equiv_L} = \{a,b\}^* \cup \{a,b\}^*c\{a,b,c\}^* = \{a,b,c\}^* = \Sigma^*$$

Das leere Wort ist in $[a]_{\equiv_L}$ beinhaltet sowie alle Wörter, die kein c enthalten. $[c]_{\equiv_L}$ beinhaltet alle Wörter, die mind. ein c enthalten, an egal welcher Stelle. Die Vereinigung beider stellt also Σ^* dar.

b)