

## Zusammenfassung

Ein Pattern ist ein Wort über einem Alphabet, welches aus einer Variablenmenge  $X$  und einer Menge an Terminalsymbolen  $\Sigma$  besteht. Die Menge aller Wörter in  $\Sigma^*$ , die dadurch entsteht, dass man alle Variablenvorkommen in einem Pattern  $p$  durch Wörter aus  $\Sigma^*$  ersetzt, wird als die von  $p$  erzeugte Patternsprache bezeichnet. Verschiedene Vorkommen der gleichen Variable müssen hierbei stets durch das gleiche Wort ersetzt werden. Das Membership-Problem für Patternsprachen besteht darin, für ein Pattern  $p$  und ein Wort  $w \in \Sigma^*$  zu entscheiden, ob  $w$  in der von  $p$  erzeugten Patternsprache liegt. Das Membership-Problem für Patternsprachen ist NP-vollständig.

In dieser Arbeit werden drei Aspekte des Membership-Problems für Patternsprachen untersucht. Im ersten Teil wird das Problem unter komplexitätstheoretischen Gesichtspunkten beleuchtet. Es wird gezeigt, dass das Membership-Problem bereits für die Klasse der Schönbergpattern (siehe [Sch06]) NP-vollständig ist. Des Weiteren wird die Komplexität des Membership-Problems auf laflängenkodierten Instanzen für verschiedene Klassen von Pattern untersucht. Im zweiten Teil wird der algorithmische Aspekt des Problems betrachtet. Insbesondere werden einige Vorschläge zur Verbesserung der Laufzeit des Membership-Tests aus dem Projekt `pattern` (siehe [Mem00]) gemacht. Der dritte Teil der Arbeit beschäftigt sich mit dem Membership-Problem für Wörter mit kurzer Sequenzlänge. Die Sequenzlänge eines Wortes ist die Anzahl der Teilwörter maximaler Länge dieses Wortes, welche nur ein einziges Symbol enthalten. Es wird ein Algorithmus zur Entscheidung des Membership-Problems vorgestellt, der das Problem schneller entscheidet als der Ansatz, auf welchem der `pattern`-Algorithmus basiert, sofern die Anzahl der verschiedenen Variablen in dem Pattern, für welches das Problem entschieden werden soll, deutlich größer ist als die Sequenzlänge des Wortes.