

Dokumentation zum Konvertierer und Interpreter für Turingmaschinen

Sven Fiergolla

November 22, 2017

Einführung

Wie in “A Universal Turing Machine with Two Internal States” von Claude E. Shannon beschrieben, lässt sich jede Turingmaschine in eine TM mit nur zwei Zuständen überführen.

Im folgenden wird die Implementierung, der im Paper beschriebenen Konvertierung, dokumentiert und erläutert. Zudem wird der Interpreter erläutert.

Klassen & Abhängigkeiten

Konvertierer (Package construction)

Für die Konvertierung einer TM nach dem beschriebenen Verfahren werden eine Reihe neuer Symbole benötigt, da die Information des aktuellen Zustandes in die Symbole übertragen wird. Anschließend müssen die Übergänge der TM und das Startsymbol (das Symbol unter dem Lesekopf zum ersten Schritt) angepasst werden. Dazu existieren die Klassen `ComplexSymbol` und `TMConstructor`.

`ComplexSymbol`

Nach Shannons Verfahren, müssen zu jedem elementaren Symbol der ursprünglichen TM, pro Zustand, 4 neue Symbole erstellt werden. Dazu werden die Konstanten aus der Dummy-Klasse `ComplexSymbol` verwendet.

`TM2Generator`

Die Klasse `TM2Generator` kann mit Hilfe der Funktion `readTMfromFile(String path)` bzw. über den Konstruktor, eine Turingmaschine im beschriebenen `.tur`-Format einlesen. Nun lässt sich die Funktion `generate2StateTM()` anwenden, welche 4 weitere Funktionen aufruft:

`generateComSymbolTable()`

Interpreter

Application

Dateiformat .tur