

# GTI Tool

## Benutzerhandbuch

Christian Fehler, Benjamin Mies, Simon Meurer  
Fachbereich 12 - Informatik und Elektrotechnik  
Universität Siegen

30. Juli 2009

# Inhaltsverzeichnis

<b>1</b>	<b>Wie erstelle ich eine neue Grammatik?</b>	<b>3</b>
1.1	Neue Grammatik Datei anlegen . . . . .	3
1.2	Anlegen von Produktionen . . . . .	4
1.3	Was fange ich mit einer Grammatik an? . . . . .	6
1.3.1	Einheitsproduktionen entfernen . . . . .	7
1.3.2	$\epsilon$ -Produktionen entfernen . . . . .	7
1.3.3	Links-Rekursion eliminieren . . . . .	8
1.3.4	Links-Faktorisierung . . . . .	10
1.3.5	Recursive Descent Parser erstellen . . . . .	11
<b>2</b>	<b>Wie erstelle ich einen Automaten?</b>	<b>12</b>
2.1	Neue Automat Datei anlegen . . . . .	13
2.2	Zustände und Übergänge anlegen . . . . .	14
2.3	Was fange ich mit einem Automaten an? . . . . .	15
2.3.1	Bearbeiten per Übergangstabelle . . . . .	15
2.3.2	Vorlage für... . . . .	16
2.3.3	Wort-Navigation . . . . .	16
2.3.4	Minimieren . . . . .	18
2.3.5	Umwandeln in... . . . .	20
2.3.6	Umwandeln in... (Potenzmenge) . . . . .	24
2.3.7	Umwandeln in regulären Ausdruck . . . . .	24
2.3.8	Erreichbare Zustände . . . . .	25
<b>3</b>	<b>Wie erstelle ich einen Kellerautomaten?</b>	<b>27</b>
3.1	Navigation . . . . .	28
3.2	Kellerautomat aus Grammatik erstellen . . . . .	28
<b>4</b>	<b>Wie erstelle ich einen regulären Ausdruck?</b>	<b>31</b>
4.1	Neue .regex Datei anlegen . . . . .	31
4.2	Regulären Ausdruck eingeben . . . . .	31
4.3	Das Informationsfenster . . . . .	32
4.4	Was macht man mit regulären Ausdrücken? . . . . .	32
4.4.1	Übersetzen in Kernsyntax . . . . .	32
4.4.2	Einen regulären Ausdruck umwandeln . . . . .	33

<i>INHALTSVERZEICHNIS</i>	3
4.4.3 Exportieren nach $\text{\LaTeX}$ . . . . .	37
<b>5 Einstellungen</b>	<b>39</b>
5.1 Oberfläche anpassen . . . . .	39
5.2 Automaten Einstellungen anpassen . . . . .	40
5.3 Grammatik Einstellungen anpassen . . . . .	41
<b>6 Sonstiges</b>	<b>43</b>
6.1 Algorithmen . . . . .	43
6.2 Zweite Ansicht . . . . .	43
6.3 Austausch von Dateien . . . . .	45
6.4 Bild exportieren . . . . .	46

# Kapitel 1

## Wie erstelle ich eine neue Grammatik?

Wir wollen das jetzt einmal an Hand folgenden Beispiels aus den Vorlesungsunterlagen durchgehen:

$$\begin{aligned}G &= (\Sigma, N, S, P) \text{ mit} \\ \Sigma &= \{\mathbf{0}, \mathbf{1}, \mathbf{x}, \mathbf{y}, -, +, *, (\, )\} \\ N &= \{\mathbf{E}\} \\ S &= \mathbf{E} \\ P &= \{\mathbf{E} \rightarrow \mathbf{0}, \mathbf{E} \rightarrow \mathbf{1}, \mathbf{E} \rightarrow \mathbf{x}, \mathbf{E} \rightarrow \mathbf{y}, \mathbf{E} \rightarrow (-\mathbf{E}), \\ &\quad \mathbf{E} \rightarrow (\mathbf{E}+\mathbf{E}), \mathbf{E} \rightarrow (\mathbf{E}-\mathbf{E}), \mathbf{E} \rightarrow (\mathbf{E}*\mathbf{E})\}\end{aligned}$$

### 1.1 Neue Grammatik Datei anlegen

Am Anfang müssen wir dazu erst einmal den „Neu...“ Dialog öffnen. Zu finden ist dieser in der Toolbar oder im Menüeintrag „Datei“. Dort wählen wir aus, dass wir eine Grammatik erstellen wollen und klicken auf „Weiter“.

Im nächsten Dialog werden wir gefragt, von welchem Typ unsere neue Grammatik sein soll. In unserem Beispiel handelt es sich um eine „Kontextfreie Grammatik“, welche wir auswählen und mit „Weiter“ bestätigen.

Als nächstes müssen wir die benötigten Nichtterminalzeichen (N), Terminalzeichen ( $\Sigma$ ) und das Startzeichen (S) angeben. Als Zeichen können alle Symbole verwendet werden, die nicht zur Syntax der Eingabe gehören. Das bedeutet „{, }“ und „“ können hier nicht verwendet werden. Symbole, welche länger als ein Zeichen sind, müssen in Anführungszeichen angegeben werden.

In dem Dialog sind an allen Stellen schon vordefinierte Werte eingetragen. Dabei handelt es sich um die Vorgaben, die man als Standardwerte in den Einstellungen eingetragen hat.

Wie man diese Standardwerte ändert, kann man in Kapitel 5 nachlesen. Wir tragen jetzt als erstes die von uns benötigten Nichtterminalzeichen ein. Dabei handelt es sich bei uns lediglich um das Symbol **E**. Also schreiben wir in das Feld „{E}“.

Nun legen wir das Startzeichen fest. Das ist bei uns das **E**, welches wir in das Feld für das Startzeichen schreiben.

Zum Schluss müssen wir noch die Terminalzeichen festlegen. Dazu tragen wir in das entsprechende Feld „{0, 1, x, y, -, +, \*, (, )}“ ein.

Damit haben wir alle benötigten Information eingegeben und können die neue Grammatik durch Klick auf „Fertig“ erstellen.

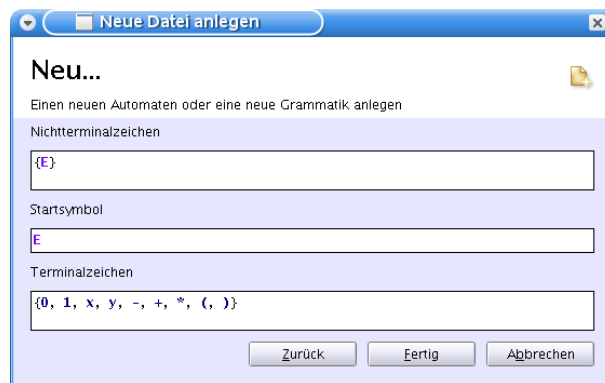


Abbildung 1.1: Grammatik - Neu Dialog

Alle Einstellungen, die wir gerade im letzten Dialog für die neue Datei getroffen haben, lassen sich über „Dokument editieren“ nachträglich ändern.

## 1.2 Anlegen von Produktionen

Um eine neue Produktion anzulegen, können wir den entsprechenden Button in der Toolbar verwenden oder über den Kontextmenüeintrag gehen.

Im Dialog für neue Produktionen müssen wir zunächst einmal angeben, für welches Nichtterminalzeichen wir eine Produktion anlegen möchten. Dieses können wir aus einer Liste der verfügbaren Zeichen auswählen. Da wir in unserem Beispiel nur ein Nichtterminalzeichen haben, ist schon das richtige ausgewählt, und wir können diesen Schritt überspringen.

Jetzt müssen wir noch das Produktions-Wort angeben. Als Hilfestellung wird in dem Dialog angezeigt, welche Nichtterminalzeichen und Terminalzeichen uns zur Verfügung stehen. Wir fangen an mit der Produktion „**E** → (**E**+**E**)“.

so geben wir als Produktions Wort „ $(E + E)$ “ ein. Als weitere Hilfestellung wird im unteren Dialog die resultierende Produktion im Ganzen angezeigt. Wir bestätigen den Dialog noch mit „OK“ und die neue Produktion erscheint in unserer Liste.

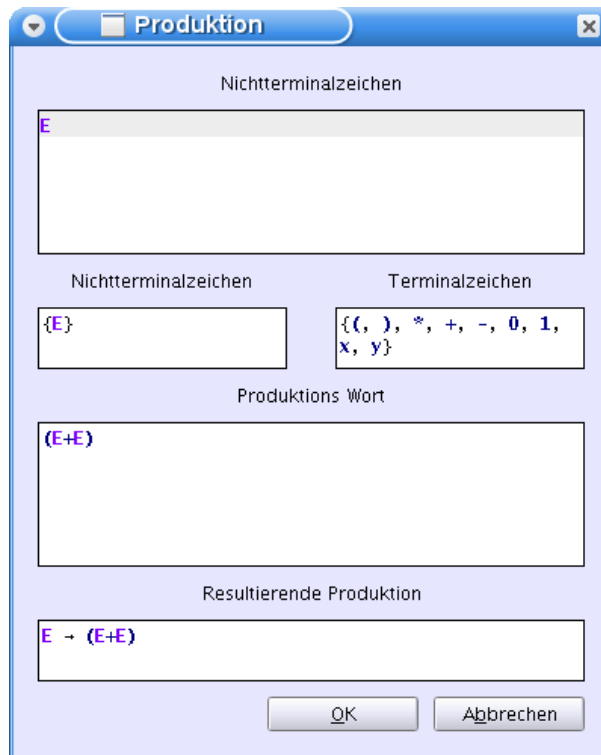


Abbildung 1.2: Grammatik - Produktions Dialog

Wir können die Produktion selbstverständlich auch editieren und wieder löschen. Beide Funktionen sind über das Kontextmenü für die ausgewählte Produktion verfügbar.

Diesen Vorgang wiederholen wir jetzt für unsere komplette Menge „P“. Wenn wir alle Produktionen angelegt haben, sind wir auch mit dem Anlegen der Grammatik fertig. Es besteht jetzt noch die Möglichkeit, die Grammatik zu validieren, um zu sehen, ob man beim Erstellen irgendwelche Fehler gemacht hat. Diese Funktion erreicht man über das Kontextmenü oder über den Menüpunkt „Ausführen“.

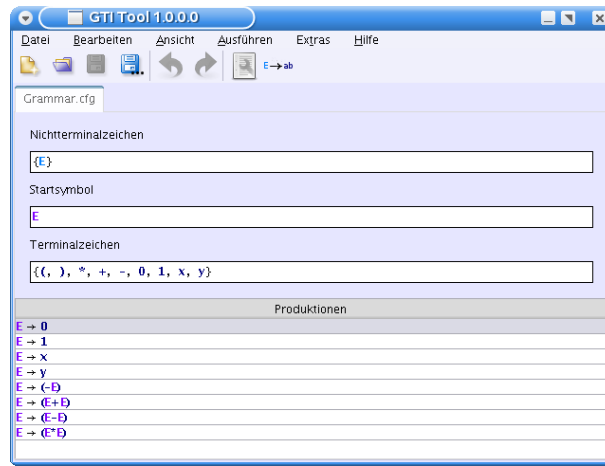


Abbildung 1.3: Grammatik - Kontextfreie Grammatik

### 1.3 Was fange ich mit einer Grammatik an?

Wir haben jetzt eine Grammatik angelegt und stellen uns jetzt vielleicht die Frage, was wir damit eigentlich anfangen können. Es bestehen zur Zeit zwei Verwendungsmöglichkeiten für eine Grammatik.

Zunächst einmal kann man die Grammatik als Vorlage für eine neue Grammatik nutzen. Das ist zum Beispiel sinnvoll, wenn wir eine kontextfreie Grammatik haben und jetzt eine reguläre Grammatik mit den gleichen Produktionen erstellen wollen. Dazu öffnet man den Menüpunkt „Ausführen“ und wählt unter „Vorlage für...“ den gewünschten Typ der neuen Datei aus. Dabei ist zu beachten, dass die Grammatik nicht konvertiert wird, sondern nur als Vorlage verwendet wird.

Die andere Verwendungsmöglichkeit beinhaltet, sich aus der Grammatik einen entsprechenden Automaten generieren zu lassen. Es besteht die Möglichkeit, sich für eine reguläre Grammatik den entsprechenden NDEA, und für eine kontextfreie Grammatik den entsprechenden Kellerautomat erzeugen zu lassen. Zu finden ist die Umwandlung im Menüeintrag „Ausführen“ unter „Umwandeln in...“.

Es ist zu beachten, dass beim Umwandeln einer Grammatik keine Validierungsfehler vorhanden sein dürfen. Bei der Funktion „Vorlage für...“ spielen Fehler allerdings keine Rolle.

### 1.3.1 Einheitsproduktionen entfernen

Hierzu nutzen wir die folgende Beispielgrammatik:

$$G = (\Sigma, N, S, P) \text{ mit}$$

$$\Sigma = \{\mathbf{a, b, c, d}\}$$

$$N = \{\mathbf{A, S}\}$$

$$S = \mathbf{S}$$

$$P = \{\mathbf{S} \rightarrow \mathbf{Aa}, \mathbf{S} \rightarrow \mathbf{b}, \mathbf{A} \rightarrow \mathbf{Ac}, \mathbf{A} \rightarrow \mathbf{Sd}, \mathbf{A} \rightarrow \epsilon, \mathbf{A} \rightarrow \mathbf{S}\}$$

Nun findet man im Menü unter „Ausführen“ den Punkt „Einheitsproduktionen entfernen“. Es erscheint das folgende Fenster.

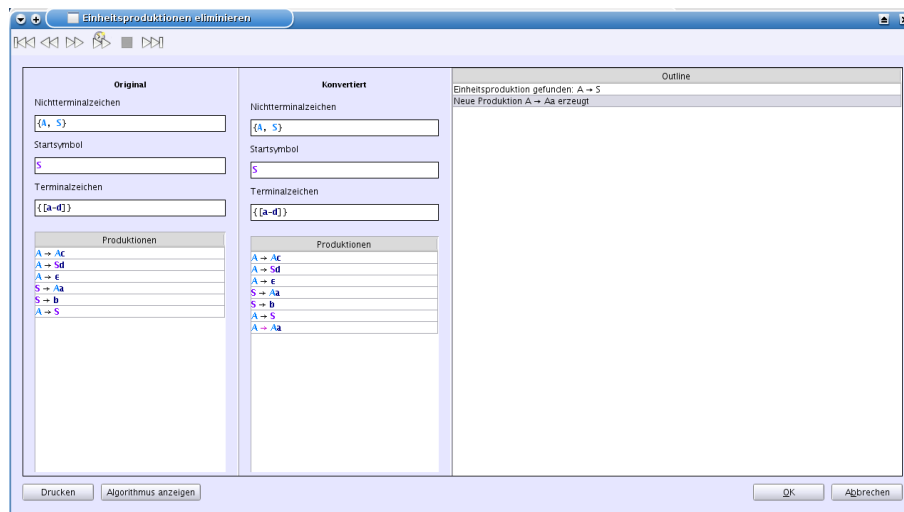


Abbildung 1.4: Grammatik - Einheitsproduktionen entfernen

Ganz links befindet sich die ursprüngliche Grammatik. Rechts daneben findet man die neue Grammatik. Diese ist am Anfang gleich der Ursprünglichen und wird erst im Verlauf des Algorithmus verändert.

Zunächst sucht der Algorithmus nun nach einer Einheitsproduktion und findet die Produktion  $\mathbf{A} \rightarrow \mathbf{S}$ .

Nun wird für jede Produktion von  $\mathbf{S}$  eine entsprechende Produktion für  $\mathbf{A}$  hinzugefügt. Dies wäre in unserem Beispiel zunächst die Produktion  $\mathbf{A} \rightarrow \mathbf{Aa}$  und danach die Produktion  $\mathbf{A} \rightarrow \mathbf{b}$ .

Zum Schluss werden die Einheitsproduktionen entfernt. Also in unserem Fall  $\mathbf{A} \rightarrow \mathbf{S}$ .

### 1.3.2 $\epsilon$ -Produktionen entfernen

Hierzu nutzen wir die folgende Beispielgrammatik:



$G = (\Sigma, N, S, P)$  mit

$\Sigma = \{a, b, c, d\}$

$N = \{A, S\}$

$S = S$

$P = \{S \rightarrow Aa, S \rightarrow b, A \rightarrow Ac, A \rightarrow Sd, A \rightarrow \epsilon\}$

Unter dem Menüpunkt „Ausführen“ findet man den Punkt „ $\epsilon$ -Produktionen entfernen“. Es erscheint das folgende Fenster.

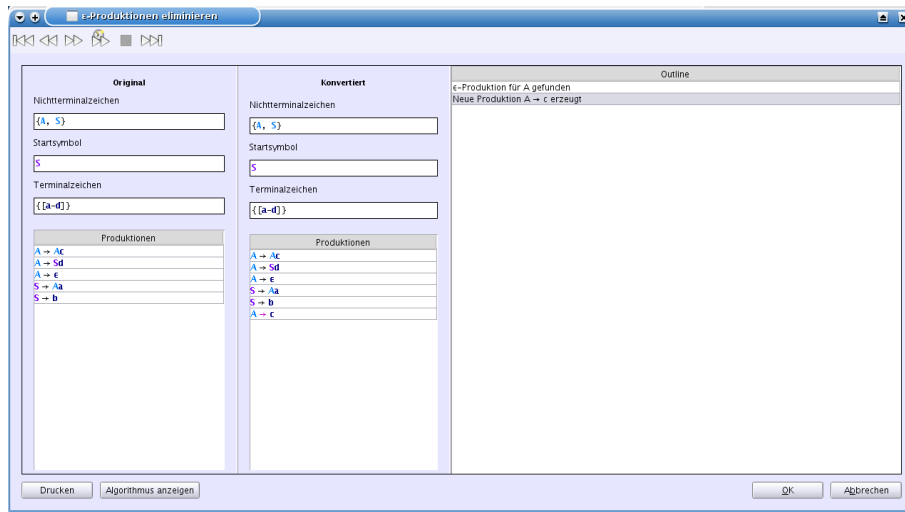


Abbildung 1.5: Grammatik -  $\epsilon$ -Produktionen entfernen

Dieses Fenster ist im Prinzip das gleiche, wie beim Entfernen der Einheitsproduktionen.

Zunächst sucht der Algorithmus nun nach einer  $\epsilon$ -Produktion und findet die eine für **A**.

Nun wird für jede Produktion, wo **A** auf der rechten Seite vorkommt, eine gleiche Produktion, ohne **A** auf der rechten Seite angelegt. In unserem Beispiel wird zunächst die Produktion **A** → **c** und dann **S** → **a** angelegt.

Zum Schluss werden die  $\epsilon$ -Produktionen entfernt, außer der für das Startsymbol. Also in unserem Fall **A** →  $\epsilon$ .

### 1.3.3 Links-Rekursion eliminieren

Hierzu nutzen wir eine Beispielgrammatik aus der Compilerbau 1-Vorlesung:

$G = (\Sigma, N, S, P)$  mit

$\Sigma = \{a, b, c, d\}$

$N = \{A, S\}$

$S = S$

$P = \{S \rightarrow Aa, S \rightarrow b, A \rightarrow Ac, A \rightarrow Sd, A \rightarrow \epsilon\}$

Im Menü findet man unter „Ausführen“ den Punkt „Links-Rekursion eliminieren“. Es erscheint das folgende Fenster.

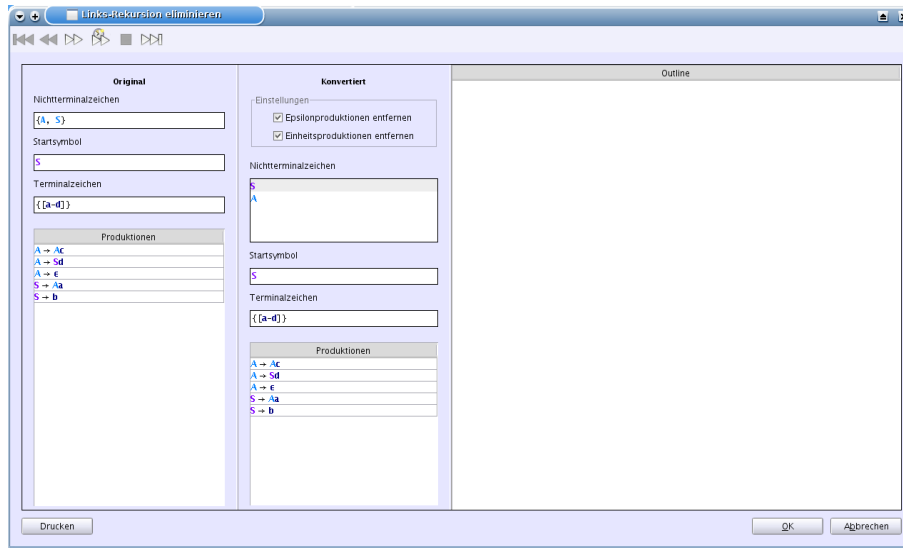


Abbildung 1.6: Grammatik - Links-Rekursion eliminieren

Dieses Fenster ähnelt den Fenstern zum Entfernen der Einheits- bzw.  $\epsilon$ -Produktionen.

Oben im Fenster der konvertierten Grammatik kann man einstellen ob zunächst die Epsilon- bzw. Einheitsproduktionen entfernt werden sollen. Diese Möglichkeit wurde gegeben, da der Algorithmus eigentlich eine Grammatik ohne Epsilon- bzw. Einheitsproduktionen verlangt. Aber im Beispiel aus der Vorlesung Compilerbau I wurde die  $\epsilon$ -Produktion in der Grammatik belassen mit dem Hinweis, dass dies nichts ausmacht. Da aber das Ergebnis dann nicht exakt das selbe wäre, ist hier die Möglichkeit gegeben die Produktionen zu belassen. Für andere Beispiele sollte man aber diese beiden Optionen angestellt lassen, da sonst der Algorithmus falsche Ergebnisse liefern kann.

In unserem Beispiel schalten wir die Option Epsilonproduktionen entfernen aus.

Als nächstes kann die Reihenfolge der Nichtterminalzeichen eingestellt werden, da auch diese sich auf die Ergebnisse des Algorithmus auswirken. Die Reihenfolge kann natürlich genau wie die beiden Optionen zum Entfernen von Produktionen nur am Anfang verändert werden.

In der Vorlesung wurde die Reihenfolge **S, A** gewählt, daher schieben wir das **A** einfach nach oben.

Nun kann der Algorithmus ablaufen:

- Im initialen Schritt passiert nichts mit der Grammatik, da  $i = j$  ist und

für  $S$  keine direkte Linksrekursion existiert.

- Nun kommen durch den Durchlauf im Algorithmus zwei Produktionen ( $A \rightarrow Abd$ ,  $A \rightarrow bd$ ) hinzu und die Produktion  $A \rightarrow Sd$  wird entfernt.
- Im letzten Schritt dieses Beispiels ist  $i = j$  und es wird daher nur noch die direkte Linksrekursion von  $A$  entfernt.

### 1.3.4 Links-Faktorisierung

Hierzu nutzen wir folgende Beispielgrammatik aus der Compilerbau 1-Vorlesung:

$G = (\Sigma, N, S, P)$  mit

$\Sigma = \{a, b, e, i, t\}$

$N = \{E, S\}$

$S = S$

$P = \{S \rightarrow iEtSeS, S \rightarrow iEtS, S \rightarrow a, E \rightarrow b\}$

Nun findet man im Menü unter „Ausführen“ den Punkt „Links-Faktorisierung“. Es erscheint das folgende Fenster.

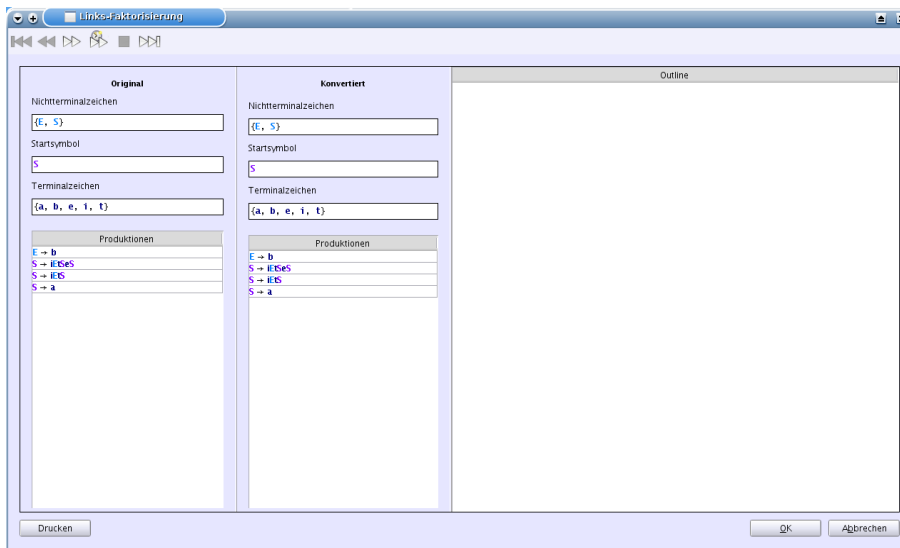


Abbildung 1.7: Grammatik - Links-Faktorisierung

Ganz links befindet sich die ursprüngliche Grammatik. Rechts daneben findet man die konvertierte Grammatik.

Wenn man den Algorithmus dann ablaufen lässt, ist der Algorithmus bereits nach einem Schritt fertig. Der Algorithmus findet als längstes gemeinsames Präfix  $iEtS$  und erzeugt ein neues Nichtterminalzeichen  $S'$ . Als Produktionen kommen hinzu:  $S \rightarrow iEtSS'$ ,  $S' \rightarrow \epsilon$ ,  $S' \rightarrow eS$ , entfernt wurde die Produktion  $S \rightarrow iEtSeS$ .

### 1.3.5 Recursive Descent Parser erstellen

Hierzu nutzen wir eine Beispielgrammatik aus der Compilerbau 1-Vorlesung:

$$G = (\Sigma, N, S, P) \text{ mit}$$


$$\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{e}, \mathbf{i}, \mathbf{t}\}$$

$$N = \{\mathbf{E}, \mathbf{S}\}$$

$$S = \mathbf{S}$$

$$P = \{\mathbf{S} \rightarrow \mathbf{iEtSeS}, \mathbf{S} \rightarrow \mathbf{iEtS}, \mathbf{S} \rightarrow \mathbf{a}, \mathbf{E} \rightarrow \mathbf{b}\}$$

Nun findet man im Menü unter „Ausführen“ den Punkt „Recursive Descent Parser erstellen“. Es erscheint das folgende Fenster.



```

void E0 {
  case {
    match("b");
  }
}

void S0 {
  case {
    match("i");
    E0;
    match("t");
    S0;
    match("e");
    S0;
  }
  case {
    match("i");
    E0;
    match("t");
    S0;
  }
  case {
    match("a");
  }
}
  
```

Abbildung 1.8: Grammatik - Recursive Descent Parser

Es wurde der Recursive Descent Parser in einer einfachen Programmiersprache erstellt.

Es wird für jedes Nonterminal eine Funktion erstellt. Also in unserem Beispiel zunächst eine für **E**. Darin werden dann verschiedene Fälle behandelt. Da es für **E** nur die Produktion  $\mathbf{E} \rightarrow \mathbf{b}$  gibt, gibt es auch nur einen Fall. In diesem wird das Eingabezeichen **b** gelesen, angedeutet durch das `match("b")`.

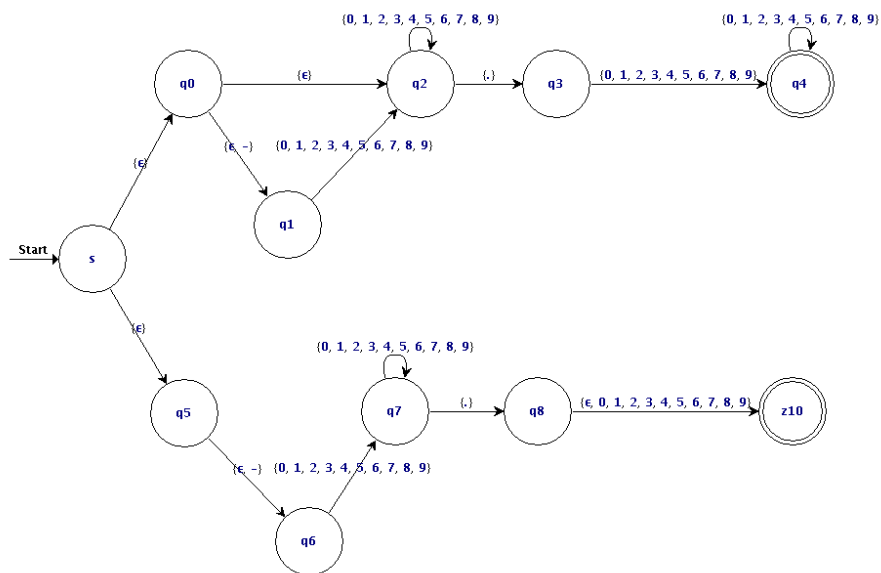
Danach wird eine Funktion für **S** erstellt. Diese beinhaltet drei Fälle. Zunächst wird im ersten Fall die erste Produktion von **S** behandelt. Als erstes wird **i** gelesen, dann wird die Funktion für das Nonterminal **E** aufgerufen und so weiter.

Den Recursive Descent Parser kann man entweder als Text speichern oder drucken.

## Kapitel 2

# Wie erstelle ich einen Automaten?

Wir wollen uns anhand des folgenden Beispiels anschauen, wie ein Automat erstellt wird:



## 2.1 Neue Automat Datei anlegen

Am Anfang müssen wir dazu erst einmal den „Neu...“ Dialog öffnen. Dort wählen wir aus, dass wir einen Automaten erstellen wollen und klicken auf „Weiter“.

Im nächsten Dialog werden wir gefragt, von welchem Typ unser neuer Automat sein soll. In unserem Beispiel handelt es sich um einen „ $\epsilon$ -NDEA“, welchen wir auswählen und mit „Weiter“ bestätigen.

Dann müssen wir das gewünschte Alphabet für den neuen Automaten angeben. Im Alphabet können alle Symbole verwendet werden, die nicht zur Syntax der Eingabe gehören. Das bedeutet, „{, }“ und „“ können hier nicht verwendet werden. Symbole, welche länger als ein Zeichen sind, müssen in Anführungszeichen angegeben werden.

In dem Dialog sind an allen Stellen schon vordefinierte Werte eingetragen. Dabei handelt es sich um die Vorgaben, die man als Standardwerte in den Einstellung eingetragen hat. Wie man diese Standardwerte ändert, kann man in Kapitel 5 nachlesen.

Wir müssen noch die von uns benötigten Symbole angeben. Dazu tragen wir in das Feld Alphabet „{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -}“ ein. Damit haben wir alle benötigten Information eingegeben und können den neuen Automaten durch Klick auf „Fertig“ erstellen.

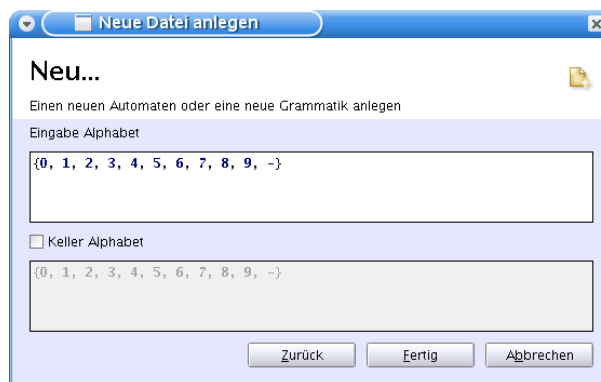







Abbildung 2.1: Automat - Toolbar

Das für den Automaten angegebene Alphabet, lässt sich jeder Zeit über den Button „Dokument editieren“ nachträglich ändern.

## 2.2 Zustände und Übergänge anlegen

Um einen Automaten zu bearbeiten, müssen wir zuerst den entsprechenden Modus auswählen. Folgende Modi sind in der Toolbar verfügbar:

-  Maus Modus
-  Neuer Zustand
-  Neuer Übergang
-  Neuer Start Zustand
-  Neuer Akzeptierender Zustand

Da wir zu Beginn den Startzustand anlegen wollen, wählen wir „Neuer Start Zustand“ und klicken auf eine freie Fläche im unteren Teil, um den Zustand dort zu erstellen. Um den Zustand nachträglich zu bearbeiten, müssen wir in den Maus Modus wechseln. Den Konfigurationsdialog erreicht man dann über Doppelklick oder das Kontextmenü. Über das Kontextmenü lässt sich ein Zustand auch wieder löschen.

Auf diese Weise lassen sich alle Zustände anlegen, die für den Beispielautomaten benötigt werden. Nachdem wir alle Zustände angelegt haben, fehlen noch die Übergänge zwischen den Zuständen. Daher wechseln wir in den „Neuer Übergang“ Modus. Um jetzt einen neuen Übergang anzulegen, klicken wir auf einen Zustand und ziehen, bei gedrückter Maustaste, die Maus auf einen anderen Zustand. Beim Loslassen öffnet sich der Konfigurationsdialog für Übergänge. In diesem Dialog ziehen wir jetzt die Symbole, die in dem Übergang enthalten sein sollen, aus der linken Liste mit allen Symbolen in die rechte Liste, welche die Übergangsmenge repräsentiert. Als Hilfestellung wird im unteren Dialog der entstehende Übergang angezeigt. Nach bestätigen durch „OK“ wird der Übergang angelegt.

Die Übergänge lassen sich auf die gleiche Weise bearbeiten und löschen, wie es bei den Zuständen möglich ist. Es gibt allerdings eine Besonderheit: Wenn man die Maus anstatt über einem Zustand, über einem leeren Bereich los lässt, entsteht beim Anlegen des Übergangs zusätzlich ein neuer Zustand.

Diesen Vorgang wiederholen wir jetzt für alle Übergänge, die für den Beispielautomaten benötigt werden. Wenn wir damit fertig sind, können wir den Automaten noch validieren, um zu sehen, ob uns beim Anlegen der Zustände und Übergänge irgendwelche Fehler unterlaufen sind. Diese Funktion erreicht man über das Kontextmenü oder über den Menüpunkt „Ausführen“.

Sollten die Zustände nicht ganz so optimal platziert sein, kann man versuchen, über die Auto-Layout Funktion ein besseres Ergebnis zu erreichen. Diese ist über „Ausführen“ und „Auto Layout“ oder über das Kontextmenü erreichbar.

## 2.3 Was fange ich mit einem Automaten an?

Für einen erstellten Automaten gibt es folgende Verwendungsmöglichkeiten.

### 2.3.1 Bearbeiten per Übergangstabelle

Die Übergänge eines Automaten können über die Tabelle bearbeitet werden. Diese ist im rechten Teil des Fensters zu finden. Es ist möglich, neue Übergänge anzulegen. Bestehende Übergänge können gelöscht oder editiert werden.

Betrachten wir ein einfaches Beispiel. Wir legen zwei Zustände **z0** und **z1** an und arbeiten auf dem Alphabet  $\{0, 1\}$ . Die Übergangstabelle besteht somit aus vier Spalten, in der ersten Spalte werden die Zustände dargestellt, in der zweiten die Übergänge mit  $\epsilon$  und in den anderen die Symbole des Alphabets, also in unserem Fall je eine Spalte für **0** und **1**.

Um von **z0** nach **z1** einen  $\epsilon$ -Übergang anzulegen, muss die erste Zeile mit **z0** bearbeitet werden. Da ein  $\epsilon$ -Übergang angelegt werden soll, muss die zweite Spalte bearbeitet werden. Um dies umzusetzen, muss auf die entsprechende Zelle doppelt geklickt werden. Es steht ein Parser zur Verfügung, in den mit Komma getrennte Zustände eingetragen werden können. Die Zustände müssen vorhanden sein. In unserem Beispiel tragen wir also in der zweiten Spalte, in der ersten Zeile den Zustand **z1** ein und bestätigen die Eingabe mit Enter. Es wird, wie erwartet, ein  $\epsilon$ -Übergang von **z0** nach **z1** angelegt.

Als nächstes sollen zwei Übergänge auf einmal mit Symbol **0** angelegt werden. Beide sollen von Zustand **z1** ausgehen, der erste soll bei **z0** enden, der zweite bei **z1**. Um dies umzusetzen, muss die Zeile von **z1** und die Spalte von **0** bearbeitet werden. Da der Übergang zu den beiden Zuständen gehen soll, muss **z0, z1** eingetragen werden.

Das Löschen und Erweitern von Übergängen erfolgt analog.



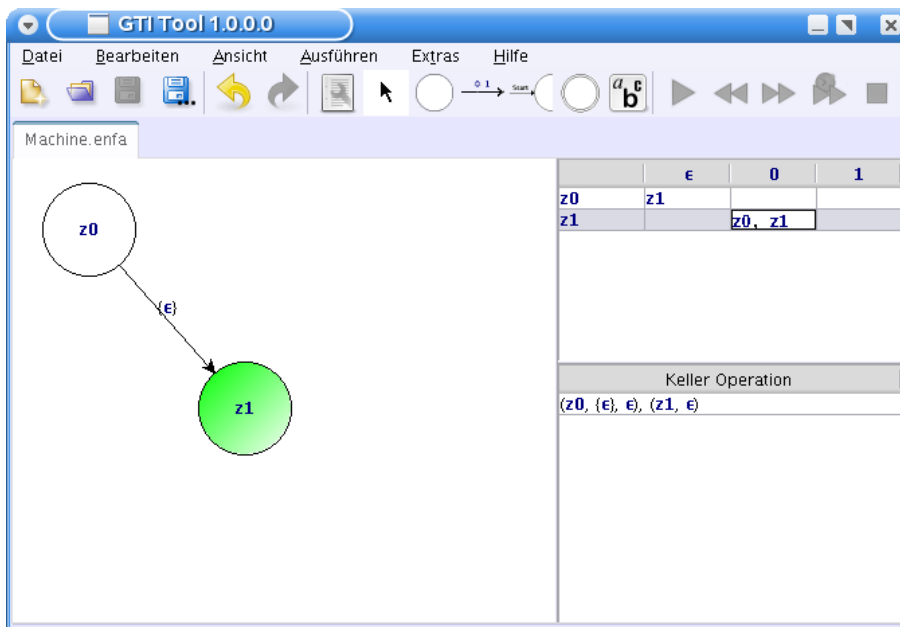


Abbildung 2.2: Automat - Übergangstabelle bearbeiten

### 2.3.2 Vorlage für...

Ein Automat kann als Vorlage für einen neuen Automaten benutzt werden. Dies sollte nicht mit „Umwandeln in...“ verwechselt werden. Es wird eine neue Datei angelegt, die alle Zustände und Übergänge des Ausgangsautomaten enthält. Allerdings kann man den Automatentypen neu festlegen.

### 2.3.3 Wort-Navigation

Wir können für einen Automaten eine Wort-Navigation starten. Das bedeutet, dass wir ein Wort eingeben können, welches wir den Automaten verarbeiten lassen wollen und können dann schrittweise vor und zurück navigieren. Dabei werden die aktuell aktiven Zustände und Übergänge farblich hervorgehoben. Am Ende können wir dann sehen, ob der Automat das von uns gewählte Wort akzeptiert oder nicht. Dies wird über ein Statusfeld im unteren Fenster signalisiert. Es gibt auch die Möglichkeit, zu erkennen, ob ein Wort zu einem früheren Zeitpunkt der Navigation akzeptiert würde. Dies kann man anhand eines Rahmens um das aktuelle Wort Feld erkennen. Ein roter Rahmen steht dafür, dass das Wort bis zum aktuellen Symbol von unserem Automaten nicht akzeptiert wird, ein grüner Rahmen hingegen gibt zu verstehen, dass es akzeptiert wird.

Während man sich im Wort-Navigations-Modus befindet, lässt sich der Automat nicht weiter bearbeiten. Das bedeutet, wir können Zustände und Übergänge weder anlegen noch löschen oder bearbeiten. Dies ist erst nach Verlassen dieses Modus wieder möglich.

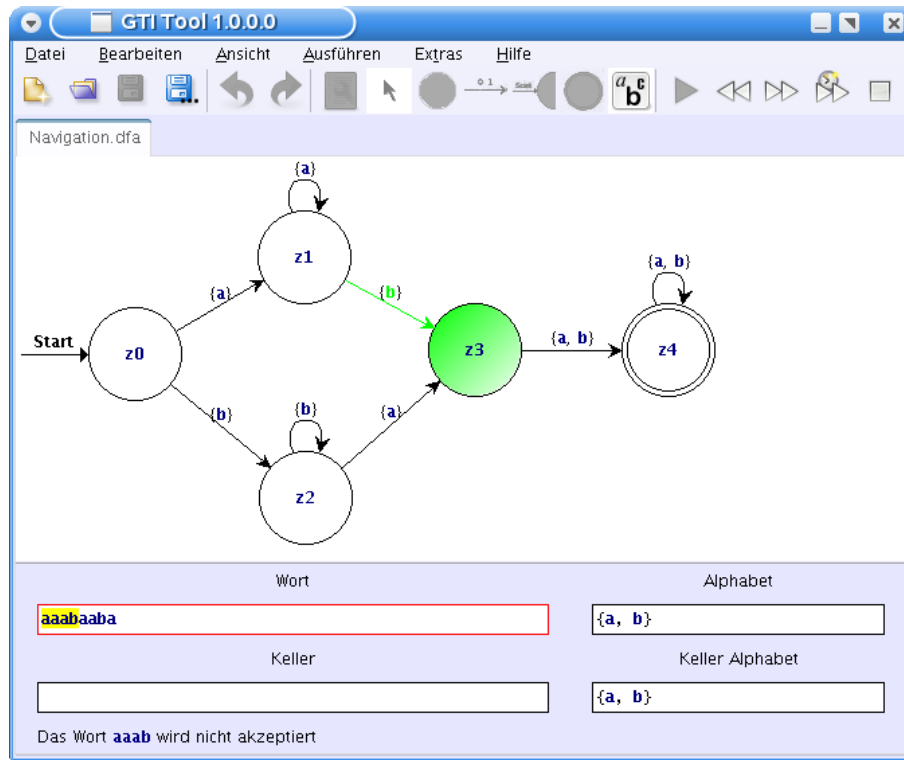


Abbildung 2.3: Automat - Wort-Navigation

Zur Wort-Navigation gelangt man über den Menüpunkt „Ausführen“ und „Wort eingeben“ oder direkt über den Button in der Toolbar. In dem Feld „Wort“ können wir jetzt ein Wort eingeben. Als Hilfestellung sehen wir rechts neben dem Eingabefeld das aktuelle Alphabet mit den gültigen Symbolen.

Nachdem wir das Wort eingegeben haben, startet man die Navigation über „Start“ in der Toolbar oder im Kontextmenü. Jetzt kann man mit „Schritt vor“ und „Schritt zurück“ durch das Wort navigieren. Es existiert auch noch ein automatischer Modus. Dabei wird nach einer kurzen Verzögerung das nächste Symbol gelesen. Dieser Modus lässt sich durch den Button „Automatische Schritte“ aktivieren und deaktivieren.

Wenn man wissen möchte, über welche Pfade man zu den momentan aktiven Zuständen gelangt ist, kann man sich dies über „Extras“ und „Zustands Pfad“ anzeigen lassen. Es öffnet sich ein neuer Dialog mit einer Tabelle, in welcher für jeden aktiven Zustand der aktuelle Pfad angezeigt wird. Dabei wird als erstes der kürzeste Pfad angezeigt, und dann alle weiteren.

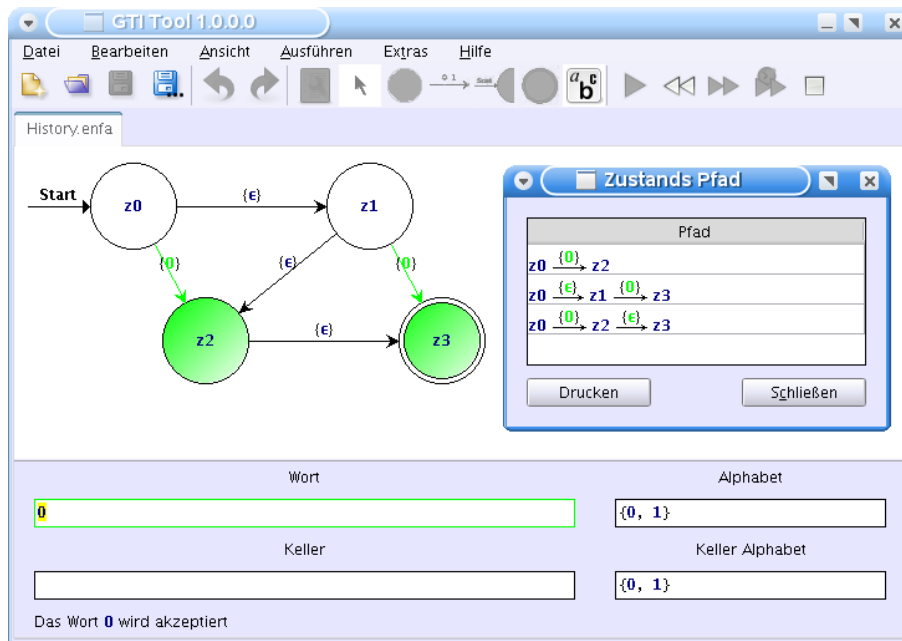


Abbildung 2.4: Automat - Minimieren

Um die Wort-Navigation zu beenden, klickt man einfach wieder auf den Button „Wort eingeben“ in der Toolbar oder man wählt im Menü „Ausführen“ den Punkt „Automat bearbeiten“. Jetzt befinden wir uns wieder im normalen Modus und wir können den Automaten wieder normal bearbeiten.

### 2.3.4 Minimieren

Es besteht auch die Möglichkeit sich aus einem DFA den minimalen Automaten berechnen zu lassen. Für diese Funktion darf der Automat allerdings keine Fehler mehr enthalten, was wir durch eine Validierung ausschließen können. Das Minimieren startet man über den Menüpunkt „Ausführen“ und „Minimieren“.

Es öffnet sich ein neuer Dialog mit einer Ansicht des Automaten und einer Tabelle. Der Automat sieht im Moment noch genau wie unser Ausgangsautomat aus. Wenn man allerdings jetzt auf „Schritt vor“ klickt, werden als erster Schritt

alle nicht erreichbaren Zustände entfernt. In der Outline wird angegeben, um welche Zustände es sich dabei handelt. Der zweite Schritt besteht darin, die erste Einteilung der Äquivalenzklassen vorzunehmen. Dazu unterteilen wir die Zustände des Automaten in akzeptierende und nicht akzeptierende Zustände. In der Automatenansicht werden immer die Zustände in der selben Farbe dargestellt, welche sich in einer Äquivalenzklasse befinden. Jetzt haben alle nicht akzeptierenden Zustände die gleiche Farbe. Gleiches gilt für alle akzeptierenden Zustände, welche natürlich in einer anderen Farbe dargestellt werden.

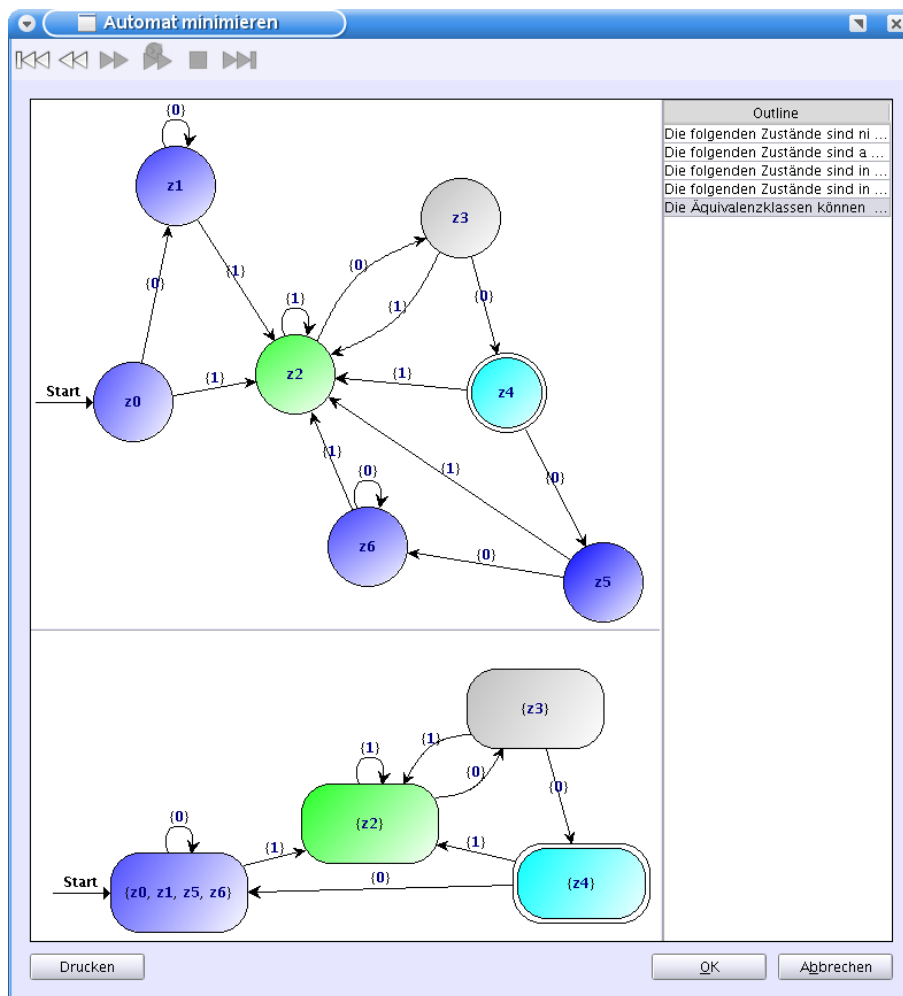


Abbildung 2.5: Automat - Minimieren

Man kann jetzt durch Klicken auf „Schritt vor“ die Äquivalenzklassen weiter verfeinern. In der Outline wird angegeben, welche Zustände in eine eigene Äquivalenzklasse gekommen sind, und diese werden jetzt auch in einer anderen Farbe dargestellt. Gleichzeitig werden auch die Übergänge hervorgehoben, welche bei diesem Verfeinerungsschritt eine Rolle gespielt haben. Dabei beziehen sich die hervorgehobenen Übergänge immer auf den aktuell ausgewählten Tabelleneintrag.

Auf diese Weise kann man den Automaten jetzt Schritt für Schritt weiter verfeinern, bis keine Verfeinerung der Äquivalenzklassen mehr möglich ist. Wenn das der Fall ist, wird auch eine zweite Automatenansicht eingeblendet, dem entstandenen minimierten Automaten. Durch bestätigen mit „OK“ wird jetzt eine neue Datei mit dem minimierten Automaten erstellt, mit welchem man dann weiter arbeiten kann.

Natürlich kann der Minimierungsprozess jederzeit abgebrochen werden. In diesem Fall wird keine neue Datei angelegt, man kann den Ausgangsautomaten weiter bearbeiten.

Wenn man in den Navigationsbereich des Dialogs schaut, stellt man fest, dass noch weitere Button verfügbar sind, auf welche jetzt noch nicht eingegangen wurde. Zunächst einmal gibt es die Button „Bis zum Ende vor“ und „An den Anfang zurück“. Mit „Bis zum Ende vor“ kann man alle Zwischenschritte überspringen, und gelangt direkt zur Ansicht des minimalen Automaten. „An den Anfang zurück“ bewirkt genau das Gegenteil, und springt zur Ausgangsansicht zurück. Dann gibt es noch einen „Schritt zurück“ Button, welcher einzelne Verfeinerungsschritte rückgängig macht. Und schlussendlich gibt es noch einen Button „Automatische Schritte“, welcher nach einer kurzen Verzögerung automatisch den nächsten Schritt macht, bis der Minimierungsprozess abgeschlossen ist. Diese Funktion kann durch den „Stop“ Button wieder deaktiviert werden.

Wenn man die neue Datei mit dem minimalen Automaten erstellen möchte, ohne sich die Minimierung im Detail anzuschauen, besteht jederzeit die Möglichkeit, den Dialog mit „OK“ zu bestätigen. Die Minimierung wird dann im Hintergrund beendet, man gelangt sofort zur Ansicht des neu entstandenen Automaten und kann mit diesem weiterarbeiten.

### 2.3.5 Umwandeln in . . .

Es besteht auch die Möglichkeit, zwischen verschiedenen Automatentypen umzuwandeln. Dabei stehen drei Umwandlungen zur Verfügung, die sich in den verwendeten Schritten unterscheiden.

- $\text{NDEA} \rightarrow \text{DEA}$
- $\epsilon\text{-NDEA} \rightarrow \text{DEA}$
- $\epsilon\text{-NDEA} \rightarrow \text{NDEA}$

Betrachten wir als erstes einen einfachen NDEA mit zwei Zuständen  $z_0$  und  $z_1$  und zwei Übergängen, der auf dem Alphabet  $\{a, b\}$  arbeitet. Der Zustand  $z_0$  ist ein Start-Zustand, der Zustand  $z_1$  ein akzeptierender Zustand. Der erste Übergang besteht zwischen  $z_0$  und  $z_1$  mit Symbol  $a$ , der zweite geht von  $z_0$  in sich selbst über, ebenfalls mit  $a$ . Der Automat erkennt somit Wörter von der Form  $a^n$  für  $n \geq 1$ . Über den Eintrag „Umwandeln in...“ im Menüeintrag „Ausführen“ kann ausgewählt werden, in welchen Automaten umgewandelt wird. Da wir einen NDEA angelegt haben, ist hier nur der DEA verfügbar. Wird dieser ausgewählt, steht die schon aus anderen Ansichten bekannte Oberfläche zur Verfügung. Auch hier kann in einzelnen Schritten, sowie automatisch navigiert werden. Wird der Dialog mit „OK“ bestätigt, steht in der Hauptansicht der umgewandelte Automat zur Verfügung. Mit „Abbrechen“ wird der Dialog einfach nur geschlossen.

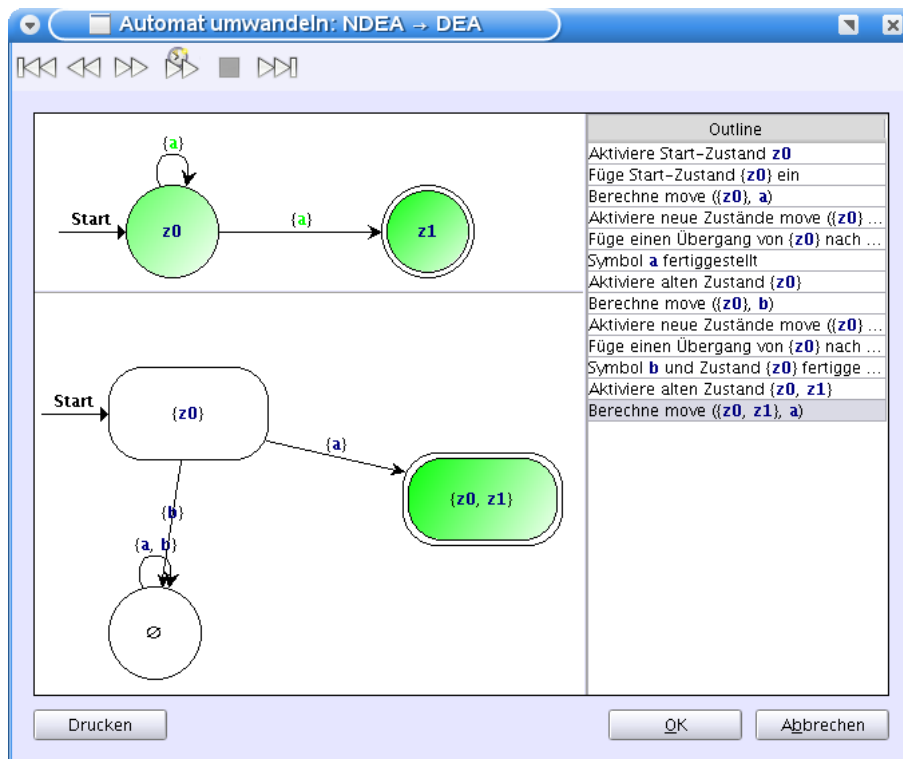


Abbildung 2.6: Automat - Umwandeln in...

Werden die Schritte einzeln durchgegangen, ergibt sich folgender Ablauf:

1. Der Start-Zustand **z0** wird aktiviert.
2. In der unteren Ansicht wird der Start-Zustand **{z0}** angelegt.
3. Die Übergänge mit dem ersten Symbol aus dem Alphabet, also **a**, werden hervorgehoben, um anzuzeigen, welche Zustände mit dem Symbol zu erreichen sind.
4. In unserem Beispiel sind das die Zustände **z0** und **z1**, die hervorgehoben werden.
5. In der unteren Ansicht wird ein Übergang von **{z0}** nach **{z0, z1}** mit Symbol **a** angelegt. Da dieser zweite Zustand in unserem Beispiel noch nicht existiert, wird er angelegt.
6. Dieser Schritt zeigt an, dass das Symbol **a** fertiggestellt ist.
7. Da für den Zustand **{z0}** noch nicht alle Symbole betrachtet wurden, wird er aktiviert.
8. Es wird die gleiche Berechnung wie bei Schritt drei durchgeführt, allerdings für Symbol **b**. Da es in unserem Beispiel keinen Übergang von den betroffenen Zuständen, bei uns nur **z0**, gibt, werden in der oberen Ansicht keine Übergänge hervorgehoben.
9. Wie in Schritt vier werden die mit Symbol **b** erreichbaren Zustände in der oberen Ansicht hervorgehoben. In unserem Beispiel existiert kein Übergang von **z0** mit Symbol **b**, deshalb werden in der oberen Ansicht keine Zustände hervorgehoben.
10. Da wir in diesem Beispiel in einen DEA umwandeln, wird auch für diesen Fall ein Zustand angelegt und zwar der nicht akzeptierende Zustand  $\emptyset$ . Wird dieser Zustand während der Wort-Navigation erreicht, kann das Wort nicht mehr akzeptiert werden. Aus diesem Grund wird direkt der Übergang von diesem Zustand in sich selbst mit allen Symbolen des Alphabets angelegt. Es wird ein Übergang mit Symbol **b** zu diesem Zustand  $\emptyset$  angelegt.
11. Wie in Schritt sechs wird angezeigt, dass das Symbol **b** fertig verarbeitet wurde. Da dieses Symbol das letzte im Alphabet ist, ist der Zustand **{z0}** fertiggestellt.
12. In diesem Schritt wird der nächste, bis jetzt noch nicht verarbeitete, Zustand **{z0, z1}** in der unteren Ansicht hervorgehoben. Gleichzeitig werden in der oberen Ansicht die entsprechenden Zustände **z0** und **z1** hervorgehoben.
13. Es folgt wieder das Hervorheben der Übergänge mit dem Symbol **a**, wie dies in Schritt drei der Fall war.

14. Wie in Schritt vier werden die Zustände **z0** und **z1** hervorgehoben.
15. In Schritt fünf wurde der Übergang zum Zustand  $\{\mathbf{z0}, \mathbf{z1}\}$  mit Symbol **a** angelegt, gleiches geschieht jetzt in diesem Schritt. Da wir allerdings im Moment den gleichen Zustand bearbeiten, wird der Übergang von diesem Zustand in sich selbst angelegt.
16. Das Symbol **a** ist fertiggestellt.
17. Da der Zustand  $\{\mathbf{z0}, \mathbf{z1}\}$  noch nicht fertiggestellt ist, wird er in beiden Ansichten wie gehabt hervorgehoben.
18. Alle Übergänge, die von den Zuständen **z0** und **z1** mit Symbol **b** ausgehen, werden hervorgehoben. In unserem Beispiel kommt das nicht vor.
19. Die mit dem Symbol **b** erreichbaren Zustände werden hervorgehoben, in unserem Beispiel ist das wieder kein Zustand.
20. Deshalb wird in diesem Schritt wieder ein Übergang mit Symbol **b** zu dem Zustand  $\emptyset$  angelegt.
21. Der Zustand  $\{\mathbf{z0}, \mathbf{z1}\}$  ist mit dem letzten Symbol **b** fertiggestellt. Da keine weiteren Zustände hinzugekommen sind, ist die Umwandlung fertiggestellt.

Bei Betrachten des entstehenden DFA's erkennt man, dass dieser die gleichen Wörter erkennt wie der ursprüngliche NFA. Der Dialog kann mit „OK“ bestätigt werden, womit ein neuer DFA angelegt wird. Es ist jetzt möglich, die Namen der Zustände neu zu vergeben, wenn man mit dem Automaten weiterarbeiten möchte und nicht die längeren Namen dargestellt haben möchte. Dazu kann man im Menü „Extras“ den Eintrag „Zustandsnamen neu vergeben“ anklicken, wodurch die Zustandsnamen neu, nach dem Muster **z0**, **z1** . . . , vergeben werden.

Zu den anderen zwei Umwandlungarten gibt es noch kleinere Änderungen, die bei dem durchgespielten Beispiel nicht betrachtet werden konnten. Wird ein  $\epsilon$ -NDEA in einen NDEA umgewandelt, entfällt das Anlegen des Zustandes  $\emptyset$ , es wird einfach kein Übergang mit dem entsprechenden Symbol angelegt, da bei einem NFA nicht alle Symbole vorhanden sein müssen.

Wird ein  $\epsilon$ -NDEA umgewandelt, kommen jeweils zwei Schritte dazu. So wird nach dem Aktivieren der alten Zustände, bei unserem Beispiel die Schritte eins, sieben, etc., der  $\epsilon$ -Abschluss berechnet, es werden also auch alle Zustände aktiviert, die mit  $\epsilon$ -Übergängen zu erreichen sind. Gleiches gilt für das Anzeigen der neuen Zustände, bei unserem Beispiel in den Schritten vier, neun etc., auch hier werden in einem einzelnen Schritt alle Zustände hervorgehoben, die mit  $\epsilon$ -Übergängen zu erreichen sind.

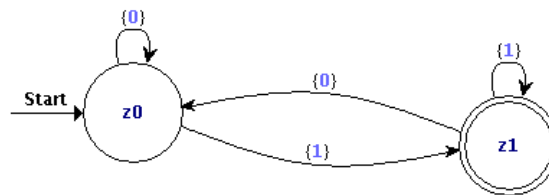


### 2.3.6 Umwandeln in... (Potenzmenge)

Der einzige Unterschied zwischen den Menüpunkten „Umwandeln in...“ und „Umwandeln in... (Potenzmenge)“ ist, dass alle Zustände bei Verwendung der Potenzmenge bereits zu Beginn angelegt werden, auch solche, die nicht benötigt werden, die also nicht erreichbar sind. In dem in Abschnitt 2.3.5 gewählten Beispiel wäre das der Zustand  $\{z1\}$ , da dieser nicht zu erreichen ist und nur einen Übergang mit allen Symbolen zu dem Zustand  $\emptyset$  hat. Wie solche Zustände im Anschluss entfernt werden können, kann im Abschnitt 2.3.8 nachgelesen werden.

### 2.3.7 Umwandeln in regulären Ausdruck

Es besteht die Möglichkeit einen DFA in einen regulären Ausdruck umzuwandeln. Dazu betrachten wir den Beispielautomat aus der Vorlesung:



Für diesen Beispielautomat liefere der Algorithmus wie folgt ab:

- Zunächst wird ein Ausdruck für den gesamten Automaten angelegt. In unserem Fall wurde  $L_{z_0 z_1}^3 = L_{z_0 z_1}^2 | L_{z_0 z_1}^2 (L_{z_1 z_1}^2)^* L_{z_1 z_1}^2$  angelegt.
- Nun werden die einzelnen Sprachen konkretisiert. In unserem Beispiel zunächst  $L_{z_0 z_1}^2$ . Dies wird in  $L_{z_0 z_1}^1 | L_{z_0 z_0}^1 (L_{z_0 z_0}^1)^* L_{z_0 z_1}^1$  umgewandelt.
- Im nächsten Schritt wird dann  $L_{z_0 z_1}^1$  in **1** umgewandelt.
- Dies wird solange fortgesetzt bis alle Sprachen zu regulären Ausdrücken umgewandelt sind.
- In unserem Fall ist das Ergebnis  $((1|0|\epsilon)(0|\epsilon)^*1)|(1|(0|\epsilon)(0|\epsilon)^*1)((1|\epsilon)|0(0|\epsilon)^*1) * ((1|\epsilon)|0(0|\epsilon)^*1)$

**Achtung:** Wie im Beispiel zu sehen, wird der reguläre Ausdruck selbst bei einem kleinen Automaten schon relativ groß. Daher ist bei größeren Automaten mit erheblichem Rechenaufwand zu rechnen.

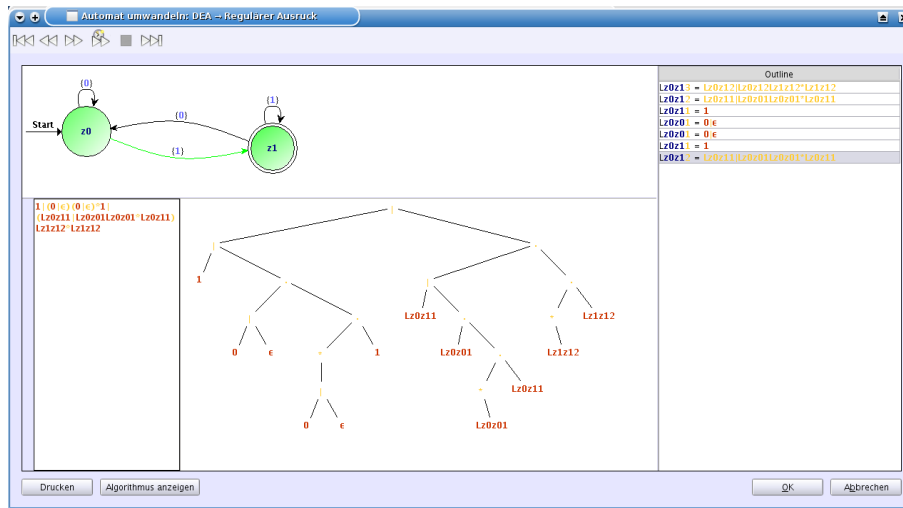


Abbildung 2.7: Automat - Umwandeln in regulären Ausdruck

### 2.3.8 Erreichbare Zustände

Über den Menüeintrag „Extras“ kann der Punkt „Erreichbare Zustände“ ausgewählt werden. In dem erscheinenden Dialog kann, wie in den anderen auch, durch die einzelnen Schritte navigiert werden. Es geht hierbei darum, in möglichst kleinen Schritten zu erkennen, welche Zustände erreichbar sind und welche nicht. Der verwendete Algorithmus geht dabei in folgenden drei Schritten vor:

- Ein Zustand, der noch nicht berechnet wurde, wird ausgewählt (am Anfang wird mit dem Start Zustand begonnen)
- Die von dem ausgewählten Zustand aus zu erreichenden Zustände werden hervorgehoben, und falls sie noch nicht berechnet wurden, zu der Menge der noch zu berechnenden Zustände hinzugefügt
- Die bis jetzt erreichbaren Zustände werden hervorgehoben. Die Menge der noch nicht berechneten Zustände angezeigt

Der Algorithmus läuft so lange, bis die Menge der noch nicht berechneten Zustände leer ist. Am Ende werden alle erreichbaren Zustände farblich hervorgehoben. Wie in den anderen Dialogen besteht jederzeit die Möglichkeit, den Dialog abubrechen, wobei er dann einfach geschlossen wird. Man hat auch jederzeit die Möglichkeit, mit einem Klick auf „OK“ die komplette Berechnung durchzuführen und einen Automaten anzulegen, der nur noch die erreichbaren Zustände enthält.

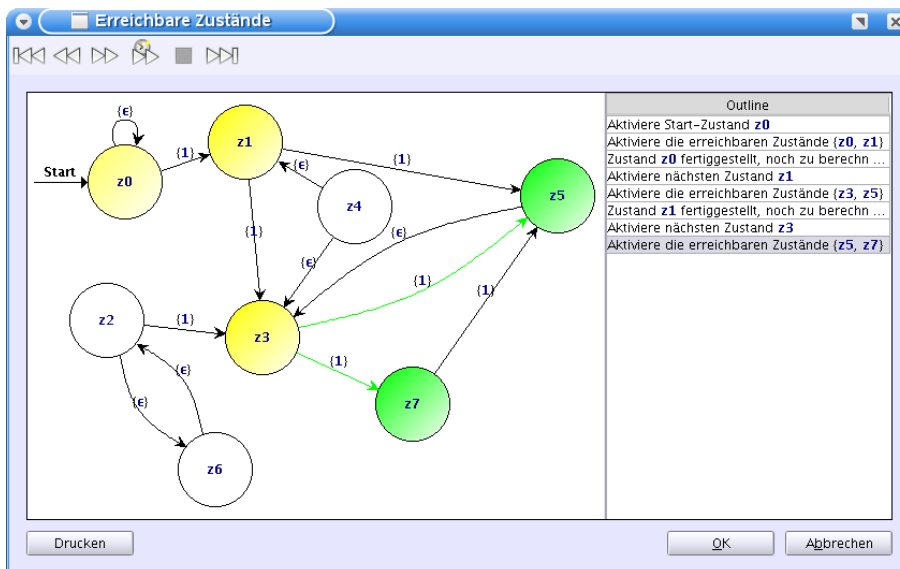


Abbildung 2.8: Automat - Erreichbare Zustände

## Kapitel 3

# Wie erstelle ich einen Kellerautomaten?

Wie ein normaler Automat angelegt wird, wurde bereits besprochen. In diesem Kapitel geht es darum, einen Kellerautomaten anzulegen, der die Sprache  $a^n b^n$  für  $n \geq 0$  erkennt. Als erstes muss der Neu-Dialog geöffnet werden, im ersten Tab muss „Automat“ gewählt werden, anschließend der „Kellerautomat“. Für das Beispiel muss sowohl als Eingabe-, wie auch als Kelleralphabet das Alphabet  $\{\mathbf{a}, \mathbf{b}\}$  gewählt werden. Dazu kann das Alphabet bei beiden Eingabefeldern einzeln eingestellt werden, oder nur bei „Eingabe Alphabet“, wobei dann der Haken bei „Keller Alphabet“ entfernt werden muss. Dadurch werden automatisch die gleichen Alphabete verwendet.

Ist die Datei angelegt, muss überlegt werden, welche Zustände und Übergänge benötigt werden, um die Sprache zu erkennen. Dabei kommt man dazu, dass mindestens folgende Zustände und Übergänge nötig sind, um die Sprache zu erkennen:

- Ein Zustand **z0**, der sowohl Start-, wie auch akzeptierenden Zustand ist
- Ein Zustand **z1**
- Ein akzeptierender Zustand **z2**
- Einen Übergang von **z0** nach **z1**, der das erste Symbol **a** vom Eingabewort liest und auf den Keller schreibt
- Einen Übergang von **z1** nach **z1**, der alle weiteren Symbole **a** vom Eingabewort liest und auf den Keller schreibt
- Einen Übergang von **z1** nach **z2**, der das erste **b** vom Eingabewort liest und das oberste **a** vom Keller löscht
- Einen Übergang von **z2** nach **z2**, der alle weiteren Symbole **b** vom Eingabewort liest und die entsprechenden Symbole **a** vom Keller löscht

Damit das Wort akzeptiert wird, muss sich somit bei der Wort-Navigation ergeben, dass das Eingabewort komplett gelesen wurde. Zusätzlich muss sich der Automat entweder im Zustand **z0** oder im Zustand **z2** befinden, wobei der Keller leer sein muss.

Wie die Zustände und Übergänge angelegt werden, wurde bereits in diesem Handbuch beschrieben, deshalb wird an dieser Stelle nur auf die Besonderheiten des Kellerautomaten eingegangen. Es werden insgesamt vier Übergänge benötigt, um die Sprache zu erkennen. Alle Übergänge müssen Veränderungen am Keller durchführen. Wir nehmen als Beispiel den Übergang von **z0** nach **z1**, bei dem das Symbol **a** vom Eingabeband gelesen und auf den Keller geschrieben werden muss. Beim Anlegen oder Editieren dieses Übergangs besteht die Möglichkeit, das „Keller Lese-Wort“ und das „Keller Schreib-Wort“ zu editieren. In diesem Fall soll das Symbol **a** auf den Keller geschrieben werden, es muss also beim „Keller Schreib-Wort“ das Symbol **a** eingetragen werden. Das „Keller Lese-Wort“ muss leer bleiben, da nichts vom Keller gelesen werden soll.

### 3.1 Navigation

Wurden auf diese Weise alle Zustände und Übergänge angelegt, kann der Automat validiert werden und sollte keine Fehler mehr aufweisen. Bei der Wort-Navigation ist jetzt nicht nur das Eingabewort interessant, sondern auch der dargestellte Keller. Wird zum Beispiel das Wort **aaabbb** gewählt, wird im ersten Schritt das Symbol **a** gelesen und auf den Keller geschrieben. Anschließend werden die restlichen zwei Symbole **a** auf den Keller gepackt und anschließend durch die Übergänge von **z1** nach **z2** bzw. von **z2** nach **z2** wieder entfernt. Am Ende wurde das komplette Wort **aaabbb** gelesen, der Automat befindet sich im Zustand **z2** und der Keller ist leer, das Wort wird somit akzeptiert.

### 3.2 Kellerautomat aus Grammatik erstellen

Im folgenden soll ein Sonderfall der Navigation mit dem Kellerautomat betrachtet werden. Dazu muss die Grammatik erstellt werden, die Wörter der Form  $a^n b^n$  für  $n \geq 0$  erkennt. Wie in Kapitel 1 angegeben, müssen die beiden Produktionen  $E \rightarrow \epsilon$  und  $E \rightarrow aEb$  angelegt werden. Anschließend wandeln wir die Grammatik in einen Kellerautomaten um, dies erfolgt mit „Umwandeln in...“ im Menüpunkt „Ausführen“. Bei dieser Art der Umwandlung entsteht immer ein Kellerautomat mit zwei Zuständen **s** und **f**. Der Unterschied zu normalen Automaten ist, dass jetzt vom Zustand **f** mehrere Übergänge in sich selbst existieren, so dass eine Beschriftung des Überganges keinen Sinn hätte. Welche Übergänge genau vorhanden sind, kann in der „Keller Operationen“ Tabelle nachgesehen werden.

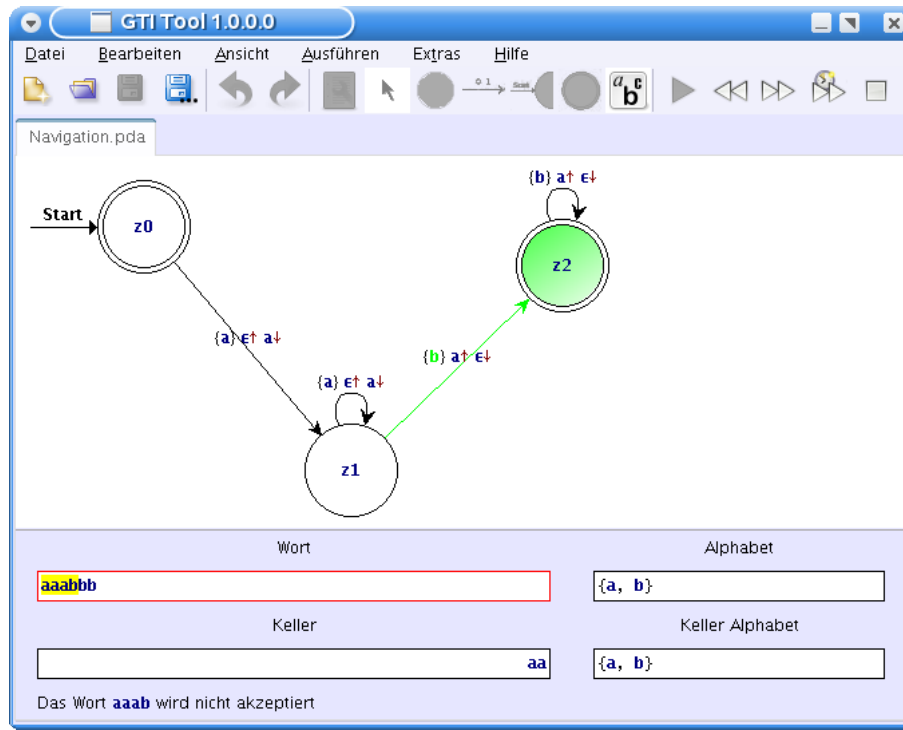


Abbildung 3.1: Automat - Kellerautomat Navigation

Wird für diesen Automaten die Wort-Navigation mit dem Wort **aaabbb** gestartet, ergibt sich im zweiten Schritt der Navigation, dass der Übergang, der verwendet werden kann, nicht mehr eindeutig ist. Oben auf dem Keller befindet sich das Nichtterminalzeichen **E**, welches entweder durch  $\epsilon$  oder durch **aEb** ersetzt werden könnte. Da wir das Wort **aaabbb** erkennen wollen, muss der Übergang **E**  $\rightarrow$  **aEb** ausgewählt werden. Der nächste Schritt ist wieder eindeutig, da nur der Übergang ausgewählt werden kann, der **a** vom Eingabewort liest und **a** vom Keller entfernt. Im nächsten Schritt muss wieder ausgewählt werden, welcher Übergang verwendet werden soll. Da als nächstes Terminalzeichen wieder ein **a** auf dem Eingabewort steht, muss wieder der Übergang **E**  $\rightarrow$  **aEb** ausgewählt werden. Genau wie zuvor ist der nächste Schritt wieder eindeutig. Da als nächstes das Terminalzeichen **b** auf dem Keller steht, muss diesmal der Übergang **E**  $\rightarrow$   $\epsilon$  ausgewählt werden. Durch diesen Übergang wird das oben auf dem Keller stehende Nichtterminalzeichen **E** entfernt. Mit den nächsten zwei eindeutigen Schritten werden die Terminalzeichen **b** auf dem Eingabewort gelesen und vom Keller entfernt. Da am Ende der Keller leer ist und der Zustand **f** aktiv ist, wird das Wort akzeptiert.

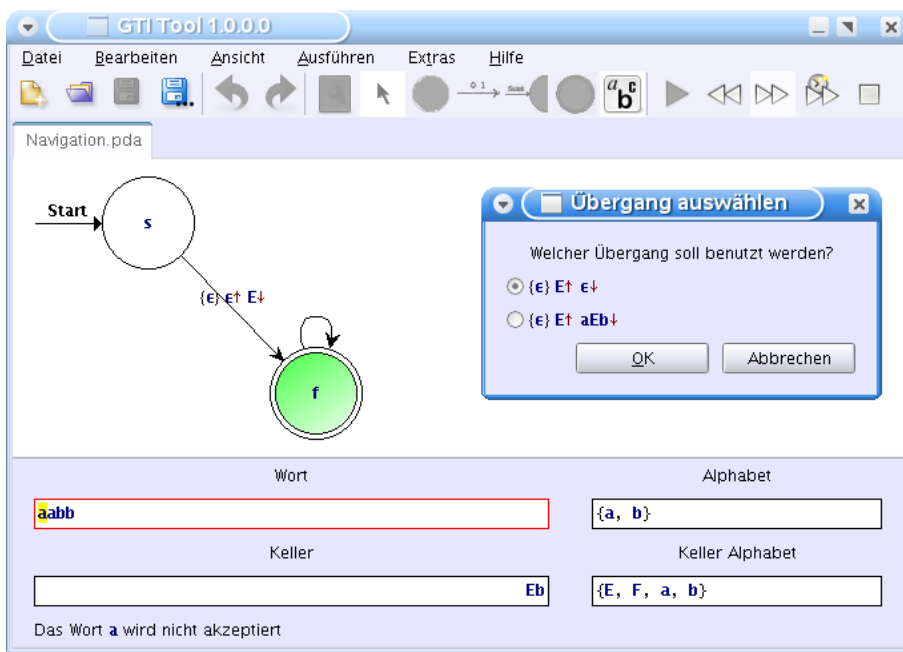


Abbildung 3.2: Automat - Nicht deterministische Kellerautomat Navigation

## Kapitel 4

# Wie erstelle ich einen regulären Ausdruck?

Wir wollen uns dies anhand des folgenden Beispiels anschauen:  $(a|b)^*abb$

### 4.1 Neue .regex Datei anlegen

Zu Beginn müssen wir dazu, wie auch beim Automat und bei Grammatiken, den „Neu...“ Dialog öffnen. Zu finden ist dieser in der Toolbar oder im Menüeintrag „Datei“. Dort wählen wir „Regulärer Ausdruck“ an und klicken auf „Weiter“.

Danach müssen wir im nächsten Dialog nur noch ein Alphabet festlegen, auf welchem der reguläre Ausdruck definiert ist. Hier ist es auch möglich eine Klasse von Zeichen anzugeben, zum Beispiel  $[a-z]$ .

In unserem Beispiel würde also ein Alphabet bestehend aus **a** und **b** genügen. Man kann dieses Alphabet auch im Nachhinein noch über den Button „Dokument editieren“ verändern. Nachdem wir ein Alphabet festgelegt haben bestätigen wir dies mit „Fertig“.

### 4.2 Regulären Ausdruck eingeben

Nun ist ein Bildschirm wie in Abbildung 4.1 zu sehen.

Oben rechts ist das Eingabefeld für den regulären Ausdruck. Darunter findet man nochmal zur Orientierung das Alphabet, dass wir gerade eingegeben haben. Wie bereits oben erwähnt kann das Alphabet nur über den Button „Dokument editieren“ verändert werden. Im Eingabefeld tragen wir nun unseren Beispielausdruck  $(a|b)^*abb$  ein.

Direkt bei der Eingabe wird im Graph-Fenster in der Mitte der reguläre Ausdruck als Baum dargestellt.



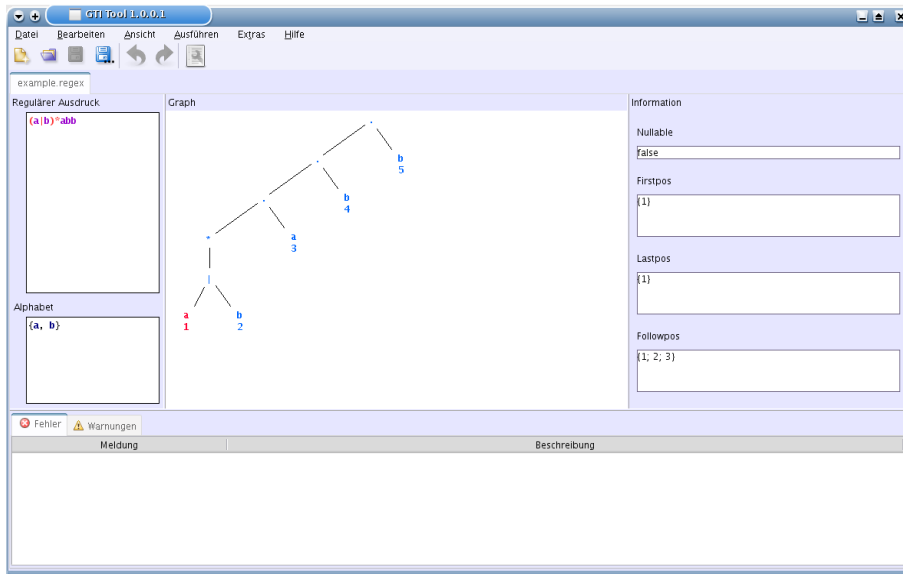


Abbildung 4.1: Regulärer Ausdruck - Eingabefenster

**Beachte:** Um ein  $\epsilon$  einzugeben kann man sowohl das Zeichen an sich eingeben, aber, da die meisten Anwender nicht über ein griechisches Tastaturlayout verfügen, auch indem man die Zeichenfolge epsilon eingibt.

### 4.3 Das Informationsfenster

Im Menü findet sich unter „Ansicht“ der Eintrag „Regulärer Ausdruck Info“. Wenn dieser Punkt aktiviert wird, erscheint auf der rechten Seite, neben dem Graph-Fenster, ein Informationsfenster.

Dieses Informationsfenster beinhaltet die Informationen zum aktuell ausgewählten Knoten im Graph-Fenster. Diese Informationen bestehen aus den in der Vorlesung Compilerbau I definierten Funktionen nullable, firstpos, lastpos und followpos. Außerdem wird jetzt auch unter den Blattknoten die jeweilige Position angegeben.

### 4.4 Was macht man mit regulären Ausdrücken?

#### 4.4.1 Übersetzen in Kernsyntax

Ein regulärer Ausdruck kann im GTI Tool folgende Erweiterungen enthalten:

- $\alpha+ = \alpha\alpha^*$

- $\alpha^? = \alpha|\epsilon$
- $[\alpha_1 \dots \alpha_n] = (\alpha_1 | \dots | \alpha_n)$
- $[\alpha_i - \alpha_k] = [\alpha_i \alpha_{i+1} \dots \alpha_n]$  wenn  $i < k$  und  $i, k \in \mathbb{N}$

Also wäre zum Beispiel auch **[a-z]** ein regulärer Ausdruck, wenn das Alphabet die Zeichen von **a** bis **z** umfasst.

Wenn man nun einen Ausdruck mit diesen Erweiterungen erstellt hat, kann man diesen „in Kernsyntax übersetzen“. Dies funktioniert über den Menüpunkt „Ausführen“ → „Übersetzen in Kernsyntax“.

Dann wird eine neue Datei erstellt, die den Ausdruck in Kernsyntax beinhaltet. Dabei wird das Alphabet einfach übernommen.

#### 4.4.2 Einen regulären Ausdruck umwandeln

Einen regulären Ausdruck kann man in einen Automaten umwandeln. Dazu gibt es drei verschiedene Algorithmen. Zum einen der „Thompson-Algorithmus“, der aus dem regulären Ausdruck einen  $\epsilon$ -NDEA konstruiert und zwei Algorithmen, um aus dem regulären Ausdruck einen NDEA bzw. direkt einen DEA zu konstruieren.

##### Regulären Ausdruck in $\epsilon$ -NDEA umwandeln

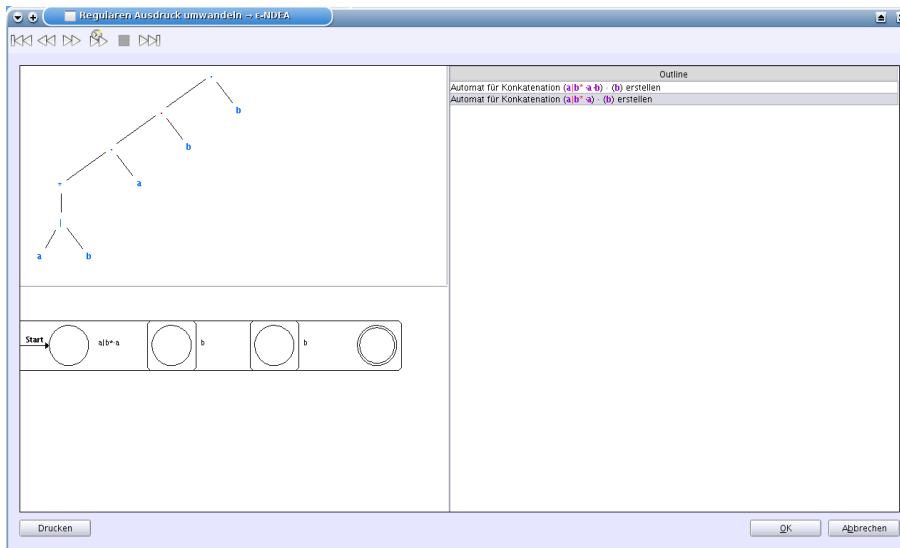
Im Menü findet sich unter „Ausführen“ der Punkt „Umwandeln in...“ → „ $\epsilon$ -NDEA“. Es öffnet sich ein Fenster, in welchem in der oberen Ansicht der reguläre Ausdruck als Graph dargestellt ist. Links ist die Outline zu sehen, in der die Informationen über den aktuellen Schritt zu finden sind und unten ist ein noch leerer Automat in welchem im Verlauf des Algorithmus der  $\epsilon$ -NDEA erstellt wird.

Oben befindet sich die Navigationsleiste, mit der durch den Algorithmus navigiert werden kann.

Der Algorithmus geht im Prinzip den Graph von oben nach unten durch und legt dabei jeweils eine Blackbox für alle nicht Blattknoten an.

Wenn wir nun den Algorithmus für unser Beispiel durchgehen, kommen wir zu folgender Reihenfolge:

1. Zunächst wird der Automat für die Konkatenation **(a|b)\*ab·b**, welche im Graph-Fenster markiert wird, erstellt. Dieser besteht aus zwei Blackboxen, eine für den regulären Ausdruck **(a|b)\*ab** und eine für das Symbol **b**. Es werden dabei auch der Start- und der Endzustand des Automaten erstellt. Diese ändern sich auch während des Algorithmus nicht mehr.
2. Aus der Blackbox für **(a|b)\*ab** werden zwei Blackboxen: Eine für **(a|b)\*a** und eine für **b**.
3. Aus der Blackbox für **(a|b)\*a** werden zwei Blackboxen: Eine für **(a|b)\*** und eine für **a**.

Abbildung 4.2: Regulärer Ausdruck - Umwandeln in  $\epsilon$ -NDEA

4. Aus der Blackbox für  $(a|b)^*$  wird der Automat für einen Kleene-Abschluss konstruiert und dadurch eine Blackbox für den Ausdruck  $a|b$  erstellt.
5. Aus der Blackbox für  $a|b$  wird der Automat für eine Disjunktion erstellt und damit zwei Blackboxen für  $a$  und  $b$ .
6. Nun werden nacheinander von links nach rechts (in der Baumansicht des regulären Ausdruck) die Blackboxen für die einzelnen Symbole durch einen Übergang mit dem jeweiligen Symbol gebildet.

**Achtung!** Vor der Umwandlung wird der reguläre Ausdruck in Kernsyntax umgewandelt. Außer Charakterklassen, diese bleiben erhalten und werden im Algorithmus wie ein Symbol behandelt.

### Regulären Ausdruck in DEA umwandeln

Im Menü findet sich unter „Ausführen“ der Punkt „Umwandeln in...“ → „DEA“. Es öffnet sich ein Fenster, das sehr ähnlich dem Fenster der Umwandlung in einen  $\epsilon$ -NDEA ist. Zusätzlich ist hier aber neben dem Graph-Fenster auch noch das Informationsfenster für die Knoten im Graphen eingeblendet. Außerdem wurde der Ausdruck um ein Endzeichen „#“ erweitert.

Wenn wir den Algorithmus an unserem Beispiel durchgehen, erhalten wir folgenden Ablauf:

- Zunächst wird der Startzustand, aus allen Positionen die in *firstpos* des gesamten Ausdruck enthalten sind, gebildet. Die Positionen kann man sich

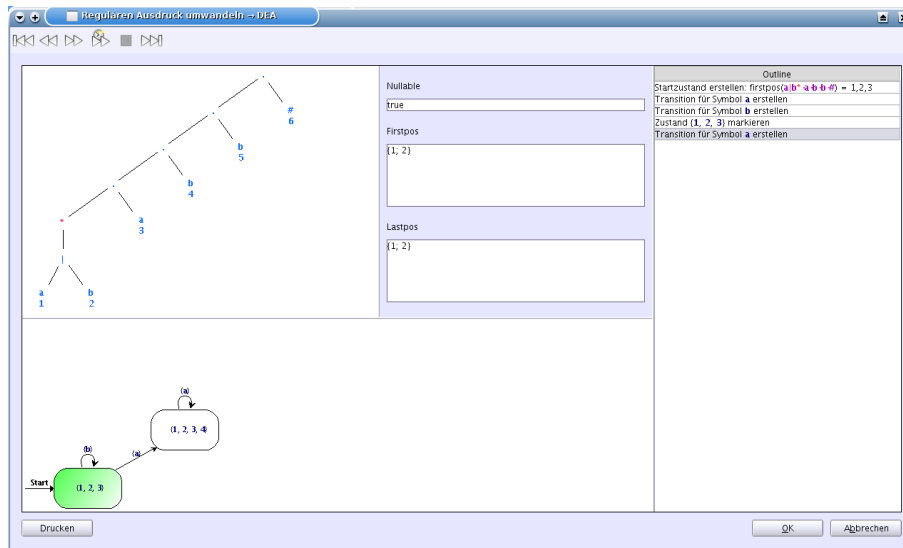


Abbildung 4.3: Regulärer Ausdruck - Umwandeln in DEA

auch anschauen, indem man den obersten Knoten im Graph markiert. In unserem Beispiel wird der Zustand  $\{1,2,3\}$  als Startzustand zum Automaten hinzugefügt.

- Danach läuft der Algorithmus ab. Es wird für jedes Symbol, das im Alphabet vorhanden ist, geschaut, ob man eine weitere Position erreicht (*followpos*). In diesem Fall wird für das Symbol **a** die Position **4** als Nachfolger gefunden. Daher wird ein Übergang **a** vom Startzustand zu einem neuen Zustand  $\{4\}$  angelegt.
- Als nächstes kommt das Symbol **b**. Für dieses wird ein Übergang zum Zustand  $\{1,2,3\}$  angelegt.
- Nun wird dieser Zustand markiert, da für ihn, wenn unser Alphabet nur die Symbole **a** und **b** umfasst, alle Symbole durchgegangen wurden.
- Nun kommen wir zum nächsten Zustand  $\{1,2,3,4\}$ . Für diesen wird das Symbol **a** getestet und es wird ein Übergang auf sich selbst angelegt.
- Danach ist **b** an der Reihe. Für welches ein Übergang in einen neuen Zustand  $\{1,2,3,5\}$  angelegt wird.
- Dies geht dann immer so weiter, bis der letzte Zustand  $\{1,2,3,6\}$  abgearbeitet wurde. Es gibt nun keine unmarkierten Zustände mehr und es werden die Endzustände gesucht. Diese sind alle, die die Position aus lastpos des gesamten Ausdrucks beinhalten. In unserem Fall ist dies der zuletzt erstellte Zustand  $\{1,2,3,6\}$ .

**Achtung 1:** Auch hier wird, wie bei der Umwandlung in einen  $\epsilon$ -NDEA, der Ausdruck erst in Kernsyntax übersetzt, nur dass er hier komplett übersetzt wird.

**Achtung 2:** Wenn das Alphabet viele Symbole umfasst, kann es zu langen Berechnungszeiten kommen, daher empfiehlt es sich, dass Alphabet nicht zu groß zu wählen.

### Regulären Ausdruck in NDEA umwandeln

Im Menü findet sich unter „Ausführen“ der Punkt „Umwandeln in...“  $\rightarrow$  „NDEA“. Es öffnet sich ein Fenster, das sehr ähnlich dem Fenster der Umwandlung in einen DEA ist. Außerdem wurde auch hier der Ausdruck um ein Endzeichen „#“ erweitert.

Der Algorithmus läuft im Prinzip genau so wie der Algorithmus zum Erstellen eines DEA, nur dass hier nicht Positionsmengen als Zustände dienen, sondern einzelne Positionen.

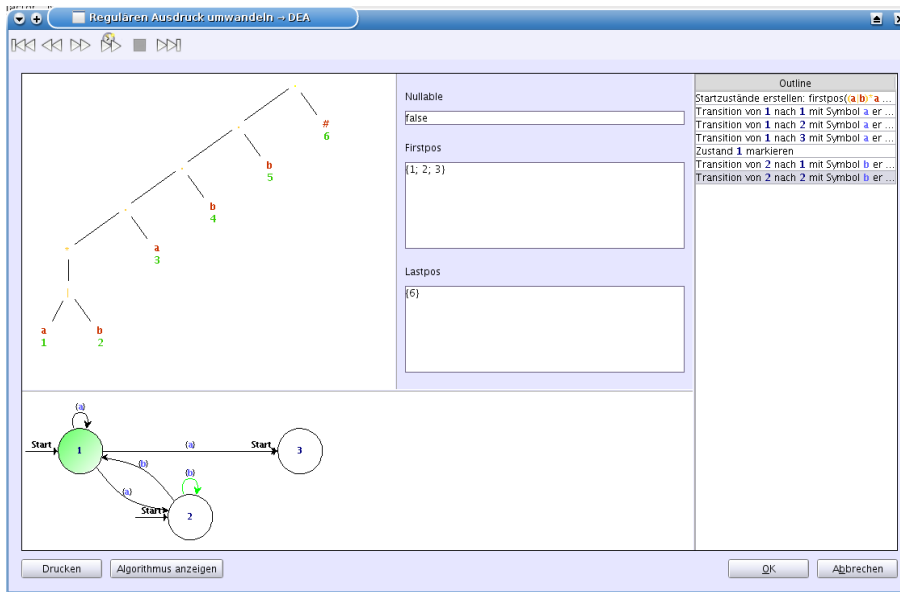


Abbildung 4.4: Regulärer Ausdruck - Umwandeln in NDEA

Wenn wir den Algorithmus an unserem Beispiel durchgehen, erhalten wir folgenden Ablauf:

- Zunächst wird für jede Position aus *firstpos* des gesamten Ausdrucks ein Startzustand angelegt.

- Danach läuft der Algorithmus ab. Es wird für jeden Zustand (also für die Position) die Funktion *followpos* generiert, diese kann man sich ansehen, indem man im regulären Ausdruck auf die jeweilige Position klickt. Für jede Position in *followpos* wird dann ein Zustand angelegt und ein Übergang mit dem Symbol an der Position des ausgehenden Zustandes. In unserem Beispiel würde zunächst ein Übergang vom Zustand **1** zu sich selbst mit Symbol **a** angelegt.
- Im Anschluss wird von **1** zu **2** sowie **3** jeweils ein Übergang mit **a** gemacht.
- Dann wird der Zustand **1** markiert.
- Dies wiederholt sich für die anderen Zustände.
- Zum Schluss wird der Zustand mit der Position des Endmarkers zum Endzustand. In unserem Beispiel ist das der Zustand **6**.

### 4.4.3 Exportieren nach L<sup>A</sup>T<sub>E</sub>X

Man kann einen regulären Ausdruck auch nach L<sup>A</sup>T<sub>E</sub>X exportieren.

Dazu gibt es im Menü den Eintrag „Datei“ → „Exportieren nach LaTeX“. Nach einem Klick auf den Eintrag erscheint ein Dialog, in welchem man den Ort und den Dateinamen der zu erstellenden L<sup>A</sup>T<sub>E</sub>X-Datei angibt.

Dann wird die Datei erstellt. Dies würde in unserem Beispiel zu folgendem Code führen:

```
\documentclass[a4paper,11pt]{article}
\usepackage{german}
\usepackage[utf8]{inputenc}
\usepackage{tree-dvips}
\begin{document}
%Erstellen des Tabulars fuer den Baum
\begin{tabular}{cccccccc}
%Erstellen der Knoten fuer den Baum
& & & & & & \node{r0}{\$\dot$}\\[4ex]
& & & & & \node{r1}{\$\dot$} & & & \node{r2}{b}\\[4ex]
& & & \node{r3}{\$\dot$} & & & & \node{r4}{b}\\[4ex]
& \node{r5}{*} & & & & & \node{r6}{a}\\[4ex]
& \node{r7}{\$|$\}\\[4ex]
\node{r8}{a} & & & & & \node{r9}{b}
\end{tabular}
%Verbinden der Knoten
\nodeconnect{r0}{r1}
\nodeconnect{r1}{r3}
\nodeconnect{r3}{r5}
\nodeconnect{r5}{r7}
\nodeconnect{r7}{r8}
\nodeconnect{r7}{r9}
```

```
\nodeconnect{r3}{r6}  
\nodeconnect{r1}{r4}  
\nodeconnect{r0}{r2}  
  
\end{document}
```

Dies ist ein komplettes  $\text{\LaTeX}$ -Dokument, das heißt diese Datei kann so für sich genutzt werden. Man kann aber auch nur den Inhalt von „document“ in eine andere  $\text{\LaTeX}$ -Datei einfügen.

**Achtung 1:** Es wird das Package „tree-dvips“ für  $\text{\LaTeX}$  genutzt. Das Package muss daher auf dem Rechner vorhanden sein, um die Datei nutzen zu können.

**Achtung 2:** Wenn man den Inhalt in anderen  $\text{\LaTeX}$ -Dokumenten nutzen will, muss unbedingt die Zeile

```
\usepackage{tree-dvips}
```

zur Präambel des Dokuments hinzugefügt werden.

**Achtung 3:** Wenn die erstellte  $\text{\LaTeX}$ -Datei mit `pdflatex` ausgeführt wird, werden die Kanten nicht mit gezeichnet! Daher sollte man zunächst „`latex datei.tex`“ ausführen und danach „`dvipdf datei.dvi`“ oder ähnliches nutzen.

# Kapitel 5

## Einstellungen

Die Oberfläche des GTI Tools kann den eigenen Bedürfnissen angepasst werden. Dazu kann man die Einstellungen im Menüpunkt „Bearbeiten“ öffnen. Alle Einstellungen werden erst bei Klick auf „Übernehmen“ oder „OK“ übernommen. Per Klick auf „Standard“ kann man die Einstellungen auf die Standardwerte zurücksetzen.

### 5.1 Oberfläche anpassen

Auf dem Tab „Allgemein“ kann die Sprache eingestellt werden, zur Auswahl stehen im Moment Deutsch und Englisch. Ist „Default“ ausgewählt, wird die Systemsprache verwendet, sollte diese Deutsch oder Englisch sein. Ist dies nicht der Fall, wird Englisch verwendet.

GTI Tool verwendet standardmäßig „TinyLaF“ als „Look & Feel“, es kann aber auch jedes andere auf dem System zur Verfügung stehende Look & Feel benutzt werden.

Die Option „Wort Modus“ bezieht sich auf die Darstellung des Wortes während der Wort-Navigation. Ist „Linksbündig“ ausgewählt, wird das Wort normal linksbündig dargestellt. Während der Navigation werden schon gelesene Symbole hervorgehoben. Ist dagegen „Rechtsbündig“ ausgewählt, wird das Wort genau wie der Keller rechtsbündig dargestellt. Eine weitere Änderung ist, dass die gelesenen Symbole nicht hervorgehoben, sondern ausgeblendet werden.

Die Einstellungsmöglichkeit „Zoom Faktor“ bezieht sich auf den Graphen des Automaten, es handelt sich dabei um die für jede Datei benutzte Voreinstellung. Der Zoom Faktor kann aber auch für jede Datei einzeln eingestellt werden.

Mit Hilfe der Einstellung „Fehlerzustand in Regulären Ausdruck in DEA-Umwandlung anzeigen“ kann man einstellen, ob der Fehlerzustand im Umwandlungsalgorithmus angezeigt wird, oder nicht. Im erstellten DEA wird der



Zustand unabhängig von der Einstellung angelegt.

Auf dem Tab „Ansicht“ kann die Einstellung „Auto Step“ angepasst werden. Im GTI Tool kann an verschiedenen Stellen, zum Beispiel bei der Wort-Navigation, diese Navigation auch automatisch ablaufen. Der hier eingestellte Wert wird dabei als Zeit zwischen zwei Schritten verwendet.

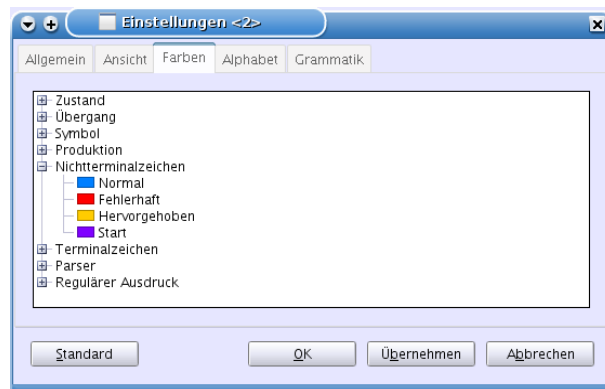


Abbildung 5.1: Einstellungen

Der Tab „Farben“ bietet die Möglichkeit, die verwendeten Farben seinen Bedürfnissen anzupassen. Zur Auswahl stehen sowohl die beim Automaten, wie auch die bei der Grammatik verwendeten Farben, aber auch einige andere. Der Tooltip, der auf jeder Farbe angezeigt wird, gibt genauere Informationen darüber, wo die Farbe verwendet wird.

## 5.2 Automaten Einstellungen anpassen

Für einen Automaten gibt es verschiedene Einstellungsmöglichkeiten, so kann eingestellt werden, wie Übergänge angelegt werden können und wie sich die Auswahl der aktiven Buttons zum Anlegen von Übergängen und Zuständen verhält. Das Anzeigen der Komponenten, die nur bei Kellerautomaten einen richtigen Sinn ergeben, kann angepasst werden und es kann eine Vorauswahl des verwendeten Alphabetes getroffen werden.

Die Einstellung „Übergang“ bietet zwei Möglichkeiten. Wird „Maus-Drag Modus“ ausgewählt, kann ein Übergang zwischen zwei Zuständen per „Drag and Drop“ angelegt werden. Dazu muss der Drag auf dem Anfangszustand beginnen, der Drop auf dem Endzustand oder auf einer freien Stelle, wobei dann ein neuer Zustand angelegt wird, erfolgen. Die zweite Möglichkeit ist die Einstellung „Maus-Klick Modus“. Ist diese ausgewählt, muss auf dem Anfangszustand

ein Mausklick erfolgen, anschließend kann auf dem Endzustand oder auf einer freien Stelle wieder mit der Maus geklickt werden, um den Übergang anzulegen.

Der Punkt „Maus Auswahl“ bietet die Möglichkeit, auszuwählen, welcher Button nach dem Anlegen eines Zustandes oder eines Überganges aktiviert wird. Ist „Ohne Rücksprung zur Maus“ ausgewählt, bleibt der zuletzt aktive Button unverändert. In diesem Modus können also mehrere Übergänge nacheinander angelegt werden, ohne, dass jedesmal wieder der Button „Neuer Übergang“ aktiviert werden muss. Wenn „Mit Rücksprung zur Maus“ ausgewählt ist, wird nach jedem Anlegen wieder der Maus Button aktiviert. Dieser Modus ist besser dazu geeignet, wenn nur einzelne Zustände oder Übergänge angelegt werden müssen, da nach dem Anlegen direkt wieder Komponenten im Graphen ausgewählt werden können.

Mit der Option „Kellerautomat Modus“ kann eingestellt werden, welche Informationen der Benutzer angezeigt bekommt, wenn er einen Automaten ohne Keller bearbeitet. Ist „Alle Komponenten anzeigen“ ausgewählt, werden auch auf einem Automaten ohne Keller die Information zu diesem angezeigt. Dies betrifft die „Keller Operationen Tabelle“, das Eingeben der Wörter, die vom Keller gelesen, bzw. auf ihn geschrieben werden im Übergangs Dialog und noch einige andere Komponenten. Ist „Nur anzeigen, wenn Kellerautomat ausgewählt“ aktiviert, werden diese Komponenten bei Automaten ohne Keller ausgeblendet, bzw. deaktiviert.

Im Tab „Alphabet“ befindet sich die Möglichkeit, das Eingabe-Alphabet, sowie das Keller-Alphabet zu bearbeiten. Bei den hier eingestellten Werten handelt es sich um die Voreinstellung für neue Dateien. Beide Alphabete können auf jeder Datei einzeln angepasst und gespeichert werden. Die Eingabe kann in geschweiften Klammern erfolgen und muss auf jeden Fall mit Komma getrennt werden. Die verwendeten Symbole dürfen nur ein Zeichen enthalten, oder müssen in Anführungszeichen stehen. Somit wäre zum Beispiel {0, "if", "then"} eine gültige Eingabe. Außerdem gibt es die Möglichkeit ganze Klassen von geordneten Zeichen als Alphabet anzugeben. Beispiele hierfür wären [a-z], [0-9], aber auch [a-Z].

Mit dem Haken bei „Keller Alphabet“ kann man angeben, dass man ein eigenes Keller-Alphabet benutzen will. Ist der Haken nicht gesetzt, wird als Keller-Alphabet das Eingabe-Alphabet verwendet. Die Eingabe des Keller-Alphabets erfolgt auf gleiche Weise wie die des Eingabe-Alphabets.

### 5.3 Grammatik Einstellungen anpassen

Im Tab „Grammatik“ können die Voreinstellungen für Nichtterminalzeichen, Terminalzeichen und für das verwendete Startsymbol, das aus der Menge der Nichtterminalzeichen sein muss, eingestellt werden. Die Eingabe muss genauso wie beim Alphabet erfolgen. Zu beachten ist, dass die Menge der Nichtterminal-

zeichen und die Menge der Terminalzeichen disjunkt sein müssen. Auch hierbei handelt es sich nur um die Voreinstellung die im „Neu-Dialog“ verwendet wird.

# Kapitel 6

## Sonstiges

### 6.1 Algorithmen

In allen Algorithmen-Fenstern gibt es einen Button „Algorithmus anzeigen“. Dann erscheint ein Fenster ähnlich dem Folgenden.

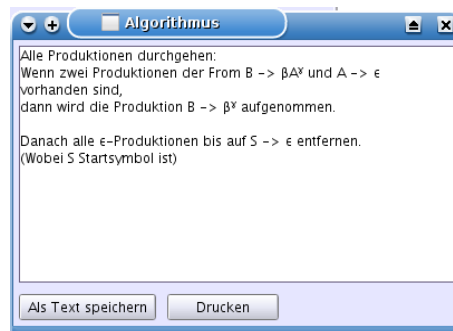


Abbildung 6.1: Algorithmenfenster

In diesem Fenster kann man sich den Algorithmus, welcher gerade ausgeführt wird in Kurzform noch einmal anschauen. Während das Fenster geöffnet ist kann man trotzdem den Algorithmus laufen lassen. Es bleibt dann im Vordergrund.

Diesen Text kann man entweder als Textdatei speichern oder drucken.

### 6.2 Zweite Ansicht

Im Menüpunkt „Ansicht“ ist die Option „Zweite Ansicht“ zu finden. Wird dieser Menüeintrag ausgewählt, wird eine zweite Ansicht eingeblendet, in der Dateien geöffnet werden können. Dabei ist darauf zu achten, dass keine Dateien geöffnet

werden können, die bereits in der anderen Ansicht geöffnet sind. Wird trotzdem eine solche Datei geöffnet, wird die Datei ausgewählt. Da immer nur eine Datei aktiv sein kann, wird die aktive Ansicht durch eine dunklere Umrandung dargestellt, als das bei der nicht aktiven der Fall ist. Die Button Stati beziehen sich dann auf diese aktive Ansicht. Die aktive Ansicht kann durch einfaches Anklicken einer Komponente in der Ansicht geändert werden.

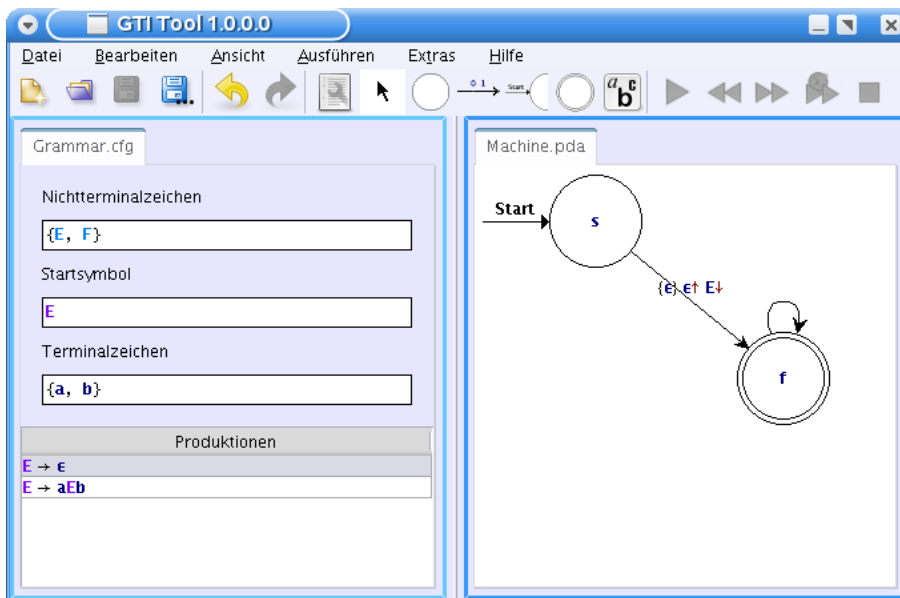


Abbildung 6.2: Zweite Ansicht

Um die Dateien in der anderen Ansicht zu öffnen, gibt es zwei verschiedene Möglichkeiten. Zum einen kann die zweite Ansicht durch Anklicken aktiviert, anschließend eine neue Datei erstellt oder eine bestehende geöffnet werden. Eine andere Möglichkeit ist das Verschieben von geöffneten Dateien. Dies ist möglich, indem auf einem Tab das Popupmenü geöffnet wird. Über „Nach rechts verschieben“ oder „Nach links verschieben“ können die Dateien in die andere Ansicht verschoben werden. Eine zweite Möglichkeit ist, das ebenfalls auch für das normale Umsortieren von Tabs verfügbare, Verschieben per Drag and Drop. Dazu kann ein Tab einfach in die andere Ansicht gezogen und dort an der richtigen Stelle wieder losgelassen werden.

Wird die zweite Ansicht wieder geschlossen, werden alle geöffneten Dateien auf die normale Ansicht verschoben. Beim Verlassen des Programmes wird gespeichert, welche Dateien auf welcher Ansicht geöffnet waren.

### 6.3 Austausch von Dateien

Das GTI Tool bietet die Möglichkeit, geöffnete Dateien zu versenden und Dateien von anderen GTI Tool Benutzern zu empfangen. Zu erreichen ist der Austausch über den Hauptmenüeintrag „Extras“. Die Dateien werden verschlüsselt übertragen.

Wenn eine Datei verschickt werden soll, muss diese als erstes selektiert werden. Anschließend muss der Austausch Dialog geöffnet werden. Dieser bietet die Möglichkeit, zwischen „Senden“ und „Empfangen“ zu wählen. Um die Datei zu verschicken, muss also „Senden“ ausgewählt werden, außerdem muss der „Host“ angegeben werden, auf dem ein GTI Tool empfangsbereit ist. Zusätzlich ist der „Port“ anzugeben, standardmäßig ist Port „64528“ eingestellt. Der Port muss also nur geändert werden, wenn der Empfänger der Datei ihn ebenfalls geändert hat. Beim Senden kann zusätzlich ein Kommentar vergeben werden, den der Empfänger angezeigt bekommt, wenn die Datei empfangen wurde, der aber sonst keinen Einfluss hat. Wenn alle Werte eingestellt sind, kann die Datei mit „Ausführen“ verschickt werden.

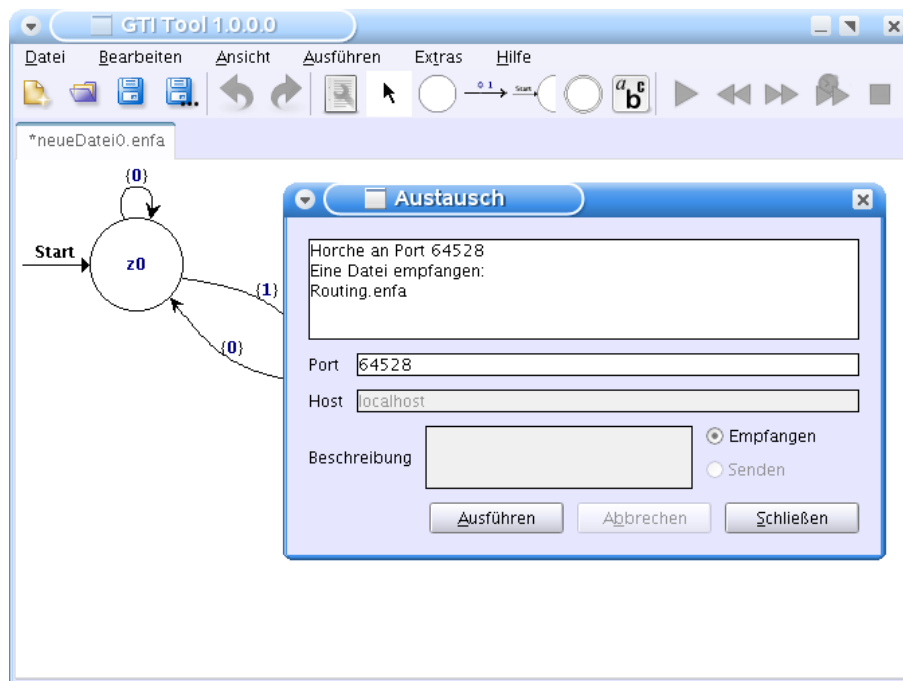


Abbildung 6.3: Austausch von Dateien

Das Empfangen einer Datei funktioniert sehr einfach, es muss nur der Port

eingestellt werden, auf dem der Server auf eingehende Dateien horchen soll. Per „Ausführen“ horcht der Server auf dem eingestellten Port auf eintreffende Dateien. Per „Abbrechen“ kann der Server wieder gestoppt werden, sollte keine Datei empfangen werden. Wird eine Datei empfangen, wird diese automatisch geöffnet und der Server beendet. Der Austausch Dialog wird nicht geschlossen, so dass weitere Dateien empfangen oder verschickt werden können. Wird eine Datei empfangen, kann der vom Sender eingestellte Kommentar im Nachrichtenfenster eingesehen werden.

## 6.4 Bild exportieren

Im Menüpunkt „Datei“ besteht die Möglichkeit, den Automaten Graphen in ein Bild zu exportieren. Der Eintrag ist nur aktiv, wenn eine Automaten Datei selektiert ist. Wird der Eintrag „Bild exportieren“ angeklickt, öffnet sich ein Speichern Dialog, mit dem das Bild gespeichert werden kann. Damit das Ergebnis nicht zu viele freie Bereiche enthält, wird nur der vom Automaten benutzte Bereich exportiert, zusätzlich zu einem 20 Pixel breiten Rand.