
Aufgaben zur Klausur **Compilerbau** im WS 2017/18 (BInf, BMinf, BTimf, BWInf)

Zeit: 75 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Nutzen Sie die Rückseiten der Klausur zur Entwicklung der Lösungen und übertragen die fertigen Lösungen in das Aufgabenblatt.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg!

Diese Klausur besteht einschließlich dieses Deckblattes aus 7 Seiten.

Aufgabe 1:

Definieren Sie die Ableitung Δ einer regulären Menge r nach einem Zeichen a :

.....
.....

Definieren Sie die Ableitung Δ einer regulären Menge r nach einem Wort w :

.....
.....
.....

Berechnen Sie zu dem regulären Ausdruck $r = (y|xz)^*$ über dem Alphabet $\{x, y, z\}$ die Ableitung $\Delta_x(r)$.

Der vereinfachte Ausdruck für die Ableitung:

.....
.....

Berechnen Sie zu dem regulären Ausdruck $r = (y|xz)^*$ über dem Alphabet $\{x, y, z\}$ die Ableitung $\Delta_{xzy}(r)$.

Der vereinfachte Ausdruck für die Ableitung:

.....
.....

Aufgabe 2:

Die Implementierung einer Programmiersprache mittels eines Interpreters für eine virtuelle Maschine gelten allgemein als wenig Laufzeit effizient. Perl, Tcl, Ruby, Javascript und andere Skriptsprachen sind aber gerade mit dieser Technik realisiert und zwar auf äußerst effiziente Art. Warum trifft das Vorurteil, Interpretierer sind ineffizient, bei diesen Sprachen nicht zu.

.....

.....

.....

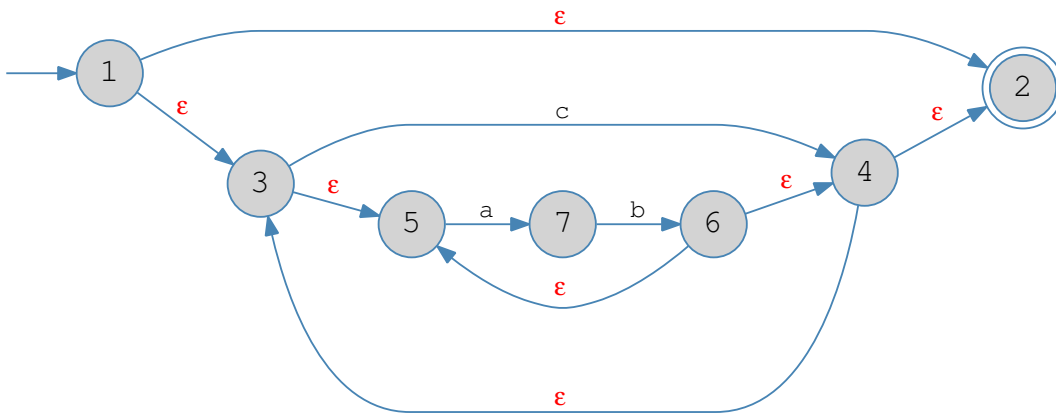
.....

.....



Aufgabe 3:

Gegeben sei der folgende nichtdeterministische endliche Automat mit dem Eingabealphabet $I = \{a, b, c\}$.



Konstruieren Sie hierfür den zugehörigen deterministischen Automaten. Nutzen Sie hierfür den Platz auf der vorigen Seite oder die Rückseiten der Klausur.

Geben Sie die Zustandsmengen für den deterministischen Automaten an:

1)

2)

3)

4)

5)

6)

Welches sind die Endzustände des deterministischen Automaten?

1)

2)

3)

4)

Ist dieses der minimale deterministische Automat?

ja nein

Begründung:

.....

Aufgabe 4:

Skizzieren Sie, welcher Assembler-Code für einen von-Neumann-Rechner für die folgende Funktion sinnvollerweise erzeugt wird. Verwenden Sie für die Befehle die in der Vorlesung vorgestellte virtuelle Maschine und deren Instruktionssatz. Nehmen Sie an, dass es für die arithmetischen Operationen $+$, $-$, $*$, $=$, \neq , **div**, **mod** entsprechende Instruktionen *addi*, *subi*, *muli*, *eqi*, *nei*, *divi*, *modi* gibt. Machen Sie keine Programmtransformationen, um den Code zu optimieren.

```
function g(x : int) : int
  if x mod 2 = 0
  then x div 2
  else 3 * x + 1
;
begin
  var
  x, i : int := 42, 0;
  while x  $\neq$  1 do
    x, i := g(x), i + 1
  endwhile
end
```

Verwenden Sie in den Instruktionen die Namen der Variablen als symbolische Adressen für die zugehörigen Speicheradressen.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....