Klausurkorrektur - Endterm - Anweisungen

Allgemein

Abgaben müssen **handschriftlich** sein. Getippte Lösungen werden mit **0P** bewertet!

Es gibt bei den meisten Aufgaben mehrere Varianten.

Aufgabe 1

Bei den Ja/Nein Fragen muss offensichtlich sein, ob die Antwort Ja oder Nein ist, sonst **0P**. Für falsche Ja/Nein Antworten **0P**.
Generell gilt: vage/unklare Antworten gehen zu Lasten des Antwortenden

a)
0P falls die Sprache nicht die angegebenen Bedingungen erfüllt

Es ist nicht möglich, genau **1P** zu erhalten

b)
0P falls inkorrektes Objekt angegeben
1P für kleinere Mängel

c)
0P falls die vorgegebenen Einschränkungen nicht erfüllt wurden
0P falls die Grammatik nicht terminiert (also keine Wörter erzeugt)
1P für kleinere Mängel

d)
1P für Beschreibung der Konsequenzen der Einschränkung aus der Aufgabenstellung, wenn die Menge der Sprachen nicht angegeben wird (trifft nur auf einige Varianten zu).
2P für kleinere Mängel

e)
0P falls über „Eindeutigkeit" des minimalen DFA argumentiert wird (die gilt nur bis auf Isomorphismen, das passt nicht zur Aufgabe)
2P falls teilweise das richtige Argument genannt wird, aber unklar ist, wie es sich auf die Aufgabe bezieht

Aufgabe 2

a)
-1P pro falscher / fehlender Produktion
Reihenfolge und Anordnung der Produktionen ist egal

b)
+0.5P pro richtige Zelle, abgerundet
keine Folgefehler (weder in Bezug auf a) noch auf tiefere Zellen)

c), d)
Folgefehler wegen b) sind okay
1P richtige Antwort („Ja“/„Nein“) und Begründung unter Zuhilfenahme der Tabelle

Beispiele:

- Falsche Antwort -> **0P**
- „Ja, weil es eine Ableitung gibt.“ -> **0P** (kein Bezug auf Tabelle)

e)
+1P Für die Idee zusätzlich Informationen zu speichern
+1P Modifikation funktioniert und macht Konstruieren der Syntaxbäume leichter

Beispiele:

- „Man speichert zusätzlich die ganze Ableitung“ → **1P,** es ist unklar, wie man kombiniert Ableitungen kombiniert?
- „Man kann den CYK-Algorithmus modifizieren, so dass er alle Syntaxbäume berechnet (siehe Vorlesung).“ → **0P**, die Frage wurde nicht beantwortet

Aufgabe 3

a)
0P wenn kein Ansatz von Produktkonstruktion erkennbar
+1P Startzustand korrekt eingezeichnet
+1P wenn akzeptierende Zustände korrekt sind und es mindestens 4 Zustände gibt
+1P wenn jeder Zustand genau eine Transition jedes Typs besitzt und es mindestens 4 Zustände gibt
+3P - 0.5P pro falscher/fehlender Transition (abgerundet, Minimum: **+0P)**

b)
0P wenn keine Produktkonstruktion verwendet / modifiziert wird
0P wenn sich Zustände ändern
0P wenn nicht klar ist, welche Zustände am Ende final sind
2P für korrekte Lösungen

Es ist nicht möglich, genau **1P** zu erhalten.

Beispiele für sinnvolle Lösungsansätze:

- Produktkonstruktion + passend negieren (vorher und/oder nachher)
- „Wir ändern die akzeptierenden Zustände, sodass ...“

Aufgabe 4

<u>PL:</u>
0P falls das Wort nicht in L ist, oder zu kurz.
+2P Richtiges Wort (also wie in ML)
+2P Es gibt eine Fallunterscheidung und mindestens ein Fall ist sinnvoll argumentiert
+2P Ganze Fallunterscheidung ist korrekt

Wenn man in einem Fall $|vwx| = n$ annimmt, und w auch pumpt, sodass genau n mal der Buchstabe entsteht, ist dieser Fall nicht sinnvoll argumentiert.

<u>CFG:</u>
-1P wenn die Grammatik nicht terminiert, aber insgesamt sinnvoll ist und es offensichtlich ist, wie dieses Problem zu beheben wäre
+1P Disjunktion erkannt und in der Grammatik umgesetzt
+1P Eine unendliche, nichttriviale Teilmenge der Sprache
    +1P auch, wenn die Kleene-Hülle falsch als balancierte Klammern implementiert wurde (trifft nur auf einige Varianten zu)
+2P Korrekte Sprache
    +1P stattdessen, wenn nur kleine Mängel

Aufgabe 5

S implizit als Startsymbol zu verwenden, ist in Ordnung.

a)
0P wenn der PDA ungültig ist (z.B. er liest mehrere Buchstaben in einer Transition) oder großflächig falsch
-1P wenn die Grammatik falsch in Normalform gebracht, aber dann richtig zu einem PDA konvertiert wurde
-1P wenn der Startzustand nicht markiert wurde
-2P für falsche/fehlende Transitionen

b)
0P wenn das Wort falsch ist oder die Begründung fehlt.
1P wenn das Wort korrekt ist und die Begründung angedeutet wird
2P wenn das Wort korrekt ist und die Begründung stimmt.

c)
0P wenn die CFG → PDA Konstruktion angewandt wurde (nicht deterministisch)
+1P wenn *deterministischer* PDA
-1P wenn Startzustand fehlt (aber offensichtlich)
+2P wenn der PDA sich die Anzahl der ‚*a*' merkt
+2P wenn die Anzahl der ‚*ba*' passt (genau 0 bis zu Anzahl der ‚*a*' sind möglich)
  +1P stattdessen , wenn off-by-one
  +**0P** für off-by-half
+1P wenn tatsächlich korrekt

Testwörter:
- *c, aac, aacba, acba*: müssen akzeptiert werden
- *a, acbaba, abac,* (alles mit zwei oder mehr ‚*c*'): müssen abgelehnt werden

Aufgabe 6

0P für falsche Entscheidung (Antwort „Wahr“ wenn falsch oder umgekehrt).
0P wenn die Antwort „falsch“ ist aber kein Gegenbeispiel angegeben wird.
0P für richtige Antwort ohne Begründung oder inhaltslose Begründungen

Beispiele:

- „*L ist rekursiv aufzählbar weil dies äquivalent zu semi-entscheidbar ist, und L ist semi-entscheidbar*“
  0P, hier fehlt eine Begründung, wieso *L* semi-entscheidbar ist.

- „*L ist rekursiv weil L' rekursiv ist*“
  0P, es gibt keine Erklärung, was L mit L' zu tun hat.

Aufgabe 7

a)
0P für einen Ausdruck, der die falsche Sprache darstellt.
1P für eine Lösung mit kleineren Mängeln.
1P für einen Ausdruck, der die korrekte Sprache darstellt, aber zu lang ist.
2P für einen korrekten und kurzen Ausdruck. Keine Angaben über die Länge nötig.

b)
0P für Ausdrücke, die nur die Wörter mit *genau* Länge 8 und mindestens 10 darstellen.
1P für Ausdrücke, die die korrekte Sprache darstellen (inklusive des leeren Wortes), aber zu lang sind. Insbesondere sind alle Lösungen der Form $\Sigma^0 \cup \Sigma^1 \cup ... \cup \Sigma^8 \cup \Sigma^{10}\Sigma^*$ zu lang.
2P für kurze Ausdrücke, die die korrekte Sprache bis auf das leere Wort darstellen (d.h. kein Abzug, wenn man ε vergisst).

c)
Aufteilung: **5P** für die Beschreibung der Reduktion, **3P** für Korrektheit und Laufzeit

<u>Reduktion:</u> (bis zu **5P**)
0P wenn es keinen Bezug auf Teilaufgaben a) und b) gibt, und die Funktion f nicht definiert wird
3P für die Aussage, dass die Formel als $r_1 \mid r_2$ konstruiert wird, wobei r_1 und r_2 aus a) und b) gewonnen werden (ohne nähere Beschreibung)

2P wenn der Bezug zu a) hergestellt wird, aber nicht zu b)

<u>Korrektheit und Laufzeit:</u> (bis zu **3P**)
+1P für ein Argument, dass die Reduktion in polynomieller Zeit läuft
+2P für ein Argument, dass die Eingabeformel genau dann erfüllbar ist, wenn der reguläre Ausdruck nicht universell ist.
    +1P stattdessen, wenn nur eine Richtung betrachtet wird.

Aufgabe 8

a)

0P wenn der reguläre Ausdruck nicht fast azyklisch ist
0P wenn keine verschachtelten Sternchen (etwa *(a*b*)**) im Ausdruck vorkommen.
Sonst:
4P - 1P pro kleinem Fehler, z.B. wenn ε fehlt

b)

0P wenn keine allgemeine Konstruktion (für beliebigen fast azyklischen NFA)
0P wenn ein unendlich langer regulärer Ausdruck konstruiert wird

<u>Lösungsansatz 1:</u>
+2P wenn für jeden akzeptierenden Pfad im NFA (ohne Schleifen) ein regulärer Ausdruck erzeugt wird, und diese dann vereinigt werden,
+2P für Konvertierung von Schleifen mit *a,b,...,z* zu *(a*b*...z*)**
+1P für Begründung / Erklärung von *(a*b*...z*)**
+**1P** für Begründung vom Rest der Konstruktion

- z.B. „es gibt nur *endlich* viele Pfade“
- z.B. wenn die Fast-azyklisch-Eigenschaft des NFA sinnvoll verwendet wird
- z.B. wenn sinnvoll argumentiert wird, wieso L(r) = L(N), also: „Korrekt, weil es für jeden akzeptierenden Pfad einen Teilausdruck gibt, der...*“*

<u>Lösungsansatz 2:</u>
+5P NFA → RE und dann

- Distributivgesetz, um | auf höchstmögliche Ebene zu bringen und dann $(\alpha^*\beta^*)^*$ Trick;
- oder das Distributivgesetz und den $(\alpha^*\beta^*)^*$ Trick wiederholen, bis der Ausdruck fast azyklisch ist.

-1P wenn keine Reihenfolge / keine Wiederholung der Umformungen

+1P für $(\alpha^*\beta^*)^*$ Begründung, nur $(\alpha|\beta)^* = (\alpha^*\beta^*)^*$ reicht nicht
höchstens 3P, wenn Distributivgesetz fehlt

Aufgabe 9

a)
-1P für kleinere Mängel (fehlender Startzustand, Probleme bei ε)

b)
0P für intuitive Begründungen (etwa „Der DFA muss sich merken, welche Zeichen vorgekommen sind“). Gefragt ist ein formaler Beweis.

+1P Zählen von Äquivalenzklassen/Residualsprachen als Beweisstrategie identifiziert.
+2P Die Repräsentanten der Äquivalenzklassen (korrespondierend mit Potenzmengen von Σ_n) werden identifiziert.
+2P Korrekter Beweis, dass die Äquivalenzklassen unterschiedlich sind.

c)
0P für intuitive Begründungen
Falls bei c) ein Beweis steht, der nur für b) funktioniert, wird er für b) gewertet.
Für b) gibt es mindestens so viele Punkte wie für c).