Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2011 Prof. R. Wattenhofer / T. Langner, J. Seidel, J. Smula

# Diskrete Ereignissysteme

## Prüfung

Montag, 6. Februar 2012, 9:00 – 12:00 Uhr

**Nicht öffnen oder umdrehen, bevor die Prüfung beginnt!**

Die Prüfung dauert 180 Minuten und es gibt insgesamt 180 Punkte. Die Anzahl der Punkte pro Teilaufgabe steht jeweils in Klammern bei der Aufgabe. Sie dürfen die Prüfung auf englisch oder deutsch beantworten. Begründen Sie alle Ihre Antworten und beschriften Sie Skizzen und Zeichnungen verständlich. Schreiben Sie zu Beginn Ihren Namen und Ihre Legi-Nummer in das folgende dafür vorgesehene Feld.

| **Name** | **Legi-Nr.** |
|---|---|
| | |

## Punkte

| **Aufgabe** | **Erreichte Punktzahl** | **Maximale Punktzahl** |
|---|---|---|
| 1 | | 15 |
| 2 | | 58 |
| 3 | | 30 |
| 4 | | 47 |
| 5 | | 30 |
| Summe | | 180 |

## 1 Multiple Choice (15 Punkte)

Beurteilen Sie ob die folgenden Aussagen richtig oder falsch sind und kreuzen Sie die entsprechenden Felder an. Eine richtig beurteilte Aussage gibt **1 Punkt**, eine nicht beurteilte Aussage **0 Punkte**, eine nicht richtig beurteilte Aussage **-1 Punkt**. Die gesamte Aufgabe wird mit minimal 0 Punkten bewertet.

| **Aussage** | **Wahr** | **Falsch** |
|---|---|---|
| Die Sprache $\mathcal{L}_1 = \{a^n b^n \mid n \geq 0\}$ ist kontextsensitiv. | ☐ | ☐ |
| Für alle Probleme aus der Klasse $\mathsf{P}$ kann man die Korrektheit einer Lösung in polynomieller Zeit (in der Eingabegrösse) verifizieren. | ☐ | ☐ |
| Alle Sprachen, die von von einem nicht-deterministischen PDA erkannt werden können, können auch von einem deterministischen PDA erkannt werden. | ☐ | ☐ |
| Das Halteproblem ist $\mathsf{NP}$-vollständig ($\mathsf{NP}$-complete). | ☐ | ☐ |
| Das Pumping-Lemma kann verwendet werden, um zu beweisen, dass eine Sprache regulär ist. | ☐ | ☐ |
| Die Linearität des Erwartungswertes gilt *nicht* für eine Summe von *abhängigen* Zufallsvariablen. | ☐ | ☐ |
| Jede Markov-Kette hat eine eindeutige stationäre Verteilung. | ☐ | ☐ |
| Für das Ski Rental-Problem gibt es einen *randomisierten* Online-Algorithmus, dessen (erwartete) Kompetitivität besser ist als die Kompetitivität des besten *deterministischen* Online-Algorithmus. | ☐ | ☐ |
| Eine Transition in einem Petrinetz erhält die Anzahl der Tokens, produziert also gleich viele Tokens wie sie konsumiert. | ☐ | ☐ |
| Wenn in einem Petrinetz mehrere Transitionen gleichzeitig aktiviert sind, ist es unbestimmt, welche als nächstes feuern wird. | ☐ | ☐ |
| Jede irreduzible Markov-Kette ist ergodisch und aperiodisch. | ☐ | ☐ |
| Der Erwartungswert der Anzahl der Jobs im System (Warteschlange + Server) und der Erwartungswert der Jobs in der Warteschlange unterscheiden sich immer um genau 1. | ☐ | ☐ |
| Wenn in einer $M/M/1$-Warteschlange die Verkehrsdichte $\rho = 1$ ist, konvergiert das System. | ☐ | ☐ |
| Ein Petrinetz befindet sich in einem Deadlock genau dann, wenn es keine aktivierte Transition mehr gibt. | ☐ | ☐ |
| Die Routingstrategie FIFO ist in Ringnetzwerken für geeignete Konstanten $(r, b)$ *nicht stabil* (unstable). | ☐ | ☐ |

# 2 Formale Sprachen (58 Punkte)

## 2.1 Automaten

Im Folgenden bezeichnet $\#_x(w)$ die Anzahl der Vorkommen des Zeichens $x$ im Wort $w$.

**a)** [5] Zeichnen Sie einen NFA für den regulären Ausdruck $(ab \cup (b \cup bab)^*)^*$ über dem Alphabet $\Sigma = \{a, b\}$.

**b)** [5] Geben Sie nun einen DFA für den regulären Ausdruck aus **a)** an.

**c)** [5] Ist die Sprache $\mathcal{L}_1 = \{w \in \Sigma^* \mid \#_0(w) \text{ ist gerade}, \#_1(w) \text{ ist ungerade}\}$ über dem Alphabet $\Sigma = \{0, 1\}$ regulär? Begründen Sie Ihre Antwort.

**d)** [8] Beweisen Sie, dass die Sprache $\mathcal{L}_2 = \{\$a\$b \mid a, b \in \mathbb{Z},\ a = 10 \cdot b\}$ über dem Alphabet $\Sigma = \{0, \ldots, 9, \$\}$ nicht regulär ist.

**e)** Gegeben sei die Sprache $\mathcal{L}_3 = \{w \in \Sigma^* \mid \#_a(w) + \#_b(w) = \#_c(w)\}$ über dem Alphabet $\Sigma = \{a, b, c\}$.

(i) [5] Zeichnen Sie einen (möglicherweise nichtdeterministischen) PDA für $\mathcal{L}_3$.

(ii) [5] Geben Sie eine kontextfreie Grammatik zu der Sprache $\mathcal{L}_3$ an.

**f)** [10] Wandeln Sie den in Abbildung 1 gegebenen NFA $A$ in einen äquivalenten DFA um und minimieren Sie ihn soweit wie möglich.

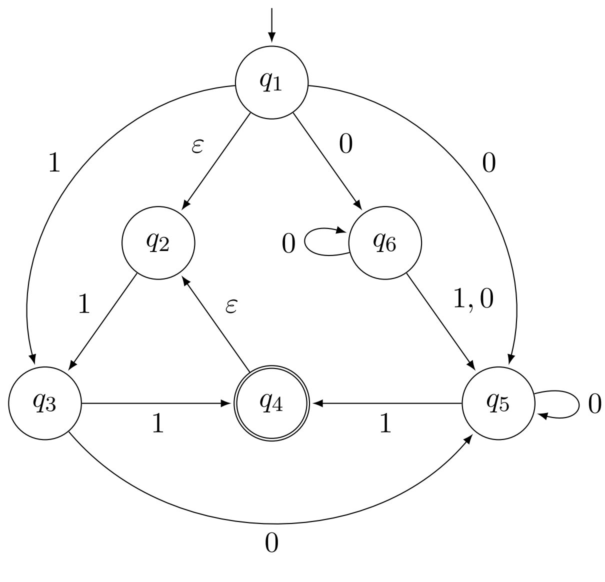



Abbildung 1: NFA $A$.

## 2.2 Abschlusseigenschaften von kontextfreien Sprachen

In den Übungen zur Vorlesung diskutierten wir die Abschlusseigenschaften von regulären Sprachen. Diese sind abgeschlossen unter Durchschnittsbildung, Vereinigung und Konkatenation. In dieser Aufgabe betrachten wir stattdessen kontextfreie Sprachen.

**a)** [5] Beweisen Sie, dass kontextfreie Sprachen unter Vereinigung abgeschlossen sind, d. h. wenn $\mathcal{L}_1$ und $\mathcal{L}_2$ kontextfrei sind, so ist auch $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ kontextfrei, wobei $\mathcal{L}_1 \cup \mathcal{L}_2 = \{w \mid w \in \mathcal{L}_1 \text{ oder } w \in \mathcal{L}_2\}$.

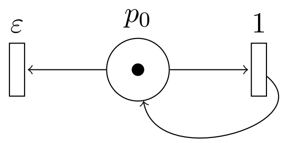
**b)** [5] Zeigen Sie, dass der Schnitt zweier kontextfreien Sprachen nicht kontextfrei sein muss, d. h. es gibt zwei kontextfreie Sprachen $\mathcal{L}_1$, $\mathcal{L}_2$ wobei $\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2$ nicht kontextfrei ist.

**c)** [5] Beweisen oder widerlegen Sie (unter Verwendung von **a)** und **b)**): Kontextfreie Sprachen sind abgeschlossen unter Komplementbildung, d. h. wenn $\mathcal{L}$ kontextfrei ist, so ist auch $\overline{\mathcal{L}}$ kontextfrei.

## 3 Petrinetze & Sprachen (30 Punkte)

### 3.1 Petrinetz-Sprachen

In den folgenden Teilaufgaben soll jeweils ein Petrinetz $P$ inklusive einer passenden Startverteilung von Tokens angegeben werden, das die Wörter der Sprache $\mathcal{L}$ über dem Alphabet $\Sigma = \{0, 1\}$ generiert. Die Transitionen sollen dabei so mit jeweils einem Symbol aus $\Sigma$ oder dem leeren Wort $\varepsilon$ beschriftet werden, dass ein Wort $w$ von $P$ genau dann akzeptiert wird, wenn es einer gültigen Feuersequenz $\sigma_w$ entspricht und $P$ nach Ausführen von $\sigma_w$ tot ist.
*Beispiel:* Das folgende Petrinetz akzeptiert das Wort $v = 11$, weil es eine entsprechende Feuersequenz $\sigma_v = 1, 1, \varepsilon$ gibt, nach deren Ausführung das Netz tot ist.




(Dieses Petrinetz akzeptiert übrigens die Sprache $L = 1^*$.)

Geben Sie jeweils ein entsprechendes Petrinetz $P$ an, das die Wörter der Sprache $\mathcal{L}$ über dem Alphabet $\Sigma = \{0, 1\}$ generiert.

*Hinweis:* Möglicherweise müssen Sie Inhibitorkanten verwenden.

**a)** [7] $\mathcal{L}_1 = \{0^a 1^a \mid a \geq 0\}$

**b)** [7] $\mathcal{L}_2 = \{w \in \{0, 1\}^* \mid w \text{ enthält } a \text{ Nullen und } b \text{ Einsen}, 1 \leq a \leq b\}$

### 3.2 Petrinetze und Chomsky-Hierarchie

Die Abbildung 2 zeigt die hierarchische Struktur der aus der Vorlesung bekannten Sprachklassen. Die gestrichelt umrandete Fläche zeigt, wo sich die Klasse $\mathcal{P}$ der oben eingeführten Petrinetz-Sprachen in dieser Hierarchie befindet.

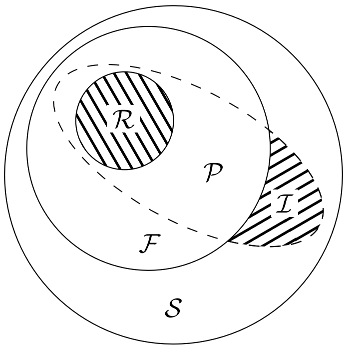



Abbildung 2: Petrinetz-Sprachen in der Chomsky-Hierarchie: Die Klasse der Petrinetz-Sprachen $\mathcal{P}$ wird durch die gestrichelt umrandete Fläche dargestellt. $\mathcal{R}$ = regulär, $\mathcal{F}$ = kontextfrei, $\mathcal{S}$ = kontextsensitiv, $\mathcal{I} = \mathcal{P} \setminus \mathcal{F}$.

**a)** [8] Beweisen Sie $\mathcal{R} \subseteq \mathcal{P}$, also dass sich die Menge $\mathcal{R}$ vollständig in der Menge $\mathcal{P}$ befindet!

**b)** [8] Beweisen Sie $\mathcal{I} \neq \emptyset$, also dass die Menge $\mathcal{I}$ nicht leer ist!

## 4 Online-Spielpartnersuche (47 Punkte)

Sie arbeiten für die Tic-Tac-Toe-Abteilung der Online-Spielbörse *Zynken* und sind damit beauftragt, einen (kompetitiven) Algorithmus ALG für das SPIELERSUCHE-Problem zu entwerfen, der geeignete Spielerpaarungen bestimmt. Dabei melden sich die Spieler über die Zeit verteilt bei der Plattform an, wobei Spieler $i$ zum Zeitpunkt $t_i$ ankommt. Wir bezeichnen mit $S$ die Menge aller Spieler und mit $n = |S|$ die Anzahl der Spieler. Ein Spieler möchte so schnell wie möglich ein Spiel beginnen und muss dafür mit einem anderen Spieler oder einem Computergegner gepaart werden. Jedem Spieler $i$ werden Kosten $c_i$ zugeordnet, wobei $c_i$ der Wartezeit von Spieler $i$ entspricht, bis er ein Spiel beginnen kann. Das Zuordnen eines Computerspielers verursacht zusätzlich „Strafkosten“ in Höhe von 1, da Spieler lieber mit menschlichen Gegnern spielen.

Eine Probleminstanz $I = t_1, \ldots, t_n$ von SPIELERSUCHE ist gegeben durch die Zeiten $t_1$ bis $t_n$, zu welchen neue Spieler erscheinen. Die Zeiten sind paarweise verschieden, $t_1 < t_2 < \ldots < t_n$.

**a)** [2] Erklären Sie (unabhängig von SPIELERSUCHE) mit Worten, was ein $k$-kompetitiver Algorithmus für konstantes $k \geq 1$ erfüllen muss.

**b)** [8] Beweisen Sie, dass es sich für keinen SPIELERSUCHE-Algorithmus lohnt, mehr als einen Spieler gleichzeitig warten zu lassen.

**c)** [10] Zeigen Sie, dass der folgende (vom Ski Rental-Problem inspirierte) SPIELERSUCHE-Algorithmus nicht $k$-kompetitiv ist für alle Konstanten $k \geq 1$:

*Algorithmus:* Ein Spieler $i$ wartet maximal 1 Zeiteinheit auf einen menschlichen Gegner, dann wird ihm ein Computergegner zugeordnet. Erscheint jedoch innerhalb der Wartezeit ein weiterer Spieler $j$, so beginnen $i$ und $j$ sofort ein Spiel.

**d)** [15] Beschreiben Sie einen $k$-kompetitiven deterministischen Algorithmus für möglichst kleines $k$ und argumentieren Sie, weshalb Ihr Algorithmus kompetitiv ist.

Nachdem die SPIELERSUCHE-Instanz in den vorherigen Aufgaben von einem Worst-Case-Gegner erzeugt wurde, wollen wir nun ein alternatives Modell betrachten. Ein vernünftiges Modell für die Ankunft von Spielern bei der Plattform ist ein Poisson-Prozess mit Rate $\lambda > 0$.

**e)** [2] Erklären Sie in zwei Sätzen, was einen Poisson-Prozess charakterisiert.

**f)** [10] Beschreiben Sie (abhängig von $\lambda$) einen möglichst guten Algorithmus und argumentieren Sie, weshalb Ihr Algorithmus gut ist.

# 5 Eine lange DISCO-Nacht (30 Punkte)

## 5.1 Besucherstrom

Sie werden von einem Unterhaltungsunternehmer gebeten, ihn bei der Dimensionierung der Räumlichkeiten seiner DISCO zu unterstützen. Das Etablissement besteht aus einer Tanzfläche, einer Bar und den Toiletten. Die Ankunft der Besucher in der DISCO kann als Poisson-Prozess mit Parameter $\lambda$ betrachtet werden. Nach dem Betreten befinden sich die Besucher zunächst auf der Tanzfläche. Die Aufenthaltsdauer dort ist exponentialverteilt mit Parameter $\mu_d$. Mit Wahrscheinlichkeit $p_v$ gefällt dem Besucher die Musik nicht und er verlässt die DISCO; mit Wahrscheinlichkeit $p_b$ wird er vom Tanzen durstig und geht an die Bar.

Besucher, die an die Bar kommen, bestellen sich ein Getränk. Die Bearbeitungsrate an der Bar (Bestellung beim Bar-Team inklusive zubereiten und trinken) beträgt $\mu_b$ Getränke pro Minute. Danach geht der Besucher mit Wahrscheinlichkeit $p_d$ gleich wieder auf die Tanzfläche. Mit Wahrscheinlichkeit $p_r$ muss der Besucher (bevor er auf die Tanzfläche zurückgeht) jedoch noch die Toiletten für eine mit $\mu_r$ exponentialverteile Dauer aufsuchen.

**a)** `[5]` Modellieren Sie die DISCO als Warteschlangen-Netzwerk.

**b)** `[10]` Geben Sie die Ankunftsrate auf der Tanzfläche in Abhängigkeit von $\lambda$, $p_b$, $p_r$, $p_v$ und $p_d$ an.

**c)** `[5]` Messungen haben ergeben, dass pro Stunde ca. 90 Personen auf die Toilette gehen. Die durchschnittliche Aufenthaltsdauer auf der Toilette beträgt 5 Minuten. Wie viele Toiletten sollten installiert werden, damit die Warteschlange davor nicht ins Unermessliche wächst? (Gehen Sie davon aus, dass eine Toilette jeweils nur von einem Benutzer gleichzeitig verwendet wird.)

**d)** `[10]` Das Ingenieurbüro „1a Toilettenplanung“ behauptet, dass die erwartete Zeitspanne von der Öffnung der DISCO bis der erste Gast auf die Toilette geht einfach berechnet werden kann; sie sei $\lambda + \mu_d + \mu_b$. Diese Aussage ist natürlich falsch. Finden Sie die Fehler!