

Aufgabe 1: Digitale Disruption

Nennen Sie eine IT Technologie, die die klassische Medienlandschaft stark verändert hat und skizzieren Sie diese Veränderung (mit maximal 100 Wörtern):

IT Technologie:

Smartphone

Veränderung des Medienbereichs durch diese Technologie:

Durch (fast) jederzeit und überall mögliche Verbindung zu digitalen Inhaltsseiten entsteht der Anspruch an Inhaltslieferanten, 7/24 aktuelle Inhalte zur Verfügung zu stellen. Dieses verändert klassische Redaktionsabläufe und Prozesse zur Erstellung dieser Inhalte.

Name: _____ Matr.-Nr: _____

Aufgabe 2: Grammatiken

Gegeben sei eine Grammatik wie folgt (ε ist Leersymbol):

G
mit

Startsymbol = *Start*

Terminale = {*a*, *b*, *c*}

Nicht-Terminale = {*X*, *Y*}

Regeln = { *Start* $\rightarrow$ *Xb*,
Start $\rightarrow$ *XbY*,
X $\rightarrow$ *ab*,
X $\rightarrow$ *aXb*,
Y $\rightarrow$ *c*,
Y $\rightarrow$ *cY* }

Entscheiden Sie, ob folgende Wörter in der durch die Grammatik definierten Sprache enthalten sind:

- ε ja / nein
- aabccb ja / nein
- abcd ja / nein
- aabbaacb ja / nein
- acdc ja / nein
- aabbbc ja / ~~nein~~
- abc ja / nein
- abbccc ja / ~~nein~~
- bc ja / nein
- abb ja / ~~nein~~

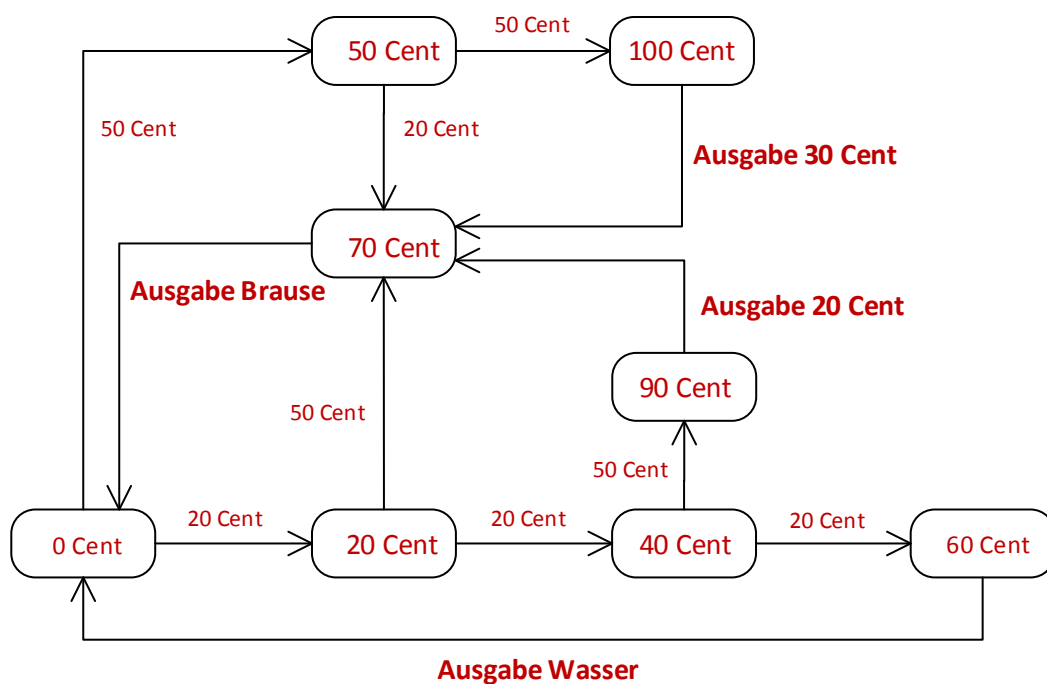
Bestimmen Sie die Wörter, die durch die Grammatik akzeptiert werden in der Art $a^n b^n$:

$a^n b^n bc^*$ mit $n > 0$

Name: _____ Matr.-Nr: _____

Aufgabe 3: Endliche Automaten

Gegeben sei ein Getränkeautomat, der 20 Cent und 50 Cent nimmt. Werden 60 Cent eingeworfen, kommt eine Flasche Wasser, bei 70 Cent eine Brause, bei mehr als 70 Cent ebenfalls eine Brause und der überzahlte Betrag heraus. Skizzieren Sie den entsprechenden endlichen Automaten.



Aufgabe 4: Aussagenlogik

Überprüfen Sie den Wahrheitswert der jeweiligen aussagenlogischen Ausdrücke mit Hilfe einer Wahrheitstabelle:

$$\neg(P \wedge Q) \Leftrightarrow ((\neg P \vee \neg Q) \wedge (P \vee \neg P))$$

P	Q	$P \wedge Q$	$\neg(P \wedge Q)$	$\neg P$	$\neg Q$	$(\neg P \vee \neg Q)$	$(P \vee \neg P)$	$((\neg P \vee \neg Q) \wedge (P \vee \neg P))$	$\dots \Leftrightarrow \dots$
0	0	0	1	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	1	1
1	1	1	0	0	0	0	1	0	1

$$(P \wedge Q) \Leftrightarrow ((P \wedge \neg R) \wedge (P \wedge Q)) \vee (P \wedge Q)$$

P	Q	R	$\neg R$	$P \wedge \neg R$	$P \wedge Q$	$(P \wedge \neg R) \wedge (P \wedge Q)$	$\dots \vee (P \wedge Q)$	$\dots \Leftrightarrow \dots$
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0	1
0	1	0	1	0	0	0	0	1
0	1	1	0	0	0	0	0	1
1	0	0	1	1	0	0	0	1
1	0	1	0	0	0	0	0	1
1	1	0	1	1	1	1	1	1
1	1	1	0	0	1	0	1	1

Name: _____ Matr.-Nr: _____

Aufgabe 5: Prolog Unifikation

Gegeben seien jeweils 2 Prolog-Ausdrücke. Sie sollen entscheiden, ob diese beiden Ausdrücke unifizierbar sind und, falls dieses der Fall ist, den durch Substitution erzeugten Ausdruck angeben. Sollte die Unifikation misslingen, geben Sie den Grund an. Als Beispiele sind gegeben (Konstanten fangen mit kleinem, Variablen mit großem Buchstaben an):

- $\text{test}(X, \text{foo}, Y) = \text{test}(M, N, M)$ ☺ mit $\text{test}(X, \text{foo}, X)$.
- $\text{test}(\text{kai}, X) = \text{test1}(\text{kai}, \text{udo})$ ☹ weil $\text{test} \neq \text{test1}$.

Hier nun die Ausdrücke:

- $\text{coautor}(\text{Koenig}, \text{roland}) = \text{coautor}(\text{esther}, \text{Seifert})$.

☺ mit $\text{Koenig} = \text{esther}$, $\text{Seifert} = \text{roland}$.

- $\text{member}(X, [2, 4, 6]) = \text{member}(4, [2, X, 6])$.

☺ mit $X = 4$.

- $\text{member}(X, [2, 4, 6]) = \text{member}(4, X)$.

☹ $4 \neq [2, 4, 6]$.

- $\text{verwandt}(\text{paul}, \text{paula}) = \text{verwandt}(\text{paula}, \text{paul})$.

☹ $\text{paul} \neq \text{paula}$.

Aufgabe 6: Prolog Programmanalyse

Gegeben ein Prolog-Programm `foo(List1, List2)` wie folgt

```
foo([], []).  
foo([a|Bar],[x|XBar]) :- foo(Bar,XBar).  
foo([_|Bar],XBar) :- foo(Bar,XBar).
```

Beschreiben Sie kurz, was das Programm macht:

Das Programm `foo` evaluiert zu wahr, wenn `List2` aus einer Liste von der Konstanten `x` besteht mit Anzahl der Elemente `List2` identisch mit der Anzahl des Auftretens der Konstanten `a` in `List1`. (Wg. schlampiger Programmierung durch KVL evaluiert ein Aufruf `foo(Irgendwas, [])` ebenfalls zu true.)

Zu was unifiziert `List` in den folgende Beispielen:

```
foo([a, b, b, a], List).
```

`List = [x, x].`

```
foo([a, c, d, c], List).
```

`List = [x].`

```
foo([1,2,3,4], List).
```

`List = [].`

Bewertung der Aufgaben

Aufgabe 1:	irgendwie zutreffende Beschreibung	=	1 Punkt
	gute Beschreibung (maximal 2 Punkte)	=	2 Punkte
Aufgabe 2:	pro richtige Wortentscheidung	=	0.5 Punkte
	richtige regulärer Ausdruck (maximal 6 Punkte)	=	1 Punkt
Aufgabe 3:	richtiger Automat	=	4 Punkte
	fast richtiger Automat	=	3 Punkt
	richtiger Ansatz	=	2 Punkt
	fast richtiger Ansatz (maximal 4 Punkte)	=	1 Punkt
Aufgabe 4:	pro richtiges Ergebnis / Beweis	=	2 Punkte
	pro fast richtiges Ergebnis / Beweis (maximal 4 Punkte)	=	1 Punkt
Aufgabe 5:	pro richtige Unifikation (inkl. Begr.) (maximal 4 Punkte)	=	1 Punkt
Aufgabe 6:	richtige Beschreibung von foo	=	1 Punkt
	fast richtige Beschreibung von foo	=	0.5 Punkte
	pro richtig gelöstes Beispiel	=	0.5 Punkte
	(maximal 2.5 Punkte)		
Maximale Punkte aller Aufgaben:		=	22.5 Punkte

Notengebung: Minimum(Aufrunden(Summe aller Punkte) - 5), 15)

Name: _____ Matr.-Nr: _____