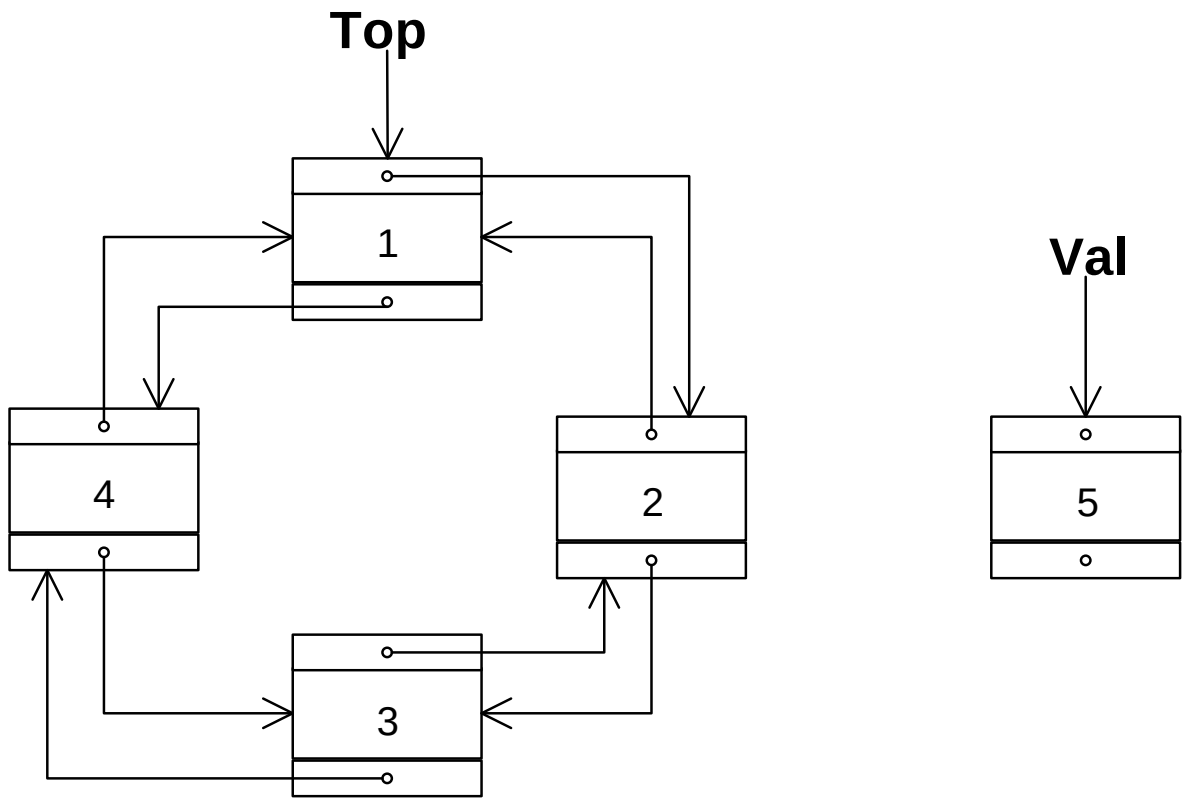
Aufgabe 1: Ring

Gegeben sei ein Ring aus doppelt verketteten Elementen. Jedes Element hat einen Vorgänger (`prev`) und einen Nachfolger (`next`). `Top` zeigt auf das erste Element des Rings, `Val` zeigt auf das einzufügende Element. Skizzieren Sie das Einfügen des Elementes, auf das `Val` zeigt, vor das Top-Element mit Pseudocode, der nur von `Top` und `Val` abhängt:



.

```
Val.next = Top;
Val.prev = Top.prev;
Top.prev.next = Val;
Top.prev = Val;
```

Name:_____ Matr.-Nr: _____

Aufgabe 2: Grammatiken

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

G = <Startsymbol, Terminale, Nicht-Terminale, Regeln>
mit

Startsymbol = *Start*

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

Nicht-Terminale = {*X, Y* }

Regeln = { $Start \rightarrow X$,
$Start \rightarrow XY$,
$X \rightarrow ac$,
$X \rightarrow aX$,
$Y \rightarrow dc$,
$Y \rightarrow dY$}

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

- ε ~~ja~~ / nein
- aacddddddddc ja / ~~nein~~
- abcd ~~ja~~ / nein
- acd ~~ja~~ / nein
- acdc ja / ~~nein~~
- aaaaacdc ja / ~~nein~~
- acccc ~~ja~~ / nein
- aaaac ja / ~~nein~~
- kai ~~ja~~ / nein
- abba ~~ja~~ / nein

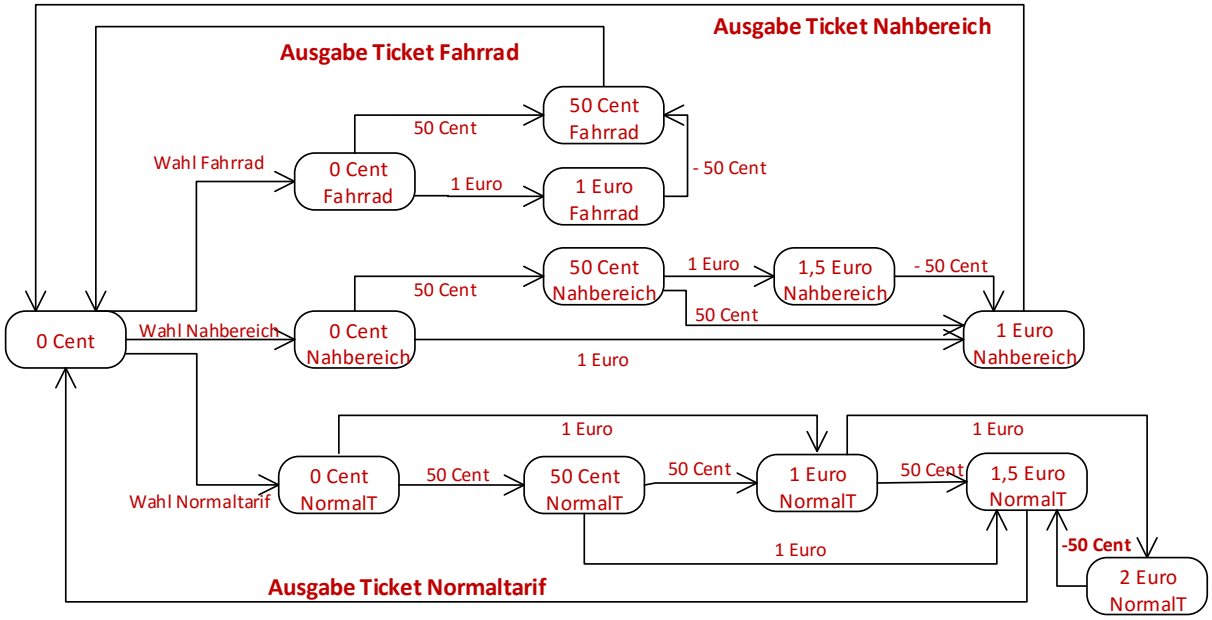
Bestimmen Sie die Wörter, die durch die Grammatik akzeptiert werden in der Art $a^n b^n$:

$a^* c (d^* c)^?$

Name:_____ Matr.-Nr: _____

Aufgabe 3: Endliche Automaten

Gegeben sei ein Fahrkartenautomat, der 50 Cent und 1 Euro Münzen nimmt. Man kann auswählen zwischen einer Karte für das Fahrrad (kostet 50 Cent), für einen Nahbereich (kostet 1 Euro) und für einen Normaltarif (kostet 1,5 Euro). Der Automat gibt überzahltes Geld zurück. Skizzieren Sie den entsprechenden endlichen Automaten.



Name:_____ Matr.-Nr: _____

Aufgabe 4: Aussagenlogik

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

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

P	Q	¬P	(¬P ∨ Q)	¬Q	(P ∧ ¬Q)	¬(P ∧ ¬Q)	(¬(P ∧ ¬Q) ∨ P)	((¬(P ∧ ¬Q) ∨ P) ∧ (¬P ∨ Q))
0	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1	1
1	0	0	0	1	1	0	1	0
1	1	0	1	0	0	1	1	1

P	Q	(¬P ∨ Q) ⇔ ((¬(P ∧ ¬Q) ∨ P) ∧ (¬P ∨ Q))
0	0	1
0	1	1
1	0	1
1	1	1

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

P	Q	R	¬R	(¬R ∨ Q)	((¬R ∨ Q) ∨ R)	(Q ∨ R)	((Q ∨ R) ∨ ¬R)	((¬R ∨ Q) ∨ R) ∧ ((Q ∨ R) ∨ ¬R)	(((¬R ∨ Q) ∨ R) ∧ ((Q ∨ R) ∨ ¬R) ∧ P)	P ⇔ …
0	0	0	1	1	1	0	1	1	0	1
0	0	1	0	0	1	1	1	1	0	1
0	1	0	1	1	1	1	1	1	0	1
0	1	1	0	1	1	1	1	1	0	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	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):

- test (X, foo, Y) = test (M, N, M) ☺ mit test (X, foo, X).
- test (kai, X) = test1(kai, udo) ☹ weil test ≠ test1.

Hier nun die Ausdrücke:

- append([1,2], [a,b], L) = append(X, Y, [1,a,2,b]).

☺ mit L = [1, a, 2, b], X = [1, 2], Y = [a, b].

- reverse([a, b, c, d], L) = append([d, c], [b, a], L).

☹ reverse ≠ append.

- eltern(Udo, Eva, Kinder) = eltern(udo, eva, hans, maria).

☹ eltern/3 ≠ eltern/4.

- eltern(Udo, Eva, Kinder) = eltern(udo, eva, [hans, maria]).

☺ Udo = udo, Eva = eva, Kinder = [hans, maria].

Name:_____ Matr.-Nr: _____

Aufgabe 6: Prolog Programmanalyse

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

```
foo(_,[],[]).
foo(X,[X|Rest], Result) :- foo(X, Rest, Result).
foo(X,[Y|Rest], [Y|Result]) :- X\=Y, foo(X, Rest, Result).
```

Beschreiben Sie kurz, was das Programm macht unter der Bedingung, dass Elem und List1 keine Variablen sind:

Das Programm `foo` evaluiert zu wahr, wenn `List2` aus einer Liste besteht, in der alle Vorkommen von `Elem` in `List1` fehlen, die restlichen Elemente in `List2` mit dem Vorkommen und der Reihenfolge der Elemente in `List1` identisch sind.

Zu was unifiziert List in den folgende Beispielen:

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

foo(b,[a, c, d, c], List).
List = [a, c, d, c].

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

Name:_____ Matr.-Nr: _____

Bewertung der Aufgaben

Aufgabe 1:	irgendwie brauchbare Idee	=	1 Punkt
	halb richtige Idee	=	2 Punkte
	fast richtiger Code	=	3 Punkte
	richtiger Code	=	4 Punkte
	(maximal 4 Punkte)		
Aufgabe 2:	pro richtige Wortentscheidung	=	0.5 Punkte
	richtige regulärer Ausdruck	=	1 Punkt
	(maximal 6 Punkte)		
Aufgabe 3:	richtiger minimaler Automat	=	4 Punkte
	fast richtiger Automat	=	3 Punkt
	richtiger Ansatz	=	2 Punkt
	fast richtiger Ansatz	=	1 Punkt
	(maximal 4 Punkte)		
Aufgabe 4:	richtiges Ergebnis	=	2 Punkte
	fast richtiges Ergebnis	=	1 Punkt
	(maximal 4 Punkte)		
Aufgabe 5:	richtiges Ergebnis	=	1 Punkt
	(maximal 4 Punkte)		
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:		=	24,5 Punkte

Notengebung: Min(Max(Aufrunden(Summe aller Punkte) - 8),0),15)

Name:_____ Matr.-Nr: _____