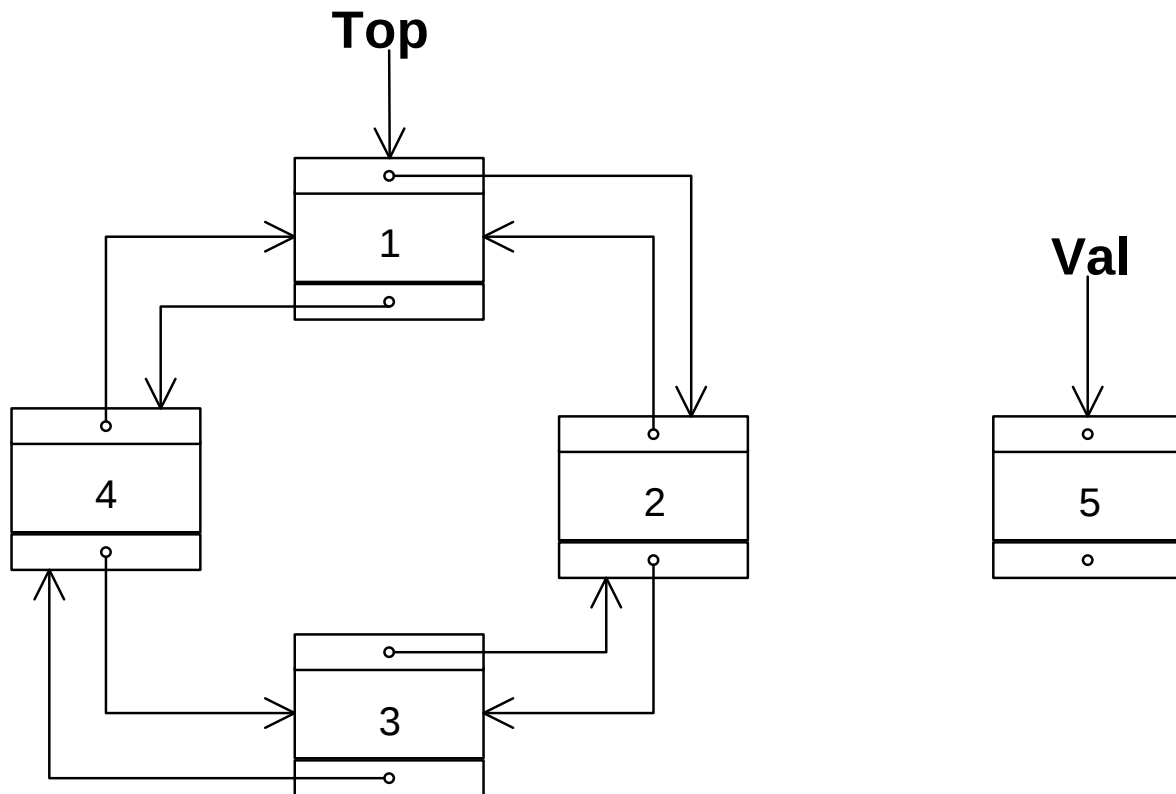


Aufgabe 1: Ring

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:



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 |
| • aacdddddddc | 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 $a^n b^n$:

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.

Aufgabe 4: Aussagenlogik

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

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

P	Q	
0	0	
0	1	
1	0	
1	1	

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

P	Q	R	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
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` $\neq$ `test1`.

Hier nun die Ausdrücke:

- `append([1,2], [a,b], L)` = `append(X, Y, [1,a,2,b])`.
- `reverse([a, b, c, d], L)` = `append([d, c], [b, a], L)`.
- `eltern(Udo, Eva, Kinder)` = `eltern(udo, eva, hans, maria)`.
- `eltern(Udo, Eva, Kinder)` = `eltern(udo, eva, [hans, maria])`.

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:

Zu was unifiziert `List` in den folgende Beispielen:

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

`List =`

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

`List =`

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

`List =`