

Aufgabe 2)

a)

i)

Syntax:

" $p.q : -p$ " ist kein syntaktisch korrektes Wort, da es nicht mit einem "." endet.

ii)

Syntax:

$$\begin{aligned} S_2 &\rightarrow A.S_2 \rightarrow B.S_2 \rightarrow p.S_2 \\ &\rightarrow p.A.S_2 \rightarrow p.B : -B.S_2 \rightarrow p.q : -p.S_2 \\ &\rightarrow p.q : -p.A. \rightarrow p.q : -p.B : -B. \rightarrow p.q : -p.s : -r. \end{aligned}$$

Hiermit ist gezeigt das " $p.q : -p.s : -r.$ " ein syntaktisch korrektes Wort ist.

Semantik:

$$W(p.q : -p.s : -r.) = W(p.q : -p.) \cup \emptyset = W(p.) \cup \emptyset \cup \emptyset = \{p\}$$

iii)

Syntax:

$$S_2 \rightarrow A. \rightarrow B : -B.$$

Da es kein Terminal " a " gibt, kann man ab hier nicht weiter ableiten. Deswegen ist das Wort " $a : -r.$ " syntaktisch falsch.

b)

Zwei Wörter mit gleicher Semantik müssen nicht die gleiche Syntax haben. Zum Beispiel die Wörter "Fehrbetätigung" und "Drücker" in der deutschen Sprache meinen im richtigen Kontext beide das Gleiche, aber ihre Syntax ist unterschiedlich.

c)

Damit ein Wort semantisch Korrekt ist, muss es syntaktisch Korrekt sein, weil es sonst nicht zur Sprache gehört.

Aufgabe 4)

a)

$$G(L_2) = (\{S, A, B, X\}, \{a, b\}, P, S)$$

$$P = \{ \begin{array}{l} S \rightarrow B \mid X \mid \epsilon \\ B \rightarrow bB \mid b \\ X \rightarrow aXb \mid Xab \mid abX \mid A \\ A \rightarrow aA \mid a \end{array} \}$$

b)

$$\begin{array}{l} S_1 = \{ a, [S_2] \} \\ S_2 = [(b \mid a, S_2)] \end{array}$$

c)

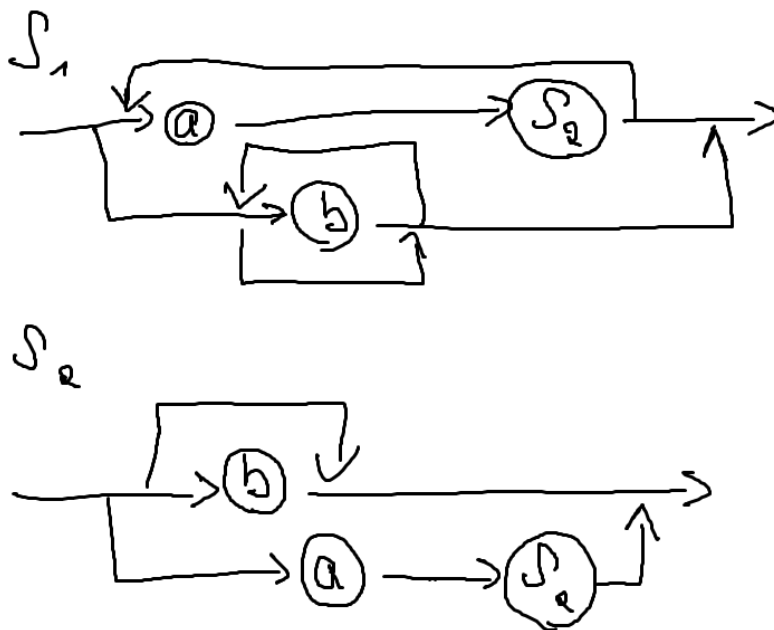


Abbildung 1

Aufgabe 6)

a)

i)

$$(0101010101)_2 = (2^8 + 2^6 + 2^4 + 2^2 + 2^0)_{10} = (341)_{10}$$

ii)

$$(1000000000)_2 = K_2(1000000000)_2 = K_1(1000000000)_2 + 1 = -(1000000000)_2 = -(1024)_{10}$$

iii)

$$(0100101110)_2 = (2^8 + 2^5 + 2^3 + 2^2 + 2^1)_{10} = (302)_{10}$$

iv)

$$(1100011001)_2 = K_2(1100011001)_2 = -(0011100111)_2 = -(231)_{10}$$

v)

$$(1000101110)_2 = K_2(1000101110)_2 = -(0111010010)_2 = -(466)_{10}$$

b)

i)

Da beim "-" in Java standardmäßig die beiden Ganzzahlen als int interpretiert werden und die kleinstmögliche, mit einem int darstellbare, Zahl $-2^{31} = -2147483648$ ist und $-1000000000 - 1100000000 - 2000000000$ kleiner als -2^{31} ist, entsteht hier ein Unterlauf, welcher zur Folge hat das aus der Subtraktion, welcher ja eigentlich kleiner 0 sein sollte, das Ergebnis 1994967296 hervorgeht. Das ist größer 0 und deshalb ergibt der Ausdruck true.

ii)

$$\begin{aligned} & -(0 - 2000000000 - 1470000000 - 483000 - 648)_{10} = -(-(2147483648)_{10}) \\ & = -(10...0)_2 = -(K_1(10...0)_2 + 1) = -(-(01...1)_2 + 1) = -(-(10...0)_2) = (10...0)_2 \\ & = K_1(10...0)_2 + 1 = -((01...1)_2 + 1) = -(10...0)_2 = -(2147483648)_{10} < 0 \end{aligned}$$

Mathematisch macht die Rechnung natürlich keinen direkten Sinn, jedoch aufgrund des beschränkten Speicherplatzes auf einem Rechner kann man die kleinste Zahl im K_2 immer negieren und man erhält wieder denselben Wert, weil es beim int, mit seinen 32 Bit, kein positives Äquivalent zur kleinsten Zahl (-2^{31}) gibt. Versucht man die kleinste darstellbare Zahl für 32 Bit zu negieren, bekommt man als Ergebnis die umgekehrte Zahl, also 2^{31} . Diese würde dann jedoch 33 Bit belegen, weshalb es zu einem Überlauf kommt, wobei das erste Bit der 33 Bit Zahl abgeschnitten wird und man wieder bei der kleinsten 32 Bit Zahl landet. Da diese ja kleiner 0 ist, ergibt der Ausdruck true.