Zusatz 12. Übung

Dominic Deckert

17. Juli 2017

1 a)

h	LOAD 1; LOAD 2; LEQ; JMC **h.1**;
	LOAD 1; STORE 1; WRITE 1; JMP **0**;
h.1	LOAD 2; LOAD 1; STORE 2; STORE 1;
	JMP **h**;

1 b)

```
int main(){
    int x1, x2, x3;
    scanf("%d", &x1);   scanf("%d", &x2);
    while(x2>0){
        if(x1>=x2)x1=x1-1;
        else{
            x3=x1;
            x1=x2;
            x2=x3;}
    printf("%d", x1);
    return 0;
}
```

2

$$
\begin{aligned}
&\left\{\begin{pmatrix} \delta(\gamma(x_3), \gamma(\gamma(\alpha)), \sigma(\gamma(x_2), x_1)) \\ \delta(\gamma(x_3), \gamma(x_3), \sigma(\gamma(\alpha), \gamma(\gamma(x_2)))) \end{pmatrix}\right\} \\
\overset{Dek.}{\Longrightarrow} &\left\{\begin{pmatrix} \gamma(x_3) \\ \gamma(x_3) \end{pmatrix}, \begin{pmatrix} \gamma(\gamma(\alpha)) \\ \gamma(x_3) \end{pmatrix}, \begin{pmatrix} \sigma(\gamma(x_2), x_1) \\ \sigma(\gamma(\alpha), \gamma(\gamma(x_2))) \end{pmatrix}\right\} \\
\overset{Dek.}{\Longrightarrow} &\left\{\begin{pmatrix} x_3 \\ x_3 \end{pmatrix}, \begin{pmatrix} \gamma(\alpha) \\ x_3 \end{pmatrix}, \begin{pmatrix} \gamma(x_2) \\ \gamma(\alpha) \end{pmatrix}, \begin{pmatrix} x_1 \\ \gamma(\gamma(x_2)) \end{pmatrix}\right\} \\
\overset{Dek.}{\Longrightarrow} &\left\{\begin{pmatrix} x_3 \\ x_3 \end{pmatrix}, \begin{pmatrix} \gamma(\alpha) \\ x_3 \end{pmatrix}, \begin{pmatrix} x_2 \\ \alpha \end{pmatrix}, \begin{pmatrix} x_1 \\ \gamma(\gamma(x_2)) \end{pmatrix}\right\}
\end{aligned}
$$

2

$$\overset{Elim.}{\Longrightarrow} \left\{ \begin{pmatrix} \gamma(\alpha) \\ x_3 \end{pmatrix}, \begin{pmatrix} x_2 \\ \alpha \end{pmatrix}, \begin{pmatrix} x_1 \\ \gamma(\gamma(x_2)) \end{pmatrix} \right\}$$

$$\overset{Vert.}{\Longrightarrow} \left\{ \begin{pmatrix} x_3 \\ \gamma(\alpha) \end{pmatrix}, \begin{pmatrix} x_2 \\ \alpha \end{pmatrix}, \begin{pmatrix} x_1 \\ \gamma(\gamma(x_2)) \end{pmatrix} \right\}$$

$$\overset{Sub.}{\Longrightarrow} \left\{ \begin{pmatrix} x_3 \\ \gamma(\alpha) \end{pmatrix}, \begin{pmatrix} x_2 \\ \alpha \end{pmatrix}, \begin{pmatrix} x_1 \\ \gamma(\gamma(\alpha)) \end{pmatrix} \right\}$$

3 a)

Sei a ein Datentyp beliebig aber fest.
Induktionsanfang:
Sei xs :: [a] mit xs = []. Dann gilt

$$
\begin{aligned}
breadth(toTree xs) &= breadth(toTree\, [\,])\\
&\overset{Z12}{=} breadth(Leaf)\\
&\overset{Z4}{=} 1\\
&\overset{Z16}{=} pow(0)\\
&\overset{Z8}{=} pow(length\, [\,]) = pow(length\, xs)
\end{aligned}
$$

3 b)

Induktionsvoraussetzung:
Sei xs :: [a] so, dass gilt:
IV: $breadth(toTree\, xs) = pow(length\, xs)$

Es sei x :: a beliebig aber fest. Behauptung: Dann gilt
$breadth(toTree(x : xs)) = pow(length(x : xs))$

3 b)

$$
\begin{aligned}
breadth(toTree(x : xs)) &\overset{Z13}{=} breadth(Branch\, x\, (toTree xs)(toTree xs)) \\
&\overset{Z5}{=} breadth(toTree\, xs) + breadth(toTree\, xs) \\
&\overset{IV}{=} 2 * pow(length\, xs) \\
&\overset{Z17}{=} pow(1 + length\, xs) \\
&\overset{Z9}{=} pow(length(x : xs))
\end{aligned}
$$