KI
Musterlösung 3. Übung

1. Aufgabe

a.

p	q	r	$\overbrace{p \mid q \mid r}^{a}$	$\overbrace{r => (p \mid q)}^{b}$	$\overbrace{(q \,\&\, r) => p}^{c}$	$\overbrace{\sim p \mid q \mid r}^{d}$	a & b & c & d	(q => p) | ~(q => (p | r))
0	0	0	0	1	1	1	0	1
0	0	1	1	0	1	1	0	1
0	1	0	1	1	1	1	1	1
0	1	1	1	1	0	1	0	0
1	0	0	1	1	1	0	0	1
1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

b.

p	q	r	$\overbrace{q \mid r}^{a}$	$\overbrace{q => \sim p}^{b}$	$\overbrace{\sim(r \,\&\, p)}^{c}$	a & b & c	~p
0	0	0	0	1	1	0	1
0	0	1	1	1	1	1	1
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0
1	0	1	1	1	0	0	0
1	1	0	1	0	1	0	0
1	1	1	1	0	0	0	0

c.

p	q	$\overbrace{p => q}^{a}$	a & q	p & q
0	0	1	0	0
0	1	1	1	0
1	0	0	0	0
1	1	1	1	1

2. Aufgabe

a.

p	q	r	c	(p & q & r) => c
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

b.

p	q	r	c	c | ~p | ~q | ~r
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

3. Aufgabe

a.

Wenn der Händler ehrlich ist, ist das Saatgut, das Horst bei ihm kauft, gut. $\rightarrow \{\sim H, S\}$
Wenn das Saatgut und das Wetter im Sommer gut sind, verdient Horst ausreichend Geld. $\rightarrow \{\sim S, \sim W_1, G\}$
Wenn er ausreichend Geld hat, füttert er den Wolf. $\rightarrow \{\sim G, W_2\}$
Wenn er den Wolf gefüttert hat und in der Stadt Kirmes ist, geht er in die Stadt und lässt Wolf und Borsti allein. $\rightarrow \{\sim W_2, \sim K, A\}$
Wenn der Wolf gefüttert worden ist, ist er satt. $\rightarrow \{\sim W_2, W_3\}$
Wenn der Wolf satt ist oder der Bauer anwesend ist, wird Borsti nicht gefressen. $\rightarrow \{\{\sim W_3, \sim B\}, \{A, \sim B\}\}$

$H \rightarrow$ Händler ist ehrlich
$S \rightarrow$ Saatgut ist gut
$W_1 \rightarrow$ Wetter ist gut
$G \rightarrow$ Horst verdient Geld
$W_2 \rightarrow$ Wolf ist gefüttert
$K \rightarrow$ In der Stadt ist Kirmes
$A \rightarrow$ Wolf und Borsti sind allein
$W_3 \rightarrow$ Wolf ist satt
$B \rightarrow$ Borsti wird gefressen

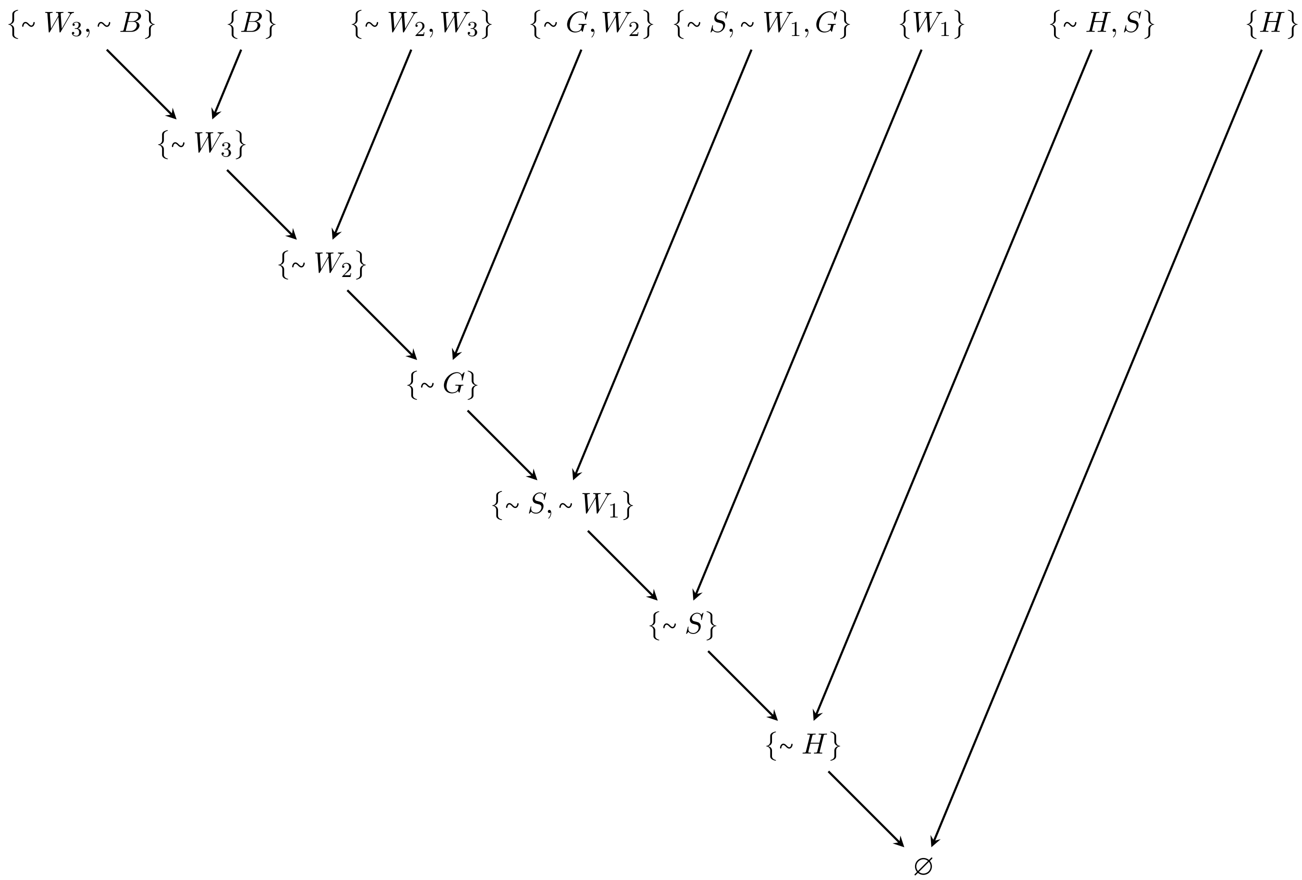
b.

$\{K\}$ muss zu der Hornklauselmenge hinzugefügt werden (in der Stadt ist Kirmes).

Außerdem fügen wir die negierte Anfrage $(\{B\}, \{W_1\}, \{H\})$ hinzu:

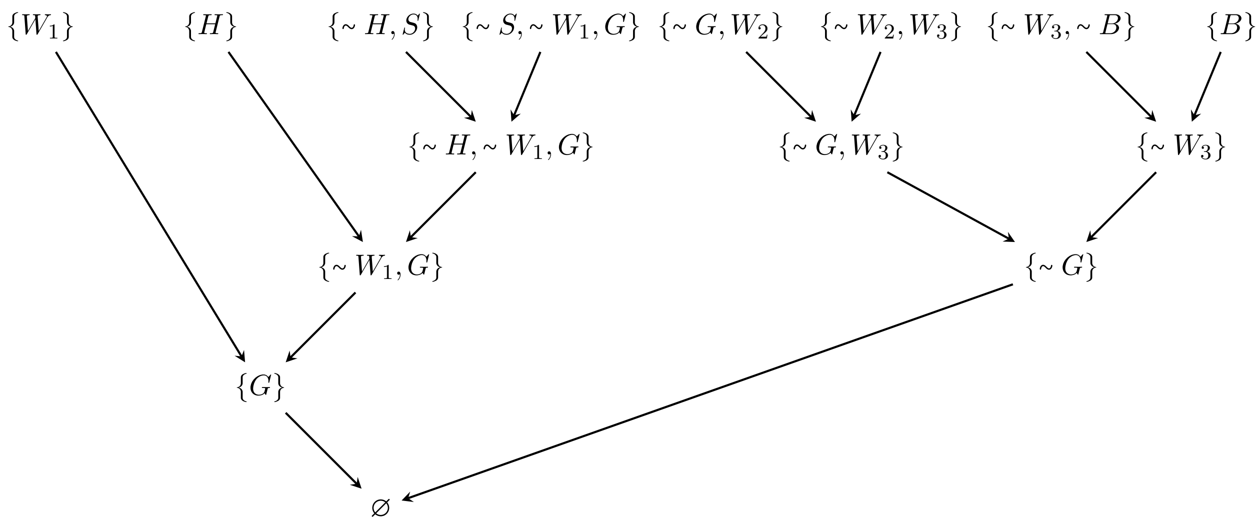
$\sim ((W_1 \ \& \ H) \ \Rightarrow \ \sim B)$
$\equiv \ \sim (\sim (W_1 \ \& \ H) \mid \ \sim B)$
$\equiv \ \sim (\sim B \mid \ \sim W_1 \mid \ \sim H)$
$\equiv \ B \ \& \ W_1 \ \& \ H$

Gezeigt wird, dass die leere Menge aus den Klauseln und der Negation der Anfrage resolviert wird. Dabei wird mit einer Zielklausel begonnen. (Zielklauseln sind Klauseln mit nur negierten Literalen. In diesem Fall kommt nur $\{\sim W_3, \sim B\}$ in Frage). In jeden Resolutionsschritt entsteht dabei eine neue Zielklausel:



Da die gegebenen Klauseln vereinigt mit der negierten Anfrage nicht erfüllbar sind, wird Borsti nicht gefressen.

Alternativ hätte man auch andere Klauseln mit der jeweiligen Zielklausel resolvieren können. Es ist auch möglich, nicht nur mit Zielklauseln zu resolvieren. Das ist dann aber nicht SLD-Resolution:



c.

```
fof(haendler_honest, axiom, (
    honest(haendler)
    => good(saat)
)).

fof(horst_earns_money, axiom, (
    (good(saat)
    & good(weather))
    => earns_money(horst)
)).

fof(feed_wolf, axiom, (
    earns_money(horst)
    => feed(wolf)
)).

fof(wolf_borsti_alone, axiom, (
    (feed(wolf)
    & kirmes(stadt))
    => alone(wolf, borsti)
)).

fof(wolf_full, axiom, (
    feed(wolf)
    => full(wolf)
)).

fof(borsti_is_eaten, axiom, (
    (full(wolf)
    | ~alone(wolf, borsti)
    | (full(wolf) & ~alone(wolf, borsti)))
    => alive(borsti)
)).

fof(is_kirmes, axiom, (
    kirmes(stadt)
)).

fof(borsti_alive, conjecture, (
    (good(weather)
    & honest(haendler))
    => alive(borsti)
)).
```