Exercise 1 *(8.4.2)*

1.

$$
\begin{aligned}
\{\neg(\exists x)^{-}(\forall y)^{-}P(x,y) \vee (\forall x)Q(x)\} \wedge \{(\forall x)^{-}A(x) \supset (\forall x)^{+}B(x)\} &\equiv \\
\{\neg(\exists x)(\forall y)P(x,y) \vee (\forall x')Q(x')\} \wedge \{(\forall x'')A(x'') \supset (\forall x''')B(x''')\} &\equiv \\
(\forall x)(\exists y)(\forall x')\{\neg P(x,y) \vee Q(x')\} \wedge (\exists x'')(\forall x''')\{A(x'') \supset B(x''')\} &\equiv \\
(\forall x)(\exists y)(\forall x')(\exists x'')(\forall x''')[\{\neg P(x,y) \vee Q(x')\} \wedge \{A(x'') \supset B(x''')\}]
\end{aligned}
$$

where I've marked the initial formula with the polarities of the quantifiers. This is just one possible order in which the quantifiers can be 'merged'. Some other results obtained by merging are $(\exists x'')(\forall x''')(\forall x)(\exists y)(\forall x')$ *and* $(\forall x')(\exists x'')(\forall x)(\forall x''')(\exists y)$*. Irrespective of the order, there are* 3 $\forall$*s and* 2 $\exists$*s.*

2.

$$
\begin{aligned}
\neg(\exists x)^{-}(\exists y)^{-}[P(x,y) \supset (\forall x)^{-}(\forall y)^{-}P(x,y)] &\equiv \\
\neg(\exists x)(\exists y)[P(x,y) \supset (\forall x')(\forall y')P(x',y')] &\equiv \\
(\forall x)(\forall y)\neg[P(x,y) \supset (\forall x')(\forall y')P(x',y')] &\equiv \\
(\forall x)(\forall y)(\exists x')(\exists y')\neg[P(x,y) \supset P(x',y')]
\end{aligned}
$$

3.

$$
\begin{aligned}
(\forall x)^{+}\{(\forall y)^{-}[((\forall z)^{-}P(x,y,z) \supset (\exists w)^{+}Q(x,y,w)) \supset R(x)] \supset S(x)\} &\equiv \\
(\forall x)(\exists y)(\exists z)(\exists w)\{[(P(x,y,z) \supset Q(x,y,w)) \supset R(x)] \supset S(x)\}
\end{aligned}
$$

Remark 1 *About half did these basic transformations completely correctly.*

As in previous exercises, it is convenient to first draw the abstract syntax tree to determine whether subformulas occur negatively or positively, to determine whether quantifiers are essentially universal or existential, giving rise to a $\forall$ *resp.* $\exists$ *in the prefix of the prenex normal form.*

Prefixes are obtained by merging the linear orders of the quantifiers along the branches in the abstract syntax tree, from the root to the leaves.

Exercise 2 *(8.9.1) The exercise was to apply the* procedure given in this section. *In particular the goal was to eliminate the primary-connective-cut as described on p. 237 (that we can reason semantically to close faster using double negation is ok, but was not the question, as that is not part of the procedure in this section). The answer should be something like the following, provided by one of you:*

~~8.9.1~~

8.9.1

$$\theta \left\{ \begin{array}{l} \neg((P \supset \neg\neg Q) \vee (Q \supset R)) \\ \neg(P \supset \neg\neg Q) \\ \neg(Q \supset R) \end{array} \right.$$

$P \supset Q$ | $\neg(P \supset Q)$

$\alpha_1 \left[\begin{array}{l} P \\ \neg\neg\neg Q \end{array} \right]$ | $\left[\begin{array}{l} Q \\ \neg R \end{array} \right] \alpha_2$

T_1 | T_2

Both α_1, α_2 were obtained by α rule application to θ therefore we "push" them up above the cut.

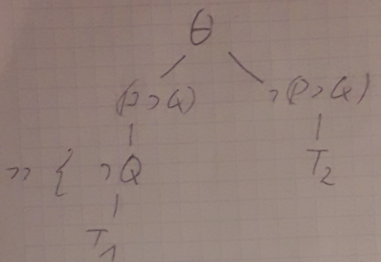
$$\theta$$

$$\left[\begin{array}{c} P \\ \neg\neg\neg Q \\ Q \\ \neg R \end{array} \right] \alpha_1 \text{ and } \alpha_2$$

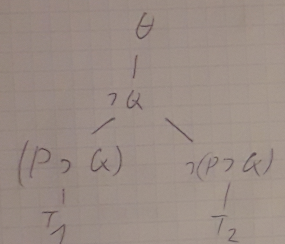
$P \supset Q$ | $\neg(P \supset Q)$

T_1 | T_2

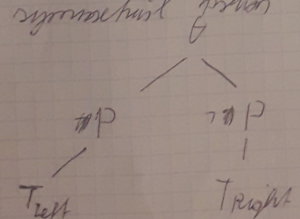
We can also push up $\neg\neg$ as it is a result of θ.



$\Downarrow$



X cut is in this case $P \supset Q$. Here $\neg X$ is α and X is β we thus apply the rules as stated in symmetrical fashion



$X = (P \supset Q)\ (\beta)$

$\neg X = \neg(P \supset Q)\ (\alpha)$

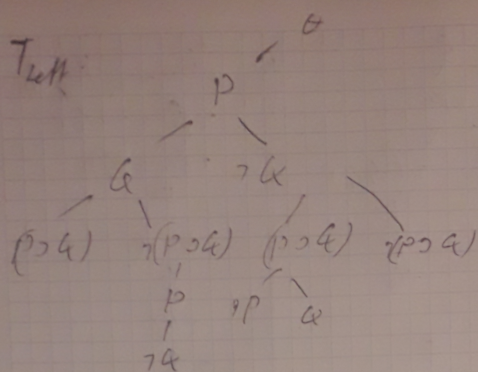
$\beta_1 = \neg P \qquad T_1 = \{\}$

$\beta_2 = Q \qquad T_2 = \{\}$

$\alpha_1 = P \qquad T_3 = \{\}$

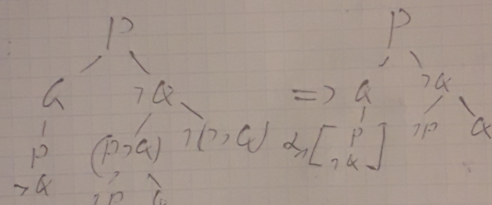
$\alpha_2 = \neg Q$

T_{left}:

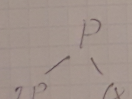


We can now eliminate cuts at the branch ends.

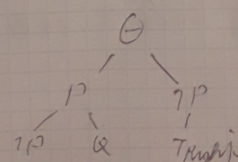
T_{left}:



push up α application: and eliminate cut



Table



T_{Right}

Θ

$\neg P$

$(P \supset Q)$ $\quad$ $\neg(P \supset Q)$

P

$\neg Q$

eliminate:

Θ

$\neg P$

$\left[\begin{matrix} P \\ \neg Q \end{matrix}\right]\alpha$

Push up α gives

Θ

P $\quad$ $\neg P$

$\neg P$ $\quad$ Q

$\Downarrow$

Θ

$\neg P$ $\quad$ Q

in table (disregarding duplicate sentences)

$\neg((P \supset \neg\neg Q) \supset (Q \supset R))$

$\neg(P \supset \neg\neg Q)$

$\neg(Q \supset R)$

P

$\neg\neg\neg Q$

Q

$\neg R$

$\neg Q$

$\neg P$ $\quad$ Q

Exercise 3 *(8.11.1) We do the subcase that $\gamma \in S_2$ in the proof of Lemma 8.11.3 in detail. That is, assuming $S \cup \{\gamma(t)\}$ is not Craig consistent, we show S is not Craig consistent either. That is, we must show that every partition S_1, S_2, i.e. pair of sets such that $S = S_1 \cup S_2$ and $\varnothing = S_1 \cap S_2$, has an interpolant, i.e. a formula Z such that $S_1 \cup \{Z\}$ and $S_2 \cup \{\neg Z\}$ are not satisfiable. We adapt the text from the book for the case $\gamma \in S_1$, as follows:*[1]

$S_1, S_2 \cup \{\gamma(t)\}$ is a partition of $S \cup \{\gamma(t)\}$, so it has an interpolant, say Z, since $S \cup \{\gamma(t)\}$ is not Craig-consistent. Now $S_1 \cup \{Z\}$ is not satisfiable. Also, $S_2 \cup \{\gamma(t)\} \cup \{\neg Z\}$ is not satisfiable, and it follows easily that $S_2 \cup \{\neg Z\}$ is not satisfiable either, since $\gamma \in S_2$.

We know that all constant, function, and relation symbols of Z are common to S_1 and $S_2 \cup \{\gamma(t)\}$. If the all occur in S_2, we are done; Z is an interpolant for S_1, S_2. So now suppose Z contains some symbol occurring in $S_2 \cup \{\gamma(t)\}$ but not in S_2. Since $\gamma \in S_2$, any such symbol must occur in t and so must be a constant or a function symbol. There may be several; for simplicity let is say Z contains just one subterm, $f(u_1, \ldots, u_n)$, where f occurs in t but not in S_2. The more general situation is treated similarly.

Let s be a new free variable, and let Z^* be like Z but with the occurrence of $f(u_1, \ldots, u_n)$ replaced by x, so $Z = Z^*\{x/f(u_1, \ldots, u_n)\}$. We claim $(\forall x)Z^*$ is an interpolant for S_1, S_2.

First, all constant, function, and relation symbols of $(\forall x)Z^*$ are common to both S_1 and S_2, because we have removed the only one that was a problem. Next, $S_1 \cup \{Z\}$ is not satisfiable, hence, neither is $S_1 \cup \{(\forall x)Z^*\}$. This follows from the validity of

$$(\forall x)Z^* \supset Z^*\{x/f(u_1, \ldots, u_n)$$

Finally, $S_2 \cup \{\neg Z\}$ is not satisfiable, and it follows that $S_2 \cup \{\neg(\forall x)Z^*\}$ is also not satisfiable. This argument needs a little more discussion than the others. (Its similarity to the proof of Lemma 8.3.1 is no coincidence.)

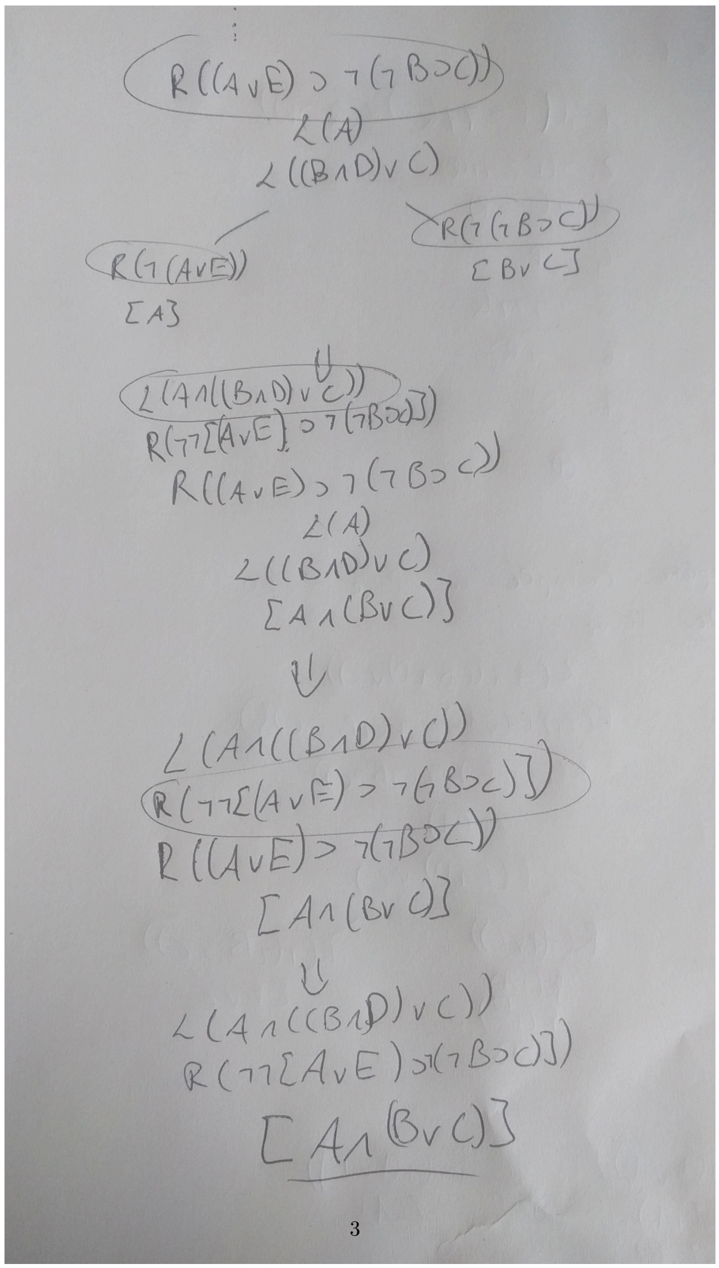
Suppose $S_2 \cup \{\neg(\forall x)Z^*\}$ is satisfiable, we show $S_2 \cup \{\neg Z\}$ also is. Suppose the members of $S_2 \cup \{\neg(\forall x)Z^*\}$ are true in the model $\langle D, I\rangle$. Then in particular, $\neg Z^{*I,A}$ is true for some assignment A. Now define a new interpretation J to be like I on all symbols except f, and set f^J to the same as f^I on all members of D except $u_1^{I,A}, \ldots, u_n^{I,A}$. Finally, set $f^J(u_1^{I,A}, \ldots, u_n^{I,A}) = x^I$. Since I and J differ only on f, and that does not occur in S_2, the members of this set will have the same truth values using either interpretation. Consequently, the members of S_2 are true in $\langle D, J\rangle$. Using Proposition 5.3.7 $[Z^*\{x/f(u_1, \ldots, u_n)\}]^{J,A} = Z^{*J,A} = Z^{*I,A} = t$.

Remark 2 *All of you doing the exercise had the correct idea to try to adapt the text in the book. But only one of you gave a correct adaptation. The others failed to check that their adaptation worked, i.e. that the sentences stay true (otherwise they would have found their adaptation to be faulty). Checking truth*

[1] The adaptation consists in working with γ in the second component instead of the first component of the partition, and correspondingly abstracting from offending symbols in Z using a $\forall$ instead of an $\exists$, since Z formula is negated in the second component.

is essential; otherwise it's not a proof. For instance, again trying to take $(\exists x)Z^*$ *as interpolant, that* $S_1 \cup \{Z\}$ *is not satisfiable would* not *imply* $S_1 \cup \{(\exists x)Z^*\}$ *is not satisfiable (we could have a satisfying assignment for* x*).*

Exercise 4 *(complete example p. 260) Caused no problems. The answer should be something like the following, provided by one of you:*



$\vdots$

$R((A \vee E) \supset \neg(\neg B \supset C))$

$L(A)$

$L((B \wedge D) \vee C)$

$R(\neg(A \vee E))$ $\quad$ $R(\neg(\neg B \supset C))$

$[A]$ $\quad$ $[B \vee C]$

$\Downarrow$

$L(A \wedge ((B \wedge D) \vee C))$

$R(\neg\neg[A \vee E] \supset \neg(\neg B \supset C)])$

$R((A \vee E) \supset \neg(\neg B \supset C))$

$L(A)$

$L((B \wedge D) \vee C)$

$[A \wedge (B \vee C)]$

$\Downarrow$

$L(A \wedge ((B \wedge D) \vee C))$

$R(\neg\neg[(A \vee E) \supset \neg(\neg B \supset C)])$

$R((A \vee E) \supset \neg(\neg B \supset C))$

$[A \wedge (B \vee C)]$

$\Downarrow$

$L(A \wedge ((B \wedge D) \vee C))$

$R(\neg\neg[A \vee E) \supset \neg(\neg B \supset C)])$

$[A \wedge (B \vee C)]$

3

Exercise 5 *(8.12.1) The answer should be something like the following, provided by one of you:*

$$
\begin{array}{c}
L((\forall x)(P(x) \supset \neg Q(x)) \wedge P(c)) \\
R(\neg\neg Q(c)) \\
R(Q(c)) \\
L((\forall x)(P(x) \supset \neg Q(x))) \\
L(P(c)) \\
L(P(c) \supset \neg Q(c)) \\
\begin{array}{cc}
L(\neg P(c)) & L(\neg Q(c)) \\
[\perp] & [\neg Q(c)]
\end{array}
\end{array}
\qquad
\begin{array}{c}
L((\forall x)(P(x) \supset \neg Q(x)) \wedge P(c)) \\
R(\neg\neg Q(c)) \\
R(Q(c)) \\
L((\forall x)(P(x) \supset \neg Q(x))) \\
L(P(c)) \\
L(P(c) \supset \neg Q(c)) \\
[\perp \vee \neg Q(c)]
\end{array}
$$

$$
\begin{array}{c}
L((\forall x)(P(x) \supset \neg Q(x)) \wedge P(c)) \\
R(\neg\neg Q(c)) \\
R(Q(c)) \\
L((\forall x)(P(x) \supset \neg Q(x))) \\
L(P(c)) \\
[\perp \vee \neg Q(c)]
\end{array}
\qquad
\begin{array}{c}
L((\forall x)(P(x) \supset \neg Q(x)) \wedge P(c)) \\
R(\neg\neg Q(c)) \\
R(Q(c)) \\
[\perp \vee \neg Q(c)]
\end{array}
$$

$$
\begin{array}{c}
L((\forall x)(P(x) \supset \neg Q(x)) \wedge P(c)) \\
R(\neg\neg Q(c)) \\
[\perp \vee \neg Q(c)]
\end{array}
$$

Exercise 6 *(8.12.2) We have to show that if $X \supset Y$ is a valid sentence, then it has an interpolant Z such that every relation symbol occurring positively in Z has a positive occurrence in both X and Y, and every relation symbol occurring negatively in Z has a negative occurrence in both X and Y. To that end, we have to show that this in fact follows from the construction of the interpolant for the biased tableau proof of $\neg(X \supset Y)$, i.e. starting from $L(X)$ and $R(\neg Y)$.*

We claim that for any calculation of Z from S by means of the calculation rules, every relation symbol occurring posi/negatively in Z has a posi/negative occurrence in X for some $L(X) \in S$ and a nega/positive occurrence in X for some $R(X) \in S$, by induction on the length of the calculation. From the claim we conclude since in the end the calculation rules compute Z from $L(X), R(\neg Y)$.

The claim is easily verified for the base cases, the 6 rules for the closed branches on page 258. Next we show it is preserved by the inference rules.

- *For the computation rule inferences of $\perp$ and $\top$ there is nothing to verify, since the formulas in S involved do not contain relation symbols.*

- *For the computation rule inference for double negations ($\neg\neg$), αs, βs, γs, and δs we conclude from the IH(s) and that the transformations on formulas in S and on Z, neither change relation symbols occurring in them nor their polarity.*

Remark 3 *Only one of the handed in solutions did do the required induction.*