Rozwiązania zadań z logiki klasycznej i operatorów sterowania

Zeimer

15 stycznia 2019

Zad. 1

(a) Chcemy ze specjalnego przypadku prawa Peirce'a ($P\bot$) oraz prawa eliminacji fałszu ($\bot E$) wyprowadzić prawo Peirce'a (P). W tym celu w systemie z regułami ($P\bot$) oraz ($\bot E$) udowodnimy $\vdash ((\varphi \to \psi) \to \varphi) \to \varphi$.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot \vdash (\varphi \to \psi) \to \varphi$}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot, \varphi \vdash \varphi \to \bot$}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot, \varphi \vdash \varphi$}
\BinaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot, \varphi \vdash \bot$}
\RightLabel{$\bot$E}
\UnaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot, \varphi \vdash \psi$}
\RightLabel{$\to$I}
\UnaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot \vdash \varphi \to \psi$}
\RightLabel{$\to E$}
\BinaryInfC{$(\varphi \to \psi) \to \varphi, \varphi \to \bot \vdash \varphi$}
\RightLabel{$P\bot$}
\UnaryInfC{$(\varphi \to \psi) \to \varphi \vdash \varphi$}
\RightLabel{$\to I$}
\UnaryInfC{$\vdash ((\varphi \to \psi) \to \varphi) \to \varphi$}
\end{prooftree}
$$

(b) Chcemy ze specjalnego przypadku prawa Peirce'a P$\bot$ oraz prawa eliminacji fałszu $\bot$E wyprowadzić prawo eliminacji podwójnej negacji $\neg\neg$E. W tym celu pokażemy w systemie z tymi regułami, że $\vdash ((\varphi \to \bot) \to \bot) \to \varphi$.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \bot) \to \bot, \varphi \to \bot \vdash (\varphi \to \bot) \to \bot$}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \bot) \to \bot, \varphi \to \bot \vdash \varphi \to \bot$}
\RightLabel{$\to$E}
\BinaryInfC{$(\varphi \to \bot) \to \bot, \varphi \to \bot \vdash \bot$}
\RightLabel{$\bot$E}
\UnaryInfC{$(\varphi \to \bot) \to \bot, \varphi \to \bot \vdash \varphi$}
\RightLabel{P$\bot$}
\UnaryInfC{$(\varphi \to \bot) \to \bot \vdash \varphi$}
\RightLabel{$\to$I}
\UnaryInfC{$\vdash ((\varphi \to \bot) \to \bot) \to \varphi$}
\end{prooftree}
$$

(c) Chcemy z prawa eliminacji podwójnej negacji $\neg\neg$E wyprowadzić szczególny przypadek prawa Peirce'a P$\bot$ oraz prawo eliminacji fałszu $\bot$E.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$\bot, \varphi \to \bot \vdash \bot$}
\RightLabel{$\neg\neg$E}
\UnaryInfC{$\bot \vdash \varphi$}
\RightLabel{$\to$I}
\UnaryInfC{$\vdash \bot \to \varphi$}
\end{prooftree}
$$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \bot) \to \varphi, \varphi \to \bot \vdash \varphi \to \bot$}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \bot) \to \varphi, \varphi \to \bot \vdash (\varphi \to \bot) \to \varphi$}
\AxiomC{}
\RightLabel{Ass}
\UnaryInfC{$(\varphi \to \bot) \to \varphi, \varphi \to \bot \vdash \varphi \to \bot$}
\RightLabel{$\to$E}
\BinaryInfC{$(\varphi \to \bot) \to \varphi, \varphi \to \bot \vdash \varphi$}
\RightLabel{$\to$E}
\BinaryInfC{$(\varphi \to \bot) \to \varphi, \varphi \to \bot \vdash \bot$}
\RightLabel{$\neg\neg$E}
\UnaryInfC{$(\varphi \to \bot) \to \varphi \vdash \varphi$}
\RightLabel{$\to$I}
\UnaryInfC{$\vdash ((\varphi \to \bot) \to \varphi) \to \varphi$}
\end{prooftree}
$$

Zad. 2

Udowodnijmy najpierw (w systemie z ¬¬E) prawo wyłączonego środka $\varphi \lor \neg\varphi$.

$$
\dfrac{\dfrac{\dfrac{}{(\varphi \lor \neg\varphi) \to \bot \vdash (\varphi \lor \neg\varphi) \to \bot}\,\text{Ass} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{(\varphi \lor \neg\varphi) \to \bot, \varphi \vdash (\varphi \lor \neg\varphi) \to \bot}\,\text{Ass} \qquad \dfrac{\dfrac{}{(\varphi \lor \neg\varphi) \to \bot, \varphi \vdash \varphi}\,\text{Ass}}{(\varphi \lor \neg\varphi) \to \bot, \varphi \vdash \varphi \lor \neg\varphi}\,\lor I_1}{(\varphi \lor \neg\varphi) \to \bot, \varphi \vdash \bot}\,\to\!\text{E}}{(\varphi \lor \neg\varphi) \to \bot \vdash \neg\varphi}\,\to\!\text{I}}{(\varphi \lor \neg\varphi) \to \bot \vdash \varphi \lor \neg\varphi}\,\lor I_2}{(\varphi \lor \neg\varphi) \to \bot \vdash \bot}\,\to\!\text{E}}{\vdash \varphi \lor \neg\varphi}\,\neg\neg\text{E}
$$

Rzut okiem na termy i ich reguły typowania pozwala stwierdzić, że termem odpowiadającym temu dowodowi jest (anotacje typów pozwolę sobie pominąć) $\mathrm{EM}_\varphi :\equiv \Delta x.x(\iota_2(\lambda y.x(\iota_1 y)))$, zaś rzut okiem na reguły redukcji $\to_\Delta$ pozwala stwierdzić, że term ten jest w postaci normalnej (reguły 1 i 3 nie pasują, zaś w regule 2 nie jest spełniona przesłanka dotycząca zmiennej wolnej).

Chcemy teraz znormalizować term $\lambda t.\lambda f.\mathrm{case}(\mathrm{EM}_\varphi, x_1.t, x_2.f)$. Normalizujemy więc:

$$
\begin{gathered}
\lambda t.\lambda f.\mathrm{case}(\mathrm{EM}_\varphi, x_1.t, x_2.f) \equiv \lambda t.\lambda f.\Delta k.(\Delta x.x(\lambda x_1.\lambda x_2.(\lambda y.x(\lambda z_1.\lambda z_2.z_1 y))))(\lambda x_1.kt)(\lambda x_2.kf) \to_\Delta \\
\lambda t.\lambda f.\Delta k.(\Delta z.(\lambda w.z(w(\lambda x_1.kt)))(\lambda x_1.\lambda x_2.x_2(\lambda y.(\lambda w.z(w(\lambda x_1.kt)))(\lambda z_1.\lambda z_2.y))))(\lambda x_2.kf) \to_\beta \\
\lambda t.\lambda f.\Delta k.(\Delta z.(\lambda w.z(w(\lambda x_1.kt)))(\lambda x_1.\lambda x_2.x_2(\lambda y.z(\lambda z_2.kt))))(\lambda x_2.kf) \to_\beta^* \lambda t.\lambda f.\Delta k.(\Delta z.z(\lambda x_2.x_2(\lambda y.z(\lambda z_2.kt))))(\lambda x_2.kf) \to_\Delta \\
\lambda t.\lambda f.\Delta k.\Delta w.(\lambda a.w(a(\lambda x_2.kf)))(\lambda x_2.x_2(\lambda y.(\lambda a.w(a(\lambda x_2.kf)))(\lambda z_2.kt))) \to_\beta^* \\
\lambda t.\lambda f.\Delta k.\Delta w.(\lambda a.w(a(\lambda x_2.kf)))(\lambda x_2.x_2(\lambda y.w(kt))) \to_\beta \lambda t.\lambda f.\Delta k.\Delta w.w((\lambda x_2.x_2(\lambda y.w(kt)))(\lambda x_2.kf)) \to_\beta \\
\lambda t.\lambda f.\Delta k.\Delta w.w((\lambda x_2.kf)(\lambda y.w(kt))) \to_\beta \lambda t.\lambda f.\Delta k.\Delta w.w(kf) \to_\Delta \lambda t.\lambda f.\Delta k.kf \to_\Delta \lambda t.\lambda f.f
\end{gathered}
$$

Jak widać, wychodzi funkcja stale zwracająca drugi argument (czyli fałsz).