Natural Deduction

Foundations for Programming Languages
MASTER IN SOFTWARE AND SYSTEMS
Universidad Politécnica de Madrid/IMDEA Software Institute

Halloween, 2017

To be turned in by November 19, 2017. Send them to me by mail to `jmarino@fi.upm.es`.

Lógica proposicional clásica

Usa las reglas de deducción natural (NK = NJ + DN) mostradas en la figura 1 para resolver los siguientes ejercicios.

Exercise 1. Para cada una de las siguientes proposiciones, estudia si son *tautologías*. Para las que no lo sean, proporciona un contraejemplo (una asignación de valores de verdad a sus variables proposicionales que hace falsa la fórmula).

a. $(p \vee p) \rightarrow p$.

b. $((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r))$.

c. $(p \rightarrow q) \leftrightarrow \neg(\neg p \wedge q)$.

d. $(p \wedge (p \vee q)) \leftrightarrow p$.

e. $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$.

f. $p \rightarrow (q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$.

g. $(p \rightarrow q) \rightarrow ((p \rightarrow \neg q) \rightarrow \neg p)$.

h. $p \rightarrow \neg\neg p$.

i. $\neg\neg p \rightarrow p$.

j. $(p \rightarrow q) \vee (q \rightarrow p)$.

k. $((p \rightarrow q) \rightarrow p) \rightarrow p$.

l. $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$.

Exercise 2. Da una demostración en NK para cada una de las tautologías del ejercicio anterior. ¿Qué fórmulas son demostrables en NJ?

Exercise 3. Demuestra el teorema de corrección (*soundness theorem*) para la lógica proposicional clásica, es decir, muestra que si hay una demostración en NK de ϕ a partir de unas premisas Γ (formalmente $\Gamma \vdash_{\mathrm{NK}} \phi$), entonces cualquier valoración que haga Γ cierta también hace cierta a ϕ (formalmente $\Gamma \models \phi$).

Exercise 4. NK también es *completo* para lógica proposicional clásica, es decir, si $\Gamma \models \phi$, entonces $\Gamma \vdash_{\mathrm{NK}} \phi$. Proporciona una estrategia de demostración para este teorema de completitud.

NJ: deducción natural para lógica proposicional intuicionista.

$$\frac{\phi \qquad \psi}{\phi \wedge \psi}\wedge i \qquad \frac{\phi \wedge \psi}{\phi}\wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi}\wedge e_2$$

$$\frac{\phi}{\phi \vee \psi}\vee i_1 \qquad \frac{\psi}{\phi \vee \psi}\vee i_2 \qquad \frac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \xi\end{array} \qquad \begin{array}{c}[\psi]\\ \xi\end{array}}{\xi}\vee e$$

$$\frac{\begin{array}{c}[\phi]\\ \psi\end{array}}{\phi \to \psi}\to i \qquad \frac{\phi \to \psi \qquad \phi}{\psi}\to e$$

$$\frac{\begin{array}{c}[\phi]\\ \psi \wedge \neg\psi\end{array}}{\neg\phi}\neg i \qquad \frac{\phi \wedge \neg\phi}{\psi}\neg e$$

NK: añadir una de las siguientes reglas para cubrir la lógica clásica.

$$\frac{\neg\neg\phi}{\phi}\,\mathrm{DN} \qquad \frac{}{\phi \vee \neg\phi}\,\mathrm{TND}$$

Figura 1: NK: sistema de deducción natural para lógica proposicional clásica.

Lógica intuicionista vs. lógica clásica

Exercise 5. Demuestra *tertium non datur* (TND) en NJ + DN.

Exercise 6. Demuestra *double negation* (DN) en NJ + TND.

Lógica proposicional en Haskell

Exercise 7. Define un tipo de datos *Prop* para representar fórmulas de lógica proposicional.

Exercise 8. Define una función $taut :: Prop \to Bool$ que decida si una fórmula proposicional es una tautología. Aplica a las fórmulas del ejercicio 1.