Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

PPA - Un asistente de demostración para lógica de primer orden con extracción de testigos usando la traducción de Friedman

Tesis de Licenciatura en Ciencias de la Computación

Manuel Panichelli

Director: Pablo Barenbaum

Buenos Aires, 2024

PPA - UN ASISTENTE DE DEMOSTRACIÓN PARA LÓGICA DE PRIMER ORDEN CON EXTRACCIÓN DE TESTIGOS USANDO LA TRADUCCIÓN DE FRIEDMAN

La princesa Leia, líder del movimiento rebelde que desea reinstaurar la República en la galaxia en los tiempos ominosos del Imperio, es capturada por las malévolas Fuerzas Imperiales, capitaneadas por el implacable Darth Vader. El intrépido Luke Skywalker, ayudado por Han Solo, capitán de la nave espacial "El Halcón Milenario", y los androides, R2D2 y C3PO, serán los encargados de luchar contra el enemigo y rescatar a la princesa para volver a instaurar la justicia en el seno de la Galaxia (aprox. 200 palabras).

Palabras claves: Guerra, Rebelión, Wookie, Jedi, Fuerza, Imperio (no menos de 5).

PPA - A PROOF-ASSISTANT FOR FIRST-ORDER LOGIC WITH WITNESS EXTRACTION USING FRIEDMAN'S TRANSLATION

In a galaxy far, far away, a psychopathic emperor and his most trusted servant – a former Jedi Knight known as Darth Vader – are ruling a universe with fear. They have built a horrifying weapon known as the Death Star, a giant battle station capable of annihilating a world in less than a second. When the Death Star's master plans are captured by the fledgling Rebel Alliance, Vader starts a pursuit of the ship carrying them. A young dissident Senator, Leia Organa, is aboard the ship & puts the plans into a maintenance robot named R2-D2. Although she is captured, the Death Star plans cannot be found, as R2 & his companion, a tall robot named C-3PO, have escaped to the desert world of Tatooine below. Through a series of mishaps, the robots end up in the hands of a farm boy named Luke Skywalker, who lives with his Uncle Owen & Aunt Beru. Owen & Beru are viciously murdered by the Empire's stormtroopers who are trying to recover the plans, and Luke & the robots meet with former Jedi Knight Obi-Wan Kenobi to try to return the plans to Leia Organa's home, Alderaan. After contracting a pilot named Han Solo & his Wookiee companion Chewbacca, they escape an Imperial blockade. But when they reach Alderaan's coordinates, they find it destroyed - by the Death Star. They soon find themselves caught in a tractor beam & pulled into the Death Star. Although they rescue Leia Organa from the Death Star after a series of narrow escapes, Kenobi becomes one with the Force after being killed by his former pupil - Darth Vader. They reach the Alliance's base on Yavin's fourth moon, but the Imperials are in hot pursuit with the Death Star, and plan to annihilate the Rebel base. The Rebels must quickly find a way to eliminate the Death Star before it destroys them as it did Alderaan (aprox. 200 palabras).

Keywords: War, Rebellion, Wookie, Jedi, The Force, Empire (no menos de 5).

AGRADECIMIENTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sapien ipsum, aliquet eget convallis at, adipiscing non odio. Donec porttitor tincidunt cursus. In tellus dui, varius sed scelerisque faucibus, sagittis non magna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris et luctus justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris sit amet purus massa, sed sodales justo. Mauris id mi sed orci porttitor dictum. Donec vitae mi non leo consectetur tempus vel et sapien. Curabitur enim quam, sollicitudin id iaculis id, congue euismod diam. Sed in eros nec urna lacinia porttitor ut vitae nulla. Ut mattis, erat et laoreet feugiat, lacus urna hendrerit nisi, at tincidunt dui justo at felis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut iaculis euismod magna et consequat. Mauris eu augue in ipsum elementum dictum. Sed accumsan, velit vel vehicula dignissim, nibh tellus consequat metus, vel fringilla neque dolor in dolor. Aliquam ac justo ut lectus iaculis pharetra vitae sed turpis. Aliquam pulvinar lorem vel ipsum auctor et hendrerit nisl molestie. Donec id felis nec ante placerat vehicula. Sed lacus risus, aliquet vel facilisis eu, placerat vitae augue.

Índice general

1. INTRODUCCIÓN

2. DEDUCCIÓN NATURAL

Vamos a comenzar por las fundaciones: Queremos armar un programa que permita escribir teoremas y demostraciones. ¿Cómo se representa una demostración en la computadora? Es necesaria una representación precisa y rigurosa.

En el área de estudio de *proof theory*, en la cuál las demostraciones son tratadas como objetos matemáticos formales, nos encontramos con los *proof calculi* o *proof systems*, que son sistemas lógicos formales que permiten demostrar sentencias. Pueden ser modelados como un tipo abstracto de datos, así siendo representados en la computadora.

Por ejemplo, supongamos que tenemos la siguiente *teoría* de exámenes en la facultad, que vamos a ir iterando a lo largo de la tesis. Por ahora, en su versión proposicional. Si un alumno reprueba un final, entonces recursa. Si un alumno falta, entonces reprueba. Con estas dos, podríamos demostrar que si un alumno falta a un final, entonces recursa. Veamos cómo podría ser una demostración en lenguaje natural.

Ejemplo 1. Si ((reprueba entonces recursa) y (falta entonces reprueba)) y falta, entonces recursa.

Demostración:

- Asumo que falta. Quiero ver que recursa.
- Sabemos que si falta, entonces reprueba. Reprobó.
- Sabemos que si reprueba, entonces recursa.
- $\therefore$ recursó.

□

¿Cómo podría ser formalizada en un *proof system*?

2.1. El sistema de deducción natural

Los *proof systems* en general están compuestos por

- **Lenguaje formal**: el conjunto L de fórmulas admitidas por el sistema. En nuestro caso, lógica de primer órden.
- **Reglas de inferencia**: lista de reglas que se usan para probar teoremas de axiomas y otros teoremas. Por ejemplo, *modus ponens* (si es cierto $A \to B$ y A, se puede concluir B) o *modus tollens* (si es cierto $A \to B$ y $\neg B$, se puede concluir $\neg A$)
- **Axiomas**: fórmulas de L que se asumen válidas. Todos los teoremas se derivan de axiomas. Por ejemplo, como estamos en lógica clásica, vale el axioma *LEM* (Law of Excluded Middle): $A \vee \neg A$

El sistema particular que usamos se conoce como **deducción natural**, introducido por Gerhard Gentzen en [Gen35] (TODO: Chequear cita). Tiene dos tipos de *reglas de inferencia* para cada operador ($\wedge$, $\vee$, $\exists$, ...), que nos permiten razonar

- **Introducción**: ¿Cómo demuestro este operador?
- **Eliminación**: ¿Cómo uso este operador para demostrar otra fórmula?

Introducimos algunas definiciones preliminares, luego vemos las reglas de inferencia, un ejemplo de una demostración, y finalmente explicamos cada regla.

Def. 3. Contexto de demostración.

- Definimos los **contextos de demostración** Γ como un conjunto de fórmulas, compuesto por las hipótesis que se asumen a lo largo de una demostración.
- Para algunas reglas es necesario conocer las variables libres de un contexto, que se definen de la forma usual:

$$fv(\Gamma) = \bigcup_{A \in \Gamma} fv(A)$$

Def. 4. Relación $\vdash$. Las reglas de inferencia de la Figura 2.1 definen la siguiente relación, que intuitivamente puede ser interpretada como "A es una consecuencia de las suposiciones de Γ"

$$\Gamma \vdash A$$

Def. 5. Sustitución. Notamos la **sustitución sin capturas** de todas las ocurrencias libres de la variable x por el término t en la fórmula A como

$$A\{x := t\}$$

Se explora en más detalle en la Subsección 2.3.3 Sustitución sin capturas

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\,\mathrm{E}\bot \qquad \frac{}{\Gamma \vdash \top}\,\mathrm{I}\top$$

$$\frac{}{\Gamma \vdash A \vee \neg A}\,\mathrm{LEM} \qquad \frac{}{\Gamma, A \vdash A}\,\mathrm{Ax}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,\mathrm{I}\wedge$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,\mathrm{E}\wedge_1 \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,\mathrm{E}\wedge_2$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,\mathrm{I}\vee_1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,\mathrm{I}\vee_2$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\,\mathrm{E}\vee$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,\mathrm{I}\to \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\,\mathrm{E}\to$$

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\,\mathrm{I}\neg \qquad \frac{\Gamma \vdash \neg A \qquad \Gamma \vdash A}{\Gamma \vdash \bot}\,\mathrm{E}\neg$$

$$\frac{\Gamma \vdash A \qquad x \notin fv(\Gamma)}{\Gamma \vdash \forall x.A}\,\mathrm{I}\forall \qquad \frac{\Gamma \vdash \forall x.A}{\Gamma \vdash A\{x := t\}}\,\mathrm{E}\forall$$

$$\frac{\Gamma \vdash A\{x := t\}}{\Gamma \vdash \exists x.A}\,\mathrm{I}\exists$$

$$\frac{\Gamma \vdash \exists x.A \qquad \Gamma, A \vdash B \qquad x \notin fv(\Gamma, B)}{\Gamma \vdash B}\,\mathrm{E}\exists$$

Fig. 2.1: Reglas de inferencia para deducción natural de lógica de primer órden

Ejemplo 2. Demostración de Ejemplo 1 en deducción natural. Como es en su versión proposicional, vamos a modelarlo para un solo alumno y materia. Notamos

- $X \equiv \text{reprueba}(juan, \text{final}(logica))$
- $R \equiv \text{recursa}(juan, logica)$
- $F \equiv \text{falta}(juan, \text{final}(logica))$

Queremos probar entonces

$$\Big((X \to R) \wedge (F \to X)\Big) \to (F \to R)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (X \to R) \wedge (F \to X)}\,\text{Ax}}{\Gamma \vdash X \to R}\,\text{E}\wedge_1 \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash (X \to R) \wedge (F \to X)}\,\text{Ax}}{\Gamma \vdash F \to X}\,\text{E}\wedge_2 \qquad \dfrac{}{\Gamma \vdash F}\,\text{Ax}}{\Gamma \vdash X}\,\text{E}\to}{\dfrac{\Gamma = (X \to R) \wedge (F \to X),\ F \vdash R}{(X \to R) \wedge (F \to X) \vdash F \to R}\,\text{I}\to}\,\text{I}\to}{\vdash \Big((X \to R) \wedge (F \to X)\Big) \to (F \to R)}$$

Fig. 2.2: Demostración de $\big((X \to R) \wedge (F \to X)\big) \to (F \to R)$ en deducción natural

Las demostraciones en deducción natural son un árbol, en el que cada juicio está justificado por una regla de inferencia, que puede tener sub-árboles de demostración. La raíz es la fórmula a demostrar. Paso por paso,

- I→: *introducimos* la implicación. Para demostrarla, asumimos el antecedente y en base a eso demostramos el consecuente. Es decir asumimos $(X \to R) \wedge (F \to X)$, y en base a eso queremos deducir $F \to R$.
- I→: Asumimos F, nos queda probar R. Renombramos el *contexto* de hipótesis como Γ.
- La estrategia para probar R es usando la siguiente cadena de implicaciones: $F \to X \to R$, y sabemos que vale F. Como tenemos que probar R, arrancamos de atrás para adelante.
- E→: *eliminamos* una implicación, la usamos para deducir su conclusión demostrando el antecedente. Esta regla de inferencia tiene dos partes, probar la implicación $(X \to R)$, y probar el antecedente (X).
  - Para probar la implicación, tenemos que usar la hipótesis *eliminando* la conjunción y especificando cuál de las dos cláusulas estamos usando.
  - Para probar el antecedente X, es un proceso análogo pero usando la otra implicación y el hecho de que vale F por hipótesis.
- Las hojas del árbol, los casos base, suelen ser aplicaciones de la regla de inferencia Ax, que permite deducir fórmulas citando hipótesis del contexto.

2.2. Reglas de inferencia

A continuación se explican brevemente las reglas de inferencia listadas en Figura 2.1.

2.2.1. Reglas base

$$\dfrac{\Gamma \vdash \bot}{\Gamma \vdash A}\,\text{E}\bot \qquad \dfrac{}{\Gamma \vdash \top}\,\text{I}\top$$

$$\dfrac{}{\Gamma \vdash A \vee \neg A}\,\text{LEM} \qquad \dfrac{}{\Gamma, A \vdash A}\,\text{Ax}$$

- E⊥: A partir de ⊥, algo que es falso, vamos a poder deducir cualquier fórmula.

- I$\top$: $\top$ trivialmente vale siempre
- LEM: El *principio del tercero excluido* que vale en lógica clásica. Incluir este axioma es lo que hace que este sistema sea clásico.
- Ax: Como ya vimos en el Ejemplo 2 , lo usamos para deducir fórmulas que ya tenemos como hipótesis.

2.2.2. Reglas de conjunciones y disyunciones

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\, \mathrm{I}\wedge$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\, \mathrm{E}\wedge_1 \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\, \mathrm{E}\wedge_2$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\, \mathrm{I}\vee_1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\, \mathrm{I}\vee_2$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\, \mathrm{E}\vee$$

- I$\wedge$: Para demostrar una conjunción, debemos demostrar ambas fórmulas.
- E$\wedge_1$/ E$\wedge_2$: A partir de una conjunción podemos deducir cualquiera de las dos fórmulas que la componen, porque ambas valen. Se modela con dos reglas.
- I$\vee_1$/ I$\vee_2$: Para demostrar una disyunción, alcanza con demostrar una de sus dos fórmulas. Se modela con dos reglas al igual que la eliminación de conjunción.
- E$\vee$: Nos permite deducir una conclusión a partir de una disyunción dando sub demostraciones que muestran que sin importar cual de las dos valga, asumiéndolas por separado, se puede demostrar.

2.2.3. Reglas de implicación y negación

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\, \mathrm{I}\to \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\, \mathrm{E}\to$$

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\, \mathrm{I}\neg \qquad \frac{\Gamma \vdash \neg A \qquad \Gamma \vdash A}{\Gamma \vdash \bot}\, \mathrm{E}\neg$$

- I$\to$: Para demostrar una implicación, asumimos el antecedente (agregándolo a las hipótesis) y en base a eso se demuestra el consecuente.
- E$\to$: también conocida como *modus ponens*. A partir de una implicación, si podemos demostrar su antecedente, entonces vale su consecuente.
- I$\neg$: Para demostrar una negación, lo hacemos por el absurdo: asumimos que vale la fórmula y llegamos a una contradicción. Esta regla también se suele llamar *reducción al absurdo* o RAA.
- E$\neg$: Podemos concluir un absurdo demostrando que vale una fórmula y su negación.

2.2.4. Reglas de cuantificadores

Las reglas de $\forall$ y $\exists$ se pueden ver como extensiones a las de $\wedge$ y $\vee$. Un $\forall$ se puede pensar como una conjunción con un elemento por cada uno del dominio sobre el cual se cuantifica, y análogamente un $\exists$ como una disyunción.

$$\frac{\Gamma \vdash A \qquad x \notin fv(\Gamma)}{\Gamma \vdash \forall x.A}\ \mathrm{I}\forall \qquad\qquad \frac{\Gamma \vdash \forall x.A}{\Gamma \vdash A\{x := t\}}\ \mathrm{E}\forall$$

- I$\forall$: Para demostrar un $\forall x.A$, quiero ver que sin importar el valor que tome x yo puedo demostrar A. Pero para eso en el contexto Γ no tengo que tenerlo ligado a nada, sino no lo estaría demostrando en general.
- E$\forall$: Para usar un $\forall x.A$ para demostrar, como vale para todo x, puedo instanciarlo en *cualquier término* t.

$$\frac{\Gamma \vdash A\{x := t\}}{\Gamma \vdash \exists x.A}\ \mathrm{I}\exists$$

$$\frac{\Gamma \vdash \exists x.A \qquad \Gamma, A \vdash B \qquad x \notin fv(\Gamma, B)}{\Gamma \vdash B}\ \mathrm{E}\exists$$

- I$\exists$: Para demostrar un $\exists$, alcanza con instanciar x en un término t para el que sea cierto.
- E$\exists$: Para usar un $\exists$ para demostrar, es parecido a E$\vee$. Como tenemos que ver que vale para cualquier x, podemos concluir B tomando como hipótesis A con x sin instanciar.

Ejemplo 3. Para ejemplificar el uso de las reglas de cuantificadores, extendemos el Ejemplo 2 a primer órden. Usamos

- a representa un alumno, m una materia y e un examen.
- $X(a, e) \equiv \text{reprueba}(a, e)$
- $R(a, m) \equiv \text{recursa}(a, m)$
- $F(a, e) \equiv \text{falta}(a, e)$

Vamos a tomar los siguientes como *axiomas*, que van a formar parte del contexto inicial de la demostración. Es lo mismo que haremos en PPA para modelar teorías de primer órden.

- Si un alumno reprueba el final de una materia, entonces recursa

$$\forall a.\ \forall m.\ (X(a, \text{final}(m)) \to R(a, m))$$

- Si un alumno falta a un examen, lo reprueba

$$\forall a.\ \forall e.\ (F(a, e) \to X(a, e))$$

Definimos

$$\Gamma_0 = \{\forall a.\ \forall m.\ X(a, \text{final}(m)) \to R(a, m), \forall a.\ \forall e.\ F(a, e) \to X(a, e)\}$$

Luego, queremos probar $\Gamma_0 \vdash \forall a.\ \forall m.\ F(a, \text{final}(m)) \to R(a, m)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_1 \vdash \forall a \forall m.\ X(a, \text{final}(m)) \to R(a,m)}\,\text{Ax}}{\Gamma_1 \vdash \forall m.\ X(a, \text{final}(m)) \to R(a,m)}\,\text{E}\forall}{\Gamma_1 \vdash X(a, \text{final}(m)) \to R(a,m)}\,\text{E}\forall \qquad \Pi}{\dfrac{\dfrac{\dfrac{\Gamma_1 = \Gamma_0, F(a, \text{final}(m)) \vdash R(a,m)}{\Gamma_0 \vdash F(a, \text{final}(m)) \to R(a,m)}\,\text{I}\to}{\Gamma_0 \vdash \forall m.\ F(a, \text{final}(m)) \to R(a,m)}\,\text{I}\forall}{\Gamma_0 \vdash \forall a.\ \forall m.\ F(a, \text{final}(m)) \to R(a,m)}\,\text{I}\forall} \qquad \Gamma_1 \vdash X(a, \text{final}(m))}{}\,\text{E}\to$$

Con

$$\Pi = \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_1 \vdash \forall a.\ \forall e.\ F(a,e) \to X(a,e)}\,\text{Ax}}{\Gamma_1 \vdash \forall e.\ F(a,e) \to X(a,e)}\,\text{E}\forall}{\Gamma_1 \vdash F(a, \text{final}(m)) \to X(a, \text{final}(m))}\,\text{E}\forall \qquad \dfrac{}{\Gamma_1 \vdash F(a, \text{final}(m))}\,\text{Ax}}{\Gamma_1 \vdash X(a, \text{final}(m))}\,\text{E}\to$$

Fig. 2.3: Demostración con cuantificadores en deducción natural

2.2.5. Ajustes para generación de demostraciones

Hipótesis etiquetadas

En las secciones anteriores presentamos a los contextos Γ como *conjuntos* de fórmulas. Pero en realidad, para mayor claridad en las demostraciones, vamos a querer que las hipótesis estén **etiquetadas**. Para permitirlo, las diferencias son las siguientes.

- Los contextos Γ son conjuntos de pares $h : A$ de etiquetas y fórmulas.
- Las reglas que hacen uso de hipótesis, lo hacen nombrándolas.

$$\dfrac{\boldsymbol{h} : A \in \Gamma}{\Gamma \vdash A}\,\text{Ax} \qquad \dfrac{\Gamma, \boldsymbol{h} : A \vdash B}{\Gamma \vdash A \to B}\,\text{I}\to \qquad \dfrac{\Gamma, \boldsymbol{h} : A \vdash \bot}{\Gamma \vdash \neg A}\,\text{I}\neg$$

(DUDA: No veo por qué con esta presentación sería necesario tener a los nombres de las reglas con las etiquetas, por ej. $\text{I}\to_h$)

Variables libres en contexto

Las reglas I$\forall$ y E$\exists$ requieren que la variable del cuantificador no esté libre en el contexto. Esto representó un problema en la generación de demostraciones. Por ejemplo cuando se citan otros teoremas y se insertan sus demostraciones, si usan las mismas variables, lo cual es usual, llevaban a conflictos. Para evitar esos problemas, lo cambiamos por algo más permisivo pero que mantiene validez: en lugar de fallar, remueve del contexto todas las hipótesis que contengan libre esa variable en la sub-demostración.

$$\dfrac{\hat{\boldsymbol{\Gamma}} \vdash A \qquad x \notin fv(\Gamma)}{\Gamma \vdash \forall x.A}\,\text{I}\forall$$

$$\dfrac{\hat{\boldsymbol{\Gamma}} \vdash \exists x.A \qquad \hat{\boldsymbol{\Gamma}}, A \vdash B \qquad x \notin fv(B)}{\Gamma \vdash B}\,\text{E}\exists$$

donde

$$\hat{\mathbf{\Gamma}} = \{A \in \Gamma \mid x \notin fv(A)\}$$

2.2.6. Reglas admisibles

Antes mencionamos *modus tollens* como regla de inferencia, pero no aparece en la Figura 2.1. Esto es porque nos va a interesar tener un sistema lógico minimal: no vamos a agregar reglas de inferencia que se puedan deducir a partir de otras, es decir, *reglas admisibles*. Nos va a servir para simplificar el resto de PPA, dado que vamos a generar demostraciones en deducción natural y operar sobre ellas. Mientras más sencillas sean las partes con las que se componen, mejor. Las reglas admisibles las podemos demostrar para cualquier fórmula, así luego podemos usarlas como *macros*.

Ejemplo 4. *Modus tollens*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (A \to B) \wedge \neg B}\,\text{Ax}}{\Gamma \vdash \neg B}\,\text{E}\wedge_2 \quad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash (A \to B) \wedge \neg B}\,\text{Ax}}{\Gamma \vdash A \to B}\,\text{E}\wedge_1 \quad \dfrac{}{\Gamma \vdash A}\,\text{Ax}}{\Gamma \vdash B}\,\text{E}\to}{\Gamma = (A \to B) \wedge \neg B, A \vdash \bot}\,\text{E}\neg}{(A \to B) \wedge \neg B \vdash \neg A}\,\text{I}\neg}{\vdash (A \to B \wedge \neg B) \to \neg A}\,\text{I}\to$$

(NOTA: cut y dnegElim las voy a contar en la sección del certifier, que es donde se usan.)

2.3. Algoritmos

A continuación describimos los algoritmos que son necesarios para la implementación de deducción natural. El chequeo de las demostraciones, alpha equivalencia de fórmulas, sustitución sin capturas, y variables libres

2.3.1. Chequeo

El algoritmo de chequeo de una demostración en deducción natural consiste en recorrer recursivamente el árbol de demostración, chequeando que todas las inferencias sean válidas. Que se usan para demostrar la fórmula que corresponde (no un I$\wedge$ para un $\vee$) y que cumplen con las condiciones impuestas.

(DUDA: Acá falta algo. Tal vez un ejemplo de una demo errónea?)

2.3.2. Alpha equivalencia

Si tenemos una hipótesis $\exists x.f(x)$, sería ideal poder usarla para demostrar a partir de ella una fórmula $\exists y.f(y)$. Si bien no son exactamente iguales, son **alpha-equivalentes**: su estructura es la misma, pero tienen nombres diferentes para variables *ligadas* (no libres)

Def. 6. Alpha equivalencia. Se define la relación $\overset{\alpha}{=}$ como la que permite renombrar variables ligadas evitando capturas. Es la congruencia más chica que cumple con

$$(\forall x.A) \overset{\alpha}{=} (\forall y.A') \iff A\{x := z\} \overset{\alpha}{=} A'\{y := z\} \text{ con } z \text{ fresca}$$
$$(\exists x.A) \overset{\alpha}{=} (\exists y.A') \iff A\{x := z\} \overset{\alpha}{=} A'\{y := z\} \text{ con } z \text{ fresca}$$

Para implementarlo, un algoritmo naïve podría ser cuadrático: chequeamos recursivamente la igualdad estructural de ambas fórmulas. Si nos encontramos con un cuantificador con variables con nombres distintos, digamos x e y, elegimos una nueva variable *fresca* (para evitar capturas) y lo renombramos recursivamente en ambos. Luego continuamos con el algoritmo. Si en la base nos encontramos con dos variables, tienen que ser iguales.

Para hacerlo un poco más eficiente, se implementó un algoritmo lineal en la estructura de la fórmula. Mantenemos dos sustituciones de variables, una para cada fórmula. Si nos encontramos con $\exists x.f(x)$ y $\exists y.f(y)$, vamos a elegir una variable fresca igual que antes (por ejemplo z), pero en vez de renombrar recursivamente, que lo hace cuadrático, insertamos en cada sustitución los renombres $x \mapsto z$ y $y \mapsto z$. Luego, cuando estemos comparando dos variables libres, chequeamos que *sus renombres* sean iguales. En este ejemplo son alpha equivalentes, pues

$$\begin{aligned}
(\exists x.f(x)) \overset{\alpha}{=} (\exists y.f(y)) &\iff f(x) \overset{\alpha}{=} f(y) && \{x \mapsto z\}, \{y \mapsto z\} \\
&\iff x \overset{\alpha}{=} y && \{x \mapsto z\}, \{y \mapsto z\} \\
&\iff z = z. &&
\end{aligned}$$

2.3.3. Sustitución sin capturas

Notamos la sustitución de todas las ocurrencias libres de la variable x por un término t en una fórmula A como $A\{x := t\}$. Esto se usa en algunas reglas de inferencia,

$$\frac{\Gamma \vdash \forall x.A}{\Gamma \vdash A\{x := t\}} \, \mathrm{E}\forall$$

Pero queremos evitar **captura de variables**. Por ejemplo, en

$$\forall y.p(x)\{x := y\},$$

si sustituimos sin más, estaríamos involuntariamente "capturando" a y. Si hiciéramos que falle, tener que escribir las demostraciones con estos cuidados puede ser muy frágil y propenso a errores, por lo que es deseable que se resuelva *automáticamente*: cuando nos encontramos con una captura, sustituimos la variable ligada de forma que no ocurra.

$$\forall y.p(x)\{x := y\} = \forall \boldsymbol{z}.p(y)$$

donde z es una variable *fresca*.

Def. 7. Sustitución sin capturas. Se define inductivamente en la estructura la fórmula. Sean y una variable y t un término cualquiera.

- Términos

$$x\{y := t\} = \begin{cases} t & \text{si } x = y \\ x & \text{si no} \end{cases}$$
$$f(t_1, \ldots, t_n)\{y := t\} = f(t_1\{y := t\}, \ldots, t_n\{y := t\})$$

- Fórmulas

$$\begin{aligned}
\bot\{y := t\} &= \bot \\
\top\{y := t\} &= \top \\
p(t_1, \ldots, t_n)\{y := t\} &= p(t_1\{y := t\}, \ldots, t_n\{y := t\}) \\
(A \wedge B)\{y := t\} &= A\{y := t\} \wedge B\{y := t\} \\
(A \vee B)\{y := t\} &= A\{y := t\} \vee B\{y := t\} \\
(A \to B)\{y := t\} &= A\{y := t\} \to B\{y := t\} \\
(\neg A)\{y := t\} &= \neg A\{y := t\} \\
(\forall x.A)\{y := t\} &= \begin{cases} \forall z.(A\{x := z\})\{y := t\} & \text{si } x = y, \text{ con } z \text{ fresca} \\ \forall x.A\{y := t\} & \text{si no} \end{cases} \\
(\exists x.A)\{y := t\} &= \begin{cases} \exists z.(A\{x := z\})\{y := t\} & \text{si } x = y, \text{ con } z \text{ fresca} \\ \exists x.A\{y := t\} & \text{si no} \end{cases}
\end{aligned}$$

Para implementarlo, cada vez que nos encontramos con una captura, vamos a *renombrar* la variable del cuantificador por una nueva, fresca. Al igual que la alpha igualdad, esto se puede implementar de forma naïve cuadrática pero lo hicimos lineal. Mantenemos un único mapeo a lo largo de la sustitución, y cada vez que nos encontramos con una variable libre, si son iguales la sustituimos por el término, y si está mapeada la renombramos.

2.3.4. Variables libres

(DUDA: Mover a la sección de LPO en la introducción? esto es well-known)

Def. 8. Variables libres de fórmulas y términos. Se definen inductivamente en su estructura de la siguiente forma.

- Términos

$$fv(x) = \{x\}$$
$$fv(f(t_1, \ldots, t_n)) = \bigcup_{i \in 1 \ldots n} fv(t_i)$$

- Fórmulas

$$
\begin{aligned}
fv(\bot) &= \emptyset \\
fv(\top) &= \emptyset \\
fv(p(t_1, \ldots, t_n)) &= \bigcup_{i \in 1 \ldots n} fv(t_i) \\
fv(A \wedge B) &= fv(A) \cup fv(B) \\
fv(A \vee B) &= fv(A) \cup fv(B) \\
fv(A \to B) &= fv(A) \cup fv(B) \\
fv(\neg A) &= fv(A) \\
fv(\forall x.A) &= fv(A) \setminus x \\
fv(\exists x.A) &= fv(A) \setminus x
\end{aligned}
$$

Def. 9. Variables libres de una demostración. Sea Π una demostración. $fv(\Pi)$ son las variables libres de todas las fórmulas que la componen. Por ejemplo, para la siguiente

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\, \mathrm{I}\wedge$$

se tiene $fv(\Pi) = fv(A) \cup fv(B)$

3. PPA EL LENGUAJE

4. EXTRACCIÓN DE TESTIGOS DE EXISTENCIALES

5. LA HERRAMIENTA PPA

6. CONCLUSIONES

BIBLIOGRAFíA

[Gen35] Gerhard Gentzen. «Untersuchungen über das logische Schließen. I». En: *Mathematische Zeitschrift* 39.1 (dic. de 1935), págs. 176-210. ISSN: 1432-1823. DOI: 10.1007/BF01201353. URL: https://doi.org/10.1007/BF01201353.