Teórica 6 : Lógica de Primer Orden

Tomás Felipe Melli

June 1, 2025

Índice

1 Introducción

Es importante destacar la diferencia esencial entre la **lógica proposicional**y la **lógica de primer orden**. En la primera, nosotros tenemos proposiciones completas que pueden resultar verdaderas o falsas y estamos sólo interesados en su valor de verdad. Lo que diferencia a esta última de la **LPO** es que en la LPO, vamos a **razonar sobre objetos individuales y sus propiedades**. En esta nueva lógica se extiende con **predicados, términos y cuantificadores**. Veamos el ejemplo :

1. **Proposicional :** $Llueve \lor \neg Llueve$
2. **LPO :** $\forall X.(EsPar(X) \implies \neg EsPar(Succ(X)))$

Vemos claramente, cómo en la **LPO** razonamos sobre los elementos y predicamos sobre ellos.
Existe un vínculo muy estrecho entre la computación y la lógica de primer orden, en particular por una pregunta que se hizo **David Hilbert** un matemático alemán allá por el siglo XX. **El problema de decisión de Hilbert**, consistía masomenos en :

> ¿Existe un método (algoritmo) general que, dado cualquier enunciado matemático formulado en lógica de primer orden, pueda decidir automáticamente si ese enunciado es verdadero o falso (o, más precisamente, si es válido o derivable)?

Este problema fue luego demostrado que **no existe**, por **Alan Turing**(al demostrar que el halting problem es indecidible) y **Alonso Church**(usando el cálculo lambda).
Sin embargo, todo esto motivó los **fundamentos de la programación lógica**. Aclaración : la progrmación lógica pertenece a los lenguajes de **progrmación declarativos**.
Con esto en mente, la **programación lógica** es un paradigma de programación donde los programas se escriben como conjuntos de hechos y reglas lógicas, y el sistema intenta satisfacer o refutar la fórmula. En caso de satisfacerla, produce una salida que verifica la propiedad P buscada.

2 Sintaxis de la Lógica de Primer Orden

Con lo anterior, vemos que la **LPO** es un sistema lógico completo que incluye varios componentes para construirse. El primero de ellos, es su **sintaxis**, para ello, se define un **Lenguaje de Primer Orden** que será todo el conjunto de reglas sintácticas de la LPO.

2.1 Lenguaje de Primer Orden

Un lenguaje de primer orden es un sistema de símbolos que permite expresar enunciados sobre un dominio (por ejemplo, números). Este lenguaje tiene reglas para construir términos (que representan objetos) y fórmulas (que expresan propiedades o relaciones entre objetos).
Tenemos :

1. Un conjunto de **símbolos de función** $\mathcal{F} = \{f, g, h, \ldots\}$ donde cada símbolo de función tiene una **aridad**(cuántos argumentos toma la función) asociada (≥ 0)
2. Un conjunto de **símbolos de predicado** $\mathcal{P} = \{P, Q, R \ldots\}$ donde cada símbolo de predicado tiene una aridad asociada (≥ 0)

Con esta sintaxis, y un **conjunto infinito numerable de variables** $\mathcal{X} = \{X, Y, Z \ldots\}$ podremos construir :

- El **conjunto de términos** que se define como

  $$t \quad ::= \quad X \quad | \quad f(t_1, \ldots, t_n)$$

  donde :

  - **X** es una variable
  - **f** es un símbolo de función de aridad n. Básicamente, esto es una función que toma n argumentos y devuelve un término (un objeto del dominio, es un constructor de términos)

Es importante destacar que los términos son las expresiones básicas del lenguaje de la lógica de primer orden, construidas a partir de variables, constantes y funciones. Estos términos representan objetos dentro del dominio y no pueden contener conectivos lógicos ni cuantificadores (esos aparecen en fórmulas, no en términos).
Veamos un ejemplo del lenguaje $\mathcal{L}_{aritmética}$ formado por:

- **Constante** 0^0 donde el superíndice nos indica la aridad del símbolo de función (las constantes son símbolos de función con aridad 0)
- Los **símbolos de función** del lenguaje son : $succ^1 \quad +^2 \quad *^2$
- Los **símbolos de predicado** son $\quad =^2 \quad <^2$

Podemos construir por ejemplo los términos

$$+(0, succ(X)) \quad *(+(X,Y)Z)$$

Dato de color : Usamos notación infija por conveniencia $0 + Succ(X) \quad (X+Y)*Z$
Ahora bien, ya podemos pasar a las **fórmulas de primer orden**. son expresiones que pueden ser verdaderas o falsas. Se construyen a partir de términos, símbolos de predicado, conectivos lógicos y cuantificadores. Vamos a extender entonces la gramática de las fórmulas de la lógica proposicional a la de primer orden (agregamos los cuantificadores y los predicados):

$$\begin{aligned} \sigma ::= & \; P(t_1, \dots, t_n) \\ & \mid \bot \\ & \mid \sigma \Rightarrow \sigma \\ & \mid \sigma \wedge \sigma \\ & \mid \sigma \vee \sigma \\ & \mid \neg\sigma \\ & \mid \forall X.\sigma \\ & \mid \exists X.\sigma \end{aligned}$$

Tenemos primero la **fórmula atómica**, donde **P es un símbolo de predicado de aridad** n (se afirma cierta propiedad sobre los n-objetos y resulta en un valor de verdad). Luego extendemos con **bottom** que representa la fórmula falsa (**contradicción**), los conectivos lógicos (implicación, conjunción, disyunción, negación) y finalmente, los **cuantificadores** : **universal y existencial** que ligan una variable **X**.
Veamos unos ejemplos de fórmulas sobre este lenguaje $\mathcal{L}_{aritmética}$

$$\begin{gathered} \forall X.\exists Y. = (+(X,Y), 0) \\ \forall X.\forall Y.\ (succ(X) = succ(Y) \Rightarrow X = Y) \\ \forall X.(X < 0 \vee X = 0 \vee 0 < X) \end{gathered}$$

Decimos que una **ocurrencia de una variable X** en una fórmula está **ligada** si est bajo el alcance de un cuantificador (existencial o universal) y **libre** si no lo está.
Decimos que si dos fórmulas sólo difieren en los nombres de sus variables ligadas, son iguales. Como $\forall X.\exists Y.P(X,Y) \equiv \forall Y.\exists X.P(Y,X)$
Vamos a notar $\sigma\{X := t\}$ como la **sustitución de las ocurrencias libres de X en σ por el término t**, evitando la captura de variables. Esto último se entenderá con el siguiente ejemplo :

$$\sigma = \forall X.P(X,Y)$$
$$\forall X.P(X,X) \texttt{ con } \sigma\ \{Y := X\}$$

El problema es que ahora X está ligada por el cuantificador universal y cambia totalmente el significado de la fórmula. La solución es **renombrar**.

3 Deducción Natural para LPO

Vamos a extender la deducción natural proposicional a primer orden. Como ya vimos, un **contexto** Γ es un conjunto finito de fórmulas y un **secuente** es de la forma $\Gamma \vdash \sigma$. Vamos a extender con las **reglas de introducción y eliminación para** $\forall$ y $\exists$

3.1 Reglas del cuantificador universal

$$\frac{\Gamma \vdash \forall X.\sigma}{\Gamma \vdash \sigma\{X := t\}}\ \forall e$$

$$\frac{\Gamma \vdash \sigma}{\Gamma \vdash \forall X.\sigma}\ \forall i,\ X \notin \text{fv}(\Gamma)$$

Por qué se exige que la variable X no pertenezca a las variables libres en Γ en la regla del cuantificador universal ?

Si podemos deducir σ con el contexto Γ y la variable X **no aparece libre** en el contexto, deducimos que $\Gamma \vdash \forall X.\sigma$. La regla nos dice que **nada en el conjunto de premisas depende de X**. La idea es que si podemos probar verdadera cierta fórmula σ sin asumir nada sobre X, entonces, podemos concluir que σ (la fórmula) lo será para todos los X. En esencia, si podemos probar σ con información sobre X, es evidente que hay restricciones y por tanto, no vale para todos los X. Veamos el siguiente ejemplo :

$$\cfrac{\cfrac{}{\text{EsPar}(N) \vdash \text{EsPar}(N)}\ \text{ax}}{\text{EsPar}(N) \vdash \forall N.\text{EsPar}(N)}\ \textbf{paso inválido}$$

Veamos un ejemplo bien hecho :

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\forall X.(P(X) \land Q(X)) \vdash \forall X.(P(X) \land Q(X))}\ \text{ax}}{\forall X.(P(X) \land Q(X)) \vdash P(\cos(X)) \land Q(\cos(X))}\ \forall e}{\forall X.(P(X) \land Q(X)) \vdash P(\cos(X))}\ \land e_1}{\forall X.(P(X) \land Q(X)) \vdash \forall X.P(\cos(X))}\ \forall i}{\vdash \forall X.(P(X) \land Q(X)) \Rightarrow \forall X.P(\cos(X))}\ \Rightarrow i$$

3.2 Reglas del cuantificador existencial

$$\frac{\Gamma \vdash \sigma\{X := t\}}{\Gamma \vdash \exists X.\sigma}\ \exists i$$

$$\frac{\Gamma \vdash \exists X.\sigma \qquad \Gamma, \sigma \vdash \tau}{\Gamma \vdash \tau}\ \exists e,\ X \notin \text{fv}(\Gamma, \tau)$$

Por qué se exige $X \notin \text{fv}(\Gamma, \tau)$

Primero miremos qué significa la regla de eliminación del existencial. Esta nos dice que *con cierto conjunto de premisas, existe un X para el cual la fórmula σ es verdadera y suponiendo una fórmula σ completamente arbitraria, podemos deducir τ y con esto concluir que* $\Gamma \vdash \tau$.
X es una variable fresca, que se usa solo como un "testigo arbitrario" de la existencia. Si X aparece libre en Γ o en τ, entonces podríamos estar deduciendo cosas que dependen de un valor específico de X, en lugar de cualquier testigo posible de la existencia.
Supongamos que X sí aparece en τ. Entonces, cuando deducimos τ a partir de σ, podríamos estar usando propiedades particulares de X, como si X = 5, y no podríamos reemplazarlo libremente por otro valor. Esto haría inválida la generalidad de la deducción.
Dicho esto, La eliminación del existencial significa: "Sé que existe un X que cumple σ. Voy a llamarlo X_0 y razonar con $\sigma[X_0]$. Si logro deducir τ sin depender de quién es X_0, entonces τ vale en general". Pero si τ depende de X_0, entonces no podés garantizar que τ vale para todos los posibles testigos. Y el paso no sería válido.
Veamos un ejemplo : Definimos: $\sigma\exists X.P(\cos(X))$

$$
\dfrac{\dfrac{\dfrac{}{\sigma \vdash \sigma}\,\text{ax} \qquad \dfrac{\dfrac{\dfrac{}{\sigma, P(\cos(X)) \vdash P(\cos(X))}\,\text{ax}}{\sigma, P(\cos(X)) \vdash P(\cos(X)) \vee Q(\cos(X))}\,\vee i_1}{\sigma, P(\cos(X)) \vdash \exists X.(P(X) \vee Q(X))}\,\exists i}{\sigma \vdash \exists X.(P(X) \vee Q(X))}\,\exists e}{\vdash \sigma \Rightarrow \exists X.(P(X) \vee Q(X))}\,\Rightarrow i
$$

4 Semántica de la LPO

4.1 Estructura de Primer Orden

Una **estructura de primer orden** es un par $\mathcal{M} = (M, I)$ donde **M es un conjunto no vacío llamada Universo**, **I es una función que le da una interpretación a cada símbolo** es decir :

$\bullet I(f) : M^n \to M \qquad \bullet\, I(P) \subseteq M^n$

Veamos un ejemplo de una estructura sobre $\mathcal{L}_{aritmética}$
Definimos $M := \mathbb{N}$ `(los elementos son números naturales)`

$$
\begin{aligned}
I(0) &= 0 \\
I(succ)(0) &= n + 1 \\
I(+)(n, m) &= n + m \\
I(*)(n, m) &= n \cdot m
\end{aligned}
$$

`donde :`

$$
\begin{aligned}
(n, m) \in I(=) &\iff n = m \\
(n, m) \in I(<) &\iff n < m
\end{aligned}
$$

Bajo esta estructura, la fórmula $\forall X. X = X + X$ es falsa.
Otro ejemplo :
Definimos la estructura: $M := \mathcal{P}(\mathbb{R})$ `(donde los elementos son subconjuntos de números reales)`

$$
\begin{aligned}
I(0) &= \emptyset \\
I(\text{succ})(A) &= \{1 + x \mid x \in A\} \\
I(+)(A, B) &= A \cup B \\
I(*)(A, B) &= A \cap B
\end{aligned}
$$

`donde :`

$$
\begin{aligned}
(A, B) \in I(=) &\iff A = B \\
(A, B) \in I(<) &\iff A \subseteq B
\end{aligned}
$$

Bajo esta estructura, la siguiente fórmula es verdadera: $\forall X.\ X = X + X$

4.2 Interpretación de Términos

Antes de definir la interpretación de términos, introducimos la **asignación**, esto es, **una función que a cada variable le asigna un elemento del universo** :

$\mathbf{a} : \mathcal{X} \to M$

Ahora bien, cada término $t \in \mathcal{T}$ **se interpreta** como un elemento $\mathbf{a}(t) \in M$, extendiendo la definición de $\mathbf{a}$ a términos :

$\mathbf{a}(f(t_1, \ldots, t_n)) = I(f)(\mathbf{a}(t_1), \ldots, \mathbf{a}(t_n))$

Esto quiere decir, $\mathbf{a}(t)$ es la evaluación del término t bajo una asignación $\mathbf{a}$ (que le da valores a las variables) y una estructura I (que le da significado a las funciones). Si el término es una función aplicada a otros términos, como $f(t_1, \ldots, t_n)$, entonces: Primero evaluamos cada sub-término: $\mathbf{a}(t_1), \ldots, \mathbf{a}(t_n)$ y l uego aplicamos la interpretación del símbolo de función f (dado por $I(f)$) a esos valores.

4.2.1 Relación de Satisfacción

Definimos una relación de **satisfacción** $\mathbf{a} \models_{\mathcal{M}} \sigma$ (la asignación de $\mathbf{a}$ (bajo la estructura $\mathcal{M}$) satisface la fórmula σ) Esto quiere decir : La fórmula σ es verdadera cuando las variables tienen los valores que indica $\mathbf{a}$, y los símbolos se interpretan según $\mathcal{M}$.)

$$
\begin{array}{ll}
\mathbf{a} \models_{\mathcal{M}} P(t_1, \ldots, t_n) & \iff (\mathbf{a}(t_1), \ldots, \mathbf{a}(t_n)) \in I(P) \\
\mathbf{a} \models_{\mathcal{M}} \sigma \wedge \tau & \iff \mathbf{a} \models_{\mathcal{M}} \sigma \text{ y } \mathbf{a} \models_{\mathcal{M}} \tau \\
\mathbf{a} \models_{\mathcal{M}} \sigma \vee \tau & \iff \mathbf{a} \models_{\mathcal{M}} \sigma \text{ o } \mathbf{a} \models_{\mathcal{M}} \tau \\
\mathbf{a} \models_{\mathcal{M}} \sigma \Rightarrow \tau & \iff \mathbf{a} \not\models_{\mathcal{M}} \sigma \text{ o } \mathbf{a} \models_{\mathcal{M}} \tau \\
\mathbf{a} \models_{\mathcal{M}} \neg\sigma & \iff \mathbf{a} \not\models_{\mathcal{M}} \sigma \\
\mathbf{a} \not\models_{\mathcal{M}} \bot & \text{(la contradicción nunca se satisface)} \\
\mathbf{a} \models_{\mathcal{M}} \forall X.\, \sigma & \iff \text{para todo } m \in M,\ \mathbf{a}[X \mapsto m] \models_{\mathcal{M}} \sigma \\
\mathbf{a} \models_{\mathcal{M}} \exists X.\, \sigma & \iff \text{existe } m \in M \text{ tal que } \mathbf{a}[X \mapsto m] \models_{\mathcal{M}} \sigma
\end{array}
$$

O sea que...

- $a \models_M P(t_1, \ldots, t_n)$: Los valores de los términos $t_1, \ldots, t_n$ bajo la asignación a, es decir, $(a(t_1), \ldots, a(t_n))$, pertenecen a la interpretación del predicado P en la estructura M. Esto significa que el predicado P es verdadero para esos valores según M.
- $a \models_M \sigma \wedge \tau$: La asignación a satisface la conjunción si **ambas** fórmulas σ y τ son verdaderas bajo a.
- $a \models_M \sigma \vee \tau$: La asignación a satisface la disyunción si **al menos una** de las fórmulas σ o τ es verdadera bajo a.
- $a \models_M \sigma \Rightarrow \tau$: La asignación a satisface la implicación si **o bien** σ no es verdadera bajo a o τ sí es verdadera bajo a.
- $a \models_M \neg\sigma$: La asignación a satisface la negación si la fórmula σ **no es verdadera** bajo a.
- $a \not\models_M \bot$: La contradicción $\bot$ **nunca** es satisfecha por ninguna asignación; es siempre falsa.
- $a \models_M \forall X.\, \sigma$: La asignación a satisface el cuantificador universal si, para **cada elemento** m en el dominio M, la fórmula σ es verdadera cuando X es asignado a m.
- $a \models_M \exists X.\, \sigma$: La asignación a satisface el cuantificador existencial si **existe al menos un elemento** m en el dominio M tal que σ es verdadera asignando X a m.

4.3 Validez y Satisfactibilidad

Decimos que una fórmula σ es :

- **Válida** si $\quad a \models_M \sigma \quad$ para toda estructura M y toda asignación a.
- **Satisfactible** si $\quad a \models_M \sigma \quad$ para alguna estructura M y alguna asignación a.
- **Inválida** si $\quad a \not\models_M \sigma \quad$ para alguna estructura M y alguna asignación a.
- **Insatisfactible** si $\quad a \not\models_M \sigma \quad$ para toda estructura M y toda asignación a.

Observaciones:

$$
\begin{array}{ll}
\sigma \text{ es Válida} & \iff \sigma \text{ no es Inválida} \\
\sigma \text{ es Satisfactible} & \iff \sigma \text{ no es Insatisfactible} \\
\sigma \text{ es Válida} & \iff \neg\sigma \text{ es Insatisfactible} \\
\sigma \text{ es Satisfactible} & \iff \neg\sigma \text{ es Inválida}
\end{array}
$$

Ejemplos :

1. $\forall X.\ X = X$ satisfactible e inválida
2. $\forall X.\ P(X) \Rightarrow \forall X.\ P(f(X))$ válida (por lo tanto satisfactible)
3. $\forall X.\ \neg P(X) \wedge \exists X.\ P(X)$ insatisfactible (por lo tanto inválida)
4. $\forall X.\ \exists Y.\ P(X,Y) \Rightarrow \exists Y.\ \forall X.\ P(X,Y)$ satisfactible e inválida
5. $\forall X.\ (P(X) \Rightarrow \sigma) \Rightarrow (\exists X.\ P(X)) \Rightarrow \sigma$ con $X \notin \mathrm{fv}(\sigma)$ válida

4.4 Modelos

- Una **sentencia** es una fórmula σ sin variables libres.
- Una **teoría de primer orden** es un conjunto de sentencias.

Una teoría T es **consistente** si $\mathcal{T} \nvdash \perp$. (no es posible derivar una contradicción $\perp$ a partir de las sentencias que contiene.)
Una estructura $\mathcal{M} = (M, I)$ es un **modelo** de una teoría $\mathcal{T}$ si $\mathcal{M} \models \sigma$ para toda fórmula $\sigma \in \mathcal{T}$. (Si todas las reglas de $\mathcal{T}$ son verdaderas en $\mathcal{M}$(cuando las interpretamos en $\mathcal{M}$), decimos que $\mathcal{M}$ es un modelo de $\mathcal{T}$)
(La asignación es irrelevante pues σ es cerrada (su verdad depende sólo de la estructura y no de la asignación)).

4.5 Corrección y Completitud

(Gödel, 1929) Dada una teoría T, son equivalentes:

1. T es consistente.
2. T tiene (al menos) un modelo.

Dada una fórmula σ, son equivalentes:

1. $\vdash \sigma$ es derivable.
2. σ es válida.

Dada una fórmula σ, son equivalentes:

1. $\vdash \neg\sigma$ es derivable.
2. σ es insatisfactible.

4.6 Problema de la Decisión

¿Existe un algoritmo que, dada cualquier fórmula σ en lógica de primer orden, pueda decidir automáticamente si σ es válida o no?

```
entrada : una fórmula σ
 salida : un booleano que indica si σ es válida
```

No hay un procedimiento automático general que pueda responder siempre correctamente a esta pregunta para todas las fórmulas σ.

5 Unificación de Términos

5.1 Algoritmo de Unificación

La forma de adaptar el algoritmo de unificación que conocemos a la lógica de primer orden es *cambiando la notación*

$$\{X \overset{?}{=} X\} \cup E \xrightarrow{\texttt{Delete}} E$$

$$\{f(t_1, \ldots, t_n) \overset{?}{=} f(s_1, \ldots, s_n)\} \cup E \xrightarrow{\texttt{Decompose}} \{t_1 \overset{?}{=} s_1, \ldots, t_n \overset{?}{=} s_n\} \cup E$$

$$\{t \overset{?}{=} X\} \cup E \xrightarrow{\texttt{Swap}} \{X \overset{?}{=} t\} \cup E \quad \text{si } t \text{ no es una variable}$$

$$\{X \overset{?}{=} t\} \cup E \xrightarrow{\texttt{Elim}}_{\{X:=t\}} E\{X := t\} \quad \text{si } X \notin \text{fv}(t)$$

$$\{f(t_1, \ldots, t_n) \overset{?}{=} g(s_1, \ldots, s_m)\} \cup E \xrightarrow{\texttt{Clash}} \textbf{falla} \quad \text{si } f \neq g$$

$$\{X \overset{?}{=} t\} \cup E \xrightarrow{\texttt{Occurs-Check}} \textbf{falla} \quad \text{si } X \neq t \text{ y } X \in \text{fv}(t)$$

5.2 Terminación del Algoritmo

Dado un conjunto de ecuaciones de unificación E, definimos:

- n_1: cantidad de variables distintas en E.
- n_2: tamaño de E, calculado como: $n_2 = \sum_{(t=s)\in E}(|t| + |s|)$
- n_3: cantidad de ecuaciones de la forma $t = X$ en E.

Podemos observar que las reglas que no producen falla achican la tripla (n_1, n_2, n_3), de acuerdo con el orden lexicográfico:

Regla	n_1	n_2	n_3
Elim	$>$		
Decompose	$=$	$>$	
Delete	$\geq$	$>$	
Swap	$=$	$=$	$>$

5.3 Corrección del Algoritmo

1. Una **sustitución** es una función S que le asocia un término $S(X)$ a cada variable X.
2. S es un **unificador** de E si, para cada ecuación $(t \overset{?}{=} s) \in E$, se cumple que: $S(t) = S(s)$
3. S es **más general** que S' si existe una sustitución T tal que:

   $$S' = T \circ S$$

4. S es un **m.g.u.** de E si:
   - S es un unificador de E
   - Para todo unificador S' de E, se cumple que S es más general que S'.

   Técnicamente, nos interesan los **m.g.u. idempotentes**, es decir, aquellos para los cuales: $S(S(t)) = S(t) \quad \text{para todo término } t$

Lemas :

- **Lema — corrección de la regla Delete:**

  $$S \text{ m.g.u. de } E \implies S \text{ m.g.u. de } \{X \overset{?}{=} X\} \cup E$$

- **Lema — corrección de la regla Swap:**

  $$S \text{ m.g.u. de } \{t \overset{?}{=} s\} \cup E \implies S \text{ m.g.u. de } \{s \overset{?}{=} t\} \cup E$$

- **Lema — corrección de la regla Decompose:**

  $$S \text{ m.g.u. de } \{t_1 \overset{?}{=} s_1, \ldots, t_n \overset{?}{=} s_n\} \cup E \implies S \text{ m.g.u. de } \{f(t_1, \ldots, t_n) \overset{?}{=} f(s_1, \ldots, s_n)\} \cup E$$

- **Lema — corrección de la regla Elim:**
  Si S es m.g.u. de $E\{X := t\}$ y $X \notin \text{fv}(t)$

  $$\implies S \circ \{X := t\} \text{ es m.g.u. de } E$$

  Usar el hecho de que si $S(X) = t$, entonces $S(s\{X := t\}) = S(s)$.

Supongamos una secuencia de pasos del algoritmo:

$$E_0 \xrightarrow{S_1} E_1 \xrightarrow{S_2} \ldots \xrightarrow{S_n} E_n = \emptyset$$

Queremos demostrar que la composición $S_n \circ \cdots \circ S_1$ es un m.g.u. de E.

Demostración por inducción en n

- **Caso base:** $n = 0$

  Entonces $E_0 = \emptyset$, y la sustitución identidad es un m.g.u. de $\emptyset$.

- **Paso inductivo:** Supongamos que para $n > 0$, se tiene:

  $$E_0 \xrightarrow{S_1} E_1 \quad E_1 \xrightarrow{S_2} \dots \xrightarrow{S_n} E_n = \emptyset$$

  Por hipótesis inductiva (HI), la composición $S_n \circ \cdots \circ S_2$ es un m.g.u. de E_1.
  Aplicando uno de los lemas de corrección correspondientes (según la regla usada para ir de E_0 a E_1), se concluye que:

  $S_n \circ \cdots \circ S_2 \circ S_1$ es un m.g.u. de E_0.