



UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática

Tesis de Licenciatura

Nociones de equivalencia y bisimulación

María Mercedes Lorenzón

Director: Abriola Sergio Alejandro

Fecha de Presentación:

a mis padres.

Agradecimientos

Me gustaría empezar esta tesis agradeciendo a todas aquellas personas que me ayudaron y acompañaron durante esta hermosa etapa.

A mi mamá que siempre me acompañó, confió en mí y me alentó a seguir.

A Lucas, por estar a mi lado siempre, por escucharme, aconsejarme y ayudarme en cada paso que doy.

A Melina y Soledad que siempre están para compartir alegrías y tristezas, logros y derrotas.

A Luz, mi querida compañera, amiga y confidente, el mejor de los regalos que me dio la carrera.

A María, Victoria, Francisco, Luis y Loli, mis hermanos, que siempre me alentaron y apoyaron.

A Marina, Cecilia y Paula que desde la distancia se hacen presentes con su apoyo y aliento.

A mi abuela Negra y mi tía Fernanda, que me prendieron mil velas y rezaron por mí.

A Magui y Anita que me escucharon y aconsejaron cada mañana.

A toda mi familia, de sangre y política que siempre me desean lo mejor.

A Manu, Carlo, y los Santis que compartimos muchas tardes de estudio y risas.

A Sergio Abriola, mi director, por su paciencia, por sus consejos y por su tiempo.

A todo el jurado por tomarse el tiempo de leer esta tesis.

A la UBA, de la cual estoy orgullosa de haber sido alumna, y a todos los docentes que fueron parte de mi formación académica.

Índice general

1. Lógica de Primer orden	9
1.1. Introducción a la lógica de primer orden	9
1.1.1. Nociones básicas de lógica de primer orden	9
1.1.2. Valuación	10
1.2. Equivalencia de modelos	12
1.2.1. Isomorfismos de modelos	12
1.2.2. Isomorfismos Parciales y Potenciales.	14
2. Lógica Modal	17
2.1. Introducción a la lógica modal	17
2.1.1. Correspondencia de modelos	18
2.2. Bisimulaciones	22
2.2.1. Noción de Morfismo en lógica modal básica	22
2.2.2. Uniones y submodelos generados	29
2.2.3. Modelos de árboles	31
2.2.4. Modelos finitos y n -Bisimulación	33
3. Lógica XPath₌	37
3.1. Introducción a la lógica XPath ₌	37
3.1.1. Correspondencia de modelos	40
3.2. XPath ₌ [↓]	41
3.2.1. Bisimulación y l -Bisimulación	42
3.2.2. Equivalencia y bisimulación	44
3.2.3. Saturación	48
3.3. XPath ₌ [↓]	50
3.4. Bisimulación y (r,s,k) -Bisimulación	56
3.4.1. Equivalencia y bisimulación	59
3.4.2. Saturación	62
3.5. Simulación	63
Conclusión	67
Apéndice	70

Introducción

La idea general de esta tesis es poder entender y ver cómo cambia la noción de equivalencia, simulación y bisimulación de los modelos de distintas lógicas tales como la lógica de primer orden, la lógica modal, y la lógica con igualdad $XPath_=_$.

Recordaremos nociones básicas de cada lógica para que el lector tenga más herramientas.

Como concepto central de este trabajo se ubica la bisimulación. La bisimulación es una noción fundamental que se establece cuando dos nodos (o estados) en un grafo etiquetado no pueden ser distinguidos por un observador externo.

Esta herramienta fue descubierta independientemente en las áreas de la informática y de la lógica filosófica en la década de los '70. En ambos contextos la bisimulación apareció como un refinamiento de la noción de morfismo, es decir, de asignaciones que preservan la estructura.

En el campo de la informática, fue desarrollada por Milner en 1971 (y refinada por Park en 1981) en el contexto de la teoría de la concurrencia como una forma de determinar comportamientos de los programas. En el caso de la filosofía, fue introducida por Van Benthem en 1976 para caracterizar el poder expresivo de la lógica modal básica en términos de un fragmento de la lógica de primer orden.

Este trabajo se divide en 3 grandes capítulos dedicados a la lógica de primer orden ([2, 12, 4]), la lógica modal ([5, 6, 13]) y la lógica $XPath_=_$ ([1, 3, 9, 10, 17]), respectivamente. Además hay un apéndice que hace un acercamiento a la teoría de juegos, usando los juegos de Ehrenfeucht - Fraïssé.([7, 14])

A lo largo de la tesis vamos a ir descubriendo que las lógicas modal y $XPath_=_$ pueden ser entendidas como un fragmento propio de la lógica de primer orden, con lo cuál el orden en su estudio es importante.

Pasaremos por varias nociones de comparación de modelos hasta llegar a las Bisimulaciones, que como ya mencionamos anteriormente, es el eje de este trabajo, y es la teoría que mejor interpreta la idea de equivalencia lógica.

Capítulo 1

Lógica de Primer orden

1.1. Introducción a la lógica de primer orden

La lógica de primer orden, también llamada lógica de predicados, es un sistema formal diseñado para estudiar la inferencia de los lenguajes de primer orden. Los lenguajes de primer orden son, a su vez, lenguajes formales con cuantificadores que alcanzan sólo a variables de individuos y con predicados y funciones cuyos argumentos son sólo constantes o variables de individuo.

Con la lógica de primer orden podemos representar situaciones que contienen relaciones entre objetos (como las relaciones «padre» e «hijo»). Ahora bien, a veces la situación también expresa relaciones entre relaciones, o propiedades (relaciones de grado 1) de relaciones. Por ejemplo, «padre es una relación familiar». A la lógica de primer orden sólo se le permite representar relaciones entre objetos, pero esto ya es suficiente para darle una enorme expresividad.

1.1.1. Nociones básicas de lógica de primer orden

Lenguajes de Primer Orden

- **Símbolos lógicos y auxiliares:** $x, \iota, \forall, \neg, \rightarrow, ()$
 $x, x', x'', x''' \dots$ símbolos de variables. Llamamos VAR al conjunto de variables
- **Símbolos de cada lenguaje:** Un lenguaje es $\mathcal{L} = C \cup P \cup F$ donde
 C es un conjunto de símbolos constantes (puede ser vacío)
 F es un conjunto de símbolos de funciones (puede ser vacío)
 P es un conjunto de símbolos de predicados ($P \neq \emptyset$)
- **Términos:** Para un lenguaje \mathcal{L} fijo definimos los términos de \mathcal{L} (notamos $TERM(\mathcal{L})$) al conjunto de todos los términos del lenguaje \mathcal{L} .

- 1 Todas las variables son términos.
 - 2 Todo símbolo constante de \mathcal{L} es un término.
 - 3 Si f es un símbolo de función n -adico de \mathcal{L} y t_1, \dots, t_n son términos de \mathcal{L} entonces $f(t_1, \dots, t_n)$ es un término de \mathcal{L}
 - 4 Nada más es un término.
- **Fórmulas:** Para un lenguaje fijo \mathcal{L} , definimos las fórmulas de \mathcal{L} (notamos $FORM(\mathcal{L})$) al conjunto de toda las fórmulas del lenguaje \mathcal{L} .
 - 1 Si P es un símbolo de predicado n -ádico de \mathcal{L} y t_1, \dots, t_n son términos de \mathcal{L} , entonces $P(t_1, \dots, t_n)$ es una fórmula de \mathcal{L} (la llamamos *atómica*)
 - 2 Si φ es una fórmula de \mathcal{L} entonces $\neg\varphi$ es una fórmula de \mathcal{L}
 - 3 Si φ y ψ son fórmulas de \mathcal{L} entonces $(\varphi \rightarrow \psi)$ es una fórmula de \mathcal{L} .
 - 4 Si φ es una fórmula de \mathcal{L} y x una variable, entonces $(\forall x)\varphi$ es una fórmula de \mathcal{L} .
 - 5 Nada más es una fórmula de \mathcal{L}

Así, una \mathcal{L} -estructura, \mathcal{A} , de un lenguaje $\mathcal{L} = C \cup P \cup F$, es un conjunto A no vacío (que llamaremos universo o dominio) y las asignaciones siguientes:

- Para cada símbolo constante $c \in C$, un elemento fijo $c_A \in A$.
- Para cada símbolo de función n -aria $f \in F$, una función $f_A : A^n \rightarrow A$.
- Para cada símbolo de predicado n -ario $p \in P$, una relación n -aria p_A en A .

Notación: $\mathcal{A} = \langle A, c_A, \dots, f_A, \dots, p_A \dots \rangle$

Podemos pensar como un caso particular la lógica de primer orden con igualdad, que es aquella que se extiende con un símbolo binario $=$ que siempre se interpreta como la relación de igualdad usual. También podemos considerar agregar otros símbolos de relación tales como \wedge , \vee , \exists , y \leftrightarrow sin que estos aumenten el poder expresivo de la lógica de primer orden.

1.1.2. Valuación

Definición 1 Dada una \mathcal{L} -estructura \mathcal{A} , con dominio A . Una **valuación** para \mathcal{A} es una función $V: Var \rightarrow A$, que luego es extendida a $\widehat{V}: term(\mathcal{L}) \rightarrow A$ de la siguiente manera:

- Si $t=c$ es una constante, entonces $\widehat{V}(t)=c_A$

- Si $t=x$ es una variable, entonces $\widehat{V}(t)=V(t)=V(x)$
- Si $t=f(t_1, \dots, t_n)$, entonces $\widehat{V}(t)=f_A(\widehat{V}(t_1), \dots, \widehat{V}(t_n))$.

Además sea V una valuación de \mathcal{A} . Definimos $V(x = a)$ como

$$V(x = a)(y) = \begin{cases} V(y) & \text{si } x \neq y \\ a & \text{si } x = y \end{cases}$$

Decimos que φ es **verdadera** en \mathcal{A} bajo la valuación V (escribimos $\mathcal{A} \models \varphi[V]$) si:

- φ es de la forma $P(t_1, \dots, t_n)$

$$\mathcal{A} \models P(t_1, \dots, t_n)[V] \Leftrightarrow \widehat{V}(t_1, \dots, t_n) \in P_A$$

- φ es de la forma $\neg\psi$

$$\mathcal{A} \models \neg\psi[V] \Leftrightarrow \text{no } \mathcal{A} \models \psi[V], \text{ o sea } \mathcal{A} \not\models \psi[V]$$

- φ es de la forma $(\psi \rightarrow \rho)$

$$\mathcal{A} \models (\psi \rightarrow \rho)[V] \Leftrightarrow \text{no } \mathcal{A} \models \psi[V] \text{ ó } \mathcal{A} \models \rho[V]$$

o sea $\mathcal{A} \not\models \psi[V]$ ó $\mathcal{A} \models \rho[V]$

- φ es de la forma $(\forall x)\psi$

$$\mathcal{A} \models \rho[V] \Leftrightarrow \text{para cualquier } a \in A, \mathcal{A} \models \psi[V(x = a)]$$

Entonces podemos decir que hay 3 niveles de verdad:

- φ es **satisfacible** si existe una \mathcal{L} -estructura \mathcal{A} y una valuación V de \mathcal{A} tal que $\mathcal{A} \models \varphi[V]$
- φ es **verdadera** (o válida) en una \mathcal{L} -estructura, \mathcal{A} (notamos $\mathcal{A} \models \varphi$) si $\mathcal{A} \models \varphi[V]$ para toda valuación v de \mathcal{A} . Decimos que \mathcal{A} es un modelo de φ .
- φ es **universalmente válida** (notamos $\models \varphi$) si $\mathcal{A} \models \varphi[V]$ para toda \mathcal{L} -estructura \mathcal{A} y toda valuación V de \mathcal{A} .

Teorema 1 (Teorema de Compacidad) Sea $\mathcal{T} \subseteq \text{Form}(\mathcal{L})$. Entonces \mathcal{T} es satisfacible si y sólo si todo subconjunto finito de \mathcal{T} es satisfacible

No daremos la demostración en esta tesis. Si el lector esta interesado puede encontrarla en [16].

El siguiente resultado nos habla del tamaño del modelo que satisface al conjunto de fórmulas:

Teorema 2 (Teorema de Löwenheim - Skolem:) *Si \mathcal{T} es un conjunto satisfacible de fórmulas de primer orden, entonces \mathcal{T} es satisfacible en algún modelo finito o numerable.*

Para la demostración ver [2]

1.2. Equivalencia de modelos

1.2.1. Isomorfismos de modelos

Definición 2 Sean \mathcal{A} y \mathcal{B} dos modelos de lógicas de Primer orden sobre el mismo lenguaje \mathcal{L} .

Decimos que $i: A \rightarrow B$ es un **isomorfismo** entre \mathcal{A} y \mathcal{B} si:

- i es biyectiva
- $(x_1, \dots, x_n) \in P_A$ si y sólo si $(i(x_1), \dots, i(x_n)) \in P_B$
- $i(f_A(x_1, \dots, x_n)) = f_B(i(x_1), \dots, i(x_n))$ para cada función n -aria $f \in F$ y elementos $x_1, \dots, x_n \in A$
- $i(c_A) = c_B$ para cada constante $c \in A$

Y en ese caso decimos que \mathcal{A} y \mathcal{B} son isomorfos y lo denotamos como $\mathcal{A} \cong \mathcal{B}$

Teorema 3 Si i es un isomorfismo entre \mathcal{A} y \mathcal{B} entonces para toda V valuación tenemos:

$$\mathcal{A}, V \models \varphi \text{ si y sólo si } \mathcal{B}, i \circ V \models \varphi$$

Demostración

Para la demostración del teorema vamos a necesitar un lema previo:

Lema 4 Si V es una valuación para \mathcal{A} y i es un isomorfismos entre \mathcal{A} y \mathcal{B} entonces $\widehat{i \circ V} = i \circ \widehat{V}$

Demostración

Por inducción en los \mathcal{L} -términos.

- Para cada constante $c \in \mathcal{L}$ tenemos $\widehat{i \circ V}(c) = c_B = i(c_A) = \widehat{i \circ c} = (i \circ \widehat{V})(c)$
- Para cada variable x : $\widehat{i \circ V}(x) = (i \circ V)(x) = i(V(x)) = i(\widehat{V})(x) = (i \circ \widehat{V})(x)$
- Para cada función n-aria $f \in \mathcal{L}$: Si $\widehat{i \circ V}(t_j) = (i \circ \widehat{V})(t_j)$ para todo $j \in [1, n]$, entonces

$$\begin{aligned} \widehat{i \circ V}(f(t_1, \dots, t_n)) &= f_B(\widehat{i \circ V}(t_1), \dots, \widehat{i \circ V}(t_n)) = f_B((i \circ \widehat{V})(t_1), \dots, (i \circ \widehat{V})(t_n)) = \\ &= f_B(i \circ (\widehat{V}(t_1)), \dots, i(\widehat{V}(t_n))) = i(f_A(\widehat{V}(t_1), \dots, \widehat{V}(t_n))) = i(\widehat{V}(f(t_1, \dots, t_n))) = \\ &= (i \circ \widehat{V})(f(t_1, \dots, t_n)). \end{aligned}$$

Esto demuestra el lema. \square

Ahora vamos a demostrar el teorema por inducción en la estructura de φ .

- Si $\varphi = R(t_1, \dots, t_n)$, entonces:

$$\begin{aligned} (\mathcal{A}, V) \models R(t_1, \dots, t_n) &\text{ si y sólo si} \\ (\widehat{V}(t_1), \dots, \widehat{V}(t_n)) &\in R_A \text{ si y sólo si} \\ (i(\widehat{V}(t_1)), \dots, i(\widehat{V}(t_n))) &\in R_B \text{ si y sólo si} \\ ((i \circ \widehat{V})(t_1), \dots, (i \circ \widehat{V})(t_n)) &\in R_B \text{ si y sólo si} \\ (\widehat{i \circ V}(t_1), \dots, \widehat{i \circ V}(t_n)) &\in R_B \text{ si y sólo si} \\ (B, i \circ V) \models R(t_1, \dots, t_n). & \end{aligned}$$

Finalmente suponemos que la propiedad se cumple para φ y ψ

- Si $\varphi = \neg\psi$, entonces:

$$\begin{aligned} (\mathcal{A}, V) \models \varphi &\text{ si y sólo si} \\ (\mathcal{A}, V) \not\models \psi &\text{ si y sólo si} \\ (\mathcal{B}, i \circ V) \not\models \psi &\text{ si y sólo si} \\ (\mathcal{B}, i \circ V) \models \varphi. & \end{aligned}$$

- Si $\varphi = \psi \rightarrow \theta$, entonces:

$$\begin{aligned} (\mathcal{A}, V) \models \varphi &\text{ si y sólo si} \\ (\mathcal{A}, V) \not\models \psi \text{ o } (\mathcal{A}, V) \models \theta &\text{ si y sólo si} \\ (\mathcal{A}, V) \models \neg\psi \text{ o } (\mathcal{A}, V) \models \theta &\text{ si y sólo si} \\ (\mathcal{B}, i \circ V) \models \neg\psi \text{ o } (\mathcal{B}, i \circ V) \models \theta &\text{ si y sólo si} \\ (\mathcal{B}, i \circ V) \not\models \psi \text{ o } (\mathcal{B}, i \circ V) \models \theta &\text{ si y sólo si} \\ (\mathcal{B}, i \circ V) \models \varphi & \end{aligned}$$

Supongamos que $\varphi = (\forall x)\psi$.

Sólo vamos a demostrar el teorema en una dirección. La otra dirección se demuestra de la misma forma pero considerando i^{-1} en lugar de i .

Si $(\mathcal{A}, V) \models \varphi$. Para cualquier $a \in A$ vale que $(\mathcal{A}, V[x/a]) \models \psi$.

Por hipótesis inductiva para cualquier $a \in A$ vale que $(\mathcal{B}, i \circ V[x/a]) \models \psi$.

Pero $i \circ V[x/a] = (i \circ V)[x/i(a)]$. Tenemos que para cualquier $b \in B$ vale que $(\mathcal{B}, (i \circ V)[x/b]) \models \psi$. Por lo tanto $(\mathcal{B}, i \circ V) \models \varphi$.

Esto termina la demostración del teorema. ■

Estamos acostumbrados a que la existencia de un isomorfismo nos dé una noción de equivalencia, pero tenemos que tener cuidado, ya que el teorema nos da información en una sola dirección, es decir, si hay un isomorfismo entonces... Esto nos dice que si hay un isomorfismo entre dos modelos de lógica de primer orden, entonces no puedo distinguirlas. Pero ¿puede la lógica de primer orden distinguir cuando dos modelos son isomorfos o no? Veamos el siguiente ejemplo:

Sea M un modelo cuyo dominio es \mathbb{R} y definamos

$$\mathcal{T} = \{\varphi \mid M \models \varphi\}$$

- 1- Por definición, $M \models \mathcal{T}$, con lo cual, \mathcal{T} es satisfacible
- 2- Por Löwenheim-Skolem, para algún N finito o numerable, $N \models \mathcal{T}$ (establece que si una teoría de primer orden es consistente, entonces tiene al menos un modelo con dominio finito o numerable).
- 3- M y N tienen cardinales distintos. Por lo tanto no pueden ser isomorfos.

¿Podemos distinguir en primer orden a M y N ? $M \models \varphi \Rightarrow \varphi \in \mathcal{T} \Rightarrow N \models \varphi$ Por otro lado $M \not\models \varphi \Rightarrow M \models \neg\varphi \Rightarrow \neg\varphi \in \mathcal{T} \Rightarrow N \models \neg\varphi \Rightarrow N \not\models \varphi$ Es decir, $M \models \varphi \Leftrightarrow N \models \varphi$. . . ¡no podemos distinguírlas!

Es ahora cuando nos preguntamos, ¿Habría una noción de equivalencia para primer orden más aproximada que la de isomorfismos?

1.2.2. Isomorfismos Parciales y Potenciales.

Definición 3 *Dados dos modelos de lógicas de primer orden \mathcal{A} y \mathcal{B} con dominios A y B respectivamente. Decimos que $p : A' \rightarrow B' (A' \subseteq A, B' \subseteq B)$ es un **isomorfismo parcial** entre \mathcal{A} y \mathcal{B} si p es un isomorfismo entre $\mathcal{A} \upharpoonright A'$ y $\mathcal{B} \upharpoonright B'$*

Definición 4 *Un **isomorfismo potencial** entre dos modelos \mathcal{A} y \mathcal{B} con dominios A y B respectivamente, es una colección \mathcal{F} de isomorfismos parciales finitos entre A y B que satisfacen:*

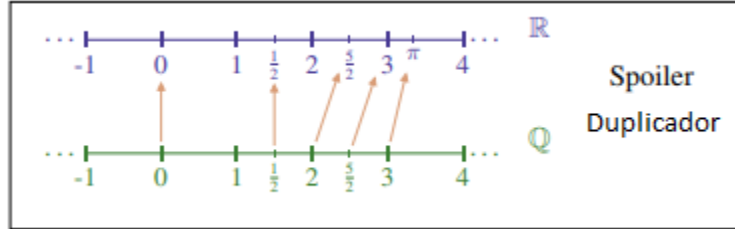


Figura 1.1: Spoiler elige un elemento y Duplicador responde con un isomorfismo parcial.

- (zig) Si $p \in \mathcal{F}$ y $x \in A$, existe $y \in B$ tal que $p \cup \{x \mapsto y\} \in \mathcal{F}$
- (zag) Si $p \in \mathcal{F}$ e $y \in B$, existe $x \in A$ tal que $p \cup \{x \mapsto y\} \in \mathcal{F}$

Observemos que si i es un isomorfismo entre A y B me puedo construir un isomorfismo potencial tomando las restricciones de i como familia de isomorfismos parciales. Teniendo en cuenta esto, queda claro que pedir que exista un isomorfismo potencial es una condición más relajada que pedir la existencia de un isomorfismo, ya que pueden ser familias de isomorfismos parciales cualesquiera.

Ejemplo 5 Veamos el siguiente ejemplo de isomorfismo potencial entre $\langle \mathbb{R}, < \rangle$ y $\langle \mathbb{Q}, < \rangle$.

Lo pensamos como un juego entre dos jugadores, Spoiler y Duplicador.

Spoiler elige un número en \mathbb{R} o en \mathbb{Q} lo más difícil que pueda. y Duplicador le responde extendiendo un isomorfismo parcial.

En la figura 5 puede ver mejor como se procede.

Como \mathbb{Q} es denso en \mathbb{R} y solo estamos teniendo en cuenta la relación de $<$, Duplicador siempre va a poder encontrar un isomorfismo parcial para responderle a Spoiler.

Así cada estrategia ganadora para Duplicador va a ser un isomorfismo potencial entre $\langle \mathbb{R}, < \rangle$ y $\langle \mathbb{Q}, < \rangle$.

Ver Apéndice para más formalidad de este juego, llamado de Ehrenfeucht - Fraïssé[7].

Ahora vamos a ver algunos resultados con esta nueva noción de isomorfismo entre modelos.

¿Es suficiente para garantizar equivalencia entre los modelos?.

Proposición 6 Dos modelos numerables potencialmente isomorfos son isomorfos.

Demostración

Solo daremos la idea de esta demostración. Tomar un isomorfismo potencial \mathcal{F} entre \mathcal{A} y \mathcal{B} y elegir un $p \in \mathcal{F}$. Por definición, p se puede extender tantas veces como uno quiera y, en el límite, esto nos da un isomorfismo. ■

Teorema 7 *Si existe un isomorfismo potencial \mathcal{F} entre \mathcal{A} y \mathcal{B} , entonces, para toda fórmula de primer orden φ*

$$\mathcal{A} \models \varphi \Leftrightarrow \mathcal{B} \models \varphi$$

Para la demostración ver [2].

Veamos qué nos dice este teorema. Lo que nos está garantizando es que si tenemos un isomorfismo potencial entre \mathcal{A} y \mathcal{B} entonces todas las sentencias que son ciertas en \mathcal{A} lo serán en \mathcal{B} y viceversa. Pero, ¿valdrá la vuelta?. Es decir, si no puedo distinguir con sentencias de primer orden \mathcal{A} de \mathcal{B} , eso me garantiza que existe un isomorfismo potencial entre ellos? Veamos el siguiente ejemplo:

Consideremos $\mathcal{S} = \langle \langle, 0, 1, 2, \dots \rangle$ y $\mathcal{S}' = \langle \langle, c, 0, 1, 2, \dots \rangle$ y tomemos a \mathbb{N} como modelo sobre la signatura \mathcal{S} Definamos $\mathcal{T}_{\mathbb{N}} = \{\varphi \mid \mathbb{N} \models \varphi\}$ y $\Sigma = \{0 < c, 1 < c, 2 < c, \dots\}$. Por compacidad, $\mathcal{T}_{\mathbb{N}} \cup \Sigma$ es satisfacible y por Löwenheim-Skolem, tiene algún modelo numerable, \mathcal{M} . Ahora, sea \mathcal{M}_0 la restricción de \mathcal{M} a la signatura \mathcal{S} Observar que $\mathbb{N} \models \varphi \Leftrightarrow \mathcal{M}_0 \models \varphi$ (si $\mathbb{N} \models \varphi \Rightarrow \varphi \in \mathcal{T}_{\mathbb{N}} \Rightarrow \mathcal{M}_0 \models \varphi$ pues $\mathcal{M}_0 \models \mathcal{T}_{\mathbb{N}}$, si $\mathbb{N} \not\models \varphi \Rightarrow \mathbb{N} \models \neg\varphi \Rightarrow \neg\varphi \in \mathcal{T}_{\mathbb{N}} \Rightarrow \mathcal{M}_0 \models \neg\varphi \Rightarrow \mathcal{M}_0 \not\models \varphi$) ¡Pero \mathbb{N} y \mathcal{M}_0 no pueden ser isomorfos! (porque no es posible asignarle un elemento a $c^{\mathcal{M}}$) Y por ser numerables, tampoco son potencialmente isomorfos (ya que si lo fueran, por lo visto anteriormente, resultarían isomorfos).

Por lo tanto tampoco logramos tener una noción de equivalencia tradicional.

Esto nos lleva a seguir estudiando los isomorfismo y obtener el siguiente resultado:

Teorema 8 (Teorema de isomorfismo de Keisler-Shelah) *\mathcal{M} y \mathcal{N} coinciden en toda sentencia de primer orden si y sólo si existe un isomorfismo entre sus ultrapotencias.*

Para demostrar y entender este teorema necesitamos varias herramientas más que exceden el nivel de este trabajo.

Para profundizar en este tema ver [4] y [12].

Así tenemos que los isomorfismos nos dan una noción de equivalencia para primer orden que no es la natural ya que hay modelos no isomorfos que primer orden no puede distinguir. Por otro lado la noción de isomorfismo potencial generaliza la idea de isomorfismos para estructuras de distintos cardinales, pero aún así hay modelos no potencialmente isomorfos que primer orden tampoco los puede distinguir. Por último llegamos a isomorfismos de las ultrapotencias que son los que capturan adecuadamente la noción de equivalencia de primer orden.

Capítulo 2

Lógica Modal

2.1. Introducción a la lógica modal

La lógica modal puede pensarse como una extensión de la lógica clásica para darle un carácter puramente intencional, es decir, para hablar sobre posibilidad y necesidad. Por ejemplo, consideremos las siguientes oraciones: "*Todas las personas son mortales*" y "*Pedro es mortal*". Estas oraciones denotan objetos por extensión, ya sea enumerando elemento por elemento ó incluyendo a todos los elementos de algún dominio determinado (en este caso las oraciones denotan conjuntos de personas). También, observemos que de estas oraciones sólo podemos afirmar si son verdad o no. En contraste, consideremos las oraciones: "*Pienso que es correcto*", "*Creo que él sabe la respuesta*", "*Esto debería ser así*" y "*Es improbable que suceda eso*".

Aquí se denotan objetos por intención, esto es, describiendo una propiedad que deben cumplir (por ejemplo, el conjunto de todas las cosas que yo creo que son verdad). En estos casos, las palabras *pienso*, *creo*, *debería* e *improbable* califican la veracidad de la oración. A este tipo de modificadores los llamamos *modalidades*.

Vamos a estudiar primero el lenguaje de la lógica monomodalproposicional, ó simplemente lógica modal.

La lógica modal es la lógica de proposiciones extendida con dos operadores: $\langle - \rangle$ denotando la existencia de alguno y $[-]$ para todos. En el caso de la lógica monomodal al existir una sola reacción binaria lo notamos \diamond y \square .

Sintaxis formal del lenguaje modal básico:

- 1- La signatura del lenguaje va a consistir de dos conjuntos infinitos numerables disjuntos entre sí
 - $Prop = \{p_1, p_2, p_3, \dots\}$ el conjunto de variables proposicionales
 - R un símbolo de relación binaria.

2- El conjunto $Form$ en la signatura $\langle Prop, R \rangle$ está definido como:

$$Form = p \mid \neg\varphi \mid \varphi \wedge \psi \mid \diamond\varphi$$

en donde $p \in Prop$ y $\varphi, \psi \in Form$

3- El operador \Box se define como $\Box\varphi = \neg\diamond\neg\varphi$

De igual forma que $\forall x\varphi$ es $\neg\exists x\neg\varphi$

Definición 5 *Un modelo de Kripke es una estructura $\langle W, R, V \rangle$ donde*

- W es un conjunto no vacío de elementos, a veces llamados mundos.
- R es la relación binaria de W
- $V : Prop \rightarrow \mathcal{P}(W)$ es una función de valuación ($V(p)$ es el conjunto de mundos donde vale p)

Semántica del lenguaje modal Básico:

Dado un modelo $\mathcal{M} = \langle W, R, V \rangle$ y un estado $w \in W$, la relación de satisfacibilidad esta determinada por lo siguiente

- $\mathcal{M}, w \models p$ si y sólo si $w \in V(p)$, para $p \in Prop$
- $\mathcal{M}, w \models \neg\varphi$ si y sólo si $\mathcal{M}, w \not\models \varphi$
- $\mathcal{M}, w \models \varphi \wedge \psi$ si y sólo si $\mathcal{M}, w \models \varphi$ y $\mathcal{M}, w \models \psi$
- $\mathcal{M}, w \models \diamond\varphi$ si y sólo si $\exists w' \in W/R(w, w')$ y $\mathcal{M}, w' \models \varphi$
- Vamos a decir que φ es válida en un modelo \mathcal{M} si y sólo si en todos los estados $w \in W$ vale $\mathcal{M}, w \models \varphi$, y en ese caso escribimos $\mathcal{M} \models \varphi$.

2.1.1. Correspondencia de modelos

En esta sección vamos a ver al modelo de Kripke como un modelo de primer orden, ya que para cada $p \in Prop$ del modelo de Kripke tenemos $V(p) \subseteq W$ (es decir, V codifica una relación unaria por cada proposición p)

Formalmente, un modelo de Kripke $\mathcal{M} = \langle W, R, V \rangle$ definido sobre una signatura $\mathcal{S} = \langle Prop, R \rangle$ se corresponde con un modelo de primer orden $\mathcal{I}^{\mathcal{M}} = \langle W, \cdot^{\mathcal{I}^{\mathcal{M}}} \rangle$ cuyo vocabulario es $\mathcal{P}^{\mathcal{S}} = \{P_i/i \in Prop \cup R\}$ donde

$$\mathcal{P}_i^{\mathcal{I}^{\mathcal{M}}} = \begin{cases} V(i) & \text{si } i \in Prop \\ R & \text{si no} \end{cases}$$

A \mathcal{P}^S se lo llama lenguaje de correspondencia de primer orden.

Así con esta correspondencia entre modelos, la lógica modal básica y la de primer orden operan sobre los mismos objetos semánticos si las miramos en el lenguaje de correspondencia. Entonces podemos ir más allá y pensar que las fórmulas de una lógica también tienen su correspondiente en la otra lógica. Es decir, podemos pensar la correspondencia como una traducción de fórmulas. Qué tal si vemos si podemos reproducir las operaciones de la lógica modal básica en la lógica de primer orden. Se sabe que hay una traducción estándar a primer orden, ST , tal que $ST_x(\varphi)$ le hace corresponder, a cada φ , una fórmula de primer orden con exactamente una variable libre, x . Recordemos las condiciones de semántica de la lógica modal:

- $\mathcal{M}, w \models p$ si y sólo si $w \in V(p)$, para $p \in Prop$
- $\mathcal{M}, w \models \neg\varphi$ si y sólo si $\mathcal{M}, w \not\models \varphi$
- $\mathcal{M}, w \models \varphi \wedge \psi$ si y sólo si $\mathcal{M}, w \models \varphi$ y $\mathcal{M}, w \models \psi$
- $\mathcal{M}, w \models \diamond\varphi$ si y sólo si $\exists w' \in W \mid R(w, w')$ y $\mathcal{M}, w' \models \varphi$

Como queremos que la fórmula y la traducción sean equivalentes, la traducción estándar a primer orden deberá cumplir:

$$\begin{aligned} ST_x(p) &\equiv Pp(x) \\ ST_x(\neg\varphi) &\equiv \neg ST_x(\varphi) \\ ST_x(\varphi \wedge \psi) &\equiv ST_x(\varphi) \wedge ST_x(\psi) \\ ST_x(\langle R \rangle \varphi) &\equiv \exists y(R(x, y) \wedge ST_y(\varphi)) \end{aligned}$$

Con esto se podría probar el siguiente resultado:

Teorema 9 *Para toda fórmula φ de la lógica modal básica, todo modelo de Kripke \mathcal{M} , todo w en el dominio de \mathcal{M} y toda valuación V ,*

$$\mathcal{M}, w \models \varphi \text{ si y sólo si } \mathcal{M}, V[x \mapsto w] \models ST_x(\varphi)$$

La traducción estándar es una herramienta que nos permite hacer una analogía fácil de muchos resultados de la lógica de primer orden, como por ejemplo, el Teorema de compacidad y el de Löwenheim-Skolem, entre otros.

Teorema 10 (Teorema de compacidad) *Sea \mathcal{T} un conjunto de fórmulas de la lógica modal básica. Si todo subconjunto finito de \mathcal{T} es satisfacible entonces, \mathcal{T} lo es.*

Demostración

Definamos, para todo conjunto de fórmulas modales Δ ,

$$ST_x = \{ST_x(\varphi) / \varphi \in \Delta\}$$

Sea \mathcal{T}_0 un subconjunto finito de \mathcal{T} ; sabemos que $ST_x(\mathcal{T}_0)$ va a ser satisfacible si y sólo si \mathcal{T}_0 lo es.

Entonces, si todo \mathcal{T}_0 es satisfacible, todo $ST_x(\mathcal{T}_0)$ lo es, y por compacidad en primer orden, también lo es $ST_x(\mathcal{T})$. Pero entonces \mathcal{T} tiene que ser satisfacible en lógica modal. ■

Teorema 11 (Teorema de Löwenheim-Skolem:) *Si \mathcal{T} es un conjunto satisfacible de formulas de la lógica modal básica, entonces \mathcal{T} es satisfacible de algún modelo de Kripke numerable (finito o infinito).*

Demostración

Esta demostración es análoga al caso de compacidad.

Si \mathcal{T} es satisfacible, $ST_x(\mathcal{T})$ también lo es. Entonces por Löwenheim-Skolem para primer orden, existe un modelo de Kripke \mathcal{M} numerable y una valuación V , tal que $\mathcal{M}, V \models ST_x(\mathcal{T})$.

Luego $\mathcal{M}, V(x) \models \mathcal{T}$. ■

Como dijimos anteriormente en esta sección vemos a la lógica modal básica como un fragmento *propio* de la lógica de primer orden, esto es porque la traducción estándar que encontramos no tiene una ‘inversa’, o sea no hay una traducción de la lógica modal básica a la lógica de primer orden. Esto es principalmente por el hecho de que realmente son distintas, por ejemplo la lógica de primer orden es indecidible mientras que la modal básica es decidible. Por lo cual no pueden ser lógicas equivalentes, y ya vimos que la lógica modal está incluida en la lógica de primer orden.

Definición 6 *Una lógica se dice **decidible** si el problema de determinar la satisfacibilidad de cada una de sus fórmulas lo es. Es decir si existe un algoritmo tal que para cada fórmula de la lógica, es capaz de decidir en un número finito de pasos si la fórmula tiene o no un modelo que la satisface.*

Lo bueno es que podemos usar más resultados de primer orden para trabajar con la lógica modal básica. Por ejemplo, por transferencia sabemos que *si \mathcal{M} y \mathcal{N} son potencialmente isomorfos, entonces para cada w existe un w' tal que*

$$\mathcal{M}, w \models \varphi \text{ si y sólo si } \mathcal{N}, w' \models \varphi$$

Si queremos hablar de una noción de equivalencia de modelos modales tenemos que tener en cuenta que las nociones de equivalencias en primer orden vistas anteriormente consideran los modelos en su totalidad, ya que la extensión de una sentencia de primer orden es todo el dominio o el conjunto vacío.

Un lenguaje modal (y, de hecho, cualquier lenguaje lógico cuyas fórmulas forman un conjunto) puede distinguir entre algunos modelos (\mathcal{M}, s) y (\mathcal{N}, t) , pero no entre todos esos pares. Por ejemplo, nuestro lenguaje modal básico puede distinguir el par de modelos que se muestra en la Figura 2.1

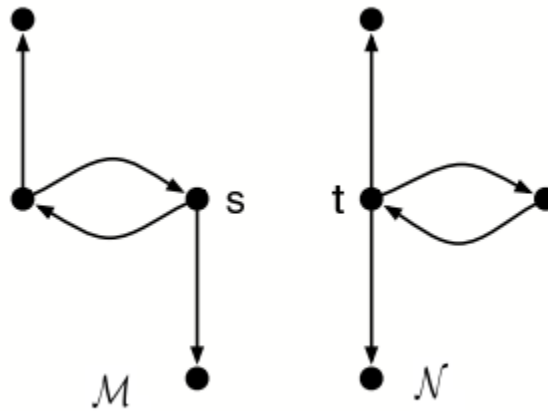


Figura 2.1: (\mathcal{M}, s) y (\mathcal{N}, t) son modalmente distinguibles

A continuación, $\Box(\Box \perp \vee \Diamond \Box \perp)$ es una fórmula modal que distingue estos modelos de Kripke: es verdadera en \mathcal{M} en s , pero falsa en \mathcal{N} en t .

Pero ahora considere el par de modelos de Kripke que se muestra en la Figura 2.2

¿Es posible distinguir (\mathcal{M}, s) modalmente de (\mathcal{K}, u) ? Es decir, ¿es posible encontrar una fórmula modal (básica) que sea verdadera en \mathcal{M} , en s , pero falsa en \mathcal{K} en u ? Tengamos en cuenta que es fácil distinguirlos si se nos permite usar lógica de primer orden: todos los puntos en \mathcal{M} (incluyendo s) son irreflexivos, mientras que el punto u en \mathcal{K} es reflexivo, por lo tanto, la fórmula de primer orden R_{xx} no es satisfasible (bajo cualquier asignación de variable) en el modelo de Kripke \mathcal{M} , pero se cumple en \mathcal{K} cuando u se asigna a x . Pero no importa lo ingenioso que sea, no encontraré ninguna fórmula en el lenguaje modal básico que distinga a estos modelos de Kripke en sus puntos designados. ¿Por qué pasa esto?

Un enfoque natural de esta pregunta es considerar su dualidad: ¿cuándo deben considerarse dos modelos de Kripke como modalmente idénticos? Por ejemplo, dada una interpretación del proceso, ¿cuándo veríamos dos diagramas de transición como

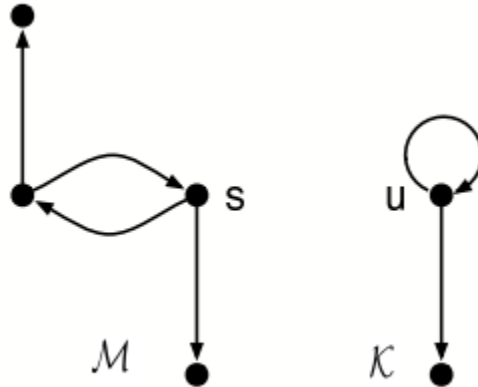


Figura 2.2: (\mathcal{M}, s) y (\mathcal{K}, u) no son modalmente distinguibles

representaciones del mismo proceso? Veamos los modelos de Kripke \mathcal{M} y \mathcal{K} de la Figura 2.2 que proporciona un ejemplo intuitivo: parece que representan el mismo proceso cuando observamos posibles acciones y puntos muertos (tenga en cuenta que en cada estado el proceso puede entrar en una situación de punto muerto), es decir, puede entrar en un estado desde el cual no puede salir. Por el contrario, \mathcal{M} y \mathcal{N} en la Figura 2.1 son diferentes, ya que no todos los estados en \mathcal{N} están amenazados con un bloqueo inmediato.

2.2. Bisimulaciones

2.2.1. Noción de Morfismo en lógica modal básica

En Matemática la noción de morfismo o de función que preserva estructuras es de fundamental importancia. Pero ¿qué noción de morfismo es apropiada para lógica modal? Es decir, ¿qué tipo de morfismo nos da una noción de invariancia? En esta sección vamos a ir introduciendo distintos conceptos que nos lleven a la respuesta. Vamos a hablar de homomorfismos, de homomorfismos fuertes, de embeddings y de isomorfismos, para finalmente llegar a morfismos acotados que es lo que más nos interesa.

Definición 7 *Un homomorfismo entre dos modelos de Kripke $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$ es una función $f : W \rightarrow W'$ tal que*

- I) *Para todo $w \in W$ y todo símbolo de proposición p , $w \in V(p)$ implica $f(w) \in V'(p)$*

II) Para todo $w, v \in W$, si Rwv entonces $R'f(w)f(v)$

Observemos que el ítem (II) nos dice que los homomorfismos entre modelos de Kripke solo preservan las relaciones en un solo sentido. es decir que **si** se relacionan en W **entonces** sus imágenes se relacionan en W' pero no sabemos nada de la recíproca. Esto es, si sus imagenes se relacionan, ¿podemos decir que sus elementos se relacionaban en el dominio?

Esto nos dá una clara idea de por qué las formulas modales no son invariantes bajo homomorfismos.

Pero entonces ¿qué pasa si definimos un homomorfismo más fuerte modificando la condición (II) para poder salvar esta debilidad? Esto nos lleva a la siguiente definición:

Definición 8 Un **homomorfismo fuerte** entre dos modelos de Kripke $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$ es una función $f : W \rightarrow W'$ tal que

I Para todo $w \in W$ y todo símbolo de proposición $p, w \in V(p)$ si y sólo si $f(w) \in V'(p)$

II Para todo $w, v \in W$, Rwv si y sólo si $R'f(w)f(v)$

Además si el homomorfismo fuerte es inyectivo decimos que es un **embedding**. Y si es biyectivo decimos que es un **Isomorfismo**. En este caso \mathcal{M} y \mathcal{N} se dicen isomorfos y lo notamos $\mathcal{M} \cong \mathcal{N}$.

Hay que hacer una observación sobre esta definición. Claramente un isomorfismo habla de una equivalencia ó de una noción de matemáticamente idénticos. Pero esta nueva definición no nos ayuda demasiado en lógica modal. Pues en las condiciones (I) y (II) se pide invarianza pero hay muchos morfismos que dan lugar a la invarianza, pero que no califican como homomorfismos fuertes. Para garantizar la invarianza modal, debemos asegurarnos de que alguna estructura de la imagen se refleje de nuevo en el dominio, pero los morfismos fuertes lo hacen de una manera demasiado severa. En el ejemplo 12 se puede ver como la función preserva invarianza pero no es un homomorfismo fuerte. Esto nos lleva a pensar en condiciones de invarianza no tan estrictas.

Definición 9 Sean $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$ modelos de Kripke sobre la misma signatura. La función $f : W \rightarrow W'$ es un **p-morfismo o morfismo acotado** (ó *bounded morphism*) si se cumple:

(Atom) $w \in V(p)$ si y sólo si $f(w) \in V'(p)$

(Zig) Si Rwv , entonces $R'f(w)f(v)$

(Zag) Si $R'f(w)v'$, entonces existe v tal que $f(v) = v'$ y Rwv

Cuando hay un morfismo acotado suryectivo entre los modelos de Kripke \mathcal{M} y \mathcal{N} lo notamos $\mathcal{M} \rightarrow \mathcal{N}$.

La idea incorporada en la condición (Zag) es absolutamente fundamental para la lógica modal: de hecho, es la idea que subyace a la noción de bisimulación que definiremos más adelante, por lo que debemos comprender de manera inmediata lo que implica. Aquí hay un ejemplo útil:

Ejemplo 12 Consideremos los modelos de Kripke $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$ donde $W = \mathbb{N}$, Rmn si y sólo si $n = m + 1$, y $V(p) = \{n \in \mathbb{N} \mid n \text{ es par} \}$ $W' = \{e, o\}$, $R' = \{(e, o), (o, e)\}$ y $V'(p) = \{e\}$. Ahora, sea $f : W \rightarrow W'$ la siguiente función:

$$f(n) = \begin{cases} e & \text{si } n \text{ es par} \\ o & \text{si } n \text{ es impar} \end{cases}$$

Se ve esto en un simple grafico en la Figura 2.3

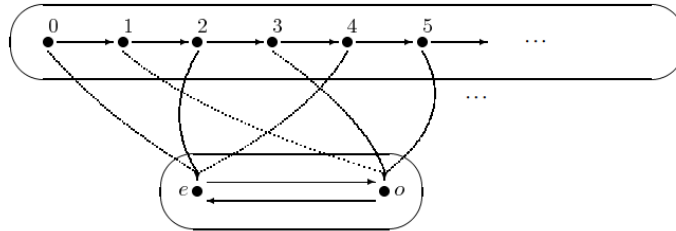


Figura 2.3: Morfismo acotado

Observemos que f no es un homomorfismo fuerte. Pues, por ejemplo, $f(1)$ y $f(4)$ se relaciona en \mathcal{N} pero 1 y 4 no lo hacen en \mathcal{M} .

Veamos ahora que si es un morfismo acotado que resulta suryectivo.

Es claro que f satisface (Atom) por definición.

Para ver (zig) consideremos un par arbitrario $(n, n + 1)$ en R . Hay dos posibilidades: n es par ó es impar. Supongamos que n es par, así $n + 1$ es impar. Y tenemos que $f(n) = e$ y $f(n + 1) = o$. o sea $R'f(n)f(n + 1)$ como queríamos. El argumento si n es impar es análogo.

Veamos la condición (Zag). Tomemos un elemento arbitrario n de W y asumamos que $R'f(n)w'$. Tenemos que encontrar un $m \in W$ tal que Rnm y $f(m) = w'$. Asumamos que n es impar (el caso con n par es similar). Como n es impar tenemos que $f(n) = o$, así, por definición de R' , tiene que ser que $w' = e$. Pero entonces $f(n + 1) = w'$ ya que $n + 1$ es par. Por lo tanto, $n + 1$ es el m que estamos buscando.

Y así concluimos que f es un morfismo acotado.

Como mencionamos anteriormente la noción de morfismo acotado surge de la idea de bisimulación, que es simplemente una relación (no necesariamente una función) entre dos modelos de Kripke en los que los estados relacionados tienen información atómica idéntica y posibilidad de transición coincidentes.

La parte interesante de la definición es la forma en que se hace precisa la noción de "posibilidad de transición coincidentes".

Definición 10 Una **bisimulación** entre dos modelos de Kripke $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$ es una relación no vacía $\mathcal{Z} \subseteq W \times W'$ tal que si $w\mathcal{Z}w'$ entonces:

(Atom) $\forall p \in VAR, w \in V(p)$ si y sólo si $w' \in V'(p)$

(Zig) Si Rw, v entonces existe v' tal que Rw', v' y $v\mathcal{Z}v'$

(Zag) Si $R'w', v'$ entonces existe v tal que Rw, v y $v\mathcal{Z}v'$

Si existe una bisimulación entre dos modelos de Kripke \mathcal{M} y \mathcal{N} , y $(w, w') \in \mathcal{Z}$ decimos que los modelos punteados \mathcal{M}, w y \mathcal{N}, w' son **bisimilares**, y lo notamos $\mathcal{M}, w \Leftrightarrow \mathcal{N}, w'$

Más aun, decimos que dos estados son bisimilares si están relacionados bajo alguna bisimulación. Dicho en palabras: dos estados son bisimilares si hacen que la misma información atómica sea verdadera y si, además, sus posibilidades de transición coinciden. Es decir, si una transición a un estado relacionado es posible en un modelo, entonces la bisimulación debe ofrecer una posibilidad de transición coincidente en el otro. En efecto, las bisimulaciones son una generalización relacional de los morfismos acotados, eliminando la direccionalidad desde el dominio al codominio (y con ello la condición homomórfica) y reemplazandola con un sistema de movimientos de combinación entre modelos. Por eso es claro que si $\mathcal{M} \cong \mathcal{N}$ entonces $\mathcal{M} \Leftrightarrow \mathcal{N}$

Volviendo a los modelos de Kripke \mathcal{M}, \mathcal{K} y \mathcal{N} considerados anteriormente (y sin tener en cuenta los símbolos de la proposición) es fácil ver que \mathcal{M} y \mathcal{K} son bisimilares: las líneas de puntos en la Figura 2.4 indican la bisimulación requerida (tengamos en cuenta que la bisimulación indicada une los dos puntos designados). Además, es fácil ver que no hay bisimulación que vincule los puntos de \mathcal{N} y \mathcal{K} . ¿Por qué no? Debido a que un movimiento de t en \mathcal{N} no tiene un movimiento coincidente en \mathcal{K} : moverse hacia abajo desde u no es una opción (los puntos finales nunca son bisimilares con los puntos que tienen sucesores) pero ninguno se está moviendo de manera reflexiva de u a sí mismo (se puede pasar de u a un sucesor, que es un punto final, pero esto no se puede hacer en \mathcal{N}). Dado cualquier modelo modal \mathcal{M} , las bisimulaciones se pueden usar de varias maneras. La llamada *contracción por bisimulación* hace que \mathcal{M} sea lo más pequeña posible. Para definir esto, tengamos en cuenta que de la definición de bisimulación se desprende que cualquier unión de bisimulaciones entre dos modelos de Kripke es en sí misma una bisimulación. Por lo tanto, la unión de todas las bisimulaciones entre dos modelos de Kripke es una bisimulación *máxima* entre ellos. Ahora formamos la

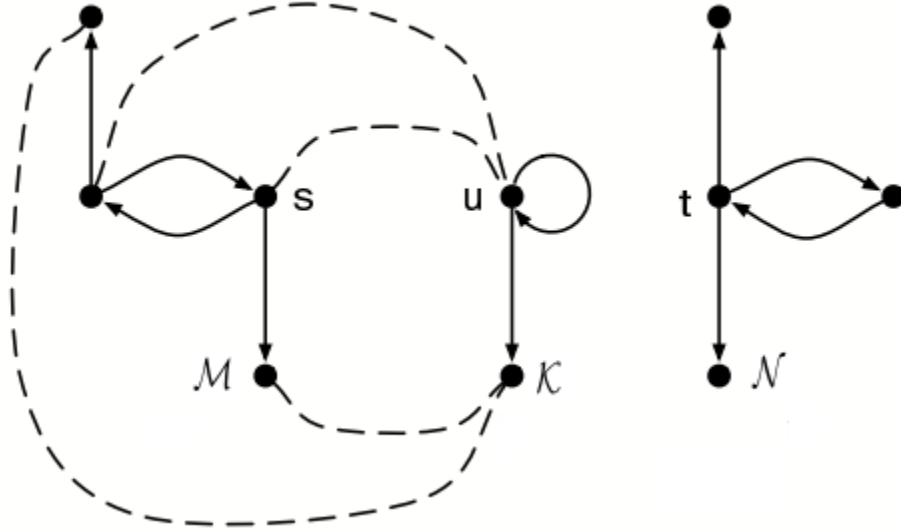


Figura 2.4: (\mathcal{M}, s) y (\mathcal{K}, u) son bisimilares, pero (\mathcal{K}, u) y (\mathcal{N}, t) no lo son.

bisimulación máxima del modelo de Kripke \mathcal{M} consigo misma (una bisimulación de un modelo consigo misma se denomina *autobisimulación*). Esta define un cociente de \mathcal{M} cuyos puntos son las clases de equivalencia, y relacionan la clase de equivalencia $|w|$ con la clase de equivalencia $|v|$ si y sólo si $|w|$ y $|v|$ contienen puntos w y v tales que Rwv . El mapeo de puntos con sus clases de equivalencia es una bisimulación. Por ejemplo, la bisimulación que se muestra en la Figura 2.4 entre \mathcal{M} y \mathcal{K} es una contracción de bisimulación. Las contracciones por bisimulación son la representación más compacta de los procesos, al menos desde un punto de vista modal. Eliminan todas las redundancias en la representación, pero también todas las simetrías estáticas (es como decir que una mariposa es un objeto redundante, ya que un ala contiene suficiente información bajo esta perspectiva).

Teorema 13 (Teorema de invarianza bajo bisimulaciones) *Si (\mathcal{M}, w) y (\mathcal{N}, w') son bisimilares, entonces son modalmente equivalentes ($\mathcal{M} \equiv \mathcal{N}$), es decir que para toda fórmula φ de la lógica modal básica se tiene*

$$\mathcal{M}, w \models \varphi \text{ si y sólo si } \mathcal{N}, w' \models \varphi$$

Demostración

Haremos la demostración por inducción en la estructura de φ . Supongamos que wZw' entonces:

- Caso Base: si φ es p . Como \mathcal{M}, w y \mathcal{N}, w' son bisimilares, por (atom) vale que $w \in V(p)$ si y sólo si $w' \in V'(p)$ para todo p , en particular para φ . Entonces $\mathcal{M}, w \models \varphi$ si y sólo si $\mathcal{N}, w' \models \varphi$
- Supongamos que vale para ψ y se quiere ver que vale para $\varphi = \neg\psi$. Si $\mathcal{M}, w \models \neg\psi$ si y sólo si $\mathcal{M}, w \not\models \psi$ así por hipótesis inductiva $\mathcal{N}, w' \not\models \psi$ si y sólo si $\mathcal{N}, w' \models \varphi$
- Supongo que vale para φ y ψ quiero ver que vale para $\varphi \wedge \psi$. $\mathcal{M}, w \models \varphi \wedge \psi$ si y sólo si $\mathcal{M}, w \models \varphi$ y $\mathcal{M}, w \models \psi$ Esto implica que $\mathcal{N}, w' \models \varphi$ y $\mathcal{N}, w' \models \psi$ o sea que $\mathcal{N}, w' \models \varphi \wedge \psi$. La otra implicación es idéntica.
- $\varphi = \Diamond\psi$: Si $\mathcal{M}, w \models \Diamond\varphi$, entonces existe v tal que Rwv y $\mathcal{M}, v \models \varphi$ luego, por (Zig), existe v' tal que $Rw'v'$ Y vZv' y por hipótesis inductiva, $\mathcal{N}, v' \models \varphi$, con lo cual $\mathcal{N}, w' \models \Diamond\varphi$. La otra implicación es idéntica. El caso $\mathcal{N}, w' \models \Diamond\varphi$ es similar usando (Zag).

■

Observemos que la recíproca del Teorema anterior no siempre es verdadera. Un ejemplo de eso son los modelos de Kripke \mathcal{M} y \mathcal{N} que se muestran en la Figura 2.5. Es claro que son equivalentes como modelos punteados sobre w y v . Veamos por qué no son bisimilares. Esto se debe a que la condición (Zag) no se cumple. Supongamos que son bisimilares, o sea que (Zag) vale. Como wZv tomo $y_1 \in W'$ tal que Rv, y_1 y y_1 está en la cadena infinita de \mathcal{N} , tiene que existir \hat{y}_1 en \mathcal{M} (que está en alguna de las cadenas finitas) tal que $\hat{y}_1 Z y_1$. Sea y_2 el siguiente nodo de la misma cadena infinita de \mathcal{N} tal que Ry_1, y_2 . Entonces tiene que existir \hat{y}_2 en \mathcal{M} tal que $\hat{y}_2 Z y_2$, y así sucesivamente. Pero los y_i son infinitos y los \hat{y}_i no. Por lo que en algún momento no voy a poder encontrar el \hat{y}_j que necesito. Luego no puede ser que valga (Zag) y por lo tanto no son bisimilares.

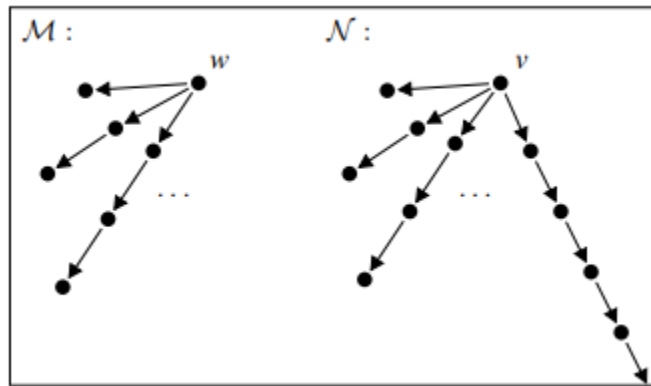


Figura 2.5: Modelos de Kripke modalmente equivalentes pero no bisimilares.

Podemos considerar algunos casos especiales donde vale la reciproca del teorema anterior:

Teorema 14 (Teorema de Hennessy - Milner) Sean \mathcal{M} y \mathcal{N} modelos de Kripke cuyas relaciones tienen imagen finita, esto es, que para cada x existen finitos y tales que xRy . Entonces (\mathcal{M}, w) y (\mathcal{N}, v) son bisimilares si y sólo si son modalmente equivalentes ($\mathcal{M} \equiv \mathcal{N}$) Es decir que en ambas valen las mismas fórmulas modales.

Demostración Resta ver la implicación hacia la derecha ya que la implicación hacia la izquierda se desprende de inmediato del teorema anterior. Basta ver que $\mathcal{Z} := \{(w, v) \mid \text{para todo } \varphi, \mathcal{M}, w \models \varphi \Leftrightarrow \mathcal{N}, v \models \varphi\}$ es una bisimulación. Para ello, supongamos que $w\mathcal{Z}v$, tenemos que ver que se cumplen las condiciones (Atom), (zig) y (Zag):

- (Atom) $w\mathcal{Z}v$ implica $\mathcal{M}, w \models p$ si y sólo si $\mathcal{N}, v \models p$ para todo p
- (Zig) Lo haremos razonando por el absurdo:
Sea Rww' tal que ningún v' cumple que Rvv' y $w'\mathcal{Z}v'$
Definamos $\mathcal{S} := \{v' \mid vRv'\}$, que como son finitos, $\mathcal{S} = \{v_1, \dots, v_k\}$ ($k > 0$).
Entonces existen $\{\varphi_1, \dots, \varphi_k\}$ con $\mathcal{M}, w' \models \varphi_i$ y $\mathcal{N}, v_i \not\models \varphi_i$. Entonces
 $\mathcal{M}, w \models \diamond(\varphi_1 \wedge \dots \wedge \varphi_k)$ y $\mathcal{N}, v \not\models \diamond(\varphi_1 \wedge \dots \wedge \varphi_k)$
Lo que es absurdo y por lo tanto demostramos lo que queríamos.
- (Zag) es análogo a (Zig)

■

Ya sabemos que la unión de dos o más bisimulaciones también resulta una bisimulación, y por lo tanto puedo pensar en una bisimulación maximal. Pasa lo mismo con la composición de bisimulaciones, si esa no resulta vacía. Más precisamente resulta el siguiente resultado:

Proposición 15 Sean $\mathcal{Z}_1 \subseteq \mathcal{M}_1 \times \mathcal{M}_2$ y $\mathcal{Z}_2 \subseteq \mathcal{M}_2 \times \mathcal{M}_3$ dos bisimulaciones. Si $\mathcal{Z}_1 \circ \mathcal{Z}_2$ es no vacío, entonces es una bisimulación.

Demostración

Sea $\mathcal{Z}_3 := \mathcal{Z}_1 \circ \mathcal{Z}_2$ y supongamos $w\mathcal{Z}_3w'$ (con lo cual, existe v tal que $w\mathcal{Z}_1v$ y $v\mathcal{Z}_2w'$)
Veamos que vale (Atom): si $w \in V_3(p)$ entonces como $v\mathcal{Z}_2w'$ y \mathcal{Z}_2 es una bisimulación, por (Atom) de \mathcal{Z}_2 tenemos que $v \in V_2(p)$ y como $w\mathcal{Z}_1v$ y \mathcal{Z}_1 es bisimulación tenemos que $w \in V_1(p)$

Probaremos que cumple (Zig) y (Zag) es análogo.

Supongamos que existe w_2 tal que $R^{\mathcal{M}_1}ww_2$. Por (Zig) de \mathcal{Z}_1 existe v_2 tal que $R^{\mathcal{M}_2}vv_2$ y $w_2\mathcal{Z}_1v_2$. Ahora, por (Zig) de \mathcal{Z}_2 tiene que existir w'_2 tal que $R^{\mathcal{M}_3}w'_2w_2$ y $v_2\mathcal{Z}_2w'_2$. Pero entonces $w_2\mathcal{Z}_3w'_2$.



Formalicemos la definición de contracción de bisimulaciones. Como dijimos antes las bisimulaciones no relacionan necesariamente todo el modelo, pueden relacionar una parte de cada modelo. Por lo que tiene sentido pensar en una bisimulación de un modelo en sí mismo. A lo que anteriormente llamamos autobisimulación, que, al igual que en las bisimulaciones tradicionales podemos pensar en un elemento maximal. Con esta idea surge la noción de contracción por bisimulación.

Definición 11 Sea $\mathcal{M} = \langle W, R, V \rangle$ y sea $\mathcal{Z}_{\mathcal{M}}$ su máxima autobisimulación. La **contracción de \mathcal{M} por bisimulación** es el modelo de Kripke $\langle W', R', V' \rangle$ donde

- $W' := W/\mathcal{Z}_{\mathcal{M}}$ (el cociente)
- $R' := \{(|w|, |v|) \text{ tal que } (w, v) \in R\}$
- $V' := \{|w| \text{ tal que } w \in V(p)\}$

Algunos resultados de contracción por bisimulaciones:

- **Proposición 16** Si $\mathcal{M} = \langle W, R, V_i \rangle$ es un modelo de Kripke y \mathcal{M}_0 es su contracción por bisimulación, entonces \mathcal{M}, w y $\mathcal{M}_0, |w|$ son bisimilares.
- **Proposición 17** La contracción por bisimulación de \mathcal{M} es un modelo de Kripke bisimilar a \mathcal{M} de cardinalidad mínima (lo notamos $\min(\mathcal{M})$)

Las demostraciones de estas propiedades se pueden ver en [2]

2.2.2. Uniones y submodelos generados

Si tenemos un par de modelos de Kripke disjuntos (es decir, un par de modelos de Kripke (W, R, V) y (W', R', V') tales que W y W' son disjuntos), entonces su unión disjunta es el modelo de Kripke $(W \cup W', R \cup R', V + V')$, donde $V + V'$ es la valuación definida por $(V + V')(p) = V(p) \cup V'(p)$, para todos los símbolos de proposición p . Es decir, formar una unión disjunta de dos modelos de Kripke significa agrupar toda la información en los dos grafos. Pero, ¿Qué pasa si los grafos no son disjuntos? Luego, simplemente tomamos copias isomorfas disjuntas de los dos modelos de Kripke y formamos la unión disjunta de las copias. Este proceso de agrupamiento se puede generalizar a muchos modelos arbitrarios, lo que nos lleva a la siguiente definición.

Definición 12 Dada una colección no vacía de modelos de Kripke $\mathcal{M}^k = \langle W^k, R^k, V^k \rangle$ todos sobre la misma signatura \mathcal{S} y donde $W^i \cap W^j = \emptyset$ para todo $i \neq j$ la **unión disjunta** $\biguplus \mathcal{M}^k = \langle W, R, V \rangle$ es el modelo de Kripke de \mathcal{S} dado por

- $W := \bigcup_k W^k$

- $R := \bigcup_k R^k$
- $V(p) := \bigcup_k V^k$

Proposición 18 *Sea \mathcal{C} una colección de modelos de Kripke disjuntos y sea \mathcal{M} un modelo de Kripke en \mathcal{C} . Para todo w en el dominio de \mathcal{M}*

$$\mathcal{M}, w \Leftrightarrow \biguplus \mathcal{C}, w$$

Demostración

No la haremos con detalle, pero sale suponiendo que W es el dominio de \mathcal{M} , y viendo que $\{(x, x) \mid x \in W\}$ es una bisimulación. ■

Corolario 19 *Satisfacción de fórmulas de la lógica modal básica es invariante respecto de uniones disjuntas.*

Las uniones disjuntas construyen modelos de Kripke más grandes a partir de (colecciones de) los más pequeños. Los submodelos generados hacen lo contrario. Surgen al restringir la atención a subgrafos de un grafo dado que se cierran en transiciones relacionales. Por ejemplo, considere los dos grafos de la Figura 2.6. Es claro que el de la derecha surge al restringir la atención a un cierto subgrafo de transición cerrada del grafo de la izquierda, es decir, el conjunto de puntos alcanzables tomando las ecuaciones de las transiciones de \mathcal{S} . Esto motiva la siguiente definición.

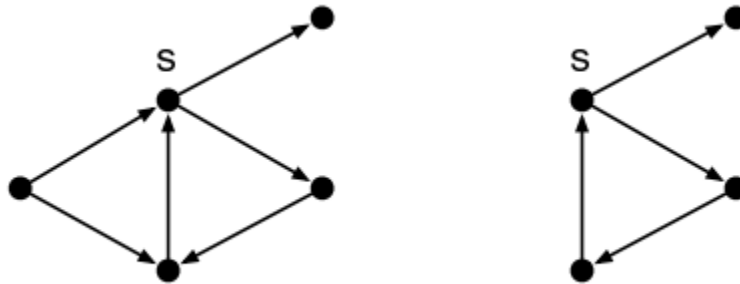


Figura 2.6: Submodelo generado de \mathcal{S}

Definición 13 *Dado $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$ decimos que \mathcal{N} es un submodelo de Kripke de \mathcal{M} si*

- 1) $W' \subseteq W$
- 2) $R' = R \cap (W' \times W')$
- 3) $V'(p) = V(p) \cap (W' \times W')$

Definición 14 Dado $\mathcal{M} = \langle W, R, V \rangle$ y $\mathcal{N} = \langle W', R', V' \rangle$
Decimos que \mathcal{N} es un submodelos de Kripke generado de \mathcal{M} si

- \mathcal{N} es un submodelo de Kripke de \mathcal{M}
- Para todo $w \in W'$, si existe $v \in W$ tal que Rwv , entonces $v \in W'$

Proposición 20 Sea \mathcal{N} un modelo de Kripke subgenerado de \mathcal{M} . Para todo w en el dominio de \mathcal{N} tenemos

$$\mathcal{M}, w \Leftrightarrow \mathcal{N}, w$$

Un submodelo de Kripke generado es bisimilar al modelo de Kripke que lo originó, al igual que con las uniones disjuntas, la relación de identidad relaciona los dos modelos de la manera apropiada. Vale la pena aclarar que cada modelo de Kripke de componente de una unión disjunta es un submodelo generado de esa unión.

2.2.3. Modelos de árboles

Muchas veces es más fácil para la interpretación y el análisis de una situación pensar en un modelo de Kripke que lo describa que se represente en forma de árbol. Pero, a priori, no todos los modelos de Kripke tiene forma de árbol, por suerte hay una propiedad muy útil que usaremos en este capítulo que nos garantiza poder pensar sin pérdida de generalidad en modelos en forma de árbol.

Definición 15 Un árbol es un grafo no dirigido, acíclico y conexo, con un nodo distintiguido llamado la raíz desde el cual puede alcanzarse cualquier otro nodo.

Es decir un árbol es un grafo que comienza con una raíz, desde la cual cualquier otro punto del grafo es alcanzado y se extiende en varias ramificaciones o líneas, cada una de las cuales puede extenderse en ramificaciones hasta terminar, finalmente en una hoja.

Proposición 21 (Propiedad de modelo de árbol) Para cualquier modelo de Kripke punteado \mathcal{M}, w existe un modelo de Kripke \mathcal{T} con forma de árbol tal que $\mathcal{T} \twoheadrightarrow \mathcal{M}$.

Es decir hay un modelo \mathcal{T} y un morfismo acotado suryectivo entre ellos.

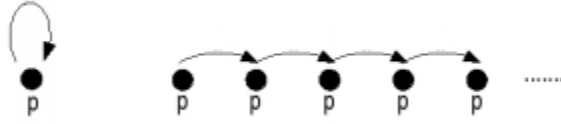


Figura 2.7: Traspaso de modelos puntados a modelos en forma de árbol.

Como consecuencia de ello, tenemos que cualquier fórmula satisfacible en \mathcal{M}, w lo será en \mathcal{T}, w .

Veamos dos ejemplos sencillos de cómo se puede construir un modelo de Kripke en forma de árbol. Ver Figura 2.7

La idea intuitiva es construir un modelo de Kripke en donde:

- Los elementos del modelo de Kripke sean secuencias finitas de sucesores desde la raíz. O sea tenemos que identificar una raíz y que cualquier otro elemento del modelo de Kripke sea alcanzado desde allí en finitos pasos.
- Las secuencias van a estar relacionadas teniendo en cuenta la relación de accesibilidad original. Esto es claramente para que sean modalmente equivalentes.
- La valuación va a depender del último elemento de la secuencia.

Formalmente podemos pensar en:

Construcción: Dado un modelo $\mathcal{M} = \langle W, R, V \rangle$ con raíz w , vamos a construir $\mathcal{M}' = \langle W', R', V' \rangle$.

- ▶ Los elementos de W' son todas las secuencias finitas $(w \xrightarrow{R} u_1 \dots \xrightarrow{R} u_n)$ tal que haya un camino $wRu_1Ru_2\dots Ru_n$ en \mathcal{M} .
- ▶ $(w \xrightarrow{R} u_1 \dots \xrightarrow{R} u_n)R'(w \xrightarrow{R'} v_1 \dots \xrightarrow{R'} v_m)$ si y sólo si $m = n + 1, v_i = u_i$ para todo $1 \leq i \leq n, R' = R$ y u_nRv_m vale en \mathcal{M} .
- ▶ V' está definida como $(w \xrightarrow{R} u_1 \dots \xrightarrow{R} u_n) \in V'(p)$ si y sólo si $u_n \in V(p)$.

Claramente \mathcal{M}' tiene forma de árbol. Tenemos que demostrar que

$$f : (w \xrightarrow{R} u_1 \dots \xrightarrow{R} u_n) \longrightarrow u_n$$

define un morfismo acotado suryectivo. A partir de la manera en la que construimos \mathcal{M}' no es difícil ver que $f : (w \xrightarrow{R} u_1 \dots \xrightarrow{R} u_n) \longrightarrow u_n$ define un morfismo acotado suryectivo. Ya que cumplen con la definición por construcción.

Es decir, supongamos que φ es satisfecha en un modelo \mathcal{M} , en un punto w : Primero construimos \mathcal{M}' , el submodelo generado a partir de w . Como los submodelos generados preservan satisfacibilidad, sabemos que $\mathcal{M}', w \models \varphi$.

Además como \mathcal{M}' es un modelo de Kripke con raíz en w , podemos generar un modelo de Kripke \mathcal{M}'' con forma de árbol (siguiendo el procedimiento que acabamos de ver), en donde además $\mathcal{M}'', w \models \varphi$. Entonces, cualquier fórmula satisfacible puede ser satisfecha en un modelo de Kripke con forma de árbol. Ya para ir terminado con el tema de bisimulación en lógica modal básica, en la siguiente sección vamos a ver algunas propiedades de bisimulación en el caso finito.

2.2.4. Modelos finitos y n -Bisimulación

En esta sección vamos a ver que el lenguaje modal básico tiene la propiedad del modelo finito, esto es que si una fórmula es satisfacible en un modelo arbitrario, también lo es en un modelo finito.

Definición 16 *Sea \mathcal{C} una clase de modelos de Kripke. Decimos que un lenguaje tiene la **propiedad de modelo finito** con respecto a \mathcal{C} si vale lo siguiente:*

Si φ es una fórmula del lenguaje que es satisfacible en un modelo de Kripke de \mathcal{C} , entonces φ es satisfecha en un modelo de Kripke finito de \mathcal{C} .

Tomando \mathcal{C} la clase de todos los modelos de Kripke de la lógica modal básica y observando que esta tiene la propiedad de modelo finito, ya no tenemos que preocuparnos por modelos de Kripke infinitos arbitrarios, porque siempre vamos a poder encontrar uno finito equivalente respecto a una cantidad finita de fórmulas. Podemos pensar que la satisfacción modal es intrínsecamente local: las modalidades analizan los estados accesibles desde el estado actual. Pero ¿Qué parte del modelo puede ver una fórmula modal desde estado actual? Obviamente, eso depende de cuán profundamente están anidadas las modalidades que contiene. Para eso debemos definir el grado de una fórmula.

Definición 17 *Definimos el **grado** de una fórmula como sigue:*

$$\begin{aligned} \text{deg}(p) &= 0 \\ \text{deg}(\neg\varphi) &= \text{deg}(\varphi) \\ \text{deg}(\varphi \wedge \psi) &= \max\{\text{deg}(\varphi), \text{deg}(\psi)\} \\ \text{deg}(\diamond\varphi) &= 1 + \text{deg}(\varphi) \end{aligned}$$

Proposición 22 *Sea un lenguaje con una signatura finita (esto es, finitos símbolos proposicionales y modalidades) tenemos:*

- 1) *Para todo n , sólo hay un conjunto finito de fórmulas de grado a lo sumo n que no son lógicamente equivalentes.*

II) Para todo n , modelo de Kripke \mathcal{M} y estado w de \mathcal{M} , el conjunto de todas las fórmulas de grado a lo sumo n que son satisfechas en w es equivalente a una sola fórmula.

Demostración

Haremos solo la demostración de I), ya que II) se deduce de inmediato de I)

Procedemos por inducción en n . El caso $n = 0$ es obvio.

Para el caso $n + 1$, observemos que toda fórmula de grado $\leq n + 1$ es una combinación de fórmulas de la forma $\diamond\psi$, donde $\text{deg}(\psi) \leq n$, ya que \neg y \vee no aumentan el grado de una fórmula. Así, como $\text{deg}(\psi) \leq n$ por hipótesis inductiva hay sólo finitas combinaciones de fórmulas no equivalentes a ψ y por lo tanto solo hay finitas combinaciones de fórmulas no equivalentes a $\diamond\psi$. Por lo tanto hay solo finitas fórmulas no equivalentes de grado a lo sumo $n + 1$. ■

Con esto podemos pensar en aproximarnos a una noción de bisimulación para modelos de Kripke finitos.

Definición 18 Sean dos modelos de Kripke \mathcal{M} y \mathcal{M}' y w y w' dos mundos de \mathcal{M} y \mathcal{M}' respectivamente. Decimos que \mathcal{M}, w y \mathcal{M}', w' son **n-bisimilares** ($\mathcal{M}, w \rightleftharpoons_n \mathcal{M}', w'$) si existe una secuencia de relaciones binarias $\mathcal{Z}_n \subseteq \dots \subseteq \mathcal{Z}_0$ con las siguientes propiedades: ($i + 1 \leq n$)

- $w \mathcal{Z}_n w'$
- Si $v \mathcal{Z}_0 v'$ entonces v y v' concuerdan en todas las proposiciones.
- Si $v \mathcal{Z}_{i+1} v'$ y Rvu , entonces existe u' con $R'v'u'$ y $u \mathcal{Z}_i u'$
- Si $v \mathcal{Z}_{i+1} v'$ y $R'v'u'$, entonces existe u con Rvu y $u \mathcal{Z}_i u'$

Se puede ver que

- * Si $\mathcal{M}, w \rightleftharpoons_n \mathcal{M}', w'$ entonces \mathcal{M}, w y \mathcal{N}, w' son bisimilares hasta el nivel n .
- * Si $\mathcal{M}, w \rightleftharpoons \mathcal{M}', w'$ entonces $\mathcal{M}, w \rightleftharpoons_n \mathcal{M}', w'$ para todo n .
- * Y es claro que la vuelta no vale, es decir que $\mathcal{M}, w \rightleftharpoons_n \mathcal{M}', w'$ para todo n **no** implica $\mathcal{M}, w \rightleftharpoons \mathcal{M}', w'$.

Recordar la figura 2.5 de la sección anterior. Sabemos que no son bisimilares y es fácil ver que son n -bisimilares para cualquier $n \in \mathbb{N}$ ya el problema que teníamos antes por el cual no se cumplía la condición (Zag) ya no ocurre porque dado un nodo solo miro fórmulas hasta de profundidad a lo sumo n , y no me tropiezo con la posibilidad de hacerlo infinitas veces.

Vamos a ver que para modelos de Kripke de signatura finita, la noción de n -bisimilaridad coincide con la de modalmente equivalentes.

Proposición 23 *Sea un lenguaje con una signatura finita, y dos modelos \mathcal{M} y \mathcal{M}' de este lenguaje. Entonces, para todo w de \mathcal{M} y w' de \mathcal{M}' los siguientes puntos son equivalentes:*

$$i \mathcal{M}, w \Leftrightarrow_n \mathcal{M}', w'$$

ii w y w' concuerdan en todas las fórmulas modales de grado a lo sumo n

De esto se sigue que para todo n ,

$$w \Leftrightarrow_n w' \text{ si y solo si } w \equiv_n w'$$

(observemos que \equiv_n se define análogamente)

Para la última propiedad que veremos tenemos que introducir las nociones de altura y restricción.

Definición 19 *Sea \mathcal{M} un modelo de Kripke con forma de árbol con raíz w . La **altura** para cada elemento del modelo se define de la siguiente manera:*

El único elemento con altura 0 es w .

Los estados con altura $n + 1$ son los inmediatos sucesores de los estados con altura n que todavía no fueron asignados con una altura menor a $n + 1$.

La altura de un modelo de Kripke \mathcal{M} es el máximo n tal que existe un estado en \mathcal{M} con altura n , si es que tal máximo existe. Si no existe, la altura de \mathcal{M} es infinita. La altura de w la notamos $height(w)$.

Restricción: *Para un natural k , la restricción de \mathcal{M} a k (notación: $\mathcal{M} \upharpoonright k$) está definida como:*

$(\mathcal{M} \upharpoonright k) = (W_k, R_k, V_k)$ donde $W_k = \{v | height(v) \leq k\}$, $R_k = R \cap (W_k \times W_k)$, y para cada p , $V_k(p) = V(p) \cap W_k$.

Proposición 24 *Sea \mathcal{M} un modelo de Kripke con raíz, y sea k un número natural. Entonces, por cada estado w de $(\mathcal{M} \upharpoonright k)$ vale que $(\mathcal{M} \upharpoonright k), w \Leftrightarrow_l \mathcal{M}, w$, donde $l = k - height(w)$.*

Capítulo 3

Lógica XPath₌

3.1. Introducción a la lógica XPath₌

XPath es un lenguaje de consulta que permite recuperar información de un documento XML. Obtiene su denominación por el uso que hace de una notación de caminos para navegar a través de la estructura jerárquica de un documento XML. XPath opera sobre la estructura lógica abstracta de un documento XML, más que en su sintaxis superficial y modela el documento XML como un árbol de nodos. Tiene como fin poder definir un subconjunto de nodos de un documento XML para su posterior procesamiento. Nos sirve para poder identificar partes específicas de un documento XML. [3]

En este capítulo vamos a estudiar los modelos de XPath con igualdad (notado XPath₌). La mayor característica de XPath₌ es que nos permite tener formulas del estilo $\langle \alpha = \beta \rangle$ donde α y β son expresiones de camino, que navegan por el árbol utilizando ejes: descendiente, hijo, ancestro, etc, y puede hacer pruebas en nodos intermedios. La fórmula es cierta en un nodo x de un árbol de datos si hay nodos y, z que pueden alcanzarse mediante las relaciones indicadas por α y β , respectivamente, y de modo que el valor de datos de y sea igual al valor de datos de z . [9]. Así, XPath₌ sirve para decir cómo debe procesar el contenido de una página XML, pero también para poder poner enlaces o cargar en un navegador zonas determinadas de una página XML, en vez de toda la página.

Los modelos de XPath son árboles con datos cuyos nodos contienen etiquetas de un alfabeto junto con una valuación de un dominio infinito. Con árboles nos referimos como antes, a una estructura con un nodo generador, que llamaremos raíz, y ramificaciones que tienen hojas que llamaremos hijos, padres vecinos, etc. Veamos más concretamente esto:

Definición 20 * Usaremos el símbolo \mathbb{A} para denotar un alfabeto finito y \mathbb{D} para un dominio infinito de valores.

* Definimos $\mathbf{Arboles}(\mathbb{A})$ como el conjunto de árboles sobre el alfabeto \mathbb{A} .

- * Decimos que \mathcal{T} es un **árbol de datos** si es un árbol de $\mathbf{Arboles}(\mathbb{A} \times \mathbb{D})$ llamamos a \mathbb{A} el conjunto de etiquetas y \mathbb{D} el conjunto de datos.
- * Dado un árbol de datos \mathcal{T} , notamos su conjunto de nodos con T . Usamos las letras x, y, z, v, w como variables para los nodos.
- * Dado un nodo $x \in T$ de \mathcal{T} , escribimos $\mathbf{etiqueta}(x) \in \mathbb{A}$ para denotar la etiqueta del nodo y $\mathbf{Data}(x) \in \mathbb{D}$ para denotar el valor del nodo.
- * Decimos que un árbol de datos se **bifurca finitamente** si cada nodo tiene finitos nodos hijos.
- * Dados dos nodos $x, y \in T$ escribimos $x \rightarrow y$ si y es un hijo de x . Y escribimos $x \xrightarrow{n} y$ si y es un descendiente de x de distancia n . (para $x \xrightarrow{1} y$ escribimos $x \rightarrow y$)
- * $(x \xrightarrow{n})$ denota el conjunto de todos los descendientes de x de distancia n y $(\xrightarrow{n} y)$ denota el único ancestro de y a distancia n (asumiendo que hay uno).

Vamos a ver con un ejemplo sencillo, las definiciones recién dadas.

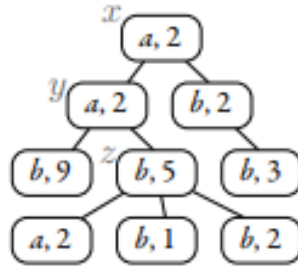


Figura 3.1: Árbol de datos de $\mathbf{Arboles}(\mathbb{A} \times \mathbb{D})$ con $\mathbb{A} = \{a, b\}$ y $\mathbb{D} = \mathbb{N}$

En la Figura 3.1 podemos tomar el nodo x cuya etiqueta es a y cuyo valor de dato es 2. Además se ve claramente que $x \rightarrow y$ y que $x \xrightarrow{2} z$ y además $(\rightarrow z)$ es y .

Consideraremos el fragmento de XPath que corresponde a la parte de XPath con igualdad y desigualdad, como ya mencionamos antes. XPath₌ es un lenguaje doble, con expresiones de camino $(\alpha, \beta, \mathcal{T} \dots)$ y expresiones de nodos $(\varphi, \psi, \eta \dots)$.

Dentro del fragmento XPath₌ podemos destacar varios fragmentos más tales como el "XPath vertical" (que denotamos XPath_↓) y cuya sintaxis se da recursivamente como sigue:

$$\begin{aligned}
 \alpha, \beta &::= o \mid [\varphi] \mid \alpha\beta \mid \alpha \cup \beta & o &\in \{\varepsilon, \uparrow, \downarrow\} \\
 \varphi, \psi &::= a \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \langle \alpha \rangle \mid \langle \alpha = \beta \rangle \mid \langle \alpha \neq \beta \rangle & a &\in \mathbb{A}
 \end{aligned}$$

Llamamos “XPath descendiente” al fragmento de XPath que solo usa la relación de navegación \downarrow (y no \uparrow) y lo notamos XPath_↓. Una fórmula de XPath_↓[†] [respectivamente XPath_↓[‡]] es simplemente una expresión de nodo o de camino en XPath_↓[†] [resp. XPath_↓[‡]]. Dado un árbol de dato \mathcal{T} se define la semántica de XPath_↓[†] como sigue:

- $\llbracket \downarrow \rrbracket^{\mathcal{T}} = \{(x, y) \mid x \rightarrow y\}$
- $\llbracket \uparrow \rrbracket^{\mathcal{T}} = \{(x, y) \mid y \rightarrow x\}$
- $\llbracket \vartheta \rrbracket^{\mathcal{T}} = \{(x, x) \mid x \in T\}$
- $\llbracket \alpha\beta \rrbracket^{\mathcal{T}} = \{(x, z) \mid (\exists y \in T)(x, y) \in \llbracket \alpha \rrbracket^{\mathcal{T}}, (y, z) \in \llbracket \beta \rrbracket^{\mathcal{T}}\}$
- $\llbracket \alpha \cup \beta \rrbracket^{\mathcal{T}} = \llbracket \alpha \rrbracket^{\mathcal{T}} \cup \llbracket \beta \rrbracket^{\mathcal{T}}$
- $\llbracket [\varphi] \rrbracket^{\mathcal{T}} = \{(x, x) \mid x \in \llbracket \varphi \rrbracket^{\mathcal{T}}\}$
- $\llbracket [a] \rrbracket^{\mathcal{T}} = \{x \in T \mid \text{etiqueta}(x) = a\}$
- $\llbracket \neg\varphi \rrbracket^{\mathcal{T}} = T \setminus \llbracket \varphi \rrbracket^{\mathcal{T}}$
- $\llbracket \varphi \wedge \psi \rrbracket^{\mathcal{T}} = \llbracket \varphi \rrbracket^{\mathcal{T}} \cap \llbracket \psi \rrbracket^{\mathcal{T}}$
- $\llbracket \langle \alpha \rangle \rrbracket^{\mathcal{T}} = \{x \in T \mid (\exists y \in T)(x, y) \in \llbracket \alpha \rrbracket^{\mathcal{T}}\}$
- $\llbracket \langle \alpha = \beta \rangle \rrbracket^{\mathcal{T}} = \{x \in T \mid (\exists y, z \in T)(x, y) \in \llbracket \alpha \rrbracket^{\mathcal{T}}, (x, z) \in \llbracket \beta \rrbracket^{\mathcal{T}}, \text{data}(y) = \text{data}(z)\}$
- $\llbracket \langle \alpha \neq \beta \rangle \rrbracket^{\mathcal{T}} = \{x \in T \mid (\exists y, z \in T)(x, y) \in \llbracket \alpha \rrbracket^{\mathcal{T}}, (x, z) \in \llbracket \beta \rrbracket^{\mathcal{T}}, \text{data}(y) \neq \text{data}(z)\}$

Además podemos pensar en el fragmento XPath_↓^{†*} [resp. XPath_↓^{‡*}] agregándole a XPath_↓[†] [resp. XPath_↓[‡]] los ejes transitivos \uparrow^* y \downarrow^* con la siguiente interpretación:

- $\llbracket \downarrow^* \rrbracket^{\mathcal{T}} = \text{Clausura reflexiva transitiva de } \llbracket \downarrow \rrbracket^{\mathcal{T}}$
- $\llbracket \uparrow^* \rrbracket^{\mathcal{T}} = \text{Clausura reflexiva transitiva de } \llbracket \uparrow \rrbracket^{\mathcal{T}}$

Como ejemplo en el árbol de la Figura 3.1 tenemos

$$\llbracket \langle \downarrow^* [b \wedge \langle \downarrow [b] \neq \downarrow [b] \rangle] \rangle \rrbracket = \{x, y, z\}$$

y se lee *Hay un nodo descendiente (o el mismo) con etiqueta b con dos hijos con etiqueta b con distinto valor de dato.*

Para un árbol de datos \mathcal{T} y $u \in T$ decimos que \mathcal{T}, u es un árbol punteado. Escribimos $\mathcal{T}, u \models \varphi$ para denotar $u \in \llbracket \varphi \rrbracket^{\mathcal{T}}$ y decimos que \mathcal{T}, u satisface φ ó que φ es verdadera en \mathcal{T}, u

Definición 21 Dadas dos expresiones de nodo φ, ψ de $XPath_{=}$. Decimos que son **equivalentes** (not. $\varphi \equiv \psi$) si y sólo si $\llbracket \varphi \rrbracket^{\mathcal{T}} = \llbracket \psi \rrbracket^{\mathcal{T}}$ para todo árbol de datos \mathcal{T} . Similarmente las expresiones de camino α, β de $Xpath_{=}$ son equivalentes (not. $\alpha \equiv \beta$) si y sólo si $\llbracket \alpha \rrbracket^{\mathcal{T}} = \llbracket \beta \rrbracket^{\mathcal{T}}$.

Dados dos árboles de datos \mathcal{T} y \mathcal{T}' y sean $u \in T, u' \in T'$. Decimos que \mathcal{T}, u y \mathcal{T}', u' son **equivalentes para $XPath_{=}^{\uparrow}$** [resp. equivalente para $XPath_{=}^{\downarrow}$] si y solo si para toda fórmula $\varphi \in XPath_{=}^{\uparrow}$ [resp. $\varphi \in XPath_{=}^{\downarrow}$] tenemos que $\mathcal{T}, u \models \varphi$ si y sólo si $\mathcal{T}', u' \models \varphi$. Y lo notamos $\mathcal{T}, u \equiv^{\uparrow} \mathcal{T}', u'$ [resp $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$].

3.1.1. Correspondencia de modelos

En esta sección mostraremos que hay una traducción que preserva la verdad de $Xpath_{=}^{\uparrow}$ a la lógica de primer orden con una signatura apropiada.[10] Para esto, primero debemos interpretar árboles de datos en estructuras relacionales, y lo haremos de la forma más estándar: usando una relación binaria que llamaremos "Hijo", una relación de equivalencia por testeo de igualdad de datos y una relación monoádica para probar etiquetas. Fijamos la signatura σ con relaciones binarias \rightsquigarrow y \approx y un predicado unario " P_a " para cada $a \in \mathbb{A}$. Cualquier árbol de datos \mathcal{T} puede verse como primer orden con una σ -estructura como la siguiente:

$$\begin{aligned} \rightsquigarrow^{\mathcal{T}} &= \{(x, y) \in T^2 \mid y \text{ es hijo de } x\} \\ \approx^{\mathcal{T}} &= \{(x, y) \in T^2 \mid data(x) = data(y)\} \\ P_a^{\mathcal{T}} &= \{x \in T \mid etiqueta(x) = a\} \end{aligned}$$

Ahora podemos dar la traducción de $XPath_{=}$ a la lógica de primer orden vía σ . La función de traducción está indexada por las variables libres de la fórmula producida, una para expresiones de nodo y dos para expresiones de camino.

$$\begin{aligned} Tr_x(a) &= P_a(x) && a \in \mathbb{A} \\ Tr_x(\varphi \top \psi) &= Tr_x(\varphi) \top Tr_x(\psi) && \top \in \{\wedge, \vee\} \\ Tr_x(\neg \varphi) &= \neg Tr_x(\varphi) \\ Tr_x(\langle \alpha \rangle) &= (\exists y) Tr_{x,y}(\alpha) && (y \text{ es una nueva variable}) \\ Tr_x(\langle \alpha = \beta \rangle) &= (\exists y)(\exists z)(y \approx z \wedge Tr_{x,y}(\alpha) \wedge Tr_{x,z}(\beta)) && (z, y \text{ son variables nuevas}) \\ Tr_x(\langle \alpha \neq \beta \rangle) &= (\exists y)(\exists z)(y \not\approx z \wedge Tr_{x,y}(\alpha) \wedge Tr_{x,z}(\beta)) && (z, y \text{ son variables nuevas}) \\ Tr_{x,y}(\varepsilon) &= (x = y) \\ Tr_{x,y}(\downarrow) &= (x \rightsquigarrow y) \\ Tr_{x,y}(\uparrow) &= (y \rightsquigarrow x) \\ Tr_{x,y}(\alpha \beta) &= (\exists z)(Tr_{x,z}(\alpha) \wedge Tr_{z,y}(\beta)) && (z \text{ es una variable nueva}) \\ Tr_{x,y}(\alpha \cup \beta) &= Tr_{x,y}(\alpha) \vee Tr_{x,y}(\beta) \\ Tr_{x,y}([\varphi]) &= Tr_x(\varphi) \wedge (x = y) \end{aligned}$$

Proposición 25 1) Si φ es una expresión de nodo de $XPath_{\underline{=}}^{\downarrow}$ entonces

$$u \in \llbracket \varphi \rrbracket^{\mathcal{T}} \text{ si y solo si } \mathcal{T}, V(x = u) \models Tr_x(\varphi)$$

2) Si α es una expresión de camino de $XPath_{\underline{=}}^{\downarrow}$ entonces

$$(u, w) \in \llbracket \alpha \rrbracket^{\mathcal{T}} \text{ si y solo si } \mathcal{T}, V(x = u, y = w) \models Tr_{x,y}(\alpha)$$

Para la demostración ver [9]

Definición 22 Para φ una fórmula de primer orden, sea $qr(\varphi)$ el cuantificador de rango, es decir la profundidad de sus cuantificadores. Definido de la siguiente manera

$$\begin{aligned} qr(Tr_x(\langle \alpha \rangle)) &= 1 + qr(Tr_x(\alpha)) \\ qr(Tr_x(\langle \alpha \odot \beta \rangle)) &= 2 + \max(qr(Tr_x(\alpha)), qr(Tr_x(\beta))) \text{ para } \odot \in \{=, \neq\} \\ qr(Tr_{x,y}(\alpha\beta)) &= 1 + \max(qr(Tr_{x,y}(\alpha)), qr(Tr_{x,y}(\beta))) \\ qr(Tr_{x,y}([\varphi])) &= qr(Tr_x(\varphi)) \end{aligned}$$

3.2. $XPath_{\underline{=}}^{\downarrow}$

En esta sección vamos a estudiar el modelo $XPath_{\underline{=}}^{\downarrow}$. Veremos la noción de bisimulación, de l -bisimulación y como se relaciona con la noción de equivalencia lógica de modelos.

Definición 23 Dada una formula φ en $XPath_{\underline{=}}^{\downarrow}$, vamos a denotar con $dd(\varphi)$ la profundidad descendiente de φ como:

$$\begin{array}{lll} dd(a) = 0 & dd(\lambda) = 0 & dd(\neg\varphi) = dd(\varphi) \\ dd(\varphi \wedge \psi) = \max\{dd(\varphi), dd(\psi)\} & & dd(\varepsilon\alpha) = dd(\alpha) \\ dd([\varphi]\alpha) = \max\{dd(\varphi), dd(\alpha)\} & & dd(\downarrow\alpha) = 1 + dd(\alpha) \\ dd(\langle \alpha \odot \beta \rangle) = \max\{dd(\alpha), dd(\beta)\} & & dd(\langle \alpha \rangle) = dd(\alpha) \end{array}$$

con $a \in \mathbb{A}$, $\odot \in \{=, \neq\}$, α cualquier expresión de camino ó el camino vacío λ .

Con esta nueva noción de profundidad de cada fórmula podemos pensar en un nuevo fragmento de $XPath_{\underline{=}}^{\downarrow}$, que llamaremos l - $XPath_{\underline{=}}^{\downarrow}$ que consiste en toda las formulas φ de $XPath_{\underline{=}}^{\downarrow}$ con $dd(\varphi) \leq l$. Entonces para este fragmento decimos que dos árboles de datos punteados \mathcal{T}, u y \mathcal{T}', u' son l -equivalentes si y sólo si toda expresión de nodos $\varphi \in l$ - $XPath_{\underline{=}}^{\downarrow}$ tenemos que $\mathcal{T}, u \models \varphi$ si y sólo si $\mathcal{T}', u' \models \varphi$. Lo notamos $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$

Vale la pena aclarar que cuando decimos que dos árboles de datos punteados son equivalentes nos referimos a que son indistinguibles por las fórmulas dadas de una lógica. No debemos confundir esto con equivalencias de fórmulas.

3.2.1. Bisimulación y l -Bisimulación

Definición 24 Sean \mathcal{T} y \mathcal{T}' dos árboles de datos. Decimos que $u \in T$ y $u' \in T'$ son **bisimilares para XPath_↓** o **↓-bisimilares** (lo notamos $\mathcal{T}, u \Leftrightarrow^\downarrow \mathcal{T}', u'$) si hay una relación $\mathcal{Z} \subseteq T \times T'$ tal que $u\mathcal{Z}u'$ y $\forall x \in T, x' \in T'$ tenemos

(Harmony) Si $x\mathcal{Z}x' \Rightarrow \text{etiqueta}(x) = \text{etiqueta}(x')$

(Zig) Si $x\mathcal{Z}x', x \xrightarrow{n} v$ y $x \xrightarrow{m} w$ entonces hay $v', w' \in T'$ tales que $x' \xrightarrow{n} v', x' \xrightarrow{m} w'$ y vale

1) $\text{Data}(v) = \text{Data}(w)$ si y sólo si $\text{Data}(v') = \text{Data}(w')$

2) $(\xrightarrow{i} v)\mathcal{Z}(\xrightarrow{i} v') \forall 0 \leq i \leq n$

3) $(\xrightarrow{i} w)\mathcal{Z}(\xrightarrow{i} w') \forall 0 \leq i \leq m$

(Zag) Si $x\mathcal{Z}x', x' \xrightarrow{n} v'$ y $x' \xrightarrow{m} w'$ entonces hay $v, w \in T$ tales que los items 1), 2) y 3) se verifican.

Para que sea más fácil de entender la Figura 3.2.1 representa la condición (Zig)

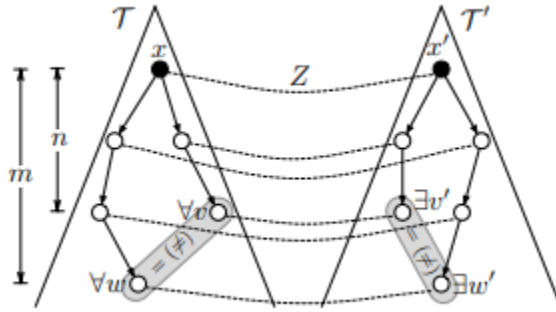


Figura 3.2: Condición (Zig) entre τ y τ'

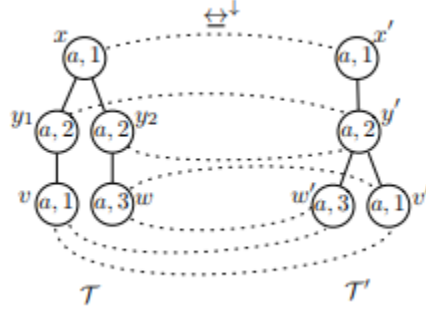
Por ejemplo, la línea punteada de los siguientes árboles de datos en la Figura 3.2.1 representa una bisimulación en XPath_↓

Definición 25 Dado un árbol de datos \mathcal{T} y $u \in T$. Denotamos con $(\mathcal{T} \mid u)$ al **sub-árbol** de \mathcal{T} inducido por $\{v \in T \mid (\exists u) (u \xrightarrow{n} v)\}$

Observemos que la raíz de $(\mathcal{T} \mid u)$ es u . Así los siguientes resultados son consecuencia inmediata de la definición de bisimulación.

Observaciones:

1) $\mathcal{T}, u \Leftrightarrow^\downarrow (\mathcal{T} \mid u), u$

Figura 3.3: $\tau \Leftrightarrow^{\downarrow} \tau'$

2) Si \mathcal{T} es un subárbol de \mathcal{T}' y $u \in T$ entonces $\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u$

Sigamos con algunas definiciones y observaciones para luego ver como se relacionan estas nociones con la de equivalencia.

Definición 26 *Dados dos árboles de $XPath_{\perp}^{\downarrow}$, \mathcal{T} y \mathcal{T}' . Decimos que $u \in T$ y $u' \in T'$ son l -bisimilares para $XPath_{\perp}^{\downarrow}$ (notamos $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u'$) si hay una familia de relaciones $(\mathcal{Z}_j)_{j \leq l}$ en $T \times T'$ tales que $u \mathcal{Z}_l u'$ y $\forall j \leq l, x \in T, x' \in T'$ tenemos*

(Harmony) Si $x \mathcal{Z}_j x' \Rightarrow etiqueta(x) = etiqueta(x')$

(Zig) Si $x \mathcal{Z}_j x', x \xrightarrow{n} v$ y $x \xrightarrow{m} w$ con $n, m \leq j$ entonces hay $v', w' \in T'$ tales que $x' \xrightarrow{n} v'$ y $x' \xrightarrow{m} w'$ y

1) $Data(v) = Data(w)$ si y sólo si $Data(v') = Data(w')$

2) $(\xrightarrow{i} v) \mathcal{Z}_{j-n+i} (\xrightarrow{i} v') \forall 0 \leq i \leq n$

3) $(\xrightarrow{i} w) \mathcal{Z}_{j-m+i} (\xrightarrow{i} w') \forall 0 \leq i \leq m$

(Zag) Si $x \mathcal{Z}_j x', x' \xrightarrow{n} v'$ y $x' \xrightarrow{m} w'$ con $n, m \leq j$ entonces hay $v, w \in T$ tales que los item 1), 2) y 3) se verifican.

Observaciones:

- 1) Claramente si $\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u' \Rightarrow \mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u' \forall l$ ya que la bisimulación \mathcal{Z} que existe por hipótesis es la que sirve para todas las \mathcal{Z}_i
- 2) Si lo pensamos como el juego de Ehrenfeuch-Fraïsse, (ver Apéndice) correspondería a un juego de l rondas, donde podemos pensar que la estrategia de Duplicador para cada ronda nos da una \mathcal{Z}_i

- 3) Supongamos que \mathcal{T} y \mathcal{T}' tienen altura como máximo l , $u \in T$ $u' \in T'$. Entonces $\mathcal{T}, u \Leftrightarrow_l^\downarrow \mathcal{T}', u'$ si y sólo si $\mathcal{T}, u \Leftrightarrow^\downarrow \mathcal{T}', u'$
- 4) Para un árbol de datos \mathcal{T} y $u \in T$. Sea $\mathcal{T} \upharpoonright_l u$ el subárbol generado por $\{v \in T \mid (\exists n \leq l) (u \xrightarrow{n} v)\}$. Así $\mathcal{T}, u \Leftrightarrow_l^\downarrow (\mathcal{T} \upharpoonright_l u), u$

3.2.2. Equivalencia y bisimulación

En esta sección vamos a ver que $\Leftrightarrow^\downarrow$ coincide con \equiv^\downarrow en árboles de ramificación finita y que $\Leftrightarrow_l^\downarrow$ coincide con \equiv_l^\downarrow . Vamos a ir enunciando resultados parciales para luego llegar al Teorema que queremos. De acá en adelante pensaremos que \mathcal{T} y \mathcal{T}' son dos árboles de datos en $XPath_{\downarrow}^*$, que $u \in T$ y que $u' \in T'$. Primero demosntremos un resultado que nos va a servir más adelante.

Lema 26 *Sea α una expresión de camino en $XPath_{\downarrow}^*$. Sea $x \xrightarrow{n} v$ y $x' \xrightarrow{n} v'$ tales que $(x, v) \in \llbracket \alpha \rrbracket^{\mathcal{T}}$ y $(x', v') \notin \llbracket \alpha \rrbracket^{\mathcal{T}'}$. Entonces hay una subfórmula φ de α y $k \in \{0, \dots, n\}$ tales que $\mathcal{T}, (\xrightarrow{k} v) \models \varphi$ y $\mathcal{T}', (\xrightarrow{k} v) \not\models \varphi$ (Con \downarrow^* hacemos referencia a los descendientes y al nodo en si mismo)*

Demostración

Sea $x = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n = v$ y $x' = v'_0 \rightarrow \dots \rightarrow v_n = v'$.

Procedemos por inducción en $|\alpha|$.

Caso base: $\alpha = \varepsilon$, entonces $x = v$ y $n = 0$. Por lo tanto tiene que ser $x' = v'$ y $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$, lo que contradice la hipótesis y entonces el entonces es trivialmente verdadero.

Si $\alpha = \downarrow$, entonces $x \rightarrow v$ y $n = 1$, lo que implica que $x' \rightarrow v'$ y $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$ lo que hace a este caso también trivial.

El caso $\alpha = \downarrow^*$ es similar.

Paso inductivo. $\alpha = [\psi]$. Como $(x', v') \notin \llbracket \alpha \rrbracket^{\mathcal{T}'}$ tenemos que $x' = v'$ y $\mathcal{T}', x' \not\models \varphi$. Tomando $k = 0$ y $\varphi = \psi$ el estado se mantiene. Observemos que ψ es un subfórmula de α .

Supongamos que $\alpha = \beta\mathcal{T}$. Hay un k tal que $(x, v_k) \in \llbracket \beta \rrbracket^{\mathcal{T}}$ y $(v_k, v) \in \llbracket \mathcal{T} \rrbracket^{\mathcal{T}}$. Como $(x', v') \notin \llbracket \alpha \rrbracket^{\mathcal{T}'}$, tenemos $(x', v'_k) \notin \llbracket \beta \rrbracket^{\mathcal{T}'}$ ó $(v_k, v') \notin \llbracket \mathcal{T} \rrbracket^{\mathcal{T}'}$. En cualquier caso aplicando fácilmente la hipótesis inductiva llegamos a lo que queremos. ■

Proposición 27 $\mathcal{T}, u \Leftrightarrow_l^\downarrow \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv_l^\downarrow \mathcal{T}', u'$

Demostración

Para probar esto demostraremos que si $\mathcal{T}, u \Leftrightarrow_l^\downarrow \mathcal{T}', u'$ entonces $\forall 0 \leq n \leq j \leq l \forall \varphi$ con $\text{dd}(\varphi) \leq l$ y $\forall \alpha$ con $\text{dd}(\alpha) \leq j$ vale

- 1) Si $x \mathcal{Z}_j x' \Rightarrow \mathcal{T}, x \models \varphi$ si y sólo si $\mathcal{T}', x' \models \varphi$
- 2) Si $x \xrightarrow{n} v$, $x' \xrightarrow{n} v'$ y $(\xrightarrow{i} v) \mathcal{Z}_{j-n+i} (\xrightarrow{i} v') \forall 0 \leq i \leq n$ entonces $(x, v) \in \llbracket \alpha \rrbracket^{\mathcal{T}}$ si y sólo si $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$

Mostraremos 1) y 2) por inducción en $|\varphi| + |\alpha|$.

Veamos primero 1). El caso base es $\varphi = a$ con $a \in \mathbb{A}$. Por (Harmony), etiqueta $(x) =$ etiqueta (x') , o sea $x \in \llbracket \varphi \rrbracket^{\mathcal{T}}$ y $x' \in \llbracket \varphi \rrbracket^{\mathcal{T}'}$. Entonces $\mathcal{T}, x \models \varphi$ si y sólo si $\mathcal{T}', x' \models \varphi$

Los casos $\varphi = \neg\psi$ y $\varphi = \gamma \wedge \psi$ son sencillos.

Hagamos directamente el paso inductivo. Supongamos que $\varphi = \langle \alpha = \beta \rangle$. Mostraremos que $\mathcal{T}, x \models \varphi$ implica $\mathcal{T}', x' \models \varphi$. La recíproca es análoga.

Asumamos que $\mathcal{T}, x \models \varphi$. Supongamos que hay $v, w \in T$ $n, m \leq j$ tales que $x \xrightarrow{n} v$, $x \xrightarrow{m} w$, $(x, v) \in \llbracket \alpha \rrbracket^{\mathcal{T}}$, $(x, w) \in \llbracket \beta \rrbracket^{\mathcal{T}}$ y $\text{data}(v) = \text{data}(w)$. Por (Zig) hay $v', w' \in T'$ tales que $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$, $(\xrightarrow{i} v) \mathcal{Z}_{j-n+1} (\xrightarrow{i} v')$, $\forall 0 \leq i < n$, $(\xrightarrow{i} w) \mathcal{Z}_{j-m+1} (\xrightarrow{i} w')$, $\forall 0 \leq i < m$ y $\text{data}(v') = \text{data}(w')$. Por hipótesis inductiva aplicada dos veces tenemos que $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$ y $(x', w') \in \llbracket \beta \rrbracket^{\mathcal{T}'}$. Entonces $\mathcal{T}', x' \models \varphi$.

El caso $\varphi = \langle \alpha \neq \beta \rangle$ y el caso $\varphi = \langle \alpha \rangle$ es similar al que recién hicimos. Analicemos ahora el caso 2). Solo vamos a demostrar por inducción que si $x \xrightarrow{n} v$, $x' \xrightarrow{n} v'$ y $(\xrightarrow{i} v) \mathcal{Z}_{j-n+i} (\xrightarrow{i} v') \forall 0 \leq i \leq n$ entonces $(x, v) \in \llbracket \alpha \rrbracket^{\mathcal{T}}$ **entonces** $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$

Caso Base: $\varphi \in \{\varepsilon, \downarrow\}$.

Si $\alpha = \varepsilon$, entonces $v = x$ y $n = 0$. Como $v' = x'$ concluimos que $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$

Si $\alpha = \downarrow$ entonces $x \rightarrow v$ en \mathcal{T} y $n = 1$. Como $x' \rightarrow v'$ tenemos que $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$.

Para el caso inductivo tenemos $x_0, \dots, x_n \in T$ y $x'_0, \dots, x'_n \in T'$ tales que

$$x = x_0 \rightarrow x_1 \rightarrow x_2 \dots \rightarrow x_n = v \text{ en } \mathcal{T}$$

$$x' = x'_0 \rightarrow x'_1 \rightarrow x'_2 \dots \rightarrow x'_n = v' \text{ en } \mathcal{T}' \text{ y } x_i \mathcal{Z}_{j-i} x'_i \forall 0 \leq i \leq n.$$

Vamos a proceder por el absurdo. Supongamos que $(x', v') \notin \llbracket \alpha \rrbracket^{\mathcal{T}'}$. Entonces, por el Lema anterior, hay una subfórmula φ para α y $k \in \{0, \dots, n\}$ tales que $\mathcal{T}, x_k \models \varphi$ y $\mathcal{T}', x'_k \not\models \varphi$ lo que contradice la hipótesis. Luego demostramos lo que queríamos. ■

Proposición 28 $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u' \Rightarrow \mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$

Demostración

Tomamos $u \in T$, $u' \in T'$ tales que $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$. Definimos $(\mathcal{Z}_i)_{i \leq l}$ como

$$x \mathcal{Z}_i x' \text{ si y sólo si } \mathcal{T}, u \equiv_i^{\downarrow} \mathcal{T}', u'$$

Veremos que \mathcal{Z} es una l -bisimulación entre \mathcal{T}, u y \mathcal{T}', u' Por hipótesis $u \mathcal{Z}_l u'$. Tomo $h \leq l$. Por construcción \mathcal{Z}_h satisface (Harmony).

Veamos que \mathcal{Z}_h satisface (Zig) ((Zag) es análogo).

Supongamos que $u \mathcal{Z}_h u'$, tenemos

$$\begin{aligned} x = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n = v \text{ en } \mathcal{T} \\ x = w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_n = w \text{ en } \mathcal{T} \end{aligned}$$

y $\text{data}(v) = \text{data}(w)$ (el caso $\text{data}(v) \neq \text{data}(w)$ es análogo) donde $m, n \leq h$. Sea $P \subseteq T^2$ definido como

$$P = \{(v, w) / x' \xrightarrow{n} v' \wedge x' \xrightarrow{m} w' \text{ y } \text{data}(v') = \text{data}(w')\}$$

Como $\mathcal{T}, x \equiv_h^\downarrow \mathcal{T}', x'$, $\text{dd}(\langle \downarrow^n = \downarrow^m \rangle) \leq h$ y $\mathcal{T}, x \models \langle \downarrow^n = \downarrow^m \rangle$ concluimos que $P \neq \emptyset$. Demostraremos que existe $(v', w') \in P$ tal que

- i) $x' = v'_0 \rightarrow v'_1 \rightarrow \dots \rightarrow v'_n = v'$ en \mathcal{T}'
- ii) $x' = w'_0 \rightarrow w'_1 \rightarrow \dots \rightarrow w'_m = w'$ en \mathcal{T}'
- iii) $(\forall i \in \{0, \dots, n\}) \mathcal{T}, v_i \equiv_{h-i}^\downarrow \mathcal{T}', v'_i$
- iv) $(\forall j \in \{0, \dots, m\}) \mathcal{T}, w_j \equiv_{h-j}^\downarrow \mathcal{T}', w'_j$

Y así \mathcal{Z}_h satisface (Zig).

Procedamos por el absurdo.

Asumamos que $\forall (v', w') \in P$ que satisface *i*), *ii*), *iii*) y *iv*) tenemos que pasar algún de estos casos:

- a) $(\exists i \in \{0, \dots, n\}) / \mathcal{T}, v_i \equiv_{h-i}^\downarrow \mathcal{T}', v'_i$
- b) $(\exists j \in \{0, \dots, m\}) / \mathcal{T}, w_j \not\equiv_{h-j}^\downarrow \mathcal{T}', w'_j$

Tomemos \top cualquier tautología tal que $\text{dd}(\top) = 0$. Para cada $(v', w') \in P$ definimos dos familias de expresión de nodo,

$$\varphi_{v',w'}^0, \dots, \varphi_{v',w'}^n \text{ y } \psi_{v',w'}^0, \dots, \psi_{v',w'}^m$$

que satisfagan que $\text{dd}(\varphi_{v',w'}^i) \leq h-i \forall i \in \{0, \dots, n\}$ y $\text{dd}(\psi_{v',w'}^j) \leq h-j \forall j \in \{0, \dots, m\}$, como los siguientes

- Asumiendo que vale *a*) y que *i* sea el número más chico tal que $\mathcal{T}, v_i \not\equiv_{h-i}^\downarrow \mathcal{T}', v'_i$. Sea $\varphi_{v',w'}^i$ tal que $\text{dd}(\varphi_{v',w'}^i) \leq h-i$ y $\mathcal{T}, v_i \models \varphi_{v',w'}^i$ pero $\mathcal{T}', v'_i \not\models \varphi_{v',w'}^i$. Para $k \in \{0, \dots, n\} \setminus \{i\}$ sea $\varphi_{v',w'}^k = \top$ y para $k \in \{0, \dots, m\}$ sea $\psi_{v',w'}^k = \top$.
- Suponiendo que *a*) no vale. Entonces vale *b*). Sea *j* el número más chico al que $\mathcal{T}, w_j \not\equiv_{h-j}^\downarrow \mathcal{T}', w'_j$. Sea $\psi_{v',w'}^j$ tal que $\text{dd}(\psi_{v',w'}^j) \leq h-j$ y $\mathcal{T}, w_j \models \psi_{v',w'}^j$ pero $\mathcal{T}', w'_j \not\models \psi_{v',w'}^j$. Para $k \in \{0, \dots, m\} \setminus \{j\}$ sea $\psi_{v',w'}^k = \top$ y para $k \in \{0, \dots, n\}$ sea $\varphi_{v',w'}^k = \top$.

Para cada $i \in \{0, \dots, n\}$ y $j \in \{0, \dots, m\}$ sea

$$\Phi^i = \bigwedge_{(v',w') \in P} \varphi_{v',w'}^i \quad y \quad \Psi^j = \bigwedge_{(v',w') \in P} \psi_{v',w'}^j \quad (1)$$

Sin embargo estas conjunciones pueden ser potencialmente infinitas, si los árboles son de ramificaciones infinitas.

Como $dd(\varphi_{v',w'}^i) \leq h - i$, ya que para todo l , \equiv_l^{\downarrow} tiene finitos índices (ver [1]) por lo que sabemos que cada conjunción es equivalente a una con finitos términos y podemos asumir que Φ^i y Ψ^j están bien definidas.

Finalmente sea $\alpha = [\Phi^0] \downarrow [\Phi^1] \downarrow \dots \downarrow [\Phi^n]$ y $\beta = [\Psi^0] \downarrow [\Psi^1] \downarrow \dots \downarrow [\Psi^m]$. Por construcción $dd(\alpha)$ y $dd(\beta) \leq h$ y entonces $dd(\langle \alpha = \beta \rangle) \leq h$. Es claro que por como lo construimos $(x, v) \in \llbracket \alpha \rrbracket^{\mathcal{T}}$ y $(x, w) \in \llbracket \beta \rrbracket^{\mathcal{T}}$ y por eso $\mathcal{T}, x \models \langle \alpha = \beta \rangle$. Llegamos entonces a que $\mathcal{T}', x' \not\models \langle \alpha = \beta \rangle$. Esto contradice que $\mathcal{T}, x \equiv_h^{\downarrow} \mathcal{T}', x'$ y así concluimos.

Supongamos ahora que $\mathcal{T}', x' \models \langle \alpha = \beta \rangle$, entonces hay $(v', w') \in P$ tal que $(x', v') \in \llbracket \alpha \rrbracket^{\mathcal{T}'}$ y $(x', w') \in \llbracket \beta \rrbracket^{\mathcal{T}'}$. En particular, i) y ii), son ciertos y tiene que ser que a) o b) valga.

En el primer caso tenemos por construcción que $(x', v') \notin \llbracket \alpha \rrbracket^{\mathcal{T}'}$. Y en el segundo caso es claro que $(x', w') \notin \llbracket \beta \rrbracket^{\mathcal{T}'}$.

En cualquier caso llegamos que a una contradicción y por lo tanto probamos lo que queríamos. ■

Todos estos resultados nos llevan a poder demostrar el siguiente teorema:

Teorema 29 1) $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$.

La recíproca también se cumple si \mathcal{T} y \mathcal{T}' son de ramificación finita.

2) $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u'$ si y sólo si $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$

Demostración

2) Es consecuencia directa de las proposiciones 27 y 28.

1) \Rightarrow) Esta implicación puede verse como consecuencia de la proposición 28. Ya que $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u' \forall l$, y por 28) sabemos que $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u' \forall l$ implica $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u' \forall l$ que a su vez implica $\mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$.

\Leftarrow) Esta implicación es similar a la de la proposición 28, pero definiendo la relación \mathcal{Z} como

$$x \mathcal{Z} x' \text{ si y sólo si } \mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$$

Además, la conjunción en (1) es finita porque \mathcal{T}' es finitamente generado y así P es finito. Observemos que el hecho de que \mathcal{T} sea finitamente generado es necesaria para probar que (Zag) se satisface. ■

3.2.3. Saturación

En el último teorema de la sección anterior se muestra que la recíproca del teorema 29 1) vale cuando los árboles son finitamente ramificados y no para cualquier árbol en general.

En esta sección queremos ampliar más esa hipótesis introduciendo una nueva noción, *Saturación*. Y mostraremos que la recíproca de ese teorema es válida para árboles de datos saturados.

Definición 27 Sea $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ tuplas de conjuntos de $XPath_{\downarrow}^{\perp}$ de expresiones de nodo. Dado un árbol de datos \mathcal{T} y $u \in T$ decimos que $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $=_{n,m}^{\downarrow}$ -**Satisfacibles** [respectivamente $\neq_{n,m}^{\downarrow}$ -**Satisfacibles**] en \mathcal{T}, u si existen $0 \rightarrow v_1 \dots \rightarrow v_n \in T$ y $w_0 \rightarrow w_1 \dots \rightarrow w_m \in T$ tales que $u = v_0 = w_0$ y

- 1) Para todo $i \in \{1, \dots, n\}$, $\mathcal{T}, v_i \models \Sigma_i$
- 2) Para todo $j \in \{1, \dots, m\}$, $\mathcal{T}, w_j \models \tau_j$
- 3) $data(v_n) = data(w_m)$ [resp. $data(v_n) \neq data(w_m)$]

Decimos que $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $=_{n,m}^{\downarrow}$ -**finitamente satisfacible** [resp. $\neq_{n,m}^{\downarrow}$ -**finitamente satisfacible**] en \mathcal{T}, u si para cada $\Sigma'_i \subseteq \Sigma_i$ y $\tau'_j \subseteq \tau_j$ finitos, tenemos que $\langle \Sigma'_1, \dots, \Sigma'_n \rangle$ y $\langle \tau'_1, \dots, \tau'_m \rangle$ son $=_{n,m}^{\downarrow}$ -**satisfacibles** [respectivamente $\neq_{n,m}^{\downarrow}$ -**satisfacibles**].

Definición 28 Decimos que un árbol de datos \mathcal{T} es \downarrow -**Saturado** si para cada $n, m \in \mathbb{N}$, cada par de tuplas $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ del conjunto de expresión de nodos de $XPath_{\downarrow}^{\perp}$, cada $u \in T$ y $* \in \{=, \neq\}$ vale lo siguiente:

Si $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $*_{n,m}^{\downarrow}$ -finitamente satisfacibles para \mathcal{T}, u entonces $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $*_{n,m}^{\downarrow}$ -satisfacibles para \mathcal{T}, u

Veamos la siguiente proposición para ver qué árboles son saturados y cómo nos puede ayudar esta nueva definición para ampliar la recíproca del teorema anterior.

Proposición 30 *Cualquier árbol de datos finitamente ramificado es \downarrow -saturado.*

Demostración

Procedamos por el absurdo. Supongamos que hay un $u \in T$ y tuplas $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ de conjuntos de nodos de expresión de $XPath_{\downarrow}^{\perp}$ que son $=_{n,m}^{\downarrow}$ -finitamente satisfacibles para \mathcal{T}, u pero que no son $=_{n,m}^{\downarrow}$ -satisfacibles para \mathcal{T}, u (el caso $\neq_{n,m}^{\downarrow}$ es análogo)

Sea

$$P = \{(v, w) \in T^2 / u \xrightarrow{n} v \wedge u \xrightarrow{m} w \text{ y } data(v) = data(w)\}$$

Observemos que P es finito pues \mathcal{T} es finitamente satisfacible. Es claro que si $(v, w) \in P$ entonces $u = v_0 \rightarrow v_1 \dots \rightarrow v_n = v \in T$ y $u = w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_m = w \in T$ y así pasa una de las dos opciones

1. existe $i \in \{1, \dots, n\}$ tal que $\mathcal{T}, v_i \not\models \Sigma_i$
2. existe $j \in \{1, \dots, m\}$ tal que $\mathcal{T}, w_j \not\models \tau_j$

Definamos conjuntos $(\Sigma_{i,v,w})_{1 \leq i \leq n}$ y $(\tau_{j,v,w})_{1 \leq j \leq m}$. Cada uno con a lo sumo un elemento como el siguiente:

Si vale 1), asumamos que i_0 es el menor número que cumple 1) y definamos $\Sigma_{i_0,v,w}$ como $\{\rho\}$ con ρ algún nodo de expresión $\rho \in \Sigma_{i_0}$ tal que $\mathcal{T}, v_{i_0} \not\models \rho$ y $\Sigma_{i,v,w} = \emptyset$ para $i \in \{1, \dots, n\} \setminus \{i_0\}$ y definimos $\tau_{j,v,w} = \emptyset \forall j$

Si no vale 1), entonces vale 2). Sea j_0 el menor número que hace valer 2), y definamos $\tau_{j_0,v,w} = \{\rho\}$ para ρ algún nodo de expresión $\rho \in \tau_{j_0}$ tal que $\mathcal{T}, w_{j_0} \not\models \rho$. Definimos $\tau_{j,v,w} = \emptyset$ para $j \in \{1, \dots, m\} \setminus \{j_0\}$ y $\Sigma_{i,v,w} = \emptyset \forall i$:

Finalmente definimos los conjuntos finitos

$$\Sigma'_i = \bigcup_{(v,w) \in P} \Sigma_{i,v,w} \quad \tau'_j = \bigcup_{(v,w) \in P} \tau_{j,v,w}$$

Por construcción tenemos $\Sigma'_i \subseteq \Sigma_i$, $\tau'_j \subseteq \tau_j$ y $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ no son $=_{n,m}^{\downarrow}$ -satisfacible para \mathcal{T}, u pero eso es una contradicción ya que son finitos y $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $=_{n,m}^{\downarrow}$ -finitamente satisfacible

■

El siguiente resultado es el que estábamos buscando, la reciproca del teorema de la sección anterior.

Proposición 31 *Sea $\mathcal{T}, \mathcal{T}'$ árboles de datos $=_{\downarrow}$ -saturados, y sea $u \in T$ y $u' \in T'$. Entonces*

$$\mathcal{T}, u \equiv_{\downarrow}^{\downarrow} \mathcal{T}', u' \Rightarrow \mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$$

Demostración

Mostraremos que \mathcal{Z} definida como

$$x \mathcal{Z} x' \iff \mathcal{T}, x \equiv_{\downarrow}^{\downarrow} \mathcal{T}', x'$$

es una \downarrow -bisimulación entre \mathcal{T}, u y \mathcal{T}', u'

Claramente $u \mathcal{Z} u'$, y (Harmony) vale.

Solo nos falta mostrar que (Zig) y (Zag) valen. Veamos (Zig) y (Zag) es análogo.

Supongamos que $x \mathcal{Z} x'$, $x = v_0 \rightarrow v_1 \dots \rightarrow v_n$ y $x = w_0 \rightarrow w_1 \dots \rightarrow w_m$ son caminos de \mathcal{T} y $\text{data}(w_m) = \text{data}(v_n)$ (el caso de $\text{data}(w_m) \neq \text{data}(v_n)$ es análogo)

Para $i \in \{1, \dots, n\}$ sea $\Sigma_i = Th_{\downarrow}(\mathcal{T}, v_i)$ y para $j \in \{1, \dots, m\}$ sea $\tau_j = Th_{\downarrow}(\mathcal{T}, w_j)$ donde $Th_{\downarrow}(\mathcal{T}, v)$ es el conjunto de todos los nodos de expresión verdaderos en \mathcal{T}, v

Además sea Σ'_i un subconjunto finito de Σ_i y $\tau'_j \subseteq \tau_j$ finito.

Definimos

$$\varphi = \langle \downarrow [\wedge \Sigma'_1] \downarrow \dots \downarrow [\wedge \Sigma'_n] = \downarrow [\wedge \tau'_1] \dots \downarrow [\wedge \tau'_m] \rangle$$

Es claro que $\mathcal{T}, x \models \varphi$ y por definición de \mathcal{Z} tenemos $\mathcal{T}, x \equiv^{\downarrow} \mathcal{T}', x'$, y entonces $\mathcal{T}', x' \models \varphi$. Luego $\langle \Sigma'_1, \dots, \Sigma'_n \rangle$ y $\langle \tau'_1, \dots, \tau'_m \rangle$ son $=^{\downarrow}_{n,m}$ -satisfacilbes en x' . Esto vale para cualquier conjunto finito $\Sigma'_i \subseteq \Sigma_i$ y $\tau'_j \subseteq \tau_j$, entonces $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $=^{\downarrow}_{n,m}$ -finitamente satisfacilbes.

Como \mathcal{T}' es \downarrow -saturado, entonces $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ son $=^{\downarrow}_{n,m}$ -satisfacilbes en \mathcal{T}', x' . y por lo tanto hay un camino $x' = v'_0 \rightarrow v'_1 \dots \rightarrow v'_n$ y $x' = w'_0 \rightarrow w'_1 \dots \rightarrow w'_m$ tal que

1. $\text{data}(w'_m) = \text{data}(v'_n)$
2. Para todo $1 \leq i \leq n$, $\mathcal{T}', v'_i \models \Sigma_i = Th_{\downarrow}(\mathcal{T}, v_i)$. Veamos que esto implica que $\mathcal{T}, v_i \equiv^{\downarrow} \mathcal{T}', v'_i$. Supongamos por el absurdo que $\mathcal{T}', v'_i \models \varphi$ pero $\mathcal{T}, v_i \not\models \varphi$, entonces $\mathcal{T}, v_i \models \neg\varphi$, luego $\mathcal{T}, v_i \models \varphi$ y entonces $\neg\varphi \in Th_{\downarrow}(\mathcal{T}, v_i)$ y entonces $\mathcal{T}', v'_i \models \neg\varphi$ lo que es absurdo. Por lo tanto tiene que ser $v_i \mathcal{Z} v'_i$
3. Para $1 \leq j \leq m$, $\mathcal{T}', w'_j \models \tau_j = Th_{\downarrow}(\mathcal{T}, w_j)$ o sea $\mathcal{T}, w_j \equiv_{\downarrow} \mathcal{T}', w'_j$ y por lo tanto $w_j \mathcal{Z} w'_j$.

Y así hemos visto que \mathcal{Z} es una \downarrow -bisimulación entre \mathcal{T}, u y \mathcal{T}', u' . ■

3.3. XPath_↓[↑]

En esta sección estudiaremos la bisimulación para el fragmento XPath_↓[↑]. Que consiste en agregarle la posibilidad de preguntar sobre los antecesores de un nodo, cosa que antes no podíamos. Introduciremos nuevos conceptos debido a que tenemos nuevas herramientas y nuevas fórmulas.

Empezaremos definiendo nociones similares a las de la sección anterior, como por ejemplo la noción de profundidad de un punto de evaluación dado. Ahora en XPath_↓[↑] tenemos una noción de distancia hacia abajo (r) y hacia arriba (s), por lo que podemos definir lo siguiente:

Definición 29 Llamaremos al par (r, s) la **Profundidad vertical** de una fórmula (notamos $Vd(\varphi)$). La **profundidad de anidación** de una fórmula φ es el máximo número de nidos [] que aparecen en φ (notamos $Nd(\varphi)$) Y se definen de la siguiente manera:

$$\begin{array}{ll}
Vd(a) = (0, 0), & Vd(\lambda) = (0, 0) \\
Vd(\varepsilon\alpha) = Vd(\alpha), & Vd(\varphi \wedge \psi) = \max\{Vd(\varphi), Vd(\psi)\} \\
Vd(\neg\varphi) = Vd(\varphi) & Vd([\varphi]\alpha) = \max\{Vd(\varphi)Vd(\alpha)\} \\
Vd(\langle\alpha\rangle) = Vd(\alpha) & Vd(\downarrow\alpha) = \max\{Vd(\alpha) + (-1, 1), (0, 0)\} \\
Vd(\langle\alpha \odot \beta\rangle) = \max\{Vd(\alpha), Vd(\beta)\} & Vd(\uparrow\alpha) = \max\{Vd(\alpha) + (-1, 1), (0, 0)\}
\end{array}$$

$$\begin{array}{ll}
Nd(a) = 0, & Nd(\varepsilon) = 0 \\
Nd(\alpha\beta) = \max\{Nd(\alpha), Nd(\beta)\}, & Nd(\varphi \wedge \psi) = \max\{Nd(\varphi), Nd(\psi)\} \\
Nd(\neg\varphi) = Nd(\varphi) & Nd([\varphi]) = Nd(\varphi) + 1 \\
Nd(\langle\alpha\rangle) = Nd(\alpha) & Nd(\downarrow) = 0 \\
Nd(\langle\alpha \odot \beta\rangle) = \max\{Nd(\alpha), Nd(\beta)\} & Nd(\uparrow) = 0
\end{array}$$

Con $a \in \mathbb{A} \odot \in \{=, \neq\}$, $+$ y \max se realizan por componentes y α es cualquier expresión de camino ó la cadena vacía λ .

Así podemos definir $(r, s, k) - XPath_{\underline{\downarrow}}^{\uparrow}$ como el conjunto de fórmulas φ en $XPath_{\underline{\downarrow}}^{\uparrow}$ con $Vd(\varphi) \leq (r, s)$ coordenada a coordenada y $Nv(\varphi) \leq k$.

Definición 30 Sea \mathcal{T}, u y \mathcal{T}', u' dos árboles de datos punteados. Decimos que \mathcal{T}, u y \mathcal{T}', u' son **equivalentes para $XPath_{\underline{\downarrow}}^{\uparrow}$** (notamos $\mathcal{T}, u \equiv_{\downarrow}^{\uparrow} \mathcal{T}', u'$) si y sólo si para toda $\varphi \in XPath_{\underline{\downarrow}}^{\uparrow}$ tenemos $\mathcal{T}, u \models \varphi$ si y sólo si $\mathcal{T}', u' \models \varphi$

Y decimos que \mathcal{T}, x y \mathcal{T}', x' son (r, s) -equivalentes para $XPath_{\underline{\downarrow}}^{\uparrow}$ (not $\mathcal{T}, x \equiv_{r, s}^{\downarrow, \uparrow} \mathcal{T}', x'$) si satisfacen las mismas expresiones de nodo φ de $XPath_{\underline{\downarrow}}^{\uparrow}$ y $Vd(\varphi) \leq (r, s)$. Respectivamente definimos (r, s, k) -equivalentes, y agregamos la condición que $Nd(\varphi) \leq k$ (notamos $\mathcal{T}, x \equiv_{r, s, k}^{\downarrow, \uparrow} \mathcal{T}', x'$)

Ahora definiremos la **Forma normal de $XPath_{\underline{\downarrow}}^{\uparrow}$** que será implícitamente usada en la definición de bisimulación de esta sección.

Para facilitar la notación definimos: para $n \geq 0$, sea \downarrow^n la concatenación de n símbolos \downarrow , es decir \downarrow^0 es la cadena vacía λ , $\downarrow^1 = \downarrow$, y $\downarrow^{n+1} = \downarrow^n \downarrow$ (lo mismo se define para \uparrow^n)

Definición 31 Una expresión de camino α de $XPath_{\underline{\downarrow}}^{\uparrow}$ se dice **descendiente ó hacia abajo** [resp **ascendente**] si es de la forma $\downarrow^n [\varphi]$ [resp $[\varphi] \uparrow^n$] para algún $n > 0$ con $\varphi \in XPath_{\underline{\downarrow}}^{\uparrow}$.

Una expresión arriba-abajo es una expresión de la forma $\varepsilon, \alpha^\uparrow, \alpha^\downarrow$ o $\alpha^\uparrow\alpha^\downarrow$ donde α^\uparrow es ascendiente y α^\downarrow es descendiente. De ahora en adelante usaremos $\alpha^\uparrow, \beta^\uparrow, \gamma^\uparrow$ para hablar de expresiones ascendientes y $\alpha^\downarrow, \beta^\downarrow, \gamma^\downarrow$ para denotar expresiones descendientes. Y $\alpha^{\uparrow\downarrow}$ para las que sean expresiones arriba-abajo. Obeservemos que cualquier expresión descendiente o ascendiente es en particular una expresión arriba-abajo.

Por ejemplo, $\downarrow [\langle \downarrow \rangle]$ es una expresión descendiente y $\downarrow [\langle \downarrow \rangle] \downarrow$ no.

Definición 32 Una fórmula de $XPath_{\downarrow}^{\uparrow}$ está en su **forma normal arriba-abajo** si todo camino de expresión contenido en ella es del tipo arriba-abajo y toda prueba de datos es de la forma $\langle \varepsilon \odot \alpha^{\uparrow\downarrow} \rangle$ con $\odot \in \{=, \neq\}$

Nuestra motivación para esta forma normal es simplificar la definición de bisimulación, tratando de hacer la su configuración lo más simple posible. De hecho para el fragmento de $Xpath_{\downarrow}^{\uparrow}$ no introdujimos esta noción, pero ya veremos que facilita bastante la definición en este caso.

Definición 33 Dado un camino de expresión α , la **navegación** de α (notamos $Nav(\alpha)$) es la cadena de $\{\downarrow, \uparrow\}$ que resulta de eliminar todas las expresiones de nodos $[\psi]$ y ε de α .

Por ejemplo $Nav(\downarrow [\langle \uparrow \rangle] \downarrow [\langle \downarrow = \uparrow \rangle] \uparrow [b]) = \downarrow \uparrow \downarrow$

Definición 34 Vamos a definir las siguientes equivalencias directas para transformar cualquier expresión de camino en uno equivalente en su forma arriba-abajo.

$$\varepsilon\gamma \equiv^{\uparrow\downarrow} \gamma \quad (\varepsilon)$$

$$\alpha\xi_{-n} \downarrow \dots \downarrow \psi_{-1} \downarrow \xi_0 \uparrow \varepsilon_1 \uparrow \dots \uparrow \varepsilon_n \beta \equiv^{\uparrow\downarrow} \alpha[\langle \xi_{-n}\xi_n \downarrow \dots \downarrow \xi_{-1}\xi_1 \downarrow \xi_0 \rangle]\beta \quad (\text{Factor})$$

$$\alpha[\psi_1][\psi_2]\beta \equiv^{\uparrow\downarrow} \alpha[\psi_1 \wedge \psi_2]\beta \quad (\text{Merge})$$

$$\alpha\xi_n \downarrow \xi_{n-1} \downarrow \dots \downarrow \xi_0 \equiv^{\uparrow\downarrow} \alpha \downarrow^n [\langle \xi_0 \uparrow \xi_1 \uparrow \dots \uparrow \xi_n \rangle] \quad (\text{Shift-l})$$

$$\xi_0 \uparrow \xi_1 \dots \uparrow \xi_n \beta \equiv^{\uparrow\downarrow} [\langle \xi_0 \uparrow \xi_1 \uparrow \dots \uparrow \xi_n \rangle] \uparrow^n \beta \quad (\text{shift-r})$$

Lema 32 Sea α una expresión de camino de $XPath_{\downarrow}^{\uparrow}$ con $Vd(\alpha) = (r, s)$ y $Nd(\alpha) = k$. Entonces hay una expresión $\alpha^{\uparrow\downarrow}$ que cumple:

$$1) \alpha^{\uparrow\downarrow} \equiv^{\uparrow\downarrow} \alpha$$

$$2) Vd(\alpha^{\uparrow\downarrow}) = (r, s)$$

$$3) Nd(\alpha^{\uparrow\downarrow}) = k + r + s + 1$$

Demostración

Primero aplicaremos la regla (factor) tantas veces como sea posible. Es claro que si $\text{Nav}(\alpha)$ es del tipo $\uparrow^n \downarrow^m$ para algún $n, m \geq 0$ entonces (factor) no puede ser aplicado y así terminamos.

Por lo tanto supongamos que $\text{Nav}(\alpha)$ tiene $\downarrow \uparrow$. Sea

$$\begin{aligned} \alpha &= \mathcal{T}_{\uparrow} \alpha_1 \mathcal{T}_{\downarrow} \\ \alpha_1 &= \mathcal{T}_1 \underbrace{\xi_{-n_1}^1 \downarrow \dots \downarrow \xi_0^1 \uparrow \dots \uparrow \xi_{n_1}^1}_{(factor)} \\ &\quad \mathcal{T}_2 \underbrace{\xi_{-n_2}^2 \downarrow \dots \downarrow \xi_0^2 \uparrow \dots \uparrow \xi_{n_2}^2}_{(factor)} \\ &\quad \vdots \\ &\quad \mathcal{T}_{m-1} \underbrace{\xi_{-n_m}^m \downarrow \dots \downarrow \xi_0^m \uparrow \dots \uparrow \xi_{n_m}^m}_{(factor)} \end{aligned}$$

con $\text{Nav}(\mathcal{T}_{\uparrow}), \text{Nav}(\mathcal{T}_m) \in \downarrow^*$, $\text{Nav}(\mathcal{T}_{\downarrow}), \text{Nav}(\mathcal{T}_1) \in \downarrow^*$, ξ_j^i son las cadenas vacías, ε o $[\varphi_1^{ij}][\varphi_2^{ij}] \dots [\varphi_{h_{ij}}^{ij}]$.

Además podemos asumir que m es maximal, ya que no podemos hacer (factor) en ningún \mathcal{T}_i y que el largo de cada \mathcal{T}_i es mínimo, o sea no puede ser que $\text{Nav}(\mathcal{T}_i)$ termine en \downarrow y $\text{Nav}(\mathcal{T}_{-i+1})$ empiece con \uparrow . Observemos que $\text{Nav}(\mathcal{T}_i) \in \uparrow^* \downarrow^*$.

Aplicamos la regla (factor) en los lugares marcados y obtendremos

$$\begin{aligned} \alpha &= \mathcal{T}_1 \underbrace{\langle \xi_{-n_1}^1 \xi_{n_1}^1 \downarrow \dots \downarrow \xi_{-1}^1 \xi_1^1 \downarrow \xi_0^1 \rangle}_{(factor) \text{ aplicado}} \\ &\quad \mathcal{T}_2 \underbrace{\langle \xi_{-n_2}^2 \downarrow \dots \downarrow \xi_{-1}^2 \xi_1^2 \downarrow \xi_0^2 \rangle}_{(factor) \text{ aplicado}} \\ &\quad \vdots \\ &\quad \mathcal{T}_{m-1} \underbrace{\langle \xi_{-n_m}^m \xi_{n_m}^m \downarrow \dots \downarrow \xi_{-1}^m \xi_1^m \downarrow \xi_0^m \rangle}_{(factor) \text{ aplicado}} \mathcal{T}_m \end{aligned}$$

Sea $\text{Vd}(\text{Nav}(\alpha_1)) = (r_1, s_1)$

Como $\text{Nav}(\alpha_2) = \text{Nav}(\mathcal{T}_{\uparrow} \alpha_1 \mathcal{T}_{\uparrow})$ contiene a $\downarrow \uparrow$ tenemos que $r_1 > 0$. Se ve que $\text{Vd}(\mathcal{T}_{\uparrow} \alpha_2 \mathcal{T}_{\downarrow}) = (r, s)$, $\text{Nd}(\alpha_2) \leq \text{Nd}(\alpha_1) + 1$, y $\text{Vd}(\text{Nav}(\alpha_2)) \leq (r_1 - 1, s_1)$. Si repetimos este procedimiento con α_2 y lo seguimos hasta que no podamos aplicar más la regla (factor), terminaremos en una expresión arriba-abajo α_f que cumple

- 1) $\alpha_f \equiv^{\uparrow} \alpha_1$
- 2) $\text{Vd}(\mathcal{T}_{\downarrow} \alpha_f \mathcal{T}_{\downarrow}) = (r, s)$

$$3) \text{Nd}(\alpha_f) \leq \text{Nd}(\alpha_1) + r_1$$

Después de aplicar (ε) y (merge) a $\mathcal{T}_{\uparrow}\alpha_f\mathcal{T}_{\downarrow}$ tantas veces como sea posible. Obtenemos un α' con la misma profundidad vertical y de anidación que $\mathcal{T}_{\uparrow}\alpha_f\mathcal{T}_{\downarrow}$ de la siguiente forma

$$\alpha' = \chi_1 \uparrow \chi_2 \uparrow \dots \uparrow \chi_n \downarrow \chi_{n+1} \downarrow \chi_{n+2} \downarrow \dots \downarrow \chi_{n+m}$$

donde χ_i es una cadena de la forma $[\varphi_i^1] \dots [\varphi_i^{n_i}]$ (posiblemente vacío). Ahora apliquemos (shift-l) y (shift-r) a α' para obtener una equivalencia α'' con la misma profundidad vertical (o sea $\text{Vd}(\alpha'') = \text{Vd}(\alpha') = (r, s)$) y con la misma profundidad de anidación que $\text{Nd}(\alpha') + 1$ de la forma

$$\alpha^{\uparrow\downarrow} = \chi' \uparrow^{n+m} \chi''$$

donde χ' y χ'' son la cadena vacía ó de la forma $[\varphi]$.

Observar que $\text{Nd}(\alpha^{\uparrow\downarrow}) = \text{Nd}(\alpha') + 1 \leq r + s + 1$ y entonces $\alpha^{\uparrow\downarrow}$ satisface todos los requisitos que queríamos para demostrar el lema. ■

Lema 33 Sean $\alpha^{\uparrow\downarrow}, \beta^{\uparrow\downarrow}$ expresiones de camino arriba-abajo. Y sea $\varphi = \langle \alpha^{\uparrow\downarrow} \odot \beta^{\uparrow\downarrow} \rangle$ ($\odot \in \{=, \neq\}$) con $\text{Vd}(\varphi) = (r, s)$ y $\text{Nd}(\varphi) = k$. Entonces hay una expresión de camino arriba-abajo $\mathcal{T}^{\uparrow\downarrow}$ tal que

$$1) \langle \mathcal{T}^{\uparrow\downarrow} \rangle \equiv^{\dagger} \varphi$$

$$2) \text{Vd}(\mathcal{T}^{\uparrow\downarrow}) = (r, s)$$

$$3) \text{Nd}(\mathcal{T}^{\uparrow\downarrow}) \leq k + 1$$

Demostración

Analicemos el caso donde $\alpha^{\uparrow\downarrow} = [\psi_{\alpha}] \uparrow^{n_{\alpha}} \downarrow^{m_{\alpha}} [\tau_{\alpha}]$ y $\beta^{\uparrow\downarrow} = [\psi_{\beta}] \uparrow^{n_{\beta}} \downarrow^{m_{\beta}} [\tau_{\beta}]$ donde $n_{\alpha} + m_{\alpha} > 0$, $n_{\beta} + m_{\beta} > 0$ y $\psi_{\alpha}, \psi_{\beta}, \tau_{\alpha}, \tau_{\beta}$ son formas normales arriba-abajo. (el caso $\alpha^{\uparrow\downarrow} = \varepsilon$ o $\beta^{\uparrow\downarrow} = \varepsilon$ es más simple)

Podemos suponer sin pérdida de generalidad que $n_{\alpha} \leq n_{\beta}$.

Por lo tanto tenemos $\langle \alpha^{\uparrow\downarrow} \odot \beta^{\uparrow\downarrow} \rangle \equiv^{\dagger} \langle \mathcal{T}^{\uparrow\downarrow} \rangle$ donde

$$\mathcal{T}^{\uparrow\downarrow} = [\psi_{\alpha} \wedge \psi_{\beta}] \uparrow^{n_{\alpha}} \downarrow^{m_{\alpha}} [\tau_{\alpha} \wedge \langle \varepsilon \odot \uparrow^{m_{\alpha}} \uparrow^{n_{\beta} - n_{\alpha}} \downarrow^{m_{\beta}} [\tau_{\beta}] \rangle]$$

Es claro que estas fórmulas son equivalentes. Veamos la figura 3.3.

Además, la fórmula de la derecha tiene a lo sumo un nido más que la de la izquierda y la profundidad vertical es a lo sumo (r, s) . Esto concluye la demostración. ■

Proposición 34 Sea $\varphi \in (r, s, k) - X\text{Path}_{\underline{\downarrow}}^{\uparrow}$, entonces hay $\varphi^{\uparrow\downarrow} \in X\text{Path}_{\underline{\downarrow}}^{\uparrow}$ en la forma normal arriba-abajo que hace valer lo siguiente:

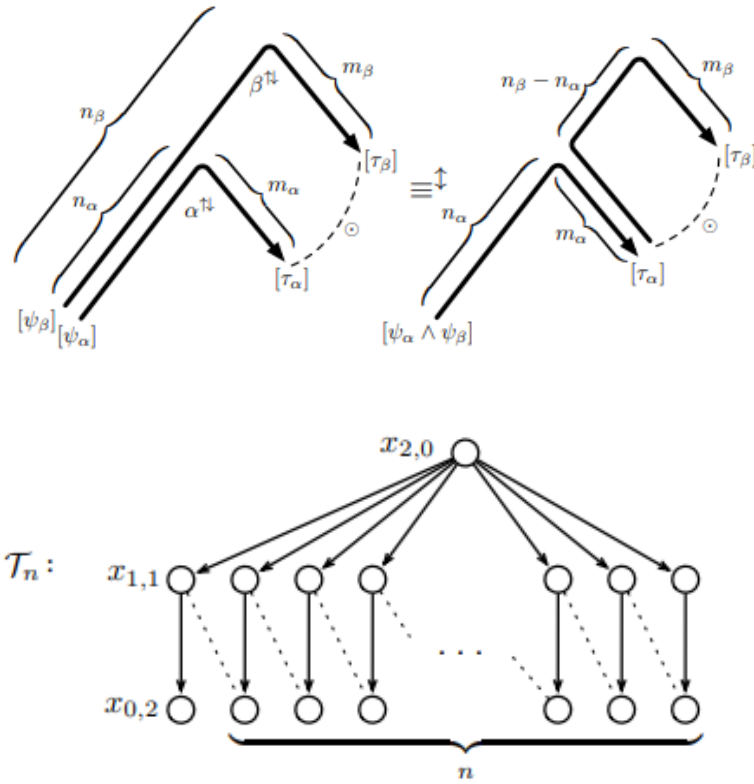


Figura 3.4: Modelo de verificación $\psi_j^i \forall i \geq n$, pero no se verifica para ψ_l con $l < n$. Las líneas punteadas representan valores de datos iguales.

- 1) $\varphi^{\downarrow\uparrow} \equiv \varphi$
- 2) $Vd(\varphi^{\downarrow\uparrow}) = (r, s)$
- 3) $Nd(\varphi^{\downarrow\uparrow}) \leq k(r + s + 2)$

Demostración En las expresiones de la definición 34 anteriores cada ξ_i es la cadena vacío ó de la forma ε o $[\varphi_1][\varphi_2] \dots [\varphi_n]$, α y β pueden ser expresiones de camino en la cadena vacía y \mathcal{T} es cualquier expresión de camino.

La idea es que (*factor*) convierte una expresión que va n veces hacia abajo y n veces hacia arriba en una expresión de nodo, y cuando hacemos esto cualquier prueba hecha en el i -ésimo nodo, al ir hacia abajo, se fusiona con la prueba $(n - i)$ -ésima cuando sube. Por ejemplo,

$$\downarrow [\neg a] \downarrow [c] \uparrow [\neg b] \uparrow \equiv^{\downarrow} [(\downarrow [\neg a][\neg b] \downarrow [c])]$$

Por otro lado (*shift - r*) y (*shift - l*) agrupan todos los nodos de prueba en el nodo de expresión más bajo, haciendo uso del hecho de que las relaciones de parientes son funcionales.

Así, por ejemplo

$$[a] \downarrow [b] \downarrow \equiv^{\downarrow} \downarrow \downarrow \langle \uparrow [b] \uparrow [a] \rangle y \uparrow [a] \uparrow [b] \equiv^{\downarrow} [(\uparrow [a] \uparrow [b])] \uparrow \uparrow$$

Es claro que las expresiones son equivalentes semánticamente. Por inducción en φ y usando los Lemas 32 y 33 podemos mostrar que hay un $\varphi^{\downarrow\uparrow}$ como queremos. ■

3.4. Bisimulación y (r, s, k) -Bisimulación

La ventaja de la forma normal introducida en la sección anterior es que hace posible una noción de bisimulación muy simple. La desventaja que tendremos es que como no preserva la profundidad de anidación, $\Leftrightarrow_{r,s,k}^{\downarrow}$ no corresponde precisamente con $\equiv_{r,s,k}^{\downarrow}$ aunque $\Leftrightarrow^{\downarrow}$ sí se corresponde con \equiv^{\downarrow} .

Sin embargo obtenemos que

$$\forall r, s, k \quad \Leftrightarrow_{r,s,k}^{\downarrow} \subseteq \equiv_{r,s,k}^{\downarrow} \subseteq \Leftrightarrow_{r,s,k(r+s+2)}^{\downarrow}$$

Definición 35 Sean \mathcal{T} y \mathcal{T}' dos árboles de datos. Decimos que $u \in \mathcal{T}$ y $u' \in \mathcal{T}'$ son **bisimilares para XPath₋** (notamos $\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u'$) si hay una relación $\mathcal{Z} \subseteq \mathcal{T} \times \mathcal{T}'$ tal que $u\mathcal{Z}u'$ y $\forall x \in \mathcal{T}, x' \in \mathcal{T}'$ tenemos

(Harmony) Si $x\mathcal{Z}x'$ entonces $etiqueta(x) = etiqueta(x')$

- (Zig) si $x \mathcal{Z} x'$, $y \xrightarrow{n} x$ y $y \xrightarrow{m} z$ entonces hay $y', z' \in T'$ tales que $y' \xrightarrow{n} x'$, $y' \xrightarrow{m} z'$, $z \mathcal{Z} z'$ y $\text{data}(z) = \text{data}(x)$ si y sólo si $\text{data}(x') = \text{data}(z')$
- (Zag) si $x \mathcal{Z} x'$, $y' \xrightarrow{n} x'$ y $y' \xrightarrow{m} z'$ entonces hay $y, z \in T$ tales que $y \xrightarrow{n} x$, $y \xrightarrow{m} z$, $z \mathcal{Z} z'$ y $\text{data}(z) = \text{data}(x)$ si y sólo si $\text{data}(x') = \text{data}(z')$

La figura 3.4 ilustra mejor la condición (Zig) y (Zag)

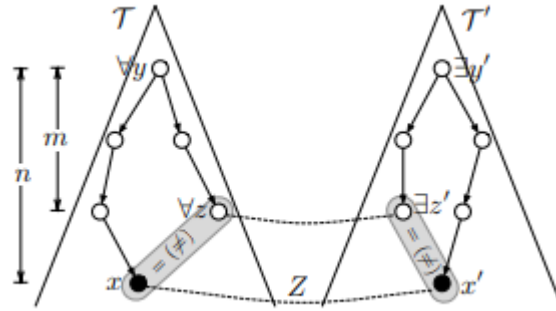


Figura 3.5: Condición (Zig) y (Zag)

Como ya mencionamos, la definición anterior se basa en gran medida en la forma normal de la proposición 34. En realidad la forma normal no es estrictamente necesaria para dar un noción de bisimulación en el fragmento vertical, y de hecho en el fragmento descendente no es necesaria ya que todos los caminos de expresión son esencialmente una repetición de una prueba de nodo y la relación "hijo". Por lo contrario en el fragmento vertical es de mucha ayuda. Para mostrar cuanto se facilita la definición vamos a ver como sería la condición (Zig) si no tenemos en cuenta la forma normal.

Si $x \mathcal{Z} x'$ y $n_1 \dots n_k$, $m_1 \dots m_k$, $\bar{n}_1 \dots \bar{n}_{\bar{k}}$ y $\bar{m}_1 \dots \bar{m}_{\bar{k}}$ son tales que

- $v_1^i \rightarrow \dots \rightarrow v_{n_i}^i$ en \mathcal{T} y $w_1^i \rightarrow \dots \rightarrow w_{m_i}^i$ en \mathcal{T} para $i \in \{1, \dots, k\}$
- $\tilde{v}_1^i \rightarrow \dots \rightarrow \tilde{v}_{\bar{n}_i}^i$ en \mathcal{T} y $\tilde{w}_1^i \rightarrow \dots \rightarrow \tilde{w}_{\bar{m}_i}^i$ en \mathcal{T} para $i \in \{1, \dots, \bar{k}\}$
- $v_1^i = w_1^i$ y $\tilde{v}_1^i = \tilde{w}_1^i$
- $w_{m_i}^i = v_{n_{i+1}}^{i+1}$ y $\tilde{w}_{\bar{m}_i}^i = \tilde{v}_{\bar{n}_{i+1}}^{i+1}$
- $x = v_{n_1}^1$ y $x = \tilde{v}_{\bar{n}_1}^1$

entonces hay $v_1^i, \dots, v_{n_i}^i$ en \mathcal{T}' , $w_1^i, \dots, w_{m_i}^i$ para $i \in \{1, \dots, k\}$ y $\tilde{v}_1^i, \dots, \tilde{v}_{\bar{n}_i}^i, \tilde{w}_1^i, \dots, \tilde{w}_{\bar{m}_i}^i$ en \mathcal{T} para $i \in \{1, \dots, \bar{k}\}$ tales que

- $v_1^i \rightarrow \dots \rightarrow v_{n_i}^i$ en \mathcal{T}' y $w_1^i \rightarrow \dots \rightarrow w_{m_i}^i$ en \mathcal{T}'

- $\tilde{v}_1^i \rightarrow \dots \rightarrow \tilde{v}_{\bar{n}_i}^i$ en \mathcal{T}' y $\tilde{w}_1^i \rightarrow \dots \rightarrow \tilde{w}_{\bar{m}_i}^i$ en \mathcal{T}'
- $v_1^i = w_1^i$ y $\tilde{v}_1^i = \tilde{w}_1^i$
- $w_{m_i}^i = v_{n_{i+1}}^{i+1}$ y $\tilde{w}_{\bar{m}_i}^i = \tilde{v}_{\bar{n}_{i+1}}^{i+1}$
- $x = v_{n_1}^1$ y $x = \tilde{v}_{\bar{n}_1}^1$
- $v_j^i \mathcal{Z} v_j^i, w_j^i \mathcal{Z} w_j^i, \tilde{v}_j^i \mathcal{Z} \tilde{v}_j^i, \tilde{w}_j^i \mathcal{Z} \tilde{w}_j^i$
- $data(w_{n_k}^k) = data(\tilde{w}_{\bar{n}_k}^k)$ si y sólo si $data(w_{n_k}^k) = data(\tilde{w}_{\bar{n}_k}^k)$

Intuitivamente esta definición establece que para todo camino p y \bar{p} que se mueva hacia arriba y abajo muchas veces en \mathcal{T} tiene que haber uno similar p' y \bar{p}' en \mathcal{T}' que también va hacia arriba y abajo muchas veces respetando la forma de p y \bar{p} respectivamente, de manera que el j -ésimo nodo de p está conectado al j -ésimo nodo de p' vía \mathcal{Z} y lo mismo para los nodos a lo largo de \bar{p} y \bar{p}' y tales que los valores de datos del último nodo de p y \bar{p} en \mathcal{T} son iguales si y solo si los valores de los últimos nodos de p' y \bar{p}' lo son.

Esta definición es la expresión natural de la bisimulación hacia abajo, pero en vez de considerar solo camino hacia abajo, se tiene en cuenta unos más generales.

Como pasa con la bisimulación hacia abajo todo nodo intermedio de \mathcal{T} tiene que estar relacionado con un correspondiente nodo en \mathcal{T}' . Como podemos ver, claramente, esta definición es más larga y hay que chequear más condiciones.

Trabajar con la forma normal nos permite tener una definición de bisimulación mucho más simple que tiene dos ventajas principales

- a) Hay solo un camino en cada árbol de datos
- b) No necesitamos nodos intermedios que están relacionados vía \mathcal{Z} .

Estas dos características son consecuencia directa de la introducir la forma normal vertical, ya que a) solo compara valores con la raíz, así hay una sola expresión de camino no vacía en cada expresión de nodo de "diamante". Y b) solo hace pruebas al principio y al final de cada camino y no en los nodos intermedios. Sin embargo observemos que gracias a la forma normal, ambas definiciones definen la misma relación de bisimulación.

Definición 36 Decimos que $u \in T$ y $u' \in T'$ son **(r,s,k)-bisimilares** para $Xpath_{\perp}^{\uparrow}$ (notamos $\mathcal{T}, u \stackrel{\uparrow}{\rightleftharpoons}_{r,s,k} \mathcal{T}', u'$) si hay una familia de relaciones $(\mathcal{Z}_{\bar{r},\bar{s}}^{\bar{k}})_{\bar{r}+\bar{s} \leq r+s, \bar{k} \leq k}$ en $T \times T'$ tales que $u \mathcal{Z}_{\bar{r},\bar{s}}^{\bar{k}} u'$ y $\forall \bar{r} + \bar{s} \leq r + s, \bar{k} \leq k, x \in T$ y $x' \in T'$ tenemos las siguientes condiciones

(Harmony) Si $x \mathcal{Z}_{\bar{r},\bar{s}}^{\bar{k}} x'$ entonces $etiqueta(x) = etiqueta(x')$.

(Zig) Si $x \mathcal{Z}_{\bar{r},\bar{s}}^{\bar{k}} x', y \xrightarrow{n} x, y \xrightarrow{m} x'$ con $n \leq \bar{s}$ y $m \leq \bar{r} + n$ entonces hay $y', z' \in T'$ tales que $y' \xrightarrow{n} x'$ y $y' \xrightarrow{m} z'$ y vale que

- 1) $data(z) = data(x) \iff data(x') = data(z')$
- 2) Si $\bar{k} > 0$, $z \mathcal{Z}_{\bar{r}', \bar{s}'}^{\bar{k}} z'$ para $\bar{r}' = \bar{r} + n - m$, $\bar{s}' = \bar{s} - n + m$

(Zag) Si $x \mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}} x', y' \xrightarrow{n} x', y' \xrightarrow{m} z'$ con $n \leq \bar{s}$ y $m \leq \bar{r} + n$ entonces hay $y, z \in T$ tales que $y \xrightarrow{n} x$ y $y \xrightarrow{m} z$ y el item 1) y 2) se cumplen.

Observemos que de la definición se desprende que si $x \mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}} x', y \xrightarrow{n} x, y' \xrightarrow{m} x'$ se sigue que $y \mathcal{Z}_{\bar{r}', \bar{s}'}^{\bar{k}-1} y'$ para $\bar{r}' = \bar{r} + n$, $\bar{s}' = \bar{s} - n$. Y lo mismo ocurre con \mathcal{Z} en lugar de $\mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}}$ en el caso de bisimulación.

Definición 37 Sea \mathcal{T} un árbol de datos y sea $u \in T$.

Denotamos con $\mathcal{T} \upharpoonright_r^s u$ al **subárbol** de \mathcal{T} inducido por

$$\{v \in T \mid (\exists m \leq s)(\exists n \leq r + m)(\exists w \in T), w \stackrel{m}{u} \wedge w \stackrel{n}{v}\}$$

Así es claro que

$$\mathcal{T}, u \stackrel{\dagger}{\equiv}_{r,s,k} (\mathcal{T} \upharpoonright_r^s u), u$$

3.4.1. Equivalencia y bisimulación

Así como vimos para $XPath^\perp$ veremos ahora que $\stackrel{\dagger}{\equiv}$ coincide con $\stackrel{\dagger}{\equiv}$ en árboles de datos con ramificaciones finitas y veremos como $\stackrel{\dagger}{\equiv}_{r,s,k}$ se relaciona con $\stackrel{\dagger}{\equiv}_{r,s,k}$

Proposición 35 $\mathcal{T}, u \stackrel{\dagger}{\equiv}_{r,s,k(r+s+2)} \mathcal{T}', u'$ implica $\mathcal{T}, u \stackrel{\dagger}{\equiv}_{r,s,k} \mathcal{T}', u'$

Demostración Mostraremos que si $\mathcal{T}, u \stackrel{\dagger}{\equiv}_{r,s,k} \mathcal{T}', u'$ vía $(\mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}})_{\bar{r} + \bar{s} \leq r + s, \bar{k} \leq k}$ entonces para todo $n \leq \bar{s}$ y $m \leq \bar{r} + n$, para toda φ en su forma vertical normal con $vd(\varphi) \leq (\bar{r}, \bar{s})$, $nd(\varphi) \leq \bar{k}$ para toda expresión hacia arriba α^\uparrow en la forma normal vertical y para toda expresión hacia abajo α^\downarrow en la forma normal vertical con $vd(\alpha^\uparrow), vd(\alpha^\downarrow) \leq (\bar{r}, \bar{s})$, $nd(\alpha^\uparrow), nd(\alpha^\downarrow) \leq \bar{k}$ vale

- 1) Si $x \mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}} x' \Rightarrow \mathcal{T}, x \models \varphi$ si y sólo si $\mathcal{T}', x' \models \varphi$
- 2) Si $y \xrightarrow{n} x, y' \xrightarrow{n} x', x \mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}} x' \Rightarrow (x, y) \in \llbracket \alpha^\uparrow \rrbracket^{\mathcal{T}}$ si y sólo si $(x', y') \in \llbracket \alpha^\uparrow \rrbracket^{\mathcal{T}'}$
- 3) Si $y \xrightarrow{m} z, y' \xrightarrow{m} z', z \mathcal{Z}_{\bar{r}', \bar{s}'}^{\bar{k}-1} z'$ con $\bar{r}' = \bar{r} + n - m$, $\bar{s}' = \bar{s} - n + m \Rightarrow (y, z) \in \llbracket \alpha^\downarrow \rrbracket^{\mathcal{T}}$ si y sólo si $(y', z') \in \llbracket \alpha^\downarrow \rrbracket^{\mathcal{T}'}$

Así usando los resultados de la Proposición 34 obtenemos lo que deseamos.

Mostraremos simultáneamente los ítems 1, 2 y 3 por inducción en $|\varphi| + |\alpha^\uparrow| + |\alpha^\downarrow|$ (Veamos 1).

El caso base es $\varphi = a$ para algún $a \in \mathbb{A}$. Por (Harmony), $Etiqueta(x) = Etiqueta(x')$ y entonces $\mathcal{T}, x \models \varphi$ si y sólo si $\mathcal{T}', x' \models \varphi$.

Los casos $\varphi = \neg\psi$ y $\varphi = \gamma \wedge \psi$ son sencillos

Supongamos ahora que $\varphi = \langle \varepsilon = \alpha^\uparrow \alpha^\downarrow \rangle$. Mostraremos que $\mathcal{T}, x \models \varphi \Rightarrow \mathcal{T}', x' \models \varphi$. Entonces asumamos que $\mathcal{T}, x \models \varphi$. Supongamos que hay $y, z \in \mathcal{T}$ y $n \leq \bar{s}$, $m \leq \bar{r} + n$ tales que $y \xrightarrow{n} x$, $y \xrightarrow{m} z$, $(x, y) \in \llbracket \alpha^\uparrow \rrbracket^{\mathcal{T}}$, $(y, z) \in \llbracket \alpha^\downarrow \rrbracket^{\mathcal{T}}$ y $data(x) = data(z)$. Por (Zig), hay $y', z' \in \mathcal{T}'$ tales que $z \mathcal{Z}_{\bar{r}'\bar{s}'}^{k-1} z'$ con $\bar{r}' = \bar{r} + n - m$, $\bar{s}' = \bar{s} - n + m$ y $data(x') = data(z')$. Así por hipótesis inductiva 2) y 3) tenemos $(x', y') \in \llbracket \alpha^\uparrow \rrbracket^{\mathcal{T}'}$ y $(y', z') \in \llbracket \alpha^\downarrow \rrbracket^{\mathcal{T}'}$. Luego $\mathcal{T}', x' \models \varphi$. La implicación $\mathcal{T}', x' \models \varphi \Rightarrow \mathcal{T}, x \models \varphi$ es análoga.

El caso $\varphi = \langle \varepsilon \neq \alpha^\uparrow \rangle$ y $\varphi = \langle \varepsilon \odot \alpha^\uparrow \rangle$, $\varphi = \langle \varepsilon \odot \alpha^\downarrow \rangle$ con $\odot \in \{=, \neq\}$ y $\varphi = \langle \alpha \rangle$ (para α en su forma normal vertical) se demuestra de forma similar.

El caso $\varphi = \langle \varepsilon \odot \varepsilon \rangle$ ($\odot \in \{=, \neq\}$) es trivial.

Analicemos ahora el ítem 2).

Sea $\varphi^\uparrow = [\psi]^\uparrow^n$ ($n \geq 0$) y sea $x_0, \dots, x_n \in T$ y $x'_0, \dots, x'_n \in T'$ tales que

$$y = x_0 \rightarrow x_1 \dots \rightarrow x_n = x \text{ en } \mathcal{T}$$

$$y' = x'_0 \rightarrow x'_1 \dots \rightarrow x'_n = x' \text{ en } \mathcal{T}' \quad y \quad x \mathcal{Z}_{\bar{r}\bar{s}}^{k-1} x'$$

Por la observación del final de la sección anterior tenemos $x_0 \mathcal{Z}_{\bar{r}'\bar{s}'}^{k-1} x'_0$ con $\bar{r}' = \bar{r} + n$, $\bar{s}' = \bar{s} - n$. Procedamos por el absurdo, es decir pensemos que $(x', y') \notin \llbracket \alpha^\uparrow \rrbracket^{\mathcal{T}'}$. Esto significa que necesariamente $\mathcal{T}, x_0 \models \psi$ pero $\mathcal{T}', x'_0 \not\models \psi$. Pero ψ es una subfórmula de α^\downarrow y $nd(\psi) \leq k - 1$ y $nd(\psi) \leq (\bar{r}', \bar{s}')$ y esto contradice la hipótesis inductiva 1. Por lo que es absurdo y luego demostramos lo que queríamos. ■

Proposición 36 $\mathcal{T}, u \equiv_{r,s,k}^\uparrow \mathcal{T}', u' \Rightarrow \mathcal{T}, u \equiv_{r,s,k}^\downarrow \mathcal{T}', u'$

Demostración Tomemos $u \in T$, $u' \in T'$ tales que $\mathcal{T}, u \equiv_{r,s,k}^\uparrow \mathcal{T}', u'$. Definamos $(\mathcal{Z}_{\bar{r},\bar{s}}^k)_{\bar{r}+\bar{s} \leq r+s, \bar{k} \leq k}$ por

$$x \mathcal{Z}_{\bar{r},\bar{s}}^k x' \text{ si y solo si } \mathcal{T}, x \equiv_{\bar{r},\bar{s},\bar{k}}^\downarrow \mathcal{T}', x'$$

Mostraremos que $\mathcal{Z}_{r,s}^k$ es una (r, s, k) -bisimulación entre \mathcal{T}, u y \mathcal{T}', u' .

Por hipótesis $u \mathcal{Z}_{r,s}^k u'$. Ahora tomamos \bar{r}, \bar{s} y \bar{k} tales que $\bar{r} + \bar{s} \leq s + r$, $\bar{k} \leq k$. Por construcción $\mathcal{Z}_{\bar{r},\bar{s}}^k$ satisface (Harmony). Veamos que $\mathcal{Z}_{\bar{r},\bar{s}}^k$ satisface (Zig), el caso (Zag) es análogo.

Supongamos que $x \mathcal{Z}_{\bar{r},\bar{s}}^k x'$ y

$$y = x_0 \rightarrow x_1 \dots \rightarrow x_n = x \text{ en } \mathcal{T}$$

$$y = z_0 \rightarrow z_1 \dots \rightarrow z_m = z \text{ en } \mathcal{T}$$

y $data(x) = data(z)$ (el caso $data(x) \neq data(z)$ se demuestra análogamente) donde $m \leq \bar{r} + n$. Sea $P \subseteq T'^2$ definido como

$$P = \{(y', x') / y' \xrightarrow{n} x' \wedge y' \xrightarrow{m} z' \wedge data(x') = data(z')\}$$

Como $\mathcal{T}, x \equiv_{\bar{r}, \bar{s}, \bar{k}}^{\uparrow} \mathcal{T}', x'$, $vd(\langle \varepsilon = \uparrow^n \downarrow^m \rangle \leq (r, s))$, $nd(\langle \varepsilon = \uparrow^n \downarrow^m \rangle) = 0$, $\mathcal{T}, x \models \langle \varepsilon = \uparrow^n \downarrow^m \rangle$ podemos concluir que $P \neq \emptyset$

Lo próximo es mostrar que existe $(y', z') \in P$ tal que

- i) $y' = x_0' \rightarrow x_1' \rightarrow \dots \rightarrow x_n = x'$ en \mathcal{T}'
- ii) $y' = z_0' \rightarrow z_1' \rightarrow \dots \rightarrow z_m = z'$ en \mathcal{T}'
- iii) $\mathcal{T}, x \equiv_{\bar{r}, \bar{s}, \bar{k}-1}^{\uparrow} \mathcal{T}', x'$
- iv) $\mathcal{T}, z \equiv_{\bar{r}', \bar{s}', \bar{k}'-1}^{\uparrow} \mathcal{T}', x'$ donde $\bar{T}' = \bar{T} + n - m$, $\bar{s}' = \bar{s} - n + m$

Pues así, por hipótesis inductiva, (Zig) es satisfacible por $\mathcal{Z}_{\bar{r}, \bar{s}}^{\bar{k}}$. Veamos que existe (y', z') como queremos. Procedemos por el absurdo. Supongamos que para todo $(y', z') \in P$ que satisface i) y ii) se tiene cualquiera de los siguientes casos:

- (a) $\mathcal{T}, x \not\equiv_{\bar{r}, \bar{s}, \bar{k}-1}^{\uparrow} \mathcal{T}', x'$ ó
- (b) $\mathcal{T}, z \not\equiv_{\bar{r}', \bar{s}', \bar{k}'-1}^{\uparrow} \mathcal{T}', z'$ con $\bar{r}' = \bar{r} + n - m$, $\bar{s}' = \bar{s} - n + m$

Tomo \top cualquier tautología tal que $vd(\top) = (0, 0)$ y $nd(\top) = 0$.

Para cada $(y', z') \in P$ definimos la expresión de nodo $\varphi_{y', z'}$ y $\psi_{y', z'}$ que satisfacen $vd(\varphi_{y', z'}) \leq (\bar{r}, \bar{s})$, $nd(\varphi_{y', z'}) = \bar{k}$ y $vd(\psi_{y', z'}) \leq (\bar{r}', \bar{s}')$, $nd(\psi_{y', z'}) \leq \bar{k}$. Ahora bien, supongamos que vale (a). Sea $\varphi_{y', z'}$ tal que $vd(\varphi_{y', z'}) \leq (\bar{r}, \bar{s})$, $nd(\varphi_{y', z'}) < \bar{k}$ y tal que $\mathcal{T}, x \models \varphi_{y', z'}$ pero $\mathcal{T}', x' \not\models \varphi_{y', z'}$ y sea $\psi_{y', z'} = \top$

Sea

$$\Phi = \bigwedge_{(y', z') \in P} \varphi_{y', z'} \quad y \quad \Psi = \bigwedge_{(y', z') \in P} \psi_{y', z'} \quad (2)$$

Como $vd(\varphi_{y', z'}) \leq (\bar{r}, \bar{s})$, $nd(\varphi_{y', z'}) < \bar{k}$ solo puede haber finitas formulas no equivalentes $\varphi_{y', z'}$, y lo mismos para $\psi_{y', z'}$. Entonces tenemos que las dos conjunciones en (2) son equivalentes a una finita. Y desde ahora podemos asumir sin pérdida de generalidad que Φ y Ψ son fórmulas bien definidas.

Finalmente sea $\alpha^\uparrow = [\Phi] \uparrow^n$ y $\alpha^\downarrow = \downarrow^m [\Psi]$.

Por construcción $vd(\alpha^\uparrow \alpha^\downarrow) \leq (\bar{r}, \bar{s})$, $nd(\alpha^\uparrow \alpha^\downarrow) \leq \bar{k}$. Más aún $\mathcal{T}, x \models \langle \varepsilon = \alpha^\rightarrow \alpha^\downarrow \rangle$ y $\mathcal{T}', x' \models \langle \varepsilon = \alpha^\uparrow \alpha^\downarrow \rangle$ pero esto contradice el hecho de que $\mathcal{T}, x \equiv_{\bar{r}, \bar{s}, \bar{k}}^{\uparrow} \mathcal{T}', x'$, por lo que es absurdo y esto concluye nuestra demostración. ■

Teorema 37 1) $\mathcal{T}, u \equiv^{\uparrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$. La otra implicación solo vale cuando \mathcal{T}' y \mathcal{T} son de ramificación finita.

2) $\mathcal{T}, u \equiv_{r, s, k(r+s+2)}^{\uparrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv_{r, s, k}^{\downarrow} \mathcal{T}', u'$

3) $\mathcal{T}, u \equiv_{r,s,k}^{\downarrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \Leftrightarrow_{r,s,k}^{\downarrow} \mathcal{T}', u'$

Demostración Para esta demostración vamos a usar las Proposiciones 35 y 36. Los ítems 2) y 3) se demuestran en 35 y 36.

El argumento de la implicación (\Rightarrow) del ítem 1) se puede ver como consecuencia del ítem 2). En efecto, $\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u'$ implica $\mathcal{T}, u_{r,s,k}^{\downarrow} \mathcal{T}', u' \forall r, s$. Que a su vez, por el ítem 2) implica $\mathcal{T}, u \equiv_{r,s}^{\downarrow} \mathcal{T}', u' \forall r, s$ que a su vez implica $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$.

El argumento de (\Leftarrow) del ítem 1) es similar al de la Proposición 36, pero trabajando con una sola relación \mathcal{Z} en vez de $(\mathcal{Z}_{\bar{r},\bar{s}}^{\bar{k}})_{\bar{r},\bar{s},\bar{k}}$.

Para la implicación inversa, definimos \mathcal{Z} por $x\mathcal{Z}x'$ si y solo si $\mathcal{T}, x \equiv^{\downarrow} \mathcal{T}', x'$. Los conjuntos en (2) son finitos porque \mathcal{T}' tiene ramificación finita y así P es finito (el hecho de que \mathcal{T} es finitamente ramificado se usa para probar que (Zag) se satisface) ■

3.4.2. Saturación

Definición 38 Dado un árbol de datos \mathcal{T} y $u \in T$ decimos que el conjunto de expresiones de nodos R de $XPath_{\downarrow}^{\downarrow}$ es $\equiv_{n,m}^{\downarrow}$ -satisfacible [respectivamente $\neq_{n,m}^{\downarrow}$] para \mathcal{T}, u si hay $v, w \in T$ tales que $v \xrightarrow{n} u$, $w \xrightarrow{m} u$, $w \models \mathcal{T}$ y $data(v) = data(w)$ [resp \neq].

Decimos que R es $\equiv_{n,m}^{\downarrow}$ finitamente satisfacible [resp $\neq_{n,m}^{\downarrow}$] para \mathcal{T}, u si para cada $R' \subset R$ finito, tenemos que R' es $\equiv_{n,m}^{\downarrow}$ -satisfacible [resp $\neq_{n,m}^{\downarrow}$] para \mathcal{T}, u

Definición 39 Decimos que un árbol de datos \mathcal{T} es \downarrow -Saturado si para cada $n, m \in \mathbb{N}$, cada par de tuplas $\langle \Sigma_1, \dots, \Sigma_n \rangle$ y $\langle \tau_1, \dots, \tau_m \rangle$ del conjunto de expresión de nodos de $XPath_{\downarrow}^{\downarrow}$, cada $u \in T$ y $* \in \{=, \neq\}$ vale lo siguiente:

Si R es $*-finitamente$ satisfacible en \mathcal{T}, u entonces R es $*_{n,m} \downarrow$ -satisfacible en \mathcal{T}, u

Proposición 38 Sean \mathcal{T} y \mathcal{T}' árboles de datos \downarrow -saturados y sea $u \in T$ y $u' \in T'$. Si $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$ entonces $\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u'$.

Demostración Sea $\mathcal{Z} \subset T \times T'$ definida como

$$x\mathcal{Z}x' \text{ si y solo si } \mathcal{T}, x \equiv^{\downarrow} \mathcal{T}', x'$$

Demostraremos que \mathcal{Z} es una \downarrow -bisimulación entre \mathcal{T}, u y \mathcal{T}', u' .

Claramente $u\mathcal{Z}u'$ y es claro que vale (Harmony).

Solo veremos que vale (Zig) y (Zag) es análogo.

Supongamos que $x\mathcal{Z}x'$, $y \xrightarrow{n} x$, $y \xrightarrow{m} z$ en \mathcal{T} con $data(x) = data(z)$ (el caso \neq es análogo). Sea $R = Th_{\downarrow}(\mathcal{T}, z)$ y sea R' un conjunto finito de R .

Definimos

$$\varphi = \langle \varepsilon = \uparrow^n \downarrow^m [\wedge R'] \rangle$$

Es claro que $\mathcal{T}, x \models \varphi$ y que por definición $\mathcal{T}, x \equiv^\downarrow \mathcal{T}', x'$ (pues $x \mathcal{Z} x'$) entonces $\mathcal{T}', x' \models \varphi$. Por lo tanto \mathcal{T}' es $=_{n,m}^\downarrow$ -satisfacible en x' . Esto vale para cualquier conjunto finito $R' \subset R$ y entonces R es $=_{n,m}^\downarrow$ -finitamente satisfacible en x' .

Por lo tanto, como \mathcal{T}' es \downarrow -saturado entonces R es $=_{n,m}^\downarrow$ -satisfacible en x' y entonces hay $y' \xrightarrow{n} x'$ y $y' \xrightarrow{m} z'$ en \mathcal{T}' tales que $data(x') = data(z')$ (análogamente para \neq) y $\mathcal{T}', z' \models R = Th_{\downarrow}(\mathcal{T}, z)$, es decir, $\mathcal{T}', z' \equiv^\downarrow \mathcal{T}, z$ y por definición de \mathcal{Z} tenemos que $z \mathcal{Z} z'$ y así se verifica (Zig). En conclusión son \downarrow -bisimilares que es lo que queríamos demostrar. ■

3.5. Simulación

En esta sección veremos la noción de Simulación directa (no simétrica) de $XPath_{\perp}^{\downarrow}$ y $XPath_{\perp}^{\downarrow}$. Obtendremos resultados similares al Teorema 29 y 37 pero relacionando cada noción de simulación con la correspondiente implicación lógica.

Definición 40 Decimos que una fórmula de $XPath_{\perp}$ es **Positiva** si no hay ningún tipo de negación (\neg) ni desigualdades ($\langle \alpha \neq \beta \rangle$).

Para $\mathcal{L} \in XPath_{\perp}$ escribimos \mathcal{L}^+ para el fragmento positivo de \mathcal{L} . Análogamente notamos \mathcal{L}^+ al fragmento positivo de \mathcal{L} en $XPath_{\perp}^{\downarrow}$. Y llamaremos $XPath_{\perp}^{\downarrow+}$ al fragmento de $XPath_{\perp}^{\downarrow}$ que tiene como lenguaje a \mathcal{L}^+ (respectivamente para $XPath_{\perp}^{\downarrow+}$)

Una simulación de $XPath_{\perp}^{\downarrow}$ [respectivamente $XPath_{\perp}^{\downarrow+}$] es simplemente una bisimulación donde la condición (Zag) y mitad de la primera parte de (Zig) son omitidas. Por eso decimos que las simulaciones no son necesariamente simétricas. Formalmente decimos:

Definición 41 Dados \mathcal{T} y \mathcal{T}' árboles de datos en $XPath_{\perp}^{\downarrow}$. Decimos que $u \in T$ es **similar** a $u' \in T'$ (notamos $\mathcal{T}, u \xrightarrow{\downarrow} \mathcal{T}', u'$) si y sólo si hay una relación $\mathcal{Z} \in T \times T'$ tal que $u \mathcal{Z} u'$ y $\forall x \in T$ y $x' \in T'$ tenemos

(Harmony) Si $x \mathcal{Z} x'$ entonces $etiqueta(x) = etiqueta(x')$

(Zag) Si $x \mathcal{Z} x'$ $x \xrightarrow{n} v$, $x \xrightarrow{m} w$ entonces hay $v', w' \in T'$ tales que $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$ y

$$1) data(v) = data(w) \Rightarrow data(v') = data(w')$$

$$2) (\xrightarrow{i} v) \mathcal{Z} (\xrightarrow{i} v') \quad \forall 0 \leq i < n$$

$$3) (\xrightarrow{i} w) \mathcal{Z} (\xrightarrow{i} w') \quad \forall 0 \leq i < m$$

Definición 42 Dados \mathcal{T} y \mathcal{T}' árboles de datos en $XPath_{\perp}^{\downarrow}$. Decimos que $u \in T$ es **similar** a $u' \in T'$ (notamos $\mathcal{T}, u \xrightarrow{\downarrow} \mathcal{T}', u'$) si y sólo si hay una relación $\mathcal{Z} \in T \times T'$ tal que $u \mathcal{Z} u'$ y $\forall x \in T$ y $x' \in T'$ tenemos

(Harmony) Si xZx' entonces $etiqueta(x) = etiqueta(x')$

(Zag) Si xZx' , $y \xrightarrow{n} x$, $y \xrightarrow{m} x'$ entonces hay $y', z' \in T'$ tales que $y' \xrightarrow{n} x'$, $y' \xrightarrow{m} z'$, zZz' y si $data(z) = data(x) \Rightarrow data(z') = data(x')$

Las relaciones $\xrightarrow{\downarrow}_l$ y $\xrightarrow{\uparrow}_{r,s,k}$ son definidas acordemente. Y definimos las implicaciones lógicas "de ida" (no simétricas) como sigue; escribimos $\mathcal{T}, u \Rightarrow^{\downarrow} \mathcal{T}', u'$ para

$$(\forall \varphi \in XPath_{\underline{=}}^{\downarrow+}) [\mathcal{T}, u \models \varphi \Rightarrow \mathcal{T}', u' \models \varphi]$$

$\Rightarrow^{\downarrow}_l$, \Rightarrow^{\uparrow} y $\Rightarrow^{\uparrow}_{r,s,k}$ se definen de forma análoga para l -XPath₌^{↓+}, XPath₌^{↑+} y (r, s, k) -XPath₌^{↑+} respectivamente.

Como en el caso de bisimulaciones veremos algunos casos donde $\xrightarrow{\downarrow}$ coincide con \Rightarrow

Teorema 39 1) Sea $\dagger \in \{\downarrow, \uparrow\}$, $\mathcal{T}, u \xrightarrow{\dagger} \mathcal{T}', u'$ implica $\mathcal{T}, u \Rightarrow^{\dagger} \mathcal{T}', u'$. La otra implicación vale cuando \mathcal{T}' es finitamente ramificada.

2) $\mathcal{T}, u \xrightarrow{\downarrow}_l \mathcal{T}', u'$ si y sólo si $\mathcal{T}, u \Rightarrow^{\downarrow}_l \mathcal{T}', u'$

3) $\mathcal{T}, u \xrightarrow{\uparrow}_{r,s,k(r+s+2)} \mathcal{T}', u'$ implica $\mathcal{T}, u \Rightarrow^{\uparrow}_{r,s,k} \mathcal{T}', u'$

4) $\mathcal{T}, u \Rightarrow^{\uparrow}_{r,s,k} \mathcal{T}', u'$ implica $\mathcal{T}, u \xrightarrow{\uparrow}_{r,s,k} \mathcal{T}', u'$

Demostración Esta demostración es una consecuencia inmediata de una adaptación de las demostraciones de las proposición 27, 28 y 35, 36, la cual no hablaremos, pero si hablaremos de la parte del "si" de la adaptación de la prueba de las proposiciones 27 y 36. Para adaptarlo definiremos la simulación como:

$$xZ_i x' \text{ si y solo si } \mathcal{T}, x \Rightarrow^{\downarrow}_i \mathcal{T}', x'$$

$$xZ_{\bar{r}, \bar{s}}^k x' \text{ si y solo si } \mathcal{T}, x \Rightarrow^{\uparrow}_{\bar{r}, \bar{s}, k} \mathcal{T}', x'$$

Respectivamente, y las condiciones (a) y (b) definidos en la demostración de la proposición 28 se convierten en:

(a) $[\exists i \in \{0, \dots, n\} \exists \varphi \in XPath_{\underline{=}}^{\downarrow+}] (dd(\varphi) \leq h - i \wedge \mathcal{T}, v_i \models \varphi \wedge \mathcal{T}', v'_i \not\models \varphi)$

(b) $[\exists j \in \{0, \dots, m\} \exists \varphi \in XPath_{\underline{=}}^{\downarrow+}] (dd(\varphi) \leq h - j \wedge \mathcal{T}, w_j \models \varphi \wedge \mathcal{T}', w'_j \not\models \varphi)$

y

(a) $[\exists i \in \{0, \dots, n\} \exists \varphi \in XPath_{\underline{=}}^{\uparrow+}] (vd(\varphi) \leq (\bar{r} + i, \bar{s} - i) \wedge nd(\varphi) \leq k - 1 \wedge \mathcal{T}, v_i \models \varphi \wedge \mathcal{T}', v'_i \not\models \varphi)$

- (b) $[\exists j \in \{0, \dots, m\} \exists \varphi \in XPath_{\underline{=}}^{\uparrow+}](vd(\varphi) \leq (\bar{r} + j', \bar{s} - j') \text{ con } j' = n - m + j \wedge nd(\varphi) \leq k - 1 \wedge \mathcal{T}, w_j \models \varphi \wedge \mathcal{T}', w_{j'} \not\models \varphi)$

■

Definición 43 Decimos que el conjunto de árboles punteados R es **definible** si hay una expresión de nodo φ que es cumplida en un árbol punteado \mathcal{T}, u si y solo si \mathcal{T} pertenece a R .

Definición 44 Decimos que \mathcal{T}' es una **subestructura** de \mathcal{T} si \mathcal{T}' es un árbol de datos que resulta de sacarle algunos nodos a \mathcal{T} , es decir $T' \subset T$ y $\forall u, v \in T'$ tenemos

1. $u \rightarrow v$ en \mathcal{T} si y solo si $u \rightarrow v$ en \mathcal{T}'
2. $etiqueta(u)$ en \mathcal{T}' es igual a $etiqueta(u)$ en \mathcal{T}
3. $data(u)$ en \mathcal{T}' es igual a $data(u)$ en \mathcal{T}

Equivalentemente, visto como una σ -estructura, \mathcal{T}' es una σ -estructura de \mathcal{T} inducida por $t' \subset T$. Podemos verificar que la identidad en T' es una simulación de $XPath_{\underline{=}}^{\downarrow}$ de \mathcal{T}' a \mathcal{T} .

Lema 40 Si \mathcal{T}' es una subestructura de \mathcal{T} y $u' \in T'$ entonces $\mathcal{T}', u' \xrightarrow{\downarrow} \mathcal{T}, u$

Lema 41 1. $\{\mathcal{T}', u' / \mathcal{T}, u \xrightarrow{\downarrow} \mathcal{T}', u'\}$ es definible por una expresión de nodo $\chi_{l, u, \mathcal{T}}^+$ de $XPath_{\underline{=}}^{\downarrow+}$ con profundidad hacia abajo $\leq l$

2. $\{\mathcal{T}', u' / \mathcal{T}, u \xrightarrow{\downarrow}_{r, s, k} \mathcal{T}', u'\}$ es definible por una expresión de nodo $\chi_{r, s, k, u, \mathcal{T}}^+$ de $XPath_{\underline{=}}^{\downarrow+}$ con profundidad vertical $\leq (r, s)$ y profundidad de anidación $\leq k$

Este Lema nos dice que las expresiones de nodo de $XPath_{\underline{=}}$ invariantes bajo simulaciones son precisamente las positivos. (ver [9]) Esto nos lleva al siguiente teorema.

Teorema 42 1) $\varphi \in XPath_{\underline{=}}^{\downarrow}$ es $\xrightarrow{\downarrow}$ -invariante (respectivamente $\xrightarrow{\downarrow}$) si y sólo si es equivalente a una expresión de nodo de $XPath_{\underline{=}}^{\downarrow+}$ (resp l - $XPath_{\underline{=}}^{\downarrow+}$)

- 2) $\varphi \in XPath_{\underline{=}}^{\downarrow}$ es $\xrightarrow{\downarrow}$ -invariante si y sólo si es equivalente a una expresión de nodo de $XPath_{\underline{=}}^{\downarrow+}$
- 3) Si $\varphi \in XPath_{\underline{=}}^{\downarrow}$ es $\xrightarrow{\downarrow}_{r, s, k}$ -invariante entonces es equivalente a una expresión de nodo de (r, s, k) - $XPath_{\underline{=}}^{\downarrow+}$
- 4) Si $\varphi \in XPath_{\underline{=}}^{\downarrow}$ es equivalente a una expresión de nodo de (r, s, k) - $XPath_{\underline{=}}^{\downarrow+}$ entonces φ es $\xrightarrow{\downarrow}_{r, s, k'}$ -invariante para $k' = k(r + s + 2)$.

Gracias a estos resultados se puede pensar en una caracterización de XPath_↓ como un fragmento de la lógica de primer orden basada en la \Leftrightarrow -invarianza de las formulas.

Teorema 43 (Caracterización) *Sea $\varphi(x)$ una fórmula de primer orden. Son equivalentes:*

- 1 φ es \Leftrightarrow -invariante sobre arboles de datos (finitos)
- 2 φ es lógicamente equivalente sobre arboles de datos (finitos) a una expresión de nodo de l -XPath_↓ donde $l = 2^{qr(\varphi)} - 1$

También podemos pensar en XPath_↓ donde vale algo un poco más debil, cualquier fórmula de primer orden $\Leftrightarrow_{rsk}^{\uparrow}$ -invariante para algún r, s, k es equivalente a una fórmula de XPath_↓. Más precisamente:

Proposición 44 *Sea $k' = k(r + s + 2)$. Si $\varphi(x)$, una fórmula de primer orden, es $\Leftrightarrow_{rsk'}^{\uparrow}$ -invariante sobre árboles de datos (finitos), entonces hay $\psi \in (r, s, k)$ -XPath_↓ tal que $Tr_x(\psi)$ es lógicamente equivalente a φ sobre árboles de datos (finitos).*

Ambas demostraciones se pueden ver en [9]

Así, con estos últimos resultados nos podemos dar una mejor idea de cual es el fragmento de la lógica de primer orden que representa a XPath_↓ y a XPath_↓ respectivamente.

Conclusión

Como hemos visto a lo largo de esta tesis, hay numerosas nociones de comparación de modelos en cada lógica, y no es tan sencillo llegar a una noción de equivalencia de modelos. A modo de resumen enumeraremos las distintas instancias trabajadas.

Lógica de Primer orden

- **Isomorfismos**

- Notación $\mathcal{A} \cong \mathcal{B}$

- Si i es un isomorfismo entre \mathcal{A} y \mathcal{B} entonces, para toda V valuación y φ fórmula tenemos:

$$\mathcal{A}, V \models \varphi \text{ si y solo si } \mathcal{B}, i \circ V \models \varphi$$

- **Isomorfismo Potencial**

- Si existe un isomorfismo potencial \mathcal{F} entre \mathcal{A} y \mathcal{B} , entonces, para toda fórmula de primer orden φ

$$\mathcal{A} \models \varphi \Leftrightarrow \mathcal{B} \models \varphi$$

- Si i es un isomorfismo entre \mathcal{A} y \mathcal{B} entonces también resulta un isomorfismo potencial entre \mathcal{A} y \mathcal{B} .

- No podemos distinguir si dos modelos que coinciden en sus sentencias son isomorfos.

- **Isomorfismo de las ultrapotencias**

- Teorema de isomorfismo de Keisler-Shelah: \mathcal{M} y \mathcal{N} coinciden en toda sentencia de primer orden si y sólo si existe un isomorfismo entre sus ultrapotencias.

- Los isomorfismos de las ultrapotencias garantizan la equivalencia de modelos.

Lógica Modal Basica

- **Homomorfismo**

-Nos da una relación debil. Los homomorfismos solo preservan las relaciones en un solo sentido.

-Las fórmulas modales no son invariantes bajo homomorfismos.

- **Homomorfismo fuerte**

Si resulta biyectivo (isomorfismo) lo notamos $\mathcal{M} \cong \mathcal{N}$

-Nos da una relación muy fuerte. Hay muchos morfismos que dan lugar a la invarianza, pero que no califican como homomorfismos fuertes.

- **p -morfismo**

-Si es suryectivo lo notamos $\mathcal{M} \rightarrow \mathcal{N}$

-Conserva la idea de invarianza pero más relajada.

- **Bisimulación**

-Notamos $\mathcal{M}, w \Leftrightarrow \mathcal{N}, w'$

-Son una generalización relacional de los morfismos acotados, eliminando la direccionalidad desde el dominio al codominio (y con ello la condición homomórfica) y reemplazandola con un sistema de movimientos de combinación entre modelos.

-Si \mathcal{M}, w y \mathcal{N}, w' son bisimilares, entonces para toda fórmula φ de la lógica modal básica se tiene

$$\mathcal{M}, w \models \varphi \text{ si y sólo si } \mathcal{N}, w' \models \varphi$$

XPath₌[↓]

- **Equivalencia para XPath₌[↓]**

-Notación $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$, o $\mathcal{T}, u \Rightarrow^{\downarrow} \mathcal{T}', u'$ para equivalencias no simetricas.

- **Bisimulación**

-Notación $\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u'$.

- **l -bisimulación**

-Notación $\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}'.u'$

- **Simulación**

-Notación $\mathcal{T}, u \Rightarrow^{\downarrow} \mathcal{T}', u'$

- **Resultados**

$\mathcal{T}, u \Leftrightarrow^{\downarrow} \mathcal{T}', u' \Rightarrow \mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u' \forall l$

$\mathcal{T}, u \Leftrightarrow_l^{\downarrow} \mathcal{T}', u' \Leftrightarrow \mathcal{T}, u \equiv_l^{\downarrow} \mathcal{T}', u'$

- $\mathcal{T}, u \Leftrightarrow^\downarrow \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv^\downarrow \mathcal{T}', u'$. La reciproca también se cumple si \mathcal{T} y \mathcal{T}' son de ramificación finita.

- $\mathcal{T}, u \equiv^\downarrow \mathcal{T}', u' \Rightarrow \mathcal{T}, u \Leftrightarrow^\downarrow \mathcal{T}', u'$ si son árboles de datos $=^\downarrow$ -saturados.

- $\mathcal{T}, u \xrightarrow{\downarrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \Rightarrow^\downarrow \mathcal{T}', u'$. La otra implicación vale cuando \mathcal{T}' es finitamente ramificada.

- $\mathcal{T}, u \xrightarrow{\downarrow} \mathcal{T}', u'$ si y sólo si $\mathcal{T}, u \Rightarrow_i^\downarrow \mathcal{T}', u'$

XPath \updownarrow

■ Equivalencia para XPath \updownarrow

-Notación $\mathcal{T}, u \equiv^\updownarrow \mathcal{T}', u'$ y $\mathcal{T}, u \Rightarrow^\updownarrow \mathcal{T}', u'$ para equivalencias no simétricas.

■ Bisimulación

-Notación $\mathcal{T}, u \Leftrightarrow^\updownarrow \mathcal{T}', u'$

■ (r,s,k)-Bisimulación

-Notación $\mathcal{T}, u \Leftrightarrow_{r,s,k}^\updownarrow \mathcal{T}', u'$

■ Simulación

-Notación $\mathcal{T}, u \xrightarrow{\updownarrow} \mathcal{T}', u'$

■ Resultados

- $\mathcal{T}, u \Leftrightarrow_{r,s,k(r+s+2)}^\updownarrow \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv_{r,s,k}^\updownarrow \mathcal{T}', u'$

- $\mathcal{T}, u \equiv_{r,s,k}^\updownarrow \mathcal{T}', u' \Rightarrow \mathcal{T}, u \Leftrightarrow_{r,s,k}^\updownarrow \mathcal{T}', u'$.

- $\mathcal{T}, u \Leftrightarrow^\updownarrow \mathcal{T}', u'$ implica $\mathcal{T}, u \equiv^\updownarrow \mathcal{T}', u'$. La otra implicación solo vale cuando \mathcal{T}' y \mathcal{T} son de ramificación finita.

- $\mathcal{T}, u \equiv^\updownarrow \mathcal{T}', u'$ entonces $\mathcal{T}, u \Leftrightarrow^\updownarrow \mathcal{T}', u'$ si son árboles de datos \updownarrow -saturados.

- $\mathcal{T}, u \xrightarrow{\updownarrow} \mathcal{T}', u'$ implica $\mathcal{T}, u \Rightarrow^\updownarrow \mathcal{T}', u'$. La otra implicación vale cuando \mathcal{T}' es finitamente ramificada.

- $\mathcal{T}, u \xrightarrow{\updownarrow}_{r,s,k(r+s+2)} \mathcal{T}', u'$ implica $\mathcal{T}, u \Rightarrow_{r,s,k}^\updownarrow \mathcal{T}', u'$

- $\mathcal{T}, u \Rightarrow_{r,s,k}^\updownarrow \mathcal{T}', u'$ implica $\mathcal{T}, u \xrightarrow{\updownarrow}_{r,s,k} \mathcal{T}', u'$

Apéndice

Juegos de Ehrenfeucht - Fraïssé

El juego de Ehrenfeucht - Fraïssé nos brinda buenas herramientas para describir la expresibilidad para lógicas sobre modelos finitos. Aunque, al menos para la lógica de primer orden, puede ser aplicada en modelos finitos e infinitos, pero para modelos infinitos hay muchas herramientas más poderosas. Tanto es así que en muchos libros apenas es mencionado en el caso infinito pero es un punto central en el caso finito.

La idea de este juego, en el caso de la lógica de primer orden, es:

Hay dos jugadores, llamados **Spoiler** y **Duplicador** (ó simplemente jugador 1 y jugador 2)

El tablero del juego consiste en dos estructuras \mathbb{A} y \mathbb{B} .

El objetivo de Spoiler es mostrar que las dos estructuras son diferentes y el de Duplicador es mostrar que son la misma. (Acá es donde entra en juego las definiciones de misma y diferente)

En el juego clásico de Ehrenfeucht - Fraïssé los jugadores juegan una cierta cantidad de rondas, y cada ronda consiste en los siguientes pasos:

- 1) Spoiler elige una estructura; \mathbb{A} ó \mathbb{B}
- 2) Spoiler hace una movida cuando elige un elemento de esa estructura $a \in \mathbb{A}$ ó $b \in \mathbb{B}$
- 3) Duplicador responde eligiendo un elemento en la estructura que Spoiler no haya elegido.

La figura 3.5 hace referencia a esta dinámica. Los movimientos de Spoiler se muestran con un círculo relleno y los de Duplicador con uno vacío. En la primer ronda Spoiler elige \mathbb{B} y selecciona $b_1 \in \mathbb{B}$, Duplicador responde con $a_1 \in \mathbb{A}$. En la siguiente ronda Spoiler cambia de estructura y elige \mathbb{A} y selecciona $a_2 \in \mathbb{A}$, por lo que Duplicador responde con $b_2 \in \mathbb{B}$. En la tercer ronda Spoiler juega $b_3 \in \mathbb{B}$ y Duplicador responde $a_3 \in \mathbb{A}$

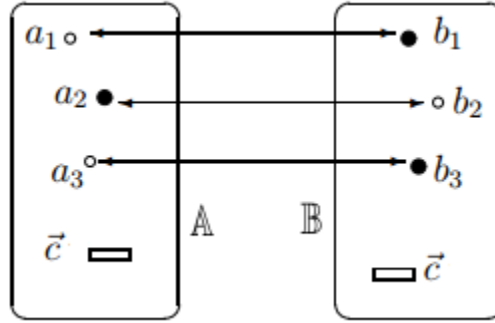


Figura 6: Juego de Ehrenfeucht - Fraïssé

Como es un juego, tiene que haber un ganador. Es aquí cuando tenemos que formalizar que significa "diferentes y mismas". Para eso vamos a necesitar la definición de isomorfismo potencial que vimos en la definición 4 del capítulo uno, pero más general.

Definición 45 Sean \mathbb{A} y \mathbb{B} dos σ -estructuras y sean $\vec{a} = (a_1, \dots, a_n)$, $\vec{b} = (b_1, \dots, b_n)$ tuplas en \mathbb{A} y \mathbb{B} respectivamente. Entonces (\vec{a}, \vec{b}) define un **isomorfismo potencial** entre \mathbb{A} y \mathbb{B} si las siguientes condiciones valen:

- Para todo $i, j \leq n$

$$a_i = a_j \text{ sii } b_i = b_j$$

- Para todo símbolo constante $c \in \sigma$ y todo $i \leq n$

$$a_i = c^{\mathbb{A}} \text{ sii } b_i = c^{\mathbb{B}}$$

- Para todo símbolo de relación P de σ y toda secuencia (i_1, \dots, i_k) de números de $[1, \dots, n]$ (no necesariamente distintos)

$$(a_{i_1}, \dots, a_{i_k}) \in P^{\mathbb{A}} \text{ sii } (b_{i_1}, \dots, b_{i_k}) \in P^{\mathbb{B}}$$

En ausencia de símbolos constantes, esta definición dice que el mapeo $a_i \mapsto b_i$, $i \leq n$ es un isomorfismo entre las subestructuras de \mathbb{A} y \mathbb{B} generadas por $\{a_1, \dots, a_n\}$ y $\{b_1, \dots, b_n\}$ respectivamente.

Así con esta definición, después de n rondas del juego de Ehrenfeucht - Fraïssé, tenemos las movidas (a_1, \dots, a_n) y (b_1, \dots, b_n) . Si c_1, \dots, c_l son símbolos constantes en σ entonces $\vec{c}_{\mathbb{A}}$ denota $(c_1^{\mathbb{A}}, \dots, c_l^{\mathbb{A}})$ y lo mismo con $\vec{c}^{\mathbb{B}}$.

Decimos que (\vec{a}, \vec{b}) es una posición ganadora para Duplicador si $((\vec{a}, \vec{c}^{\mathbb{A}}), (\vec{b}, \vec{c}^{\mathbb{B}}))$ es un isomorfismo potencial entre \mathbb{A} y \mathbb{B} . En otras palabras, el mapeo que manda cada a_i en b_i y cada $c_i^{\mathbb{A}}$ en $c_i^{\mathbb{B}}$ es un isomorfismo entre las estructuras \mathbb{A} y \mathbb{B} generado por

$\{a_1, \dots, a_n, c_1^{\mathbb{A}}, \dots, c_l^{\mathbb{A}}\}$ y $\{b_1, \dots, b_n, c_1^{\mathbb{B}}, \dots, c_l^{\mathbb{B}}\}$ respectivamente. Entonces decimos que Duplicador tiene una estrategia ganadora de n -rondas en el juego de Ehrenfeucht - Fraïssé en \mathbb{A} y \mathbb{B} si Duplicador juega de una forma que garantiza estar en una posición ganadora después de n rondas, sin importar como juegue Spoiler. De cualquier otra manera es Spoiler quien tiene una estrategia ganadora en n rondas. En otras palabras que gane Duplicador significa que hay un isomorfismo potencial entre \mathbb{A} y \mathbb{B} . Mas coloquialmente que son la “misma” estructura para la lógica de primer orden.

Por lo que si Duplicador tiene una estrategia ganadora en n rondas escribimos $\mathbb{A} \equiv_n \mathbb{B}$. Observemos que $\mathbb{A} \equiv_n \mathbb{B}$ implica $\mathbb{A} \equiv_k \mathbb{B} \forall i \leq n$

Ejemplo 45 (Estrategias ganadoras)

- *Juegos en conjuntos: En este ejemplo, el vocabulario σ es vacío. o sea que la estructura es solo un conjunto. Sea $|\mathbb{A}|, |\mathbb{B}| \geq n$. Entonces $\mathbb{A} \equiv_n \mathbb{B}$.*

La estrategia ganadora de Duplicador es la siguiente:

Supongamos que ya fueron jugadas i rondas, y que la posición del juego es $((a_1, \dots, a_i)(b_1, \dots, b_i))$. Asumamos que Spoiler elige $a_{i+1} \in \mathbb{A}$. Si $a_{i+1} = a_j$ para algún $a_j \leq i$ entonces Duplicador responde con $b_{i+1} = b_j$, de otra manera Duplicador responde con cualquier otro $b_{i+1} \in \mathbb{B} - \{b_1, \dots, b_i\}$ (que existe pues $|\mathbb{B}| \geq n$)

- *Juego en orden lineal: Este ejemplo es un poco más complejo ya que le agregamos una relación binaria $<$ a σ , la cual será interpretada como un orden lineal. o sea la respuesta que tiene que dar Duplicador tiene que preservar el orden.*

Supongamos que L_1 y L_2 son dos órdenes lineales de tamaño por lo menos n (es decir, estructuras de la forma $\langle \{1, \dots, m\}, < \rangle$ $m \geq n$).

¿Quién ganará? ¿Será $L_1 \equiv_n L_2$? Veamos qué pasa con el caso $n = 2$.

Sea l_1 tal que contiene 3 elementos (digamos $\{1, 2, 3\}$) y L_2 de 2 elementos ($\{1, 2\}$). En la primer movida, Spoiler juega $3 \in L_1$, Duplicador responde $1 \in L_2$, luego Spoiler juega $1 \in L_1$ entonces Duplicador no tiene otra alternativa más que jugar $2 \in L_2$ pero $2 > 1$, por lo que pierde.

Si Duplicador elige $2 \in L_2$ en vez de $1 \in L_2$ en la primer ronda, luego Spoiler puede elegir $3 \in L_1$ y Duplicador pierde nuevamente ya que $1 < 3$ pero $2 > 1$.

Por lo tanto $L_1 \not\equiv_2 L_2$. Por lo que Spoiler es quien tiene una estrategia ganadora en este caso.

Sin embargo Duplicador puede llegar a tener una estrategia ganadora si L_1 y L_2 son mucho más largos que la cantidad de rondas. Más precisamente, dado $n > 0$, si L_1 y L_2 son ordenes lineales de largo por lo menos 2^n entonces $L_1 \equiv_n L_2$. Ver [14].

Con la idea de este juego se puede demostrar el siguiente teorema:

Teorema 46 (Ehrenfeucht - Fraïssé) Sean \mathbb{A} y \mathbb{B} dos estructuras en un vocabulario relacional.

Son equivalentes:

- 1) \mathbb{A} y \mathbb{B} coinciden en las fórmulas de primer orden de cuantificador de rango k .
- 2) $\mathbb{A} \equiv_k \mathbb{B}$

Tal como mencionábamos al principio, los juegos de Ehrenfeucht–Fraïssé nos brindan una técnica para determinar si dos estructuras son equivalentes. La aplicación principal de esta técnica es para probar la inexpresibilidad de ciertas propiedades en lógica de primer orden. De hecho, los juegos de Ehrenfeucht–Fraïssé proporcionan una metodología completa para probar resultados de inexpresibilidad en la lógica de primer orden. Por eso, estos juegos son de especial importancia en la teoría de modelos finitos y sus aplicaciones en informática, ya que los juegos de Ehrenfeucht–Fraïssé son una de las pocas técnicas de teoría de modelos que mantiene su validez en el contexto de modelos finitos[7].

Bibliografía

- [1] Sergio Abriola, María Emilia Descotte, and Santiago Figueira. Model theory of XPath on data trees. Part II: Binary bisimulation and definability. 2017.
- [2] Carlos Areces. *Lógicas Modales*. <https://cs.famaf.unc.edu.ar/hoffmann/lmc15/lm12/t3.pdf>, 2012.
- [3] Dalila Berd. *Implementación de XPath en Haskell*. https://eprints.ucm.es/11300/1/Memoria_Dalila_Berd.pdf, 2010.
- [4] Vitaly Bergelson. *Applications of Ultrafilters and Ultraproducts*. American Mathematical Soc., 2010.
- [5] Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal Logic: Graph*. Cambridge University Press, 2001.
- [6] Patrick Blackburn, Johan FAK van Benthem, and Frank Wolter. *Handbook of modal logic*. Elsevier, 2006.
- [7] Uwe Bosse. An “Ehrenfeucht-Fraïssé game” for fixpoint logic and stratified fixpoint logic. 1992.
- [8] Calixto Badesa Cortés, Ignacio Jané Palau, and Ramón Jansana Ferrer. *Elementos de lógica formal*. Ariel, 2000.
- [9] Diego Figueira, Santiago Figueira, and Carlos Areces. Model theory of XPath on data trees. Part I: Bisimulation and characterization. 2015.
- [10] Diego Federico Figueira. *Bisimulaciones en Neighbourhood Semantics*. PhD thesis, Universidad de Buenos Aires, 2006.
- [11] Josep Maria Font, Ramon Jansana, and Don Pigozzi. A survey of abstract algebraic logic. *Studia Logica*, 2003.
- [12] Cecelia Higgins. Ultrafilters in the set theory. 2018.

- [13] George Edward Hughes, Max J Cresswell, and Mary Meyerhoff Cresswell. *A new introduction to modal logic*. Psychology Press, 1996.
- [14] Leonid Libkin. *Elements of finite model theory*. Springer Science & Business Media, 2013.
- [15] Pedro Arturo Góngora Luna. *Lógicas no clásicas: Introducción a la lógica modal*. 2010.
- [16] David Marker. *Model theory: an introduction*. Springer Science & Business Media, 2006.
- [17] Martin Otto. Bisimulation invariance and finite models. In *Logic Colloquium*, 2006.
- [18] Claudia Pons, Ricardo Rosenfeld, and Clara Patricia Smith. *Lógica para informática*. 2017.
- [19] Eric Rosen. *Modal logic over finite structures*. 1997.