1. Introducción

En nuestra experiencia, los buenos algoritmos son descubiertos al hacer un análisis formal y/o matemático del problema a resolver. En este caso nos proponemos el desarrollo de algoritmos para resolver un problema básico de la teoría de modelos finitos: decidir definibilidad de relaciones en el lenguaje de primer orden. La motivación inicial es la explotación de resultados teóricos recientes que caracterizan la definibilidad en términos semánticos presentados en Campercholi y Vaggione 2015.

2. Teoremas de definibilidad

En esta sección presentamos resultados que caracterizan la definibilidad, en diferentes fragmentos de primer orden, de relaciones sobre familias de estructuras. Estas caracterizaciones reducen la pregunta de si una relación es definible a la pregunta de si dicha relación es preservada por una colección de morfismos que depende del fragmento de primer orden considerado. Por ejemplo, ... (cf. referencia cruzada)...

Estos resultados, expuestos originalmente en Campercholi y Vaggione 2015, constituyen el punto de partida para nuestro estudio del chequeo computacional de la definibilidad. Los algoritmos desarrollados pueden encontrarse en las secciones siguientes.

2.1. Preliminares.

Sean:

$$\mathrm{Fo}\left(\mathcal{L}\right) = \{\varphi : \varphi \text{ es una } \mathcal{L}\text{-fórmula}\}$$

$$\mathrm{Op}\left(\mathcal{L}\right) = \{\varphi : \varphi \text{ es una } \mathcal{L}\text{-fórmula abierta}\}$$

$$\mathrm{Op}^{+}\left(\mathcal{L}\right) = \{\varphi : \varphi \text{ es una } \mathcal{L}\text{-fórmula abierta positiva}\}$$

$$\mathrm{E}\left(\mathcal{L}\right) = \{\varphi : \varphi \text{ es una } \mathcal{L}\text{-fórmula existencial}\}$$

$$\mathrm{E}^{+}\left(\mathcal{L}\right) = \{\varphi : \varphi \text{ es una } \mathcal{L}\text{-fórmula existencial positiva}\}$$

$$\pm\mathrm{At}\left(\mathcal{L}\right) = \mathrm{At}\left(\mathcal{L}\right) \cup \{\neg\alpha : \alpha \in \mathrm{At}\left(\mathcal{L}\right)\}$$

$$\mathrm{At}\left(\mathcal{L}\right) = \{\mathcal{L}\text{-fórmulas atómicas}\}$$

Una formula es abierta si no tiene ocurrencias de $\forall$, $\exists$. Es positiva si no tiene ocurrencias de $\neg$, $\to$, $\leftrightarrow$. Una formula es existencial si es de la forma $\exists\bar{x}\,\varphi\left(\bar{x},\bar{y}\right)$ con $\varphi\left(\bar{x},\bar{y}\right)$ una formula abierta, mientras que es existencial positiva si $\varphi\left(\bar{x},\bar{y}\right)$ es abierta positiva. Una formula es conjuncion de atomicas si es de la forma $\varphi_1\left(\bar{x}\right) \wedge \cdots \wedge \varphi_n\left(\bar{x}\right)$, donde cada $\varphi_i \in \mathrm{At}\left(\mathcal{L}\right)$. Una formula es primitiva positiva si es de la forma $\exists\bar{x}\,\varphi\left(\bar{x},\bar{y}\right)$ con $\varphi\left(\bar{x},\bar{y}\right)$ conjuncion de atomicas.

Definición 1. Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden y $R \in \mathcal{L}' - \mathcal{L}$ un símbolo de relación n-ario y $\Sigma\left(\mathcal{L}\right) \subseteq \mathrm{Fo}\left(\mathcal{L}\right)$. Diremos que R es *definible* en $\Sigma\left(\mathcal{L}\right)$ sobre una familia $\mathcal{K}$ de $\mathcal{L}'$-estructuras cuando exista una $\mathcal{L}$-fórmula $\varphi\left(x_1,\ldots,x_n\right) \in \Sigma\left(\mathcal{L}\right)$ tal que para todo $\mathbf{A} \in \mathcal{K}$ y todas $a_1,\ldots,a_n \in A$

$$\left(a_1,\ldots,a_n\right) \in R^{\mathbf{A}} \iff \varphi\left(a_1,\ldots,a_n\right)$$

Dado un conjunto de fórmulas Δ, escribiremos $\Delta(\bar{x})$ para anunciar que cada una de las fórmulas en Δ tiene sus variables libres contenidas en la tupla $\bar{x}$, y que consideramos cada $\delta \in \Delta(\bar{x})$ declarada $\delta = \delta(\bar{x})$. Si $\mathbf{A}$ es una estructura y $\bar{a}$ es una tupla de elementos de A, escribiremos $\mathbf{A} \vDash \Delta[\bar{a}]$ cuando $\mathbf{A} \vDash \delta[\bar{a}]$ para cada $\delta \in \Delta(\bar{x})$.

Si $\mathcal{L} \subseteq \mathcal{L}'$ son lenguajes de primer orden, para una $\mathcal{L}'$-estructura $\boldsymbol{A}$, usaremos $\boldsymbol{A}_{\mathcal{L}}$ para indicar el reducto de $\boldsymbol{A}$ al lenguaje $\mathcal{L}$. Si $\boldsymbol{A}, \boldsymbol{B}$ son $\mathcal{L}$-estructuras, usaremos $\boldsymbol{A} \leq \boldsymbol{B}$ para expresar que $\boldsymbol{A}$ es subestructura de $\boldsymbol{B}$.

Dados A, B conjuntos, $R^A \subseteq A^n$, $R^B \subseteq B^n$, diremos que una función $\gamma : A \to B$ preserva R si para toda tupla $(a_1, \ldots, a_n) \in A_0$ tenemos que si $(a_1, \ldots, a_n) \in R^A$ implica que $(\sigma(a_1), \ldots, \sigma(a_n)) \in R^B$.

Dada una estructura $\mathbf{A}$ y $\bar{a} = a_1, \ldots, a_n \in A$ utilizaremos $\langle \bar{a} \rangle^{\mathbf{A}}$ para denotar la subestructura generada por $\bar{a}$ en $\mathbf{A}$.

Definición 2. Dos fórmulas $\alpha(\bar{x})$ y $\beta(\bar{x})$ se dicen *equivalentes* sobre una familia de estructuras $\mathcal{K}$ si para cada $\mathbf{A} \in \mathcal{K}$ y cada $\bar{a}$ de A vale que

$$\mathbf{A} \vDash \alpha[\bar{a}] \Longleftrightarrow \mathbf{A} \vDash \beta[\bar{a}].$$

Teorema 3 (Modulo equivalencia sobre una estructura finita, la cantidad de fórmulas en $x_1, \ldots, x_n$ es finita)**.** *Sea $\mathcal{K}$ una clase finita de estructuras finitas, y sean $x_1, \ldots, x_n$ variables.*

Hay un conjunto finito de fórmulas $\Sigma(\bar{x})$ tal que para toda fórmula $\varphi(\bar{x})$ hay $\sigma(\bar{x}) \in \Sigma(\bar{x})$ tal que $\varphi(\bar{x})$ y $\sigma(\bar{x})$ son equivalentes sobre $\mathcal{K}$.

Demostración. Sea $\mathcal{K} = \{\mathbf{A}_1, \ldots, \mathbf{A}_m\}$. Veamos primero que las fórmulas son finitas modulo equivalencia. Sea $\varphi(\bar{x})$ defino $T_{\varphi\mathcal{K}} = \{\bar{a} \in A_1^n | \mathbf{A}_1 \vDash \varphi[\bar{a}]\} \times \cdots \times \{\bar{a} \in A_m^n | \mathbf{A}_m \vDash \varphi[\bar{a}]\}$ y supongamos φ equivalente ψ en $\mathcal{K}$, entonces

$$(\bar{a}_1, \ldots, \bar{a}_m) \in T_{\varphi\mathcal{K}} \Leftrightarrow (\bar{a}_1, \ldots, \bar{a}_m) \in T_{\psi\mathcal{K}} \text{ para cada } (\bar{a}_1, \ldots, \bar{a}_m) \in A_1^n \times \cdots \times A_m^n$$

lo que significa que $T_{\varphi\mathcal{K}} = T_{\psi\mathcal{K}}$. Entonces basta con contar los subconjuntos de $A_1^n \times \cdots \times A_m^n$, como $|A_1^n \times \cdots \times A_m^n| = |A_1|^n . \cdots . |A_m|^n$ y cada A_i era finito, es claro que $A_1^n \times \cdots \times A_m^n$ es finito. Por lo tanto $\mathcal{P}(A^n)$ también lo es.

Ahora veamos que existe $\Sigma(\bar{x})$. Sean $T_1, \ldots, T_k \subseteq A_1^n \times \cdots \times A_m^n$ tales que existe $\varphi_i(\bar{x})$ tal que para cada $\mathbf{A} \in \mathcal{K}$, $\mathbf{A} \vDash \varphi_i[T_i]$ y sea $\psi(\bar{x})$, tomo $T_{\psi\mathcal{K}}$y como $T_1, \ldots, T_k$ es la sucesión de todos los subconjuntos que se definen con una fórmula, hay un j tal que $T_j = T_{\psi\mathcal{K}}$. Entonces, para cada $\mathbf{A} \in \mathcal{K}$, $\mathbf{A} \vDash \psi[\bar{a}] \Leftrightarrow \mathbf{A} \vDash \varphi_j[\bar{a}]$. □

Teorema 4 (Los embeddings preservan fórmulas abiertas)**.** *Sean $\mathbf{A}$, $\mathbf{B}$ estructuras y $\gamma : \mathbf{A} \to \mathbf{B}$ una función. Son equivalentes:*

1. *γ es un embedding de A en B.*
2. *Para toda fórmula abierta $\varphi(\bar{x})$ y para cada $\bar{a}$ de A vale que:*

$$\mathbf{A} \vDash \varphi[\bar{a}] \Leftrightarrow \mathbf{B} \vDash \varphi[\gamma(\bar{a})].$$

Demostración. 1⇒2)

Sea γ un embedding de A en B, sea $\varphi(x_1, ..., x_n)$ una fórmula abierta y $\bar{a} \in A^n$, el caso base sale directo ya que los homomorfismos preservan términos. Veamos los casos inductivos:

Sea $\varphi(\bar{x}) = \neg\varphi_1(\bar{x})$ con $\varphi_1 \in F_k^\tau$:

$$\mathbf{A} \vDash \neg\varphi_1[\bar{a}] \text{ sii } \mathbf{A} \nvDash \varphi_1[\bar{a}] \text{ sii } \mathbf{B} \nvDash \varphi_1[\gamma(\bar{a})] \text{ sii } \mathbf{B} \vDash \neg\varphi_1[\gamma(\bar{a})]$$

Sea $\varphi(\bar{x}) = (\varphi_1 \eta \varphi_2)(\bar{x})$ con $\varphi_1, \varphi_2 \in F_k^\tau$:

$$\mathbf{A} \vDash (\varphi_1\eta\varphi_2)[\bar{a}] \text{ sii } \mathbf{A} \vDash \varphi_1[\bar{a}]\text{"}\eta\text{"}\mathbf{A} \vDash \varphi_2[\bar{a}] \text{ sii } \mathbf{A} \vDash \varphi_1[\gamma(\bar{a})]\text{"}\eta\text{"}\mathbf{B} \vDash \varphi_2[\gamma(\bar{a})] \text{ sii } \mathbf{B} \vDash (\varphi_1\eta\varphi_2)[\gamma(\bar{a})]$$

2⇒1)
Supongamos que para toda fórmula abierta $\varphi = \varphi(\bar{x})$ y cada $\bar{a} \in A^m$ vale que:

$$\mathbf{A} \vDash \varphi[\bar{a}] \Leftrightarrow \mathbf{B} \vDash \varphi[\gamma(\bar{a})]$$

- Veamos que γ es inyectiva:

$$\begin{array}{rcl} \gamma(a) & = & \gamma(a') \\ \mathbf{B} & \vDash & (x_1 \equiv x_2)[\gamma(a), \gamma(a')] \\ \mathbf{A} & \vDash & (x_1 \equiv x_2)[a, a'] \\ a & = & a' \end{array}$$

- Veamos que γ es un homomorfismo:
  Sea $c \in \mathcal{C}$

$$\begin{array}{rcl} \mathbf{A} & \vDash & (c \equiv x_1)[c^{\mathbf{A}}] \\ \mathbf{B} & \vDash & (c \equiv x_1)[\gamma(c^{\mathbf{A}})] \\ c^{\mathbf{B}} & = & \gamma(c^{\mathbf{A}}) \end{array}$$

  Sea $r \in \mathcal{R}_n$

$$\begin{array}{rcl} (a_1, \ldots, a_n) & \in & r^{\mathbf{A}} \\ \mathbf{A} & \vDash & r(x_1, \ldots, x_n)[a_1, \ldots, a_n] \\ \mathbf{B} & \vDash & r(x_1, \ldots, x_n)[\gamma(a_1), \ldots, \gamma(a_n)] \\ (\gamma(a_1), \ldots, \gamma(a_n)) & \in & r^{\mathbf{B}} \end{array}$$

  Sea $f \in \mathcal{F}_n$

$$\begin{array}{rcl} \mathbf{A} & \vDash & (f(x_1, \ldots, x_n) \equiv x_{n+1})[a_1, \ldots, a_n, f^{\mathbf{A}}(a_1, \ldots, a_n)] \\ \mathbf{B} & \vDash & (f(x_1, \ldots, x_n) \equiv x_{n+1})[\gamma(a_1), \ldots, \gamma(a_n), \gamma(f^{\mathbf{A}}(a_1, \ldots, a_n))] \\ f^{\mathbf{B}}(\gamma(a_1), \ldots, \gamma(a_n)) & = & \gamma(f^{\mathbf{A}}(a_1, \ldots, a_n)) \end{array}$$

- Veamos que γ es embedding:
  Sea $r \in \mathcal{R}_n$

$$\begin{array}{rcl} (\gamma(a_1), \ldots, \gamma(a_n)) & \in & r^{\mathbf{B}} \\ \mathbf{B} & \vDash & r(x_1, \ldots, x_n)[\gamma(a_1), \ldots, \gamma(a_n)] \\ \mathbf{A} & \vDash & r(x_1, \ldots, x_n)[a_1, \ldots, a_n] \\ (a_1, \ldots, a_n) & \in & r^{\mathbf{A}} \end{array}$$

□

Teorema 5. *Si* $\mathbf{A}$ *es una subestructura de* $\mathbf{B}$ *y* $\varphi(\bar{x})$ *es una fórmula abierta, entonces para cada* $\bar{a} \in A^n$ *vale que*

$$\mathbf{A} \vDash \varphi[\bar{a}] \Leftrightarrow \mathbf{B} \vDash \varphi[\bar{a}]$$

Demostración. Sea $\mathbf{A}$ una subestructura de $\mathbf{B}$ y $\varphi(\bar{x})$ una fórmula abierta.

Como $\mathbf{A}$ es subestructura es cerrada sobre r,f y $c^{\mathbf{B}} \in A$

Sea $\gamma : \mathbf{A} \to \mathbf{B}$ tal que para todo $x \in A$, $\gamma(x) = x$. Como $\mathbf{A} \leq \mathbf{B}$, es directo que γ es un embedding. Entonces por el Teorema 4, $\mathbf{A} \vDash \varphi[\bar{a}] \Leftrightarrow \mathbf{B} \vDash \varphi[\gamma(\bar{a})]$

Finalmente, por definición de γ:

$$\mathbf{A} \vDash \varphi\left[\bar{a}\right] \Leftrightarrow \mathbf{B} \vDash \varphi\left[\bar{a}\right]$$

□

Definición 6. Sea $\mathbf{A}$ una estructura y sean $a_1, \ldots, a_n \in A$. Definimos el diagrama abierto para $a_1, \ldots, a_n$ en $\mathbf{A}$ como:

$$\Delta_{\mathbf{A},\bar{a}}\left(x_1, \ldots, x_n\right) := \left\{\alpha \mid \alpha \in \mathrm{Op}\left(\mathcal{L}\right) \text{ y } \mathbf{A} \vDash \alpha\left[\bar{a}\right]\right\}$$

Teorema 7. *Sea $\mathbf{A}$ una estructura y $b_1, \ldots, b_n \in B$, son equivalentes:*

1. $\mathbf{B} \vDash \Delta_{\mathbf{A}\bar{a}}\left[\bar{b}\right]$
2. *Hay un isomorfismo γ de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\left\langle\bar{b}\right\rangle^{\mathbf{B}}$ tal que $\gamma(\bar{a}) = \bar{b}$*

Demostración. Veamos 1⇒2:

Supongamos que $\mathbf{B} \vDash \Delta_{\mathbf{A}\bar{a}}\left[\bar{b}\right]$

Si α fórmula abierta y $\mathbf{A} \vDash \alpha\left[\bar{a}\right]$ entonces $\mathbf{B} \vDash \alpha\left[\bar{b}\right]$

Tomo $\langle\bar{a}\rangle^{\mathbf{A}} = \{t^{\mathbf{A}}[a_1, \ldots, a_n] : t \in T^{\tau}, a_1, \ldots, a_n \in \bar{a}\}$ y $\left\langle\bar{b}\right\rangle^{\mathbf{A}} = \{t^{\mathbf{A}}[b_1, \ldots, b_n] : t \in T^{\tau}, b_1, \ldots, b_n \in \bar{b}\}$

Defino

$$\begin{aligned} \gamma & : A' \to B' \\ \gamma & \left(t^{\mathbf{A}}[a_1, \ldots, a_n]\right) = t^{\mathbf{B}}[b_1, \ldots, b_n] \end{aligned}$$

Es claro que γ es un homomorfismo.

- Veamos que γ es inyectivo

  Sean $a_1', a_2' \in A'$ tales que $a_1' \neq a_2'$

$$\begin{aligned} a_1' & \neq a_2' \\ t_1^{\mathbf{A}}\left[\bar{a}\right] & \neq t_2^{\mathbf{A}}\left[\bar{a}\right] \\ \mathbf{A} & \vDash \neg(t_1 \equiv t_2)\left[\bar{a}\right] \quad (2.1) \\ \mathbf{B} & \vDash \neg(t_1 \equiv t_2)\left[\bar{b}\right] \quad (2.2) \\ t_1^{\mathbf{B}}\left[\bar{b}\right] & \neq t_2^{\mathbf{B}}\left[\bar{b}\right] \\ \gamma\left(t_1^{\mathbf{A}}\left[\bar{a}\right]\right) & \neq \gamma\left(t_2^{\mathbf{A}}\left[\bar{a}\right]\right) \end{aligned}$$

- Veamos que γ es sobreyectivo

  Sea $b' \in B'$

$$\begin{aligned} b' & \in B' \\ t^{\mathbf{B}}\left[\bar{b}\right] & \in B' \end{aligned}$$

  Tomo $t^{\mathbf{A}}\left[\bar{a}\right] = a'$

$$\gamma\left(t^{\mathbf{A}}\left[\bar{a}\right]\right) = t^{\mathbf{B}}\left[\bar{b}\right]$$

Por lo tanto γ es un isomorfismo y $\gamma(a_j) = \gamma(x_j^{\mathbf{A}}[a_1, \ldots, a_n]) = x_j^{\mathbf{B}}[b_1, \ldots, b_n] = b_j$.

Ahora veamos 2⇒1:

Supongamos que hay un isomorfismo γ de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\left\langle\bar{b}\right\rangle^{\mathbf{B}}$ tal que $\gamma(\bar{a}) = \bar{b}$

Entonces, es claro que γ es un embedding de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\left\langle\bar{b}\right\rangle^{\mathbf{B}}$.

Por el Teorema 4 para toda fórmula abierta $\varphi\left(\bar{x}\right)$ y cada $\bar{a} \in A'$

$$\langle\bar{a}\rangle^{\mathbf{A}} \vDash \varphi[\bar{a}] \Leftrightarrow \left\langle\bar{b}\right\rangle^{\mathbf{B}} \vDash \varphi[\gamma(\bar{a})]$$

Por el Teorema 5 como $\langle\bar{a}\rangle^{\mathbf{A}}$ es subestructura de $\mathbf{A}$, $\left\langle\bar{b}\right\rangle^{\mathbf{B}}$ lo es de $\mathbf{B}$ y φ es abierta

$$\mathbf{A} \vDash \varphi[\bar{a}] \Leftrightarrow \mathbf{B} \vDash \varphi[\gamma(\bar{a})]$$

$$\mathbf{A} \vDash \varphi[\bar{a}] \Leftrightarrow \mathbf{B} \vDash \varphi[\bar{b}]$$

Por lo tanto si α es abierta y $\mathbf{A} \vDash \alpha[\bar{a}]$ entonces $\mathbf{B} \vDash \alpha[\bar{b}]$
Finalmente, $\mathbf{B} \vDash \Delta_{\mathbf{A}\bar{a}}[\bar{b}]$ □

Teorema 8. *Sean $\mathbf{A}, \mathbf{B}$ estructuras y $h : A \to B$ una función, son equivalentes:*

1. *h es un homomorfismo de $\mathbf{A}$ en $\mathbf{B}$.*
2. *Para toda fórmula atómica $\varphi = \varphi(\bar{x})$ y para cada $\bar{a}$ de A vale que*
$$\mathbf{A} \vDash \varphi[\bar{a}] \Longrightarrow \mathbf{B} \vDash \varphi[h(\bar{a})].$$
3. *Para toda fórmula abierta positiva $\varphi = \varphi(\bar{x})$ y para cada $\bar{a}$ de A vale que*
$$\mathbf{A} \vDash \varphi[\bar{a}] \Longrightarrow \mathbf{B} \vDash \varphi[h(\bar{a})].$$

Demostración. 1⇒2) Directo, ya que los homomorfismos preservan términos.

2⇒1) Supongamos $\mathbf{A} \vDash \varphi[\bar{a}] \Rightarrow \mathbf{B} \vDash \varphi[h(\bar{a})]$ con φ atómica. Veamos que h es homomorfismo.

Sea $c \in \mathcal{C}$

$$\mathbf{A} \vDash (c \equiv x_1)\left[c^{\mathbf{A}}\right] \Rightarrow \mathbf{B} \vDash (c \equiv x_1)\left[h\left(c^{\mathbf{A}}\right)\right] \Leftrightarrow c^{\mathbf{B}} = h\left(c^{\mathbf{A}}\right)$$

Sea $f \in \mathcal{F}_n$

$$\begin{array}{rcl} \mathbf{A} & \vDash & (f(x_1,\dots,x_n) \equiv x_{n+1})\left[a_1,\dots,a_n, f^{\mathbf{A}}(\bar{a})\right] \\ \mathbf{B} & \vDash & (f(x_1,\dots,x_n) \equiv x_{n+1})\left[h(a_1),\dots,h(\bar{a}_n), h\left(f^{\mathbf{A}}(\bar{a})\right)\right] \\ f^{\mathbf{B}}(h(\bar{a})) & = & h\left(f^{\mathbf{A}}(\bar{a})\right) \end{array}$$

Sea $r \in \mathcal{R}_n$

$$\bar{a} \in r^{\mathbf{A}} \Rightarrow \mathbf{A} \vDash r(\bar{x})[\bar{a}] \Rightarrow \mathbf{B} \vDash r(\bar{x})[h(\bar{a})] \Rightarrow h(\bar{a}) \in r^{\mathbf{B}}$$

2⇒3) Rutina.

3⇒2) Directo ya que toda atómica es abierta positiva. □

Definición 9. Sea $\mathbf{A}$ una estructura y sean $a_1,\dots,a_n \in A$. Definimos el diagrama atómico positivo de $\bar{a}$ en $\mathbf{A}$ como

$$\Delta^{+}_{\mathbf{A},\bar{a}}(x_1,\dots,x_n) := \{\alpha \mid \alpha \in \mathrm{At}(\mathcal{L}) \text{ y } \mathbf{A} \vDash \alpha[\bar{a}]\}.$$

Teorema 10. *Sean $\mathbf{B}$ una estructura y $b_1,\dots,b_n \in B$, son equivalentes:*

1. $\mathbf{B} \vDash \Delta^{+}_{\mathbf{A},\bar{a}}[\bar{b}]$.
2. *Hay un homomorfismo h de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\langle\bar{b}\rangle^{\mathbf{B}}$ tal que $h(\bar{a}) = \bar{b}$.*

Demostración. 1⇒2) Defino $h : \langle\bar{a}\rangle^{\mathbf{A}} \to \langle\bar{b}\rangle^{\mathbf{B}}$, como

$$h\left(t^{\mathbf{A}}[\bar{a}]\right) = t^{\mathbf{B}}[\bar{b}]$$

El cual es se ve fácilmente que es un homomorfismo que cumple $h(\bar{a}) = \bar{b}$.

2⇒1) Supongamos que hay un homomorfismo h de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\langle\bar{b}\rangle^{\mathbf{B}}$ tal que $h(\bar{a}) = \bar{b}$, entonces como $\mathbf{A} \vDash \Delta^{+}_{\mathbf{A},\bar{a}}[\bar{a}]$, por el Teorema 8 $\mathbf{B} \vDash \Delta^{+}_{\mathbf{A},\bar{a}}[h(\bar{a})]$, entonces $\mathbf{B} \vDash \Delta^{+}_{\mathbf{A},\bar{a}}[\bar{b}]$ □

2.2. Definibilidad de primer orden.

Teorema 11. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden, sea $R \in \mathcal{L}' - \mathcal{L}$ un símbolo de relación n-ario. Para una clase finita $\mathcal{K}$ de $\mathcal{L}'$-estructuras, los siguientes son equivalentes:*

1. *Hay una $\mathcal{L}$-fórmula que define R en $\mathcal{K}$.*
2. *Para todas $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{K}$, todo isomorfismo $\gamma : \boldsymbol{A}_{\mathcal{L}} \to \boldsymbol{B}_{\mathcal{L}}$ preserva R.*

Demostración. 1⇒2) Sea φ una $\mathcal{L}$-formula que define a R en $\mathcal{K}$, y sea $\gamma : \mathbf{A} \to \mathbf{B}$ un isomorfismo. Supongamos $\mathbf{A} \vDash \varphi[\bar{a}]$ entonces $\mathbf{B} \vDash \varphi[\gamma(\bar{a})]$ y como φ define a R en $\mathcal{K}$, entonces $\gamma(\bar{a}) \in R^{\mathbf{B}}$.

2⇒1) Para cada $\mathbf{A} \in \mathcal{K}$, cada $\bar{a} \in R^{\mathbf{A}}$, tomo $\bar{a'}$ tal que $\left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} = \mathbf{A}$. Ahora tomo $\delta_{\mathbf{A},(\bar{a},\bar{a'})}(\bar{x},\bar{y}) = \bigwedge_{\alpha \in \Delta_{(\bar{a},\bar{a'}),\mathbf{A}}} \alpha(\bar{x},\bar{y})$, que es fórmula por Teorema 3. Ahora tomo

$$\varphi = \bigvee_{\mathbf{A}\in\mathcal{K}} \left(\left(\bigvee_{\bar{a}\in R^{\mathbf{A}}} \left(\exists \bar{y}\, \delta_{\mathbf{A},(\bar{a},\bar{a'})}(\bar{x},\bar{y})\right) \right) \wedge \left(\forall x_1, \ldots, x_{|\mathbf{A}|+1} \bigvee_{i \neq j} x_i = x_j \right) \right)$$

Sea $(b_1, \ldots, b_n) \in R^{\mathbf{B}}$. Entonces como $\varphi = \cdots \vee \left(\left(\exists \bar{y}\, \delta_{\mathbf{B},(\bar{b},\bar{b'})}(\bar{x},\bar{y}) \vee \ldots \right) \wedge \left(\forall x_1, \ldots, x_{|\mathbf{B}|+1} \bigvee_{i\neq j} x_i = x_j \right) \right) \ldots$, y como $\mathbf{B} \vDash \exists \bar{y}\, \delta_{\mathbf{B},(\bar{b},\bar{b'})}(\bar{x},\bar{y}) \left[\bar{b}, \bar{b'}\right]$ y $\mathbf{B} \vDash \forall x_1, \ldots, x_{|\mathbf{B}|+1} \bigvee_{i\neq j} x_i = x_j \left[\bar{b}, \bar{b'}\right]$ entonces $\mathbf{B} \vDash \varphi\left[\bar{b}\right]$.

Sea $\mathbf{B} \vDash \varphi[b_1, \ldots, b_n]$. Entonces hay algún $\mathbf{A} \in \mathcal{K}$, y algún $\bar{a} \in R^{\mathbf{A}}$ tales que $\mathbf{B} \vDash \left(\exists \bar{y}\, \delta_{\mathbf{A},(\bar{a},\bar{a'})}(\bar{x},\bar{y})\right) \wedge \left(\forall x_1, \ldots, x_{|\mathbf{A}|+1} \bigvee_{i\neq j} x_i = x_j \right) \left[\bar{b}\right]$. Entonces hay un $\bar{z}$ en $\mathbf{B}$ tal que $\mathbf{B} \vDash \delta_{\mathbf{A},(\bar{a},\bar{a'})}(\bar{x},\bar{y}) \left[\bar{b}, \bar{z}\right]$. Como $\mathbf{B} \vDash \Delta_{(\bar{a},\bar{a'}),\mathbf{A}} \left[\bar{b}, \bar{z}\right]$, por Teorema 7 hay $\gamma : \left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} \to \left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}}$ isomorfismo tal que $\left(\gamma(\bar{a}), \gamma\left(\bar{a'}\right)\right) = \left(\bar{b}, \bar{z}\right)$ con $\left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} = \mathbf{A}$ y $\left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}} \leq \mathbf{B}$, lo que implica que $|\mathbf{A}| \leq |\mathbf{B}|$. Pero como $\mathbf{B} \vDash \forall x_1, \ldots, x_{|\mathbf{A}|+1} \bigvee_{i\neq j} x_i = x_j [b_1, \ldots, b_n]$ entonces $|\mathbf{B}| \leq |\mathbf{A}|$, luego $|\mathbf{B}| = |\mathbf{A}|$, por lo que es claro que $\left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}} = \mathbf{B}$. Como $\gamma : \mathbf{A} \to \mathbf{B}$ es un isomorfismo tal que $\left(\gamma(\bar{a}), \gamma\left(\bar{a'}\right)\right) = \left(\bar{b}, \bar{z}\right)$ y sabemos que los isomorfismos entre estructuras de $\mathcal{K}$, preservan R, como $\bar{a} \in R^{\mathbf{A}}$, $\gamma(\bar{a}) = \bar{b} \in R^{\mathbf{B}}$.

Como φ es disyunción de existenciales, por teorema puede ser convertida en una fórmula existencial. □

2.3. Definibilidad por fórmulas abiertas.

A continuación presentamos el teorema de caracterización para definibilidad por fórmulas abiertas. Antes necesitaremos algunos lemas.

Teorema 12. *Sean $\mathbf{A}, \mathbf{B}$ estructuras finitas y $a_1, \ldots, a_n \in A$, entonces existe una fórmula abierta $\varphi(\bar{x})$ tal que para todo $\bar{b} \in B^n$ son equivalentes:*

1. $\mathbf{B} \vDash \varphi\left[\bar{b}\right]$
2. *Hay un isomorfismo γ de $\langle \bar{a} \rangle^{\mathbf{A}}$ en $\left\langle \bar{b} \right\rangle^{\mathbf{B}}$ tal que $\gamma(\bar{a}) = \bar{b}$*

Demostración. Tomo $\varphi = \bigwedge_{\alpha \in \Delta_{\mathbf{A}\bar{a}}} \alpha(\bar{x})$ que es una fórmula, por el Teorema 3.

Es claro que $\mathbf{B} \vDash \bigwedge_{\alpha \in \Delta_{\mathbf{A}\bar{a}}} \alpha(\bar{x}) \left[\bar{b}\right] \Leftrightarrow \mathbf{B} \vDash \Delta_{\mathbf{A}\bar{a}} \left[\bar{b}\right]$

Entonces por Teorema 7, hay un isomorfismo γ de $\langle \bar{a} \rangle^{\mathbf{A}}$ en $\left\langle \bar{b} \right\rangle^{\mathbf{B}}$ tal que $\gamma(\bar{a}) = \bar{b}$ □

Teorema 13. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden, sea $R \in \mathcal{L}' - \mathcal{L}$ un símbolo de relación n-ario. Para una clase finita $\mathcal{K}$ de $\mathcal{L}'$-estructuras, los siguientes son equivalentes:*

1. Hay una fórmula en $\mathrm{Op}(\mathcal{L})$ que define R en $\mathcal{K}$.

2. Para todas $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{K}$, todas $\boldsymbol{A}_0 \leq \boldsymbol{A}_{\mathcal{L}}$,$\boldsymbol{B}_0 \leq \boldsymbol{B}_{\mathcal{L}}$, todo isomorfismo $\sigma : \boldsymbol{A}_0 \to \boldsymbol{B}_0$ preserva R.

Demostración. 1⇒2) Sea $\varphi(\bar{x})$ la fórmula que define R en $\mathcal{K}$, sean $\mathbf{A}, \mathbf{B} \in \mathcal{K}$ y sean $\mathbf{A}_0 \leq \mathbf{A}_{\mathcal{L}}$ y $\mathbf{B}_0 \leq \mathbf{B}_{\mathcal{L}}$ tales que $\sigma : \mathbf{A}_0 \to \mathbf{B}_0$ sea un isomorfismo. Sea $\bar{a} \in R^{\mathbf{A}_0} \cap A_0$ veamos que $\sigma(\bar{a}) \in R^{\mathbf{B}_0}$. Como $\bar{a} \in R^{\mathbf{A}_0} \Rightarrow \bar{a} \in R^{\mathbf{A}} \Leftrightarrow \mathbf{A} \vDash \varphi[\bar{a}]$, por Teorema 5, $A_0 \vDash \varphi[\bar{a}]$ y como $\mathbf{A}_0$ y $\mathbf{B}_0$ son isomorfos por σ, $\mathbf{B}_0 \vDash \varphi[\sigma(\bar{a})]$. Nuevamente por Teorema 5 $\mathbf{B} \vDash \varphi[\sigma(\bar{a})]$ y como por hipótesis φ define a R en $\mathcal{K}$, $\sigma(\bar{a}) \in R^{\mathbf{B}}$.

2⇒1) Sea $\varphi = \bigvee_{\mathbf{A}\in\mathcal{K}} \left(\bigvee_{\bar{a}\in R^{\mathbf{A}}} \left(\bigwedge_{\alpha\in\Delta_{\mathbf{A}\bar{a}}} \alpha(\bar{x})\right)\right)$, la cual es fórmula por el Teorema 3. Veamos que para cada $\mathbf{B} \in \mathcal{K}$, $\mathbf{B} \vDash \varphi[\bar{b}]$ sii $\bar{b} \in R^{\mathbf{B}}$. Sea $\mathbf{B} \in \mathcal{K}$:

⇒) Supongamos $\mathbf{B} \vDash \varphi[\bar{b}]$, entonces en particular para algún $\mathbf{A} \in \mathcal{K}$ y $\bar{a} \in R^{\mathbf{A}}$, $\mathbf{B} \vDash \bigwedge_{\alpha\in\Delta_{\mathbf{A}\bar{a}}} \alpha(\bar{x})[\bar{b}]$. Por Teorema 12, hay un isomorfismo $\gamma : \langle\bar{a}\rangle^{\mathbf{A}} \to \langle\bar{b}\rangle^{\mathbf{B}}$ tal que $\gamma(\bar{a}) = \bar{b}$. Entonces por hipótesis γ preserva R. Finalmente como $\bar{a} \in R^{\mathbf{A}}$, entonces $\gamma(\bar{a}) \in R^{\mathbf{B}}$ o sea que $\bar{b} \in R^{\mathbf{B}}$.

⇐) Supongamos $\bar{b} \in R^{\mathbf{B}}$. Entonces $\varphi = \bigvee_{\mathbf{A}\in\mathcal{K}} \left(\bigvee_{\bar{a}\in R^{\mathbf{A}}} \left(\bigwedge_{\alpha\in\Delta_{\mathbf{A}\bar{a}}} \alpha(\bar{x})\right)\right) = \cdots \vee \cdots \vee \left(\bigwedge_{\alpha\in\Delta_{\mathbf{B}\bar{b}}} \alpha(\bar{x})\right)$, evidentemente $\mathbf{B} \vDash \Delta_{\mathbf{B}\bar{b}}[\bar{b}]$, entonces $\mathbf{B} \vDash \varphi[\bar{b}]$. □

2.4. Definibilidad por fórmulas abiertas positivas.

Teorema 14. *Sean $\mathbf{A}, \mathbf{B}$ estructuras finitas y $a_1, \ldots, a_n an \in A$, entonces existe una fórmula abierta $\varphi(\bar{x})$ tal que para todo $\bar{b} \in B^n$ son equivalentes:*

1. $\mathbf{B} \vDash \varphi[\bar{b}]$
2. *Hay un homomorfismo h de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\langle\bar{b}\rangle^{\mathbf{B}}$ tal que $h(\bar{a}) = \bar{b}$*

Demostración. Tomo $\varphi = \bigwedge_{\alpha\in\Delta^{+}_{\mathbf{A}\bar{a}}} \alpha(\bar{x})$ que es una fórmula, por el Teorema 3.

Es claro que $\mathbf{B} \vDash \varphi[\bar{b}] \Leftrightarrow \mathbf{B} \vDash \Delta^{+}_{\mathbf{A}\bar{a}}[\bar{b}]$

Entonces por Teorema 10, hay un homomorfismo h de $\langle\bar{a}\rangle^{\mathbf{A}}$ en $\langle\bar{b}\rangle^{\mathbf{B}}$ tal que $h(\bar{a}) = \bar{b}$ □

Teorema 15. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden, sea $R \in \mathcal{L}' - \mathcal{L}$ un símbolo de relación n-ario. Para una clase finita $\mathcal{K}$ de $\mathcal{L}'$-estructuras, los siguientes son equivalentes:*

1. Hay una $\mathcal{L}$-fórmula abierta positiva que define R en $\mathcal{K}$.
2. Para todas $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{K}$, todas $\boldsymbol{A}_0 \leq \boldsymbol{A}_{\mathcal{L}}$,$\boldsymbol{B}_0 \leq \boldsymbol{B}_{\mathcal{L}}$, todo homomorfismo $h : \boldsymbol{A}_0 \to \boldsymbol{B}_0$ preserva R.

Demostración. Veamos 1⇒2:

Sea $\varphi(\bar{x})$ la fórmula abierta positiva que define R en $\mathcal{K}$, sean $\mathbf{A}, \mathbf{B} \in \mathcal{K}$ y sean $\mathbf{A_0} \leq \mathbf{A}_{\mathcal{L}}$ y $\mathbf{B}_0 \leq \mathbf{B}_{\mathcal{L}}$ tales que $h : \mathbf{A}_0 \to \mathbf{B}_0$ sea un homomorfismo. Sea $\bar{a} \in R^{\mathbf{A}_0} \cap A_0$ veamos que $h(\bar{a}) \in R^{\mathbf{B}_0}$:

$\bar{a} \in R^{\mathbf{A}_0} \Rightarrow \bar{a} \in R^{\mathbf{A}} \Leftrightarrow \mathbf{A} \vDash \varphi[\bar{a}]$, por Teorema 5, $\mathbf{A}_0 \vDash \varphi[\bar{a}]$ y como h es un homomorfismo de $\mathbf{A}_0$ en $\mathbf{B}_0$ y φ es abierta positiva por Teorema 8, $\mathbf{B}_0 \vDash \varphi[h(\bar{a})]$. Nuevamente por Teorema 5 $\mathbf{B} \vDash \varphi[h(\bar{a})]$ y como por hipótesis φ define a R en $\mathcal{K}$, $h(\bar{a}) \in R^{\mathbf{B}}$.

Veamos 2⇒1:

Sea $\varphi = \bigvee_{\mathbf{A}\in\mathcal{K}} \left(\bigvee_{\bar{a}\in R^{\mathbf{A}}} \left(\bigwedge_{\alpha\in\Delta^{+}_{\mathbf{A}\bar{a}}} \alpha(\bar{x})\right)\right)$, la cual es fórmula por el Teorema 3. Sea $\mathbf{B} \in \mathcal{K}$:

⇒) Supongamos $\mathbf{B} \vDash \varphi[\bar{b}]$, entonces en particular para algún $\mathbf{A} \in \mathcal{K}$ y $\bar{a} \in R^{\mathbf{A}}$, $\mathbf{B} \vDash \Delta^{+}_{\mathbf{A}\bar{a}}[\bar{b}]$. Por Teorema 14, hay un homomorfismo $h : \langle\bar{a}\rangle^{\mathbf{A}} \to \langle\bar{b}\rangle^{\mathbf{B}}$ tal que $h(\bar{a}) = \bar{b}$. Entonces por hipótesis h preserva R. Finalmente como $\bar{a} \in R^{\mathbf{A}}$, entonces $h(\bar{a}) \in R^{\mathbf{B}}$ o sea que $\bar{b} \in R^{\mathbf{B}}$.

$\Leftarrow$) Supongamos $\bar{b} \in R^{\mathbf{B}}$. Entonces $\varphi = \cdots \vee \cdots \vee \left(\bigwedge_{\alpha \in \Delta^{+}_{\mathbf{B}\bar{b}}} \alpha(\bar{x}) \right)$, evidentemente $\mathbf{B} \vDash \Delta^{+}_{\mathbf{B}\bar{b}} \left[\bar{b}\right]$, entonces $\mathbf{B} \vDash \varphi \left[\bar{b}\right]$. □

2.5. Definibilidad por formulas existenciales.

Teorema 16. *Sea* $\mathbf{A}$ *una estructura finita y* $a_1, \ldots, a_n \in A$. *Entonces hay una fórmula existencial* $\varphi(x_1, \ldots, x_n)$ *tal que para toda estructura* $\mathbf{B}$ *y para todo* $\bar{b} \in B^n$ *los siguientes son equivalentes:*

1. $\mathbf{B} \vDash \varphi \left[\bar{b}\right]$.
2. *Hay un embedding* $\gamma : \mathbf{A} \to \mathbf{B}$ *tal que* $\gamma(\bar{a}) = \bar{b}$

Demostración. Sean $a'_1, \ldots, a'_m$ tales que $\langle a_1, \ldots, a_n, a'_1, \ldots, a'_m \rangle^{\mathbf{A}} = \mathbf{A}$ y sea

$$\varphi = \exists y_1, \ldots, y_m \left(\bigwedge_{\alpha \in \Delta_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y}) \right)$$

que es fórmula por el Teorema 3.

Supongamos $\mathbf{B} \vDash \varphi \left[\bar{b}\right]$, entonces existen $b'_1, \ldots, b'_m \in B$ tales que $\mathbf{B} \vDash \bigwedge_{\alpha \in \Delta_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y}) \left[\bar{b}, \bar{b'}\right]$ por Teorema 7 hay un isomorfismo $\gamma : \left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} \to \left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}}$ tal que $\gamma\left(\bar{a}, \bar{a'}\right) = \left(\bar{b}, \bar{b'}\right)$, y como $\left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} = \mathbf{A}$ y $\left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}} \leq \mathbf{B}$, γ es un embedding de $\mathbf{A}$ en $\mathbf{B}$ tal que $\gamma(\bar{a}) = \bar{b}$.

Supongamos que hay un embedding $\gamma : \mathbf{A} \to \mathbf{B}$ tal que $\gamma(\bar{a}) = \bar{b}$, por lo tanto $\gamma : \mathbf{A} \to \mathbf{S} \leq \mathbf{B}$ es un isomorfismo. Como $\mathbf{A} \vDash \varphi[\bar{a}]$, por ser γ un isomorfismo $\mathbf{S} \vDash \varphi[\gamma(\bar{a})]$ y entonces $\mathbf{S} \vDash \varphi \left[\bar{b}\right]$. Por lo tanto existe $\bar{b'}$en $S \subseteq B$ tal que $\mathbf{S} \vDash \bigwedge_{\alpha \in \Delta_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y}) \left[\bar{b}, \bar{b'}\right]$. Como $\bigwedge_{\alpha \in \Delta_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y})$ es una fórmula abierta, por Teorema 5, $\mathbf{B} \vDash \bigwedge_{\alpha \in \Delta_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y}) \left[\bar{b}, \bar{b'}\right]$, entonces $\mathbf{B} \vDash \varphi \left[\bar{b}\right]$ □

Teorema 17. *Sean* $\mathcal{L} \subseteq \mathcal{L}'$ *lenguajes de primer orden, sea* $R \in \mathcal{L}' - \mathcal{L}$ *un símbolo de relación n-ario. Para una clase finita* $\mathcal{K}$ *de* $\mathcal{L}'$*-estructuras, los siguientes son equivalentes:*

1. *Hay una* $\mathcal{L}$*-fórmula existencial que define* R *en* $\mathcal{K}$.
2. *Para todas* $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{K}$, *todo embedding* $\gamma : \boldsymbol{A}_{\mathcal{L}} \to \boldsymbol{B}_{\mathcal{L}}$ *preserva* R.

Demostración. 1⇒2) Sea φ existencial que define a R en $\mathcal{K}$, y sea $\gamma : \mathbf{A} \to \mathbf{B}$ un embedding. Entonces $\gamma : \mathbf{A} \to \mathbf{S} \leq \mathbf{B}$ es un isomorfismo. Supongamos $\mathbf{A} \vDash \varphi[\bar{a}]$ entonces $\mathbf{S} \vDash \varphi[\gamma(\bar{a})]$. Como φ es de la forma $\varphi = \exists \bar{y}\, \psi(\bar{x}, \bar{y})$ con ψ abierta, existe un $\bar{s}$ tal que $\mathbf{S} \vDash \psi[\gamma(\bar{a}), \bar{s}]$. Como ψ es abierta y $\mathbf{S} \leq \mathbf{B}$, por el Teorema 5 $\mathbf{B} \vDash \psi[\gamma(\bar{a}), \bar{s}]$. Entonces $\mathbf{B} \vDash \varphi[\gamma(\bar{a})]$ y como φdefine a R en $\mathcal{K}$, entonces $\gamma(\bar{a}) \in R^{\mathbf{B}}$.

2⇒1) Para cada $\mathbf{A} \in \mathcal{K}$, cada $\bar{a} \in R^{\mathbf{A}}$, tomo $\bar{a'}$ tal que $\left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} = \mathbf{A}$. Ahora tomo $\delta_{\mathbf{A}, (\bar{a}, \bar{a'})}(\bar{x}, \bar{y}) = \bigwedge_{\alpha \in \Delta_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y})$, que es fórmula por Teorema 3. Ahora tomo

$$\varphi = \bigvee_{\mathbf{A} \in \mathcal{K}} \left(\bigvee_{\bar{a} \in R^{\mathbf{A}}} \left(\exists \bar{y}\, \delta_{\mathbf{A}, (\bar{a}, \bar{a'})}(\bar{x}, \bar{y}) \right) \right)$$

Sea $(b_1, \ldots, b_n) \in R^{\mathbf{B}}$. Entonces como $\varphi = \cdots \vee \exists \bar{y}\, \delta_{\mathbf{B}, (\bar{b}, \bar{b'})}(\bar{x}, \bar{y}) \vee \ldots$, y como $\mathbf{B} \vDash \exists \bar{y}\, \delta_{\mathbf{B}, (\bar{b}, \bar{b'})}(\bar{x}, \bar{y}) \left[\bar{b}, \bar{b'}\right]$, $\mathbf{B} \vDash \varphi \left[\bar{b}\right]$.

Sea $\mathbf{B} \vDash \varphi[b_1, \ldots, b_n]$. Entonces para algún $\mathbf{A} \in \mathcal{K}$, y algún $\bar{a} \in R^{\mathbf{A}}$ tales que $\mathbf{B} \vDash \exists \bar{y}\, \delta_{\mathbf{A}, (\bar{a}, \bar{a'})}(\bar{x}, \bar{y}) \left[\bar{b}\right]$. Entonces por el Teorema 16 hay $\gamma : \mathbf{A} \to \mathbf{B}$ embedding tal que $\gamma(\bar{a}) = \bar{b}$. Finalmente como $\bar{a} \in R^{\mathbf{A}}$, $\gamma(\bar{a}) \in R^{\mathbf{B}}$, $\bar{b} \in R^{\mathbf{B}}$.

Notar que como φ es disyunción de existenciales, por teorema puede ser convertida en una fórmula existencial. □

2.6. Definibilidad por formulas existenciales positivas.

Teorema 18. *Sea* $\mathbf{A}$ *una estructura finita y* $a_1, \ldots, a_n \in A$. *Entonces hay una fórmula existencial positiva* $\varphi(x_1, \ldots, x_n)$ *tal que para toda estructura* $\mathbf{B}$ *y para todo* $\bar{b} \in B^n$ *los siguientes son equivalentes:*

1. $\mathbf{B} \vDash \varphi\left[\bar{b}\right]$.
2. *Hay un homomorfismo* $h : \mathbf{A} \to \mathbf{B}$ *tal que* $h(\bar{a}) = \bar{b}$

Demostración. Sean $a'_1, \ldots, a'_m$ tales que $\langle a_1, \ldots, a_n, a'_1, \ldots, a'_m \rangle^{\mathbf{A}} = \mathbf{A}$ y sea

$$\varphi = \exists y_1, \ldots, y_m \left(\bigwedge_{\alpha \in \Delta^+_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y}) \right)$$

que es fórmula por el Teorema 3.

Supongamos $\mathbf{B} \vDash \varphi\left[\bar{b}\right]$, entonces existen $b'_1, \ldots, b'_m \in B$ tales que $\mathbf{B} \vDash \bigwedge_{\alpha \in \Delta^+_{(\bar{a}, \bar{a'}), \mathbf{A}}} \alpha(\bar{x}, \bar{y})\left[\bar{b}, \bar{b'}\right]$ por Teorema 10 hay un homomorfismo $h : \left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} \to \left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}}$ tal que $h\left(\bar{a}, \bar{a'}\right) = \left(\bar{b}, \bar{b'}\right)$, y como $\left\langle \bar{a}, \bar{a'} \right\rangle^{\mathbf{A}} = \mathbf{A}$ y $\left\langle \bar{b}, \bar{b'} \right\rangle^{\mathbf{B}} \leq \mathbf{B}$, h es un homomorfismo de $\mathbf{A}$ en $\mathbf{B}$ tal que $h(\bar{a}) = \bar{b}$.

Supongamos que hay un homomorfismo $h : \mathbf{A} \to \mathbf{B}$ tal que $h(\bar{a}) = \bar{b}$. Como $\mathbf{A} \vDash \varphi[\bar{a}]$, entonces $\mathbf{A} \vDash \Delta^+_{(\bar{a}, \bar{a'}), \mathbf{A}}\left[\bar{a}, \bar{a'}\right]$ y por Teorema 8 $\mathbf{B} \vDash \Delta^+_{(\bar{a}, \bar{a'}), \mathbf{A}}\left[h(\bar{a}), h\left(\bar{a'}\right)\right]$. Entonces $\mathbf{B} \vDash \Delta^+_{(\bar{a}, \bar{a'}), \mathbf{A}}\left[\bar{b}, \bar{b'}\right]$, por lo que claramente $\mathbf{B} \vDash \varphi\left[\bar{b}\right]$. □

Teorema 19. *Sean* $\mathcal{L} \subseteq \mathcal{L}'$ *lenguajes de primer orden, sea* $R \in \mathcal{L}' - \mathcal{L}$ *un símbolo de relación n-ario. Para una clase finita* $\mathcal{K}$ *de* $\mathcal{L}'$*-estructuras, los siguientes son equivalentes:*

1. *Hay una* $\mathcal{L}$*-fórmula existencial positiva que define* R *en* $\mathcal{K}$.
2. *Para todas* $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{K}$, *todo homomorfismo* $\gamma : \boldsymbol{A}_{\mathcal{L}} \to \boldsymbol{B}_{\mathcal{L}}$ *preserva* R.

Demostración. Igual a la prueba del Teorema 17, pero utilizando las formulas dadas por el Teorema 18. □

2.7. Definibilidad por conjunción de atómicas y primitivas positivas.

Definición 20. Sean $A_1, \ldots, A_m, B$ conjuntos, $R^{A_i} \subseteq A_i^n$ para cada $i \in [1, m]$, $R^B \subseteq B^n$ y $h : D \subseteq A_1 \times \cdots \times A_m \to B$ una función. Diremos que h *preserva* R si dados

$$\bar{a}_1 = (a_{11}, \ldots, a_{1m}), \ldots, \bar{a}_n = (a_{n1}, \ldots, a_{nm}) \in D$$

tales que

$$(a_{1i}, \ldots, a_{ni}) \in R^{A_i} \text{para cada i} \in \text{[1,m]}$$

se tiene que $(h(\bar{a}_1), \ldots, h(\bar{a}_n)) \in R^B$.

Lema 21. *Sean* $\mathbf{A}_1, \ldots, \mathbf{A}_m$ *estructuras y* $\varphi = \exists \bar{y}\, \psi(\bar{x}, \bar{y})$ *con* ψ *una conjunción de fórmulas atómicas. Entonces son equivalentes:*

1. *Para todo* $i = 1, \ldots, m$ *se da que* $\mathbf{A}_i \vDash \varphi[a_{1i}, \ldots, a_{ni}]$
2. $\mathbf{A}_1 \times \cdots \times \mathbf{A}_m \vDash \varphi[\bar{a}_1, \ldots, \bar{a}_n]$, *con* $\bar{a}_j = (a_{j1}, \ldots, a_{jm})$

Demostración. Rutina. □

Teorema 22. *Sean* $\mathcal{L} \subseteq \mathcal{L}'$ *lenguajes de primer orden, sea* $R \in \mathcal{L}' - \mathcal{L}$ *un símbolo de relación n-ario. Para una clase finita* $\mathcal{K}$ *de* $\mathcal{L}'$*-estructuras, los siguientes son equivalentes:*

1. *Para cada* $m \geq 1$, *para todas* $\mathbf{A}_1, \ldots, \mathbf{A}_m, \boldsymbol{B} \in \mathcal{K}$, *para cada* $\boldsymbol{S} \leq (\mathbf{A}_1 \times \cdots \times \mathbf{A}_m)_{\mathcal{L}}$, *todo homomorfismo* $h : \mathbf{S} \to \boldsymbol{B}_{\mathcal{L}}$ *preserva* R.

2. *Hay una conjunción finita de $\mathcal{L}$-fórmulas atómicas que define R en $\mathcal{K}$.*

Demostración. 2⇒1) Sea $\varphi(\bar{x})$ conjunción de atómicas que define a R en $\mathcal{K}$. Sean $m \geq 1$, $\mathbf{A}_1, \ldots, \mathbf{A}_m, \boldsymbol{B} \in \mathcal{K}$, $\boldsymbol{S} \leq (\mathbf{A}_1 \times \cdots \times \mathbf{A}_m)_{\mathcal{L}}$, y $h : \mathbf{S} \to \boldsymbol{B}_{\mathcal{L}}$ un homomorfismo. Sean

$$\bar{s}_1 = (s_{11}, \ldots, s_{1m}), \ldots, \bar{s}_n = (s_{n1}, \ldots, s_{nm}) \in S$$

tales que

$$(s_{1i}, \ldots, s_{ni}) \in R^{A_i} \text{ para cada } i \in \{1, \ldots, m\}.$$

Veamos que $(h(\bar{s}_1), \ldots, h(\bar{s}_n)) \in R^B$. Como $(s_{1i}, \ldots, s_{ni}) \in R^{A_i}$ y φ define a R en $\mathcal{K}$, tenemos que

$$\mathbf{A}_i \vDash \varphi[s_1, \ldots, s_{ni}] \text{ para } i \in \{1, \ldots, m\}.$$

Luego, por el Lema 21,

$$\mathbf{A}_1 \times \cdots \times \mathbf{A}_m \vDash \varphi[\bar{s}_1, \ldots, \bar{s}_n].$$

Como φ es abierta y $\boldsymbol{S} \leq (\mathbf{A}_1 \times \cdots \times \mathbf{A}_m)_{\mathcal{L}}$, el Teorema 5 implica que $\mathbf{S} \vDash \varphi[\bar{s}_1, \ldots, \bar{s}_n]$, y aplicando el Teorema 8 obtenemos que $\mathbf{B} \vDash \varphi[h(\bar{s}_1), \ldots, h(\bar{s}_n)]$. Finalmente, como φ define a R en $\mathcal{K}$ y $\mathbf{B} \in \mathcal{K}$, concluimos que $(h(\bar{s}_1), \ldots, h(\bar{s}_n)) \in R^{\mathbf{B}}$.

1⇒2) Supongamos $\mathcal{K} = \{\mathbf{A}_1, \ldots, \mathbf{A}_m\}$, R n-aria. Sean $\bar{x}_1, \ldots, \bar{x}_n \in \mathbf{A}_1^{|R^{\mathbf{A}_1}|} \times \cdots \times \mathbf{A}_m^{|R^{\mathbf{A}_m}|}$ tales que $\bar{x}_i = \left(x_{i1}^1, \ldots, x_{i|R^{\mathbf{A}_1}|}^1, \ldots, x_{i1}^m, \ldots, x_{i|R^{\mathbf{A}_m}|}^m\right)$ para cada $i \in \{1, \ldots, m\}$ con $\bigcup R^{\mathbf{A}_j} = \left\{\left(x_{1i}^j, \ldots, x_{ni}^j\right) \text{ con } i \in \left\{1, \left|R^{\mathbf{A}_j}\right|\right\}\right\}$ para cada $i \in \{1, \ldots, m\}$. Sea

$$\varphi = \bigwedge_{\alpha \in \Delta^+_{\mathbf{A}_1^{|R^{\mathbf{A}_1}|} \times \cdots \times \mathbf{A}_m^{|R^{\mathbf{A}_m}|}, (\bar{x}_1, \ldots, \bar{x}_n)}} \alpha(\bar{x})$$

. Supongamos $\mathbf{B} \vDash \varphi[b_1, \ldots, b_n]$ y veamos que $(b_1, \ldots, b_n) \in R^{\mathbf{B}}$. Como $\mathbf{B} \vDash \Delta^+_{\mathbf{A}_1^{|R^{\mathbf{A}_1}|} \times \cdots \times \mathbf{A}_m^{|R^{\mathbf{A}_m}|}, (\bar{x}_1, \ldots, \bar{x}_n)}[b_1, \ldots, b_n]$, por Teorema 10 hay h homomorfismo de $\langle \bar{x}_1, \ldots, \bar{x}_n \rangle^{\mathbf{A}_1^{|R^{\mathbf{A}_1}|} \times \cdots \times \mathbf{A}_m^{|R^{\mathbf{A}_m}|}}$ en $\langle b_1, \ldots, b_n \rangle^{\mathbf{B}}$ tal que $h(\bar{x}_i) = b_i$. Como por hipótesis h preserva R, entonces como $\left(x_{1i}^j, \ldots, x_{ni}^j\right) \in R^{\mathbf{A}_j}$, $(h(\bar{x}_1), \ldots, h(\bar{x}_n)) \in R^{\mathbf{B}}$, que es exactamente $(b_1, \ldots, b_n) \in R^{\mathbf{B}}$.

Ahora supongamos $(b_1, \ldots, b_n) \in R^{\mathbf{B}}$ y veamos que $\mathbf{B} \vDash \varphi[b_1, \ldots, b_n]$, entonces hay algún j tal que $\mathbf{A}_j = \mathbf{B}$ y hay algún k y algún j tal que

$$\left(x_{1k}^j, \ldots, x_{nk}^j\right) = (b_1, \ldots, b_n)$$

. Entonces como $\mathbf{A}_1^{|R^{\mathbf{A}_1}|} \times \cdots \mathbf{B}^{|R^{\mathbf{B}}|} \cdots \times \mathbf{A}_m^{|R^{\mathbf{A}_m}|} \vDash \varphi[\bar{x}_1, \ldots, \bar{x}_n]$ por Teorema 21 $\mathbf{B} \vDash \varphi\left[x_{1k}^j, \ldots, x_{nk}^j\right]$, por lo tanto $\mathbf{B} \vDash \varphi[b_1, \ldots, b_n]$. □

Teorema 23. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden, sea $R \in \mathcal{L}' - \mathcal{L}$ un símbolo de relación n-ario. Para una clase finita $\mathcal{K}$ de $\mathcal{L}'$-estructuras, los siguientes son equivalentes:*

1. *Para cada $m \geq 1$, para todas $\mathbf{A}_1, \ldots, \mathbf{A}_m, \boldsymbol{B} \in \mathcal{K}$, todo homomorfismo $h : (\mathbf{A}_1 \times \cdots \times \mathbf{A}_m)_{\mathcal{L}} \to \boldsymbol{B}_{\mathcal{L}}$ preserva R.*
2. *Hay una $\mathcal{L}$-fórmula primitiva positiva que define R en $\mathcal{K}$.*

Demostración. 2⇒1) Sea $\varphi = \exists \bar{y}\, \psi(\bar{x}, \bar{y})$ primitiva positiva que define a R en $\mathcal{K}$. Sean $m \geq 1$, $\mathbf{A}_1, \ldots, \mathbf{A}_m, \boldsymbol{B} \in \mathcal{K}$ y $h : (\mathbf{A}_1 \times \cdots \times \mathbf{A}_m)_{\mathcal{L}} \to \boldsymbol{B}_{\mathcal{L}}$ un homomorfismo. Sean

$$\bar{a}_1 = (a_{11}, \ldots, a_{1m}), \ldots, \bar{a}_n = (a_{n1}, \ldots, a_{nm}) \in S$$

tales que

$$(a_{1i},\dots,a_{ni})\in R^{A_i} \text{ para cada } i\in\{1,\dots,m\}.$$

Veamos que $(h(\bar{a}_1),\dots,h(\bar{a}_n))\in R^B$. Como $(a_{1i},\dots,a_{ni})\in R^{A_i}$ y φ define a R en $\mathcal{K}$, tenemos que

$$\mathbf{A}_i\vDash\varphi[a_1,\dots,a_{ni}] \text{ para } i\in\{1,\dots,m\}.$$

Luego, por el Lema 21,

$$\mathbf{A}_1\times\cdots\times\mathbf{A}_m\vDash\varphi[\bar{a}_1,\dots,\bar{a}_n].$$

Entonces hay $b_1,\dots,b_k\in A_1\times\cdots\times A_m$, tales que

$$\mathbf{A}_1\times\cdots\times\mathbf{A}_m\vDash\psi[\bar{a}_1,\dots,\bar{a}_n,b_1,\dots,b_k].$$

Como ψ es abierta positiva, aplicando el Teorema 8 obtenemos que $\mathbf{B}\vDash\varphi[h(\bar{a}_1),\dots,h(\bar{a}_n)]$. Finalmente, como φ define a R en $\mathcal{K}$ y $\mathbf{B}\in\mathcal{K}$, concluimos que $(h(\bar{a}_1),\dots,h(\bar{a}_n))\in R^{\mathbf{B}}$.

1⇒2)Supongamos $\mathcal{K}=\{\mathbf{A}_1,\dots,\mathbf{A}_m\}$, R n-aria. Sean $\bar{x}_1,\dots,\bar{x}_n\in\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}$ tales que $\bar{x}_i=\left(x^1_{i1},\dots,x^1_{i|R^{\mathbf{A}_1}|},\dots,x^m_{i1},\dots,x^m_{i|R^{\mathbf{A}_m}|}\right)$ para cada $i\in\{1,\dots,m\}$ con $\bigcup R^{\mathbf{A}_j}=\left\{\left(x^j_{1i},\dots,x^j_{ni}\right) \text{ con } i\in\{1,|R^{\mathbf{A}_j}|\}\right\}$ para cada $i\in\{1,\dots,m\}$. Sean $\bar{x'}_1,\dots,\bar{x'}_k\in\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}$, tales que $\langle\bar{x}_1,\dots,\bar{x}_n,\bar{x'}_1,\dots,\bar{x'}_k\rangle^{\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}}=\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}$. Sea

$$\varphi=\exists\bar{y}\bigwedge_{\alpha\in\Delta^+_{\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|},(\bar{x}_1,\dots,\bar{x}_n,\bar{x'}_1,\dots,\bar{x'}_k)}}\alpha(\bar{x},\bar{y})$$

. Supongamos $\mathbf{B}\vDash\varphi[b_1,\dots,b_n]$ y veamos que $(b_1,\dots,b_n)\in R^{\mathbf{B}}$. Entonces hay $b'_1,\dots,b'_k$ tal que $\mathbf{B}\vDash\Delta^+_{\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|},(\bar{x}_1,\dots,\bar{x}_n,\bar{x'}_1,\dots,\bar{x'}_k)}[b_1,\dots,b_n,b'_1,\dots,b'_k]$, por Teorema 10 hay h homomorfismo de $\langle\bar{x}_1,\dots,\bar{x}_n,\bar{x'}_1,\dots,\bar{x'}_k\rangle^{\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}}$ en $\langle b_1,\dots,b_n,b'_1,\dots,b'_k\rangle^{\mathbf{B}}$ tal que $h(\bar{x}_i)=b_i$ y $h\left(\bar{x'}_i\right)=b'_i$, claramente h es un homomorfismo de $\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}$ en $\mathbf{B}$. Como por hipótesis h preserva R, entonces como $\left(x^j_{1i},\dots,x^j_{ni}\right)\in R^{\mathbf{A}_j}$, $(h(\bar{x}_1),\dots,h(\bar{x}_n))\in R^{\mathbf{B}}$, que es exactamente $(b_1,\dots,b_n)\in R^{\mathbf{B}}$.

Ahora supongamos $(b_1,\dots,b_n)\in R^{\mathbf{B}}$ y veamos que $\mathbf{B}\vDash\varphi[b_1,\dots,b_n]$, entonces hay algún j tal que $\mathbf{A}_j=\mathbf{B}$ y hay algún k y algún j tal que

$$\left(x^j_{1k},\dots,x^j_{nk}\right)=(b_1,\dots,b_n)$$

. Entonces como $\mathbf{A}_1^{|R^{\mathbf{A}_1}|}\times\cdots\times\mathbf{B}^{|R^{\mathbf{B}}|}\times\cdots\times\mathbf{A}_m^{|R^{\mathbf{A}_m}|}\vDash\varphi[\bar{x}_1,\dots,\bar{x}_n]$ por Teorema 21 $\mathbf{B}\vDash\varphi\left[x^j_{1k},\dots,x^j_{nk}\right]$, por lo tanto $\mathbf{B}\vDash\varphi[b_1,\dots,b_n]$. □

3. Algoritmos de chequeo de definibilidad

En este capitulo describimos algoritmos para el chequeo de definibilidad de relaciones en conjuntos finitos de estructuras finitas por fragmentos de primer orden.

3.1. Definiciones e ideas preliminares. Para evitar las complicaciones resultantes de que tanto una estructura $\mathbf{A}$ y una subestructura propia $\mathbf{A}_0$ pertenezcan al conjunto $\mathcal{K}$, definiremos el concepto de clase *disjunta.* En general asumiremos que nuestra clase $\mathcal{K}$ es siempre disjunta. Notar que no hay perdida de generalidad ya que bastaria con renombrar elementos para convertir una clase no disjunta en una disjunta equivalente. Una clase $\mathcal{K}$ de estructuras será *disjunta* si dadas $\mathbf{A}, \mathbf{B} \in \mathcal{K}$ se tiene que $A \cap B \neq \emptyset$ si y solo si $\mathbf{A} = \mathbf{B}$. La clase $\mathcal{K}$ será *normal* si es disjunta, finita y cada una de las estructuras en $\mathcal{K}$ es finita.

A la hora de reducir la complejidad de nuestros algoritmos en la busqueda de homomorfismos utilizaremos conjuntos reducidos de flechas, que *generan* el resto. Dados $\mathcal{F}_0$ y $\mathcal{F}$ dos conjuntos de funciones, diremos que $\mathcal{F}_0$ *genera* a $\mathcal{F}$ si $\mathcal{F}_0 \subseteq \mathcal{F}$ y para toda $f \in \mathcal{F}$ hay $f_1, \ldots, f_n \in \mathcal{F}_0$(quizás repetidas) tales que $f = f_1 \circ \cdots \circ f_n$.

Dado un conjunto $\mathcal{K}$ de estructuras, para referirnos a las flechas nombradas en los Teoremas 11, 13, 19, 15 definimos los siguientes conjuntos:

$$\operatorname{iso}(\mathcal{K}) = \{\gamma : \gamma \text{ isomorfismo de } \mathbf{A} \text{ en } \mathbf{B} \text{ con } \mathbf{A}, \mathbf{B} \in \mathcal{K}\}$$

$$\operatorname{sub iso}(\mathcal{K}) = \{\gamma : \gamma \text{ isomorfismo de } \mathbf{A}_0 \text{ en } \mathbf{B}_0 \text{ con } \mathbf{A}_0, \mathbf{B}_0 \in \mathbb{S}(\mathcal{K})\}$$

$$\hom(\mathcal{K}) = \{\gamma : \gamma \text{ homomorfismo de } \mathbf{A} \text{ en } \mathbf{B} \text{ con } \mathbf{A}, \mathbf{B} \in \mathcal{K}\}$$

$$\operatorname{sub hom}(\mathcal{K}) = \{\gamma : \gamma \text{ homomorfismo de } \mathbf{A}_0 \text{ en } \mathbf{B}_0 \text{ con } \mathbf{A}_0, \mathbf{B}_0 \in \mathbb{S}(\mathcal{K})\}$$

3.2. Preprocesamiento. Una vez que tenemos una clase normal $\mathcal{K}$ en la que chequear definibilidad tratamos de reducir su redundancia basandonos en el siguiente resultado:

Lema 24. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden, $\mathcal{K}$ una clase normal de $\mathcal{L}'$-estructuras y $\mathbf{A}, \tilde{\mathbf{A}} \in \mathcal{K}$, con $\mathbf{A} \neq \tilde{\mathbf{A}}$, tales que hay un $\mathcal{L}$-isomorfismo $\gamma : A \to \tilde{A}$. Entonces para toda $R \in \mathcal{L}' - \mathcal{L}$ y toda $\mathcal{L}$-fórmula φ son equivalentes:*

1. *φ define a R en $\mathcal{K}$.*
2. *γ es un $\mathcal{L} \cup \{R\}$-isomorfismo y φ define a R en $\mathcal{K} - \{\tilde{\mathbf{A}}\}$.*

Demostración. 1⇒2) Trivial.

2⇒1) $\tilde{\mathbf{A}} \vDash R[\bar{a}]$ sii $\mathbf{A} \vDash R\left[\gamma^{-1}(\bar{a})\right]$ sii $\mathbf{A} \vDash \varphi\left[\gamma^{-1}(\bar{a})\right]$ sii $\tilde{\mathbf{A}} \vDash \varphi[\bar{a}]$. □

Independientemente del tipo de definibilidad que uno quiera chequear el Lema 24 sugiere que uno puede comenzar por reducir $\mathcal{K}$ eliminando copias de estructuras $\mathcal{L} \cup \{R\}$-isomorfas. El Algoritmo 1 aprovecha este resultado aprovechando ademas para chequear por $\mathcal{L}$-isomorfismos y al descubir uno de estos comprobar si es un $\mathcal{L} \cup \{R\}$-isomorfismo, lo que permite encontrar contrajemplos de definibilidad desde esta etapa.

Notar que los isomorfismos revisados por este algoritmo deben ser necesariamente revisados para comprobar definibilidad en cualquiera de los formatos, con la ventaja de que al chequear estos en primer lugar, podría reducirse la clase $\mathcal{K}$ en el proceso.

Observar además que, para poder aplicar los teoremas de definibilidad de las secciones 2.3 y 2.4, habría que verificar que cada uno de los $\mathcal{L}$-isomorfismos entre estructuras de $\mathcal{K}$ preserven R y vemos que basta con chequear sólo uno gracias al Lema 24.

Para ejemplificar el funcionamiento del algoritmo, supongamos que $\mathcal{K}$ esta dado por la Figura 3.1, donde cada tipo de isomorfismo esta representado por una forma geométrica.

Algorithm 1 Preprocesamiento

for $\mathbf{A} \in \mathcal{K}$ **do**
    for $\mathbf{B} \in \mathcal{K} - \{\mathbf{A}\}$ **do**
        if hay $\gamma : \mathbf{A} \to \mathbf{B}$ $\mathcal{L}$-isomorfismo **then**
            if γ es un $\mathcal{L} \cup \{R\}$-isomorfismo **then**
                $\mathcal{K} = \mathcal{K} - \{\mathbf{B}\}$
            else
                return γ $\qquad\triangleright$ R no es definible y γ es contraejemplo
            end if
        end if
    end for
end for
return $\mathcal{K}$ $\qquad\triangleright$ $\mathcal{K}$ sin estructuras $\mathcal{L}$-isomorfas

Figura 3.1. $\mathcal{K}$ antes del preprocesamiento

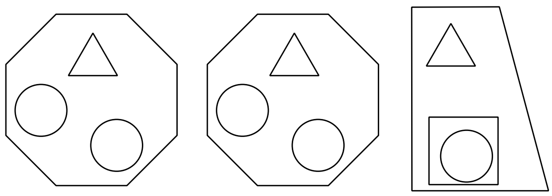
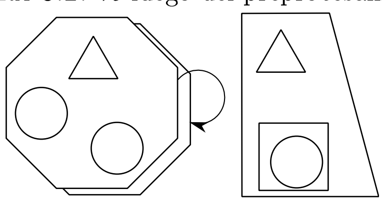


Figura 3.2. $\mathcal{K}$ luego del preprocesamiento



El algoritmo de preprocesamiento detecta el $\mathcal{L}$-isomorfismo entre los dos octagonos y revisa que sea tambien un $\mathcal{L} \cup \{R\}$-isomorfismo, dando lugar a la Figura 3.2, donde la flecha es el isomorfismo a chequear.

3.3. Definibilidad abierta. A partir del Teorema 13, habria que chequear todo isomorfismo entre subestructuras de estructuras de $\mathcal{K}$, como en el siguiente lema.

Lema 25. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden tales que $R \in \mathcal{L}' - \mathcal{L}$ es un símbolo de relación n-ario y $\mathcal{K}$ un conjunto normal de $\mathcal{L}'$-estructuras. Entonces si cada $\gamma \in \operatorname{sub iso}(\mathcal{K}_{\mathcal{L}})$ preserva R, R es definible por una $\mathcal{L}$-fórmula abierta en $\mathcal{K}$.*

Demostración. Directo del Teorema 13.

En el siguiente teorema describimos un conjunto suficiente de flechas para generar $\operatorname{sub iso}(\mathcal{K}_{\mathcal{L}})$. Notar que al tomar $\mathcal{K}$ como un conjunto normal de $\mathcal{L}$-estructuras sin estructuras isomorfas estamos pensando en la aplicacion previa del Algoritmo 1, basado en el Lema 24. $\square$

Teorema 26. *Sea $\mathcal{K}$ un conjunto normal de $\mathcal{L}$-estructuras donde no hay estructuras isomorfas. Sea $\mathcal{S} \subseteq \mathbb{S}(\mathcal{K})$ tal que contiene exactamente un representante por cada tipo de isomorfismo en $S(\mathcal{K})$. Sea $\mathcal{F} \subseteq \operatorname{sub iso}(\mathcal{K})$ tal que:*

1. *para cada $\mathbf{S} \in \mathcal{S}$ se tiene que $\operatorname{aut}(\mathbf{S}) \subseteq \mathcal{F}$,*

2. *para cada* $\mathbf{A} \in S(\mathcal{K}) - \mathcal{S}$ *hay* $\gamma, \gamma^{-1} \in \mathcal{F}$ *para algún isomorfismo* γ *de* $\mathbf{A}$ *en el representante del tipo de isomorfismo de* $\mathbf{A}$ *en* $\mathcal{S}$.

Entonces $\mathcal{F}$ *genera* $\operatorname{sub}\operatorname{iso}(\mathcal{K})$.

Demostración. Sea $\gamma \in \operatorname{sub}\operatorname{iso}(\mathcal{K})$, entonces $\gamma : \mathbf{A}_0 \leq \mathbf{A} \to \mathbf{B}_0 \leq \mathbf{B}$ isomorfismo, con $\mathbf{A}, \mathbf{B} \in \mathcal{K}$.

Si $\mathbf{A}_0, \mathbf{B}_0 \in \mathcal{S}$, como no hay estructuras isomorfas en $\mathcal{S}$, $\mathbf{A}_0 = \mathbf{B}_0$, por lo tanto $\gamma \in \operatorname{aut}(\mathbf{A}_0) \subseteq \mathcal{F}$.

Si $\mathbf{A}_0 \in \mathcal{S}$ pero $\mathbf{B}_0 \notin \mathcal{S}$, entonces $\mathbf{A}_0$ es el representante de $\mathbf{B}_0$ en $\mathcal{S}$, y por lo tanto hay un isomorfismo $\delta : \mathbf{B}_0 \to \mathbf{A}_0$ tal que $\delta, \delta^{-1} \in \mathcal{F}$. Claramente $\delta\gamma = \lambda \in \operatorname{aut}(\mathbf{A}_0) \subseteq \mathcal{F}$, por lo tanto $\gamma = \delta^{-1}\lambda$.

Si $\mathbf{A}_0, \mathbf{B}_0 \notin \mathcal{S}$, hay $\mathbf{C}_0 \in \mathcal{S}$ que es representante de $\mathbf{A}_0$ y $\mathbf{B}_0$. Luego hay $\delta : \mathbf{A}_0 \to \mathbf{C}_0$ y $\delta' : \mathbf{B}_0 \to \mathbf{C}_0$ tales que $\delta, \delta^{-1}, \delta', \delta'^{-1} \in \mathcal{F}$. Claramente $\delta'\gamma\delta^{-1} = \lambda \in \operatorname{aut}(\mathbf{C}_0) \subseteq \mathcal{F}$, por lo tanto $\gamma = \delta'^{-1}\lambda\delta$. □

Corolario 27. *Sean* $\mathcal{L} \subseteq \mathcal{L}'$ *lenguajes de primer orden tales que* $R \in \mathcal{L}' - \mathcal{L}$ *es un símbolo de relación n-ario y* $\mathcal{K}$ *un conjunto normal de* $\mathcal{L}'$*-estructuras. Supongamos* $\mathcal{F}$ *es como en el Teorema 26 para la familia* $\mathcal{K}_{\mathcal{L}}$*. Entonces si cada* $\gamma \in \mathcal{F}$ *preserva* R*, se tiene que* R *es definible por una* $\mathcal{L}$*-fórmula abierta en* $\mathcal{K}$.

Demostración. Directa ya que la composición de funciones que preservan R, trivialmente tambien preserva R. Luego aplicando el Teorema26 y el Lema 25 queda probado. □

El Teorema 26 sugiere un subconjunto de isomorfismos que bastan para revisar definibilidad abierta a través del Corolario 27.

En el Algoritmo 2 se construye $\mathcal{S}$ a la vez que se van revisando los isomorfismos en $\mathcal{F}$. Para esto recorremos las estructuras en $\mathcal{K}$ de mayor a menor tamaño. Al momento de procesar una $\mathbf{A} \in \mathcal{K}$ lo primero que se hace es buscar un $\mathcal{L}$-isomorfismo γ entre $\mathbf{A}$ y un miembro $\mathcal{S}$. Si un tal γ existe, tanto γ como su inversa son revisados (miembros de $\mathcal{F}$ correspondientes al punto (2) del Teorema 26). Si no hay un tal γ, la estructura $\mathbf{A}$ es agregada a $\mathcal{S}$ y sus $\mathcal{L}$-automorfismos son revisados (miembros de $\mathcal{F}$ correspondientes al punto (1) del Teorema 26). A continuación las subestructuras de $\mathbf{A}$ son procesadas de la misma manera.

Notar que en la linea 14 no se necesita revisar si los automorfismos son $\mathcal{L} \cup \{R\}$-automorfismos ya que, como se recorren todos su inversa también es revisada por preservación.

Además, una vez que una subestructura resulta ser $\mathcal{L} \cup \{R\}$-isomorfa a un representante en $\mathcal{S}$, sabemos que todas sus subestructuras ya tienen representante en $\mathcal{S}$ y que los isomorfismos internos preservan. Esto permite disminuir la cantidad de subestructuras a revisar, como se ve en la linea 10.

Algorithm 2 Chequeo de definibilidad abierta

$\mathcal{S} = \emptyset$
for $\mathbf{A} \in \mathcal{K}$, desde la mayor a la menor cardinalidad **do**
  $Sub = [\mathbf{C} : \mathbf{C} \in \mathbb{S}_{\mathcal{L}}(\mathbf{A})\,,$ de la mayor a menor cardinalidad]
  for $\mathbf{B} \in Sub$ **do**
    (iso, ce) = TIENEREPRESENTANTE($\mathbf{B}, \mathcal{S}$) ▷ hay un $\mathcal{L}$-iso de $\mathbf{B}$ en $\mathbf{S} \in \mathcal{S}$
    if ce $\neq$ Null **then**
      return ce ▷ ce es contraejemplo
    end if
    if iso **then**
      $Sub = [\mathbf{C} : \mathbf{C} \in Sub$ tal que $C \nsubseteq B]$
    else
      $\mathcal{S} = \mathcal{S} \cup \{\mathbf{B}\}$
      for $\alpha \in Aut_{\mathcal{L}}(\mathbf{B})$ **do**
        if α no preserva R **then**
          return α ▷ α es contraejemplo
        end if
      end for
    end if
  end for
end for
return ▷ R es abierta-definible

function TIENEREPRESENTANTE($\mathbf{A}_0, \mathcal{S}$)
  for $\mathbf{S} \in \mathcal{S}$, con $|\mathbf{A}_0| = |\mathbf{S}|$ **do**
    if hay $\gamma : \mathbf{A}_0 \to \mathbf{S}$ $\mathcal{L}$-isomorfismo **then**
      if γ no es un $\mathcal{L} \cup \{R\}$-isomorfismo **then**
        return (True, γ) ▷ γ es contraejemplo
      else
        return (True, Null) ▷ γ es isomorfismo con el representante
      end if
    end if
  end for
  return (False, Null) ▷ No tiene representante
end function

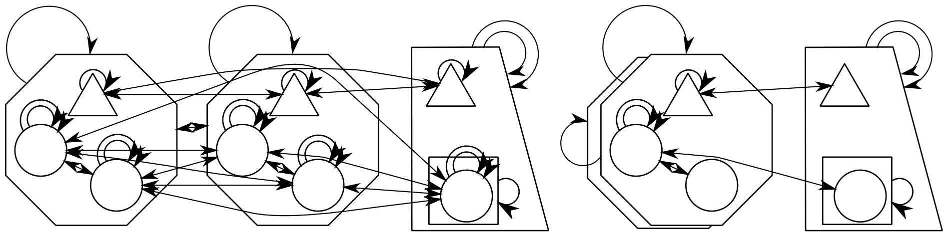
Para esquematizar la ganancia obtenida en cuanto al chequeo de una menor cantidad de flechas podemos observar en la Figura 3.4a el conjunto $\mathcal{K}$ junto con todas las flechas que deberían ser revisadas segun el Teorema 13. Mientras que en la Figura 3.4b, se puede ver las flechas que bastan para comprobar definibilidad abierta segun el Corolario 27 a la manera del algoritmo anterior.

Notar que en la Figura 3.4b, la aplicacion del Algoritmo 1 (Preprocesamiento) genera que uno de los octagonos aparezca como *sombra* del otro y baste solo revisar el isomorfismo entre ellos por el Lema 24.

3.4. Definibilidad abierta-positiva. A partir del Teorema 15, habría que chequear todo homomorfismo entre subestructuras de estructuras de $\mathcal{K}$, como en el siguiente lema.

Lema 28. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden tales que $R \in \mathcal{L}' - \mathcal{L}$ es un símbolo de relación n-ario y $\mathcal{K}$ un conjunto normal de $\mathcal{L}'$-estructuras. Entonces si cada $\gamma \in \operatorname{sub}\hom(\mathcal{K}_{\mathcal{L}})$ preserva R, R es definible por una $\mathcal{L}$-fórmula abierta positiva en $\mathcal{K}$.*

FIGURA 3.3



(A) Flechas a revisar segun el Teorema 13

(B) Flechas que bastan segun el Corolario 27

Demostración. Directo del Teorema 15.

En el siguiente teorema describimos un conjunto suficiente de flechas para generar $\operatorname{sub}\hom(\mathcal{K}_{\mathcal{L}})$. Notar que, nuevamente, al tomar $\mathcal{K}$ como un conjunto normal de $\mathcal{L}$-estructuras sin estructuras isomorfas estamos pensando en la aplicación previa del Algoritmo 1, basado en el Lema 24. □

Teorema 29. *Sea $\mathcal{K}$ un conjunto normal de $\mathcal{L}$-estructuras donde no hay estructuras isomorfas. Sea $\mathcal{S} \subseteq \mathbb{S}(\mathcal{K})$ tal que contiene exactamente un representante por cada tipo de isomorfismo en $\mathbb{S}(\mathcal{K})$. Sea $\mathcal{H} \subseteq \operatorname{sub}\hom(\mathcal{K})$ tal que:*

- *$\mathcal{F} \subseteq \mathcal{H}$ con $\mathcal{F}$ como en el Teorema 26,*
- *para cada $\mathbf{A}, \mathbf{B} \in \mathcal{S}$ todo $h : \mathbf{A} \to \mathbf{B}$ homomorfismo sobreyectivo esta en $\mathcal{H}$.*

Entonces $\mathcal{H}$ genera $\operatorname{sub}\hom(\mathcal{K})$.

Demostración. Sea $h \in \operatorname{sub}\hom(\mathcal{K})$, entonces $h : \mathbf{A}_0 \leq \mathbf{A} \to \mathbf{B}_0 \leq \mathbf{B}$ homomorfismo, con $\mathbf{A}, \mathbf{B} \in \mathcal{K}$.

Si $\mathbf{A}_0, h(\mathbf{A}_0) \in \mathcal{S}$, h es homomorfismo sobreyectivo entre estructuras de $\mathcal{S}$, por lo que $h \in \mathcal{H}$.

Si $\mathbf{A}_0 \in \mathcal{S}$ pero $h(\mathbf{A}_0) \notin \mathcal{S}$, hay $\mathbf{S}$ representante de $h(\mathbf{A}_0)$ en $\mathcal{S}$ y un isomorfismo $\delta : h(\mathbf{A}_0) \to \mathbf{S}$ tal que $\delta, \delta^{-1} \in \mathcal{F} \subseteq \mathcal{H}$. Luego $\delta^{-1}h = f$ donde f es un homomorfismo claramente sobreyectivo de $\mathbf{A}_0$ en $\mathbf{S}$, por lo que $f \in \mathcal{H}$. Finalmente $h = \delta f$.

Si $\mathbf{A}_0, h(\mathbf{A}_0) \notin \mathcal{S}$, hay respectivos $\mathbf{S}, \mathbf{S}' \in \mathcal{S}$ representantes de $\mathbf{A}_0$ y $h(\mathbf{A}_0)$. Luego hay $\delta : \mathbf{A}_0 \to \mathbf{S}$ y $\delta' : h(\mathbf{A}_0) \to \mathbf{S}'$ tales que $\delta, \delta^{-1}, \delta', \delta'^{-1} \in \mathcal{F} \subseteq \mathcal{H}$. Luego $\delta' h \delta^{-1} = f$ donde f es un homomorfismo claramente sobreyectivo de $\mathbf{S}$ en $\mathbf{S}'$, por lo que $f \in \mathcal{H}$. Finalmente $h = \delta'^{-1} f \delta$. □

Corolario 30. *Sean $\mathcal{L} \subseteq \mathcal{L}'$ lenguajes de primer orden tales que $R \in \mathcal{L}' - \mathcal{L}$ es un símbolo de relación n-ario y $\mathcal{K}$ un conjunto normal de $\mathcal{L}'$-estructuras. Supongamos $\mathcal{F}$ es como en el Teorema 29 para la familia $\mathcal{K}_{\mathcal{L}}$. Entonces si cada $\gamma \in \mathcal{F}$ preserva R, se tiene que R es definible por una $\mathcal{L}$-fórmula abierta positiva en $\mathcal{K}$.*

Demostración. Directa ya que la composición de funciones que preservan R, trivialmente tambien preserva R. Luego aplicando el Teorema29 y el Lema 28 queda probado. □

El Teorema 29 sugiere un subconjunto de homomorfismos que bastan para revisar definibilidad abierta positiva a través del Corolario 30. A su vez, a la hora de buscar homomorfismos entre estructuras aprovechamos el siguiente resultado, que nos permite organizar la busqueda de homorfismos biyectivos entre dos subestructuras en un unico sentido.

Teorema 31 (Las estructuras finitas cumplen la propiedad de Cantor-Bernstein)**.** *Dadas* $\mathbf{A}$ *y* $\mathbf{B}$ *estructuras finitas, si hay* $f : A \to B$ *y* $g : B \to A$ *homomorfismos inyectivos entonces hay un* $\gamma : \mathbf{A} \to \mathbf{B}$ *isomorfismo.*

Demostración. Sean $\mathbf{A}, \mathbf{B}$ estructuras finitas y $f : A \to B$ y $g : B \to A$ homomorfismos inyectivos. Notar que gf es una permutación de $\mathbf{A}$, y como $\mathbf{A}$ es finito hay $k \geq 1$ tal que $(gf)^k = \mathrm{Id}_A$. Luego como $g^{-1} = f(gf)^{k-1}$ vemos que g^{-1} es homomorfismo. □

En el Algoritmo 3 se construye $\mathcal{S}$ a la vez que se van revisando los homomorfismos en $\mathcal{F}$. Para esto recorremos las estructuras en $\mathcal{K}$ de mayor a menor tamaño. Al momento de procesar una $\mathbf{A} \in \mathcal{K}$, mientras que para definibilidad abierta lo primero que se hacia era buscar un $\mathcal{L}$-isomorfismo para cada $\mathbf{S} \in \mathcal{S}$, ahora se buscan homomorfismos biyectivos primero, chequeandolos por preservacion. Esto da lugar a tres casos:

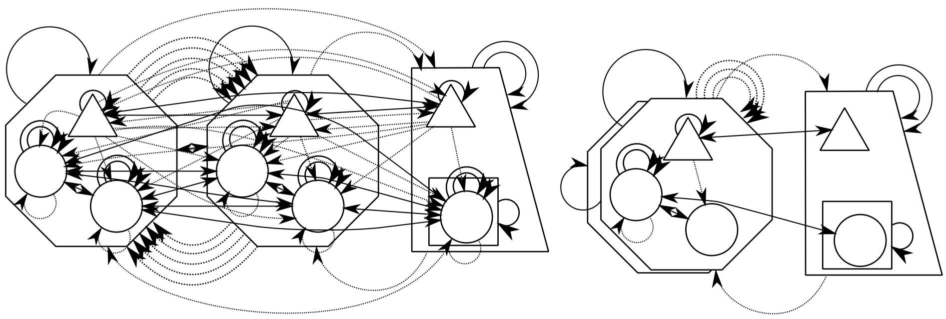
1. Uno de estos homomorfismos biyectivos es un $\mathcal{L}$-isomorfismo.
2. Hay homomorfismos biyectivos de $\mathbf{A}$ en $\mathbf{S}$ pero ninguno es un $\mathcal{L}$-isomorfismo.
3. No hay homomorfismos biyectivos de $\mathbf{A}$ en $\mathbf{S}$.

En el primer caso se procede como en el Algoritmo 2. En el segundo caso podemos estar seguros de que no hay homomorfismos biyectivos de $\mathbf{S}$ en $\mathbf{A}$, ya que por el Teorema 31, deberia haber un $\mathcal{L}$-isomorfismo entre ellos. Mientras que en el tercer caso deberemos buscar homomorfismos biyectivos de $\mathbf{S}$ en $\mathbf{A}$, para revisar su preservacion.

Esto nos permite estar seguros de que todos los homomorfismos biyectivos entre estructuras de $\mathcal{S}$ han sido chequeados. Finalmente lo unico que falta es chequear los homomorfismos sobreyectivos pero no inyectivos entre estructuras de $\mathcal{S}$. Para lo cual basta con buscar homomorfismos sobreyectivos de $\mathbf{A}$ en $\mathbf{B}$ tales que $|A| > |B|$.

Para esquematizar la ganancia obtenida en cuanto al chequeo de una menor cantidad de flechas podemos observar en la Figura 3.5a el conjunto $\mathcal{K}$ junto con todas las flechas que deberían ser revisadas segun el Teorema 15, donde las flechas punteadas son los homomorfismos. Mientras que en la Figura 3.5b, se puede ver las flechas que bastan para comprobar definibilidad abierta segun el Teorema 29 a la manera del algoritmo anterior.

FIGURA 3.4



(A) Flechas a revisar segun el Teorema 15

(B) Flechas que bastan segun el Teorema 29

3.5. Definibilidad de primer orden. Por el Teorema 11, luego de haber aplicado el Algoritmo 1 para preprocesamiento, solo basta con revisar los automorfimos para $\mathbf{A} \in \mathcal{K}$, como hacemos en el Algoritmo 4.

Algorithm 3 Chequeo de definibilidad abierta-positiva

```
S = ∅
for A ∈ K, desde la mayor a la menor cardinalidad do
    Sub = [C : C ∈ S_L (A) , de la mayor a menor cardinalidad]
    for B ∈ Sub do
        (iso,ce) = tieneRepresentante(B, S)   ▷ hay un L-iso de B en S ∈ S
        if ce ≠ null then
            return ce                          ▷ ce es contraejemplo
        end if
        if iso then
            Sub = [C : C ∈ Sub tal que C ⊈ B]
        else
            S = S ∪ {B}
            for α ∈ Aut_L(B) do
                if α no preserva R then
                    return α                   ▷ α es contraejemplo
                end if
            end for
        end if
    end for
end for
for A ∈ S do
    for B ∈ S, con |B| < |A| do
        for γ : A → B con γ L-homomorfismo sobreyectivo do
            if γ no preserva R then
                return γ                       ▷ γ es contraejemplo
            end if
        end for
    end for
end for
return                                         ▷ R es definible por una abierta positiva

function tieneRepresentante(A_0, S)
    for S ∈ S, con |A_0| = |S| do
        bihomos =[γ: γ : A_0 → S con γ L-homomorfismo biyectivo
        for γ ∈ bihomos do
            if γ es un L-isomorfismo, con S ∈ S then
                if γ no es un L ∪ {R}-isomorfismo then
                    return (True, γ)           ▷ γ es contraejemplo
                else
                    return (True, Null)        ▷ γ es isomorfismo con el representante
                end if
            else if γ no preserva R then
                return (False, γ)              ▷ γ es contraejemplo
            end if
        end for
        if bihomos = [] then
            for γ : S → A_0 con γ L-homomorfismo biyectivo do
                if γ no preserva R then
                    return (False, γ)          ▷ γ es contraejemplo
                end if
            end for
        end if
        return (False, Null)                   ▷ No tiene representante
    end for
end function
```

Algorithm 4 Chequeo de definibilidad de primer orden

```
for A ∈ 𝒦 do
    for α ∈ Aut_ℒ(A) do
        if α no es un ℒ ∪ {R}-automorfismo then
            return α                                  ▷ α es contraejemplo
        end if
    end for
end for
return                                                ▷ R es definible en primer orden
```

3.6. Definibilidad existencial. Por el Teorema 17, luego de aplicar el Algoritmo 1 de preprocesamiento, solo basta con revisar los embeddings entre $\mathbf{A}, \mathbf{B} \in \mathcal{K}$, como hacemos en el Algoritmo 5.

Algorithm 5 Chequeo de definibilidad existencial

```
for A ∈ 𝒦 do
    for B ∈ 𝒦 do
        for γ ∈ Emb_ℒ (A, B) do
            if γ no es un ℒ ∪ {R}-embedding then
                return γ                              ▷ γ es contraejemplo
            end if
        end for
    end for
end for
return                                                ▷ R es existencial definible
```

3.7. Definibilidad existencial-positiva. Por el Teorema 19, luego de aplicar el Algoritmo 1 de preprocesamiento, solo basta con revisar los homomorfismos cada par $\mathbf{A}, \mathbf{B} \in \mathcal{K}$, como hacemos en el Algoritmo 6.

Algorithm 6 Chequeo de definibilidad existencial-positiva

```
for A ∈ 𝒦 do
    for B ∈ 𝒦 do
        for α ∈ Hom_ℒ(A, B) do
            if h no preserva R then
                return h                              ▷ h es contraejemplo
            end if
        end for
    end for
end for
return                                                ▷ R es existencial positiva definible
```

3.8. Detección de homomorfismos. Para resolver CSP utilizamos Minion.

Demostración. Sea γ un endomorfismo inyectivo de $\mathbf{A}$. Como A es finito tenemos que γ es sobre y además $\gamma^{-1} = \gamma^k$ para algún $k \geq 1$. $\square$

3.8.1. CSP. Como aparece en Wikipedia 2016, un CSP (constraint satisfaction problem) es definido como una terna $\langle X, D, C\rangle$, donde

$$X = \{X_1, \ldots, X_n\} \text{ es un conjunto de variables}$$

$$D = \{D_1, \ldots, D_n\} \text{ es un conjunto con los respectivos dominios de los valores}$$

$$C = \{C_1, \ldots, C_m\} \text{ es un conjunto de restricciones}$$

Cada variable X_i se mueve en los valores del respectivo dominio no vacío D_i. Cada restricción $C_j \in C$ es un par $\langle t_j, R_j\rangle$, donde t_j es una k-upla de variables y R_j es una relación k-aria en el correspondiente dominio de cada variable. Una valuación sobre las variables es una función desde un subconjunto de las variables a un particular conjunto de valores en los correspondientes dominios de valores. Una valuación v satisface la restricción $\langle t_j, R_j\rangle$ si los valores asignados a las variables t_j satisfacen la relación R_j.

Una valuación es consistente si no viola ninguna de las restricciones. Una valuación es completa si incluye todas las variables. Una valuación es solución si es consistente y completa. En ese caso diremos que la valuación resuelve el CSP.

3.8.2. CSP para calcular homomorfismos. Sean $\mathbf{A}$,$\mathbf{B}$ $\mathcal{L}$-estructuras, queremos encontrar homomorfismos de $\mathbf{A}$ en $\mathbf{B}$ resolviendo una instancia de CSP. Tomamos

$$X = \{X_i : i \in A\}$$

$$D_i = B$$

$$C = C^R \cup C^f$$

donde C^R es tal que para cada $R \in \mathcal{L}$ n-aria, para cada $(a_1, \ldots, a_n) \in R^{\mathbf{A}}$, se da que $\left((X_{a_1}, \ldots, X_{a_n}), R^{\mathbf{B}}\right) \in C^R$ y C^f es tal que para cada $f \in \mathcal{L}$ n-aria, para cada $(a_1, \ldots, a_n, a') \in \operatorname{graph}\left(f^{\mathbf{A}}\right)$, se da que $\left((X_{a_1}, \ldots, X_{a_n}, X_{a'}), \operatorname{graph}\left(f^{\mathbf{B}}\right)\right) \in C^f$.

Supongamos que $V : X \to B$ es una valuación solución de (X, D, C), entonces el homomorfismo γ determinado por V será $\gamma(a) = V(X_a)$.

Si además quisiéramos que el homomorfismo fuera inyectivo, bastaría con agregar una restricción:

$$((X_1, \ldots, X_m), \text{para todo par } (i, j) \text{ con } i \neq j \text{ y } i, j \in \{1, \ldots, m\} \text{ se da que } X_i \neq X_j)$$

donde $X_1, \ldots, X_m$ son todas las variables en X.

Para que fuera sobreyectivo bastaría agregar la siguiente restriccion:

$$\left((X_1, \ldots, X_m), \bigcup_{i=1}^{m} \{X_i\} = B\right)$$

En el caso particular de la búsqueda de automorfismos, el lema 32 nos permite simplificar la busqueda, encontrando endomorfismos inyectivos.

Lema 32. *Sea $\mathbf{A}$ una estructura finita, entonces todo endomorfismo inyectivo de $\mathbf{A}$ es un automorfismo de $\mathbf{A}$.*

3.9. Generación de subestructuras. Dada una $\mathcal{L}$-estructura $\mathbf{A}$, generamos sus subestructuras recorriendo $\mathcal{P}(A)$, desde la mayor a la menor cardinalidad filtrando aquellos $A_0 \in \mathcal{P}(A)$ que no son cerrados bajo $\mathcal{L}$.

Además cuando detectamos que una subestructura $\mathbf{A}_0$ ya tiene representante en $\mathcal{S}$ durante nuestros algoritmos, disminuimos los subconjuntos a chequear, revisando solo $\mathcal{P}(A) - \mathcal{P}(A_0)$, ya que si $\mathbf{A}_0$ tiene representante en $\mathcal{S}$, entonces cada subestructura de $\mathbf{A}_0$ tiene representante en $\mathcal{S}$.

4. Álgebras de Lindenbaum

4.1. Definiciones e ideas básicas. Sea $\mathbf{A}$ una $\mathcal{L}$-estructura, y sea

$$\Sigma \in \left\{ \mathrm{Fo}(\mathcal{L}), \mathrm{E}(\mathcal{L}), \mathrm{E}^{+}(\mathcal{L}), \mathrm{Op}(\mathcal{L}), \mathrm{Op}^{+}(\mathcal{L}) \right\}.$$

Notar que la colección de relaciones n-arias definibles en $\mathbf{A}$ por fórmulas en Σ es cerrada bajo uniones e intersecciones (y también bajo complementación cuando $\Sigma \in \{\mathrm{Fo}(\mathcal{L}), \mathrm{Op}(\mathcal{L})\}$). Llamaremos *Álgebra de Lindenbaum* (de relaciones n-arias definibles por Σ en $\mathbf{A}$) al reticulado distributivo (o álgebra de Boole) resultante. Utilizaremos la siguiente notación:

- $\mathbf{Fo}_n(\mathbf{A})$ = Álgebra de Boole de relaciones n-arias definibles en $\mathbf{A}$,
- $\mathbf{E}_n(\mathbf{A})$ = Reticulado de relaciones n-arias definibles por existenciales en $\mathbf{A}$,
- $\mathbf{E}_n^{+}(\mathbf{A})$ = Reticulado de relaciones n-arias definibles por existenciales positivas en $\mathbf{A}$,
- $\mathbf{Op}_n(\mathbf{A})$ = Álgebra de Boole de relaciones n-arias definibles por abiertas en $\mathbf{A}$,
- $\mathbf{Op}_n^{+}(\mathbf{A})$ = Reticulado de relaciones n-arias definibles por abiertas positivas en $\mathbf{A}$

Definimos:

$$\operatorname{sub}\hom(\mathbf{A}) = \{\gamma : \gamma \text{ homomorfismo de } \mathbf{A}_0 \text{ en } \mathbf{B}_0 \text{ con } \mathbf{A}_0, \mathbf{B}_0 \in \mathbb{S}(\mathbf{A})\}$$
$$\operatorname{sub}\operatorname{iso}(\mathbf{A}) = \{\gamma : \gamma \text{ isomorfismo de } \mathbf{A}_0 \text{ en } \mathbf{B}_0 \text{ con } \mathbf{A}_0, \mathbf{B}_0 \in \mathbb{S}(\mathbf{A})\}$$

Dado que estas algebras podrian ser muy grandes ulizamos los siguientes resultados para representarlas mediante un subconjunto de sus elementos.

Teorema 33 (Teorema de representación de Stone)**.** *Toda álgebra de Boole finita $\mathbf{A}$ es isomorfa al álgebra definida por $\mathcal{P}(At)$ donde $At \subseteq A$ es el conjunto de átomos en $\mathbf{A}$.*

Teorema 34 (Teorema de representación de Birkhoff)**.** *Todo reticulado distributivo finito $\mathbf{L}$ es isomorfo al reticulado de conjuntos descendientes del poset de elementos join-irreducibles en $\mathbf{L}$.*

Lema 35. *Dada un algebra de Boole finita $\mathbf{A}$, un elemento es un atomo sii es join-irreducible.*

Los resultados anteriores nos permiten centrarnos únicamente en los elementos join-irreducibles de estas algebras. Mientras que los Lemas 36 y 37, presentados a continuacion, suguieren una manera clara de calcularlos.

Lema 36. *Una relación $r \subseteq A^n$ es join-irreducible en $\mathbf{Op}(\mathbf{A})$ sii hay $\bar{a} \in A^n$ tal que $r = \{h(\bar{a}) : h \in \operatorname{sub}\operatorname{iso}(\mathbf{A})\}$.*

Demostración. Supongamos que $r \subseteq A^n$ es join-irreducible en $\mathbf{Op}(\mathbf{A})$. Como r es cerrado bajo $\operatorname{sub}\operatorname{iso}(\mathbf{A})$ por pertenecer a $\mathbf{Op}(\mathbf{A})$, es claro que

$$r = \bigcup_{\bar{x} \in r} \{h(\bar{x}) : h \in \operatorname{sub}\operatorname{iso}(\mathbf{A})\}$$

, pero como r es join-irreducible hay una $\bar{a} \in r$ tal que $r = \{h(\bar{a}) : h \in \operatorname{sub}\operatorname{iso}(\mathbf{A})\}$.

Supongamos que $r = \{h(\bar{a}) : h \in \operatorname{sub}\operatorname{iso}(\mathbf{A})\}$ para algun $\bar{a} \in A^n$. Sean $r_1, \ldots, r_m \in \mathbf{Op}(\mathbf{A})$ tales que

$$r = r_1 \cup \cdots \cup r_m$$

, es claro que hay un r_j tal que $\bar{a} \in r_j$. Ademas como r_j es cerrado bajo $\operatorname{sub}\operatorname{iso}(\mathbf{A})$ por pertenecer a $\mathbf{Op}(\mathbf{A})$, entonces $r = r_j$. □

Lema 37. *Una relación $r \subseteq A^n$ es join-irreducible en $\mathbf{Op}^{+}(\mathbf{A})$ sii hay $\bar{a} \in A^n$ tal que $r = \{h(\bar{a}) : h \in \operatorname{sub}\hom(\mathbf{A})\}$.*

Demostración. Igual a la del Lema 36 □

4.2. Algoritmos. Con la intencion de simplificar la exposicion, supondremos que $\mathcal{K}$ tiene una unica estructura $\mathbf{A}$. La generalizacion es facilmente deducible ya que los algoritmos son casi iguales a los presentados en el Capitulo 3.

Como un algebra de Lindenbaum de relaciones definibles puede llegar a ser un objeto muy grande y dificil de manipular, nos basta generar los elementos join-irreducibles del algebra segun los Teoremas 33 para aquellas algebras de lindenbaum que son algebras de Boole y los atomos (que tambien son los elementos join-irreducibles segun el Lema 35) segun el Teorema 34, para aquellas que son reticulados.

Para generar los elementos join-irreducibles del algebra de Lindenbaum de relaciones de ancho n definibles por abiertas en $\mathbf{A}$ basta con obtener el conjunto $\mathcal{F}$ de flechas que se chequean por preservacion en los algoritmos del Capítulo 3 que genera $\operatorname{sub iso}(\mathbf{A})$ por el Teorema 26, y luego calcular $\{h(\bar{a}) : h \in \operatorname{sub iso}(\mathbf{A})\}$ para cada $\bar{a} \in A^n$, ya que por el Lema 36, todos los elementos join-irreducibles son de esta forma. De la misma manera podemos construir el algebra de Lindenbaum de relaciones de ancho n definibles por abiertas positivas en $\mathbf{A}$, basandonos en el Teorema 29 y en el Lema 37.

Los algoritmos 7 y 8, generan el conjunto $\mathcal{F}$ definidos por los Teoremas 26 y 29.

Algorithm 7 Construccion de $\mathbf{Op}_n(\mathbf{A})$

```
S = ∅
F = ∅
Sub = [C : C ∈ 𝕊_L(A), de la mayor a menor cardinalidad]
for B ∈ Sub do
    iso = tieneRepresentante(A_0, S)
    if iso ≠ Null then
        F = F ∪ {iso}
        Sub = [C : C ∈ Sub tal que C ⊈ B]
    else
        S = S ∪ {B}
        for α ∈ Aut_L(B) do
            F = F ∪ {α}
        end for
    end if
end for
return F                                        ▷ R es abierta-definible

function tieneRepresentante(A_0, S)
    for S ∈ S, con |A_0| = |S| do
        if hay γ : A_0 → S isomorfismo then
            return γ                 ▷ γ es isomorfismo con el representante
        end if
    end for
    return Null                                 ▷ No tiene representante
end function
```

Una vez calculado $\mathcal{F}$ basta con calcular $\{h(\bar{a}) : h \in \mathcal{F}\}$ para cada $\bar{a} \in A^n$, ya que $\mathcal{F}$ genera al conjunto correspondiente al tipo de definibilidad. Esto lo hacemos con el Algoritmo 9, lo cual ya nos da los elementos join-irreducibles o los atomos, segun el tipo de definibilidad en cuestion.

Algorithm 8 Construccion de $\mathbf{Op}_n^+(\mathbf{A})$

$\mathcal{S} = \emptyset$
$\mathcal{F} = \emptyset$
$Sub = [\mathbf{C} : \mathbf{C} \in \mathbb{S}_{\mathcal{L}}(\mathbf{A}),$ de la mayor a menor cardinalidad$]$
for $\mathbf{B} \in Sub$ **do**
  (iso,$\mathcal{F}_0$) = TIENEREPRESENTANTE($\mathbf{A}_0, \mathcal{S}$)
  $\mathcal{F} = \mathcal{F} \cup \mathcal{F}_0$
  if iso $\neq$ null **then**
    $\mathcal{F} = \mathcal{F} \cup \{\text{iso}\}$
    $Sub = [\mathbf{C} : \mathbf{C} \in Sub$ tal que $C \nsubseteq B]$
  else
    $\mathcal{S} = \mathcal{S} \cup \{\mathbf{B}\}$
    for $\alpha \in Aut_{\mathcal{L}}(\mathbf{B})$ **do**
      $\mathcal{F} = \mathcal{F} \cup \{\alpha\}$
    end for
  end if
end for
for $\mathbf{A} \in \mathcal{S}$ **do**
  for $\mathbf{B} \in \mathcal{S}$, con $|\mathbf{B}| < |\mathbf{A}|$ **do**
    for $\gamma : \mathbf{A} \to \mathbf{B}$ con γ homomorfismo sobreyectivo **do**
      $\mathcal{F} = \mathcal{F} \cup \{\gamma\}$
    end for
  end for
end for
return $\mathcal{F}$ ▷ R es definible por una abierta positiva

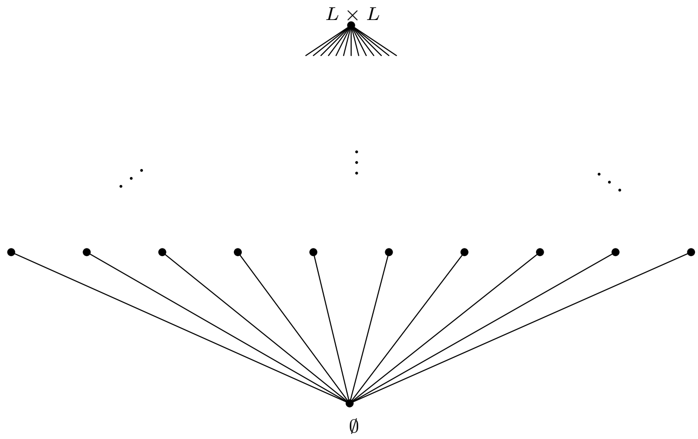
function TIENEREPRESENTANTE($\mathbf{A}_0, \mathcal{S}$)
  $\mathcal{F}_0 = \emptyset$
  for $\mathbf{S} \in \mathcal{S}$, con $|\mathbf{A}_0| = |\mathbf{S}|$ **do**
    $bihomo = False$
    for $\gamma : \mathbf{A}_0 \to \mathbf{S}$ con γ homomorfismo biyectivo **do**
      $bihomo = True$
      if γ es un $\mathcal{L}$-isomorfismo, con $\mathbf{S} \in \mathcal{S}$ **then**
        return $(\gamma, \emptyset)$ ▷ γ es isomorfismo con el representante
      else
        $\mathcal{F}_0 = \mathcal{F}_0 \cup \{\gamma\}$
      end if
    end for
    if $\neg bihomo$ **then**
      for $\gamma : \mathbf{S} \to \mathbf{A}_0$ con γ homomorfismo biyectivo **do**
        $\mathcal{F}_0 = \mathcal{F}_0 \cup \{\gamma\}$
      end for
    end if
    return (Null, $\mathcal{F}_0$) ▷ No tiene representante
  end for
end function

Algorithm 9 Saturacion por un conjunto de flechas

$\mathcal{A} = \emptyset$
for $\bar{a} \in A^n$ **do**
  $\mathcal{A} = \mathcal{A} \cup \{\text{CLAUSURA}(\bar{a}, \mathcal{E})\}$
end for
return $\mathcal{A}$ ▷ R es abierta-definible

function CLAUSURA($\bar{a}, \mathcal{E}$)
  $e = \{\bar{a}\}$
  while e crezca **do**
    for $\bar{a} \in e$ **do**
      for $\gamma \in \mathcal{E}$ **do**
        $e = e \cup \{\gamma(\bar{b})\}$

Figura 4.2. Reticulado $\mathbf{E}_2\left(\mathbf{2}\times\mathbf{2}\right)$

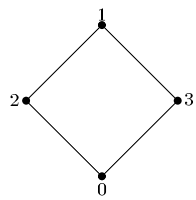


Para los demas tipos de definibilidad, el procedimiento seria analogo. Generar $\mathcal{F}$ basados en los algoritmos del capitulo 3 y saturar segun el Algoritmo 9.

4.3. Ejemplos y aplicación.

4.3.1. Utilizando el paquete Definability para SageMath. A manera de prueba de las herramientas utilizamos Definability para recorrer todas las relaciones binarias definibles en el reticulado $\mathbf{2}\times\mathbf{2}$, representado en la figura 4.1.

Figura 4.1. Reticulado $\mathbf{2}\times\mathbf{2}$



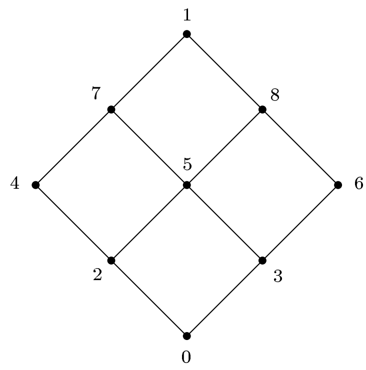
Al correr el algoritmo de generación de los elementos join-irreducibles de $\mathbf{E}_2\left(\mathbf{2}\times\mathbf{2}\right)$, Definability nos devolvió las siguientes relaciones:

- $\{(0,0)\}$
- $\{(0,1)\}$
- $\{(0,2),(0,3)\}$
- $\{(1,0)\}$
- $\{(1,1)\}$
- $\{(1,2),(1,3)\}$
- $\{(2,0),(3,0)\}$
- $\{(2,1),(3,1)\}$
- $\{(2,2),(3,3)\}$
- $\{(2,3),(3,2)\}$

Aqui se puede notar las ventajas de utilizar el Teorema 34, ya que el reticulado descripto por estos 10 elementos tiene cardinalidad $2^{10} = 1024$, ya que son todas las uniones entre elementos join-irreducibles.

En cuanto a los elementos join-irreducibles de $\mathbf{E}_2^{+}\left(\mathbf{2}\times\mathbf{2}\right)$ nos devolvió:

FIGURA 4.4. Reticulado $\mathbf{3} \times \mathbf{3}$



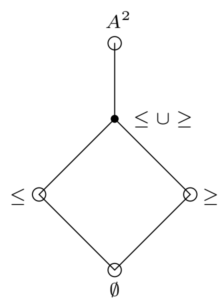
- $\{(0,0),(1,1),(2,2),(3,3)\}$
- $\{(0,0),(0,1),(0,2),(0,3),(1,1),(2,1),(2,2),(3,1),(3,3)\}$
- $\{(0,0),(1,0),(1,1),(1,2),(1,3),(2,0),(2,2),(3,0),(3,3)\}$
- $\{(0,0),(0,1),(0,2),(0,3),\ldots,(3,0),(3,1),(3,2),(3,3)\}$

Las cuales pueden ser interpretadas respectivamente como:

- Δ
- $\{(x,y) : x \leq y\}$
- $\{(x,y) : x \geq y\}$
- A^2

Lo que mediante el Teorema 34, define un subreticulado de $\mathbf{E}_2(\mathbf{2} \times \mathbf{2})$ anterior, mucho mas pequeño, como se puede ver en la Figura 4.3, donde los nodos sin rellenar son los elementos join-irreducibles devueltos por el paquete.

FIGURA 4.3. Reticulado de relaciones binarias definibles por existenciales positivas en $\mathbf{2} \times \mathbf{2}$



Luego probamos en el reticulado $\mathbf{3} \times \mathbf{3}$ representado en la figura 4.4.

Al interpretar los elementos join-irreducibles de $\mathbf{E}_2^+(\mathbf{3} \times \mathbf{3})$ calculados por el paquete, nos dio nuevamente:

- Δ
- $\{(x,y) : x \leq y\}$
- $\{(x,y) : x \geq y\}$
- A^2

Lo que representa un reticulado isomorfo al presentado en 4.3. Esto nos llevo a conjeturar el Teorema 41, que probamos a continuación.

4.3.2. Caracterización de las relaciones binarias definibles por existenciales positivas en reticulados distributivos.

Lema 38. *Sea* $\mathbf{L}$ *un reticulado y* R *una relación binaria no vacía sobre* L *preservada por endomorfismos en* $\mathbf{L}$*. Entonces vale que* $\Delta^L \subseteq R$.

Demostración. Esto es una consecuencia directa de que para cada $a \in L$ la función de L en L que vale constantemente a es un endomorfismo. $\square$

Teorema 39 (Teorema del filtro primo)**.** *Sea* $\langle L, \vee, \wedge \rangle$ *un reticulado distributivo y* F *un filtro. Supongamos* $x \in L - F$*. Entonces hay un filtro primo* P *tal que* $x \notin P$ *y* $F \subseteq P$.

Lema 40. *Sea* $\mathbf{L}$ *un reticulado distributivo y* R *una relación binaria no vacía sobre* L *preservada por endomorfismos en* $\mathbf{L}$*. Si hay* $(a, b) \in R$ *tal que* $a \nleq b$ *(respectivamente* $b \nleq a$*), entonces* $\{(x, y) : x \geq y\} \subseteq R$ *(respectivamente* $\{(x, y) : x \leq y\} \subseteq R$*).*

Demostración. Fijamos $(a, b) \in R$ tal que $a \nleq b$, y sean $c, d \in L$ tales que $c \geq d$. Veremos que $(c, d) \in R$. Por el Teorema 39 hay un filtro primo P que contiene al filtro generado por a y además $b \notin P$. Definimos $h : L \to L$ por

$$h(x) = \begin{cases} c & \text{si } x \in P, \\ d & \text{si } x \notin P. \end{cases}$$

Es fácil ver que h es un endomorfismo. Finalmente como $(a, b) \in R$ y h preserva R, $(h(a), h(b)) = (c, d) \in R$. $\square$

Teorema 41. *Sea* $\mathbf{L}$ *un reticulado distributivo y* R *una relación binaria sobre* L *definible por un fórmula existencial positiva en* $\mathbf{L}$*. Se da una de las siguientes:*

- $R = \Delta$,
- $R = \{(x, y) : x \leq y\}$,
- $R = \{(x, y) : x \geq y\}$,
- $R = \{(x, y) : x \leq y\} \cup \{(x, y) : x \geq y\}$,
- $R = L \times L$.

Demostración. Si $R \subseteq \Delta$, por el Lema 38 $R = \Delta$.

Si $R \subseteq \{(x, y) : x \leq y\}$, pero $R \nsubseteq \Delta$, hay $(a, b) \in R$ tales que $a < b$, entonces $a \ngeq b$ y por el Lema 40 $R = \{(x, y) : x \leq y\}$. Análogo para $R \subseteq \{(x, y) : x \geq y\}$.

Si $R \nsubseteq \{(x, y) : x \leq y\}$ y $R \nsubseteq \{(x, y) : x \geq y\}$, aplicando dos veces el Lema 40 $\{(x, y) : x \leq y\} \cup \{(x, y) : x \geq y\} \subseteq R$. Si además $(a, b) \in R$ pero $a \nleq b$ y $a \ngeq b$, tomo $(c, d) \in R$, si son comparables ya esta están en R por lo anterior. Si son incomparables aplico dos veces el Teorema 39 y tomo dos filtros primos P y Q tales que $a \in P$ pero $b \notin P$ y $b \in Q$ pero $a \notin Q$. Ahora definimos la siguiente función

$$h(x) = \begin{cases} c \vee d & \text{si } x \in P \cup Q \\ c & \text{si } x \in P - Q \\ d & \text{si } x \in Q - P \\ c \wedge d & \text{si } x \notin P \text{ y } x \notin Q \end{cases}$$

que fácilmente puede verse que es un endomorfismo. Por lo tanto como $(a, b) \in R$ y h preserva R, $(h(a), h(b)) = (c, d) \in R$. Por lo tanto $R = L \times L$. $\square$

5. Documentación de Definability

5.1. Introducción. Definability es un paquete que desarrollamos para SageMath, un software matemático licenciado bajo la GPL. Nuestro paquete implementa los algoritmos vistos en las secciones anteriores y permite dada una clase de estructuras, decidir definibilidad o generar las álgebras de Lindenbaum de relaciones definibles.

Nuestro paquete se basa en parte en las librerías desarrolladas por Peter Jipsen para utilizar Minion, Universal Álgebra Calculator, Mace4 y Prover9 desde SageMath que se pueden obtener en http://math.chapman.edu/∼jipsen/sagepkg/.

El paquete se desarrolla en https://github.com/pablogventura/tesis.

5.2. Instalación.

5.2.1. Requerimientos.

- SageMath 6.7 o superior.
- Minion 1.8 o superior.
- LADR versión de noviembre de 2009, (interfaz de linea de comandos de Prover9, Mace4, y otros programas)
- Git

5.2.2. Instalación del paquete en SageMath. Suponiendo que tenemos instalado SageMath en la ruta absoluta SAGEDIR, nos movemos al directorio donde se quiere descargar el paquete Definability, seguimos los siguientes comandos para instalar el paquete en nuestra instalacion de SageMath:

```
$ git clone https://github.com/pablogventura/tesis.git
$ cd tesis/package
$ ./make_spkg.sh
$ mv definability-hash.spkg SAGEDIR
$ cd SAGEDIR
$ ./sage -i definability-hash.spkg
```

5.3. Uso del paquete. Una vez instalado el paquete basta con arrancar Sage con el comando:

```
$ ./sage
```

y una vez en la consola de Sage, importamos el paquete con el comando:

```
sage: import definability
```

5.4. Generación y entrada de estructuras. Al momento de ingresar estructuras para su chequeo puede optarse por la generacion automatica de estructuras a partir de una teoria de primer orden mediante la interfaz a Mace4, o la entrada de una estructura en particular manualmente.

5.4.1. Generación de estructuras de una teoría de primer orden. Para definir un objeto del tipo FO_Theory que implementa a una teoría de primer orden basta con un nombre, una descripcion, y una lista de axiomas en la sintaxis propia de Mace4. Por ejemplo para definir la teoria de reticulados bastaria con la siguiente linea:

```
Lat = definability.FO_Theory("Lat",
                             "Lattices",
                             ['(x v y) v z = x v (y v z)',
                              'x v y = y v x',
                              '(x^y)^z = x^(y^z)',
                              'x^y = y^x',
                              '(x v y)^x = x',
                              '(x^y) v x = x'])
```

Ahora para recorrer los reticulados (filtrando isomorfismos) basta con llamar al metodo find_models con la cardinalidad buscada, el cual devuelve un generador. Por ejemplo para recorrer los reticulados de 5 elementos basta con

```
list(Lat.find_models(5))
```

que devolverá una lista con todos los reticulados de 5 elementos filtrando isomorfismos.

Nuestro paquete incluye una gran variedad de teorias de primer orden ya cargadas en el modulo fotheories. Por ejemplo: Lat para reticulados, Graph para grafos, Grp para grupos, etc. La lista completa se puede revisar mediante listar el modulo fotheories de nuestro paquete con el siguiente comando:

```
dir(definability.fotheories)
```

5.4.2. Entrada manual de una estructura. Supongamos que queremos ingresar manualmente el reticulado $\mathbf{2} \times \mathbf{2}$. Lo primero es definir el tipo de primer orden al que pertenece. Para crear un objeto de la clase FO_Type basta con pasar dos diccionarios, el primero para las funciones y el segundo para las relaciones. Los diccionarios contienen las aridades y se indexan por los simbolos correspondientes. Por ejemplo, para definir el tipo de los reticulados, junto con una relacion unaria P para chequear:

```
tlat = definability.FO_Type({'v': 2, '^': 2},{"P":1})
```

Luego se debe definir un universo para la estructura, como por ejemplo

```
universe = [0,1,2,3]
```

Luego se necesitan definir las funciones supremo e infimo, y una relacion unaria P para su chequeo.

Una funcion, implementada por la clase FO_Operation se puede definir mediante un diccionario con claves de tuplas que apuntan al valor de la funcion, o en el caso de funciones binarias, una matriz A donde el valor $A_{i,j}$ se corresponde a la funcion evaluada en (i, j). Para definir una relacion utilizando la clase FO_Relation basta con una lista de las tuplas que pertenecen a la relacion y el universo sobre el que la relacion esta definida.

Como ejemplo, definimos el infimo y el supremo utilizando cada una de las maneras para definir funciones, y definimos la relacion P:

```
meet = definability.FO_Operation({(0, 0): 0,
                                  (0, 1): 0,
                                  (0, 2): 0,
                                  (0, 3): 0,
                                  (1, 0): 0,
                                  (1, 1): 1,
                                  (1, 2): 2,
                                  (1, 3): 3,
                                  (2, 0): 0,
                                  (2, 1): 2,
                                  (2, 2): 2,
                                  (2, 3): 0,
                                  (3, 0): 0,
                                  (3, 1): 3,
                                  (3, 2): 0,
                                  (3, 3): 3})
join = definability.FO_Operation([[0,1,2,3],
                                  [1,1,1,1],
                                  [2,1,2,1],
```

```
                                [3,1,1,3]])
rP = definability.FO_Relation([(2,), (3,)], uni)
```

Finalmente, para definir el reticulado basta con hacer:

```
l2x2 = definability.FO_Model(tlat, uni, {'^': meet,
                                          'v': join},
                                         {'P': rP})
```

5.5. Chequeo de definibilidad. Una vez definidas las estructuras, para chequear definibilidad se utiliza un objeto de la clase Constellation, pasando una lista de las estructuras en $\mathcal{K}$. Esta clase implementa los algoritmos vistos en el Capitulo 3. Por ejemplo siguiendo con el ejemplo anterior:

```
c = definability.Constellation([l2x2])
```

Para chequear definibilidad estan los siguientes metodos:

- is_existential_definable
- is_existential_positive_definable
- is_open_definable
- is_positive_open_definable

los cuales toman dos tipos $\mathcal{L} \subseteq \mathcal{L}'$ para decidir definibilidad de las relaciones del tipo $\mathcal{L}' - \mathcal{L}$ en $\mathcal{L}$, y devuelven una tupla con un booleano y un contrajemplo, si es que lo hay.

Por ejemplo para chequear definibilidad abierta de la relacion P:

```
c.is_open_definable(tlat,tlat + definability.FO_Type({},{"P"
    :1}))
```

Lo cual en particular es falso, por lo que el paquete devuelve la tupla:

```
(False, Embedding([0] -> 2,
                  FO_Type({'v': 2, '^': 2},{}),
                  antitype= ['P'],
                  Injective, ))
```

Donde devuelve un contraejemplo, ya que el embedding del subreticulado formado por el elemento 0 en el subreticulado formado por el elemento 2 (un isomorfismo entre subestructuras de $\mathbf{2} \times \mathbf{2}$) no preserva P, negando el Teorema 13.

5.6. Generación de álgebras de Lindenbaum. Para la generacion de las algebras de Lindenbaum desarrolladas en el capitulo 4 el paquete dispone del modulo lindenbaum.

Este modulo contiene las siguientes funciones:

ji_of_existencial_definable_algebra: para generar los elementos join irreducibles de $\mathbf{E}_n(\mathcal{K})$

ji_of_existencial_positive_definable_algebra: para generar los elementos join irreducibles de $\mathbf{E}_n^+(\mathcal{K})$

atoms_of_open_definable_algebra: para generar los átomos de $\mathbf{Op}_n(\mathcal{K})$

ji_of_open_positive_definable_algebra: para generar los elementos join irreducibles de $\mathbf{Op}_n^+(\mathcal{K})$

Estas funciones toman un objeto del tipo Constellation que determina $\mathcal{K}$, un tipo para los morfismos en cuestion y una aridad.

Por ejemplo para obtener los elementos join-irreducibles del reticulado de relaciones ternarias definibles en $\mathbf{2} \times \mathbf{2}$ bastaria con la siguiente linea:

```
definability.lindenbaum.
    ji_of_existencial_positive_definable_algebra(c,
    definability.examples.tiporet,3)
```

Lo cual devuelve una lista de 22 relaciones join-irreducibles, que acortamos como ejemplo:

```
[Relation([0, 0, 0],
          [1, 1, 1],
          [2, 2, 2],
          [3, 3, 3]),
.
.
.
Relation([0, 0, 0],
         [0, 1, 1],
         [0, 2, 2],
         [0, 3, 3],
         [1, 0, 0],
         [1, 1, 1],
         [1, 2, 2],
         [1, 3, 3],
         [2, 0, 0],
         [2, 1, 1],
         [2, 2, 2],
         [2, 3, 3],
         [3, 0, 0],
         [3, 1, 1],
         [3, 2, 2],
         [3, 3, 3])]
```

6. Conclusiones y trabajo futuro

La ventaja de utilizar CSP es que hay múltiples circunstancias en las que uno de estos problemas se pueden resolver muy eficientemente. Esperamos explorar detalladamente estas situaciones para aprovecharlas para la busqueda de homomorfismos cuando se den propiedades especificas para las estructuras en cuestion.

Tambien esperamos explorar mejores implementaciones para la generacion de subestructuras, investigando a fondo los algoritmos utilizados por el Universal Algebra Calculator.

- Que un conjunto de flechas alcance para revisar preservación implica que genera el conjunto de flechas completo?
- Para chequear si alguien es subestructura de otro, quizá conviene recorrer mapeos que son embeddings sabiendo que la imagen va a ser una subestructura isomorfa.
- Que hace Minion cuando se le prohíben soluciones (negativetable). Quizás primero experimentar con ejemplos.
- Explorar la relación entre generadores de una subestructura y homomorfismos.
- De menor a mayor o de menor a mayor las subestructuras?

Referencias

Campercholi, Miguel y Diego Vaggione (2015). *Semantical conditions for the definability of functions and relations*. eprint: 1506.07501. URL: http://www.arxiv.org/abs/1506.07501.

Wikipedia (2016). *Constraint satisfaction problem — Wikipedia, The Free Encyclopedia*. [En línea, visitado el 11 de Enero de 2016]. URL: http://en.wikipedia.org/w/index.php?title=Constraint%20satisfaction%20problem&oldid=695651340.