UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN FILOSOFÍA DE LA CIENCIA

FACULTAD DE FILOSOFÍA Y LETRAS, FACULTAD DE CIENCIAS, INSTITUTO DE INVESTIGACIONES FILOSÓFICAS, DIRECCIÓN GENERAL DE DIVULGACIÓN DE LA CIENCIA

FILOSOFÍA DE LAS MATEMÁTICAS Y LÓGICA DE LA CIENCIA

¿Cómo un lógico podría ayudar a resolver el problema P $\stackrel{?}{=}$ NP?

TESIS QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN FILOSOFÍA DE LA CIENCIA
PRESENTA

Alejandro Javier Solares Rojas
Tutor: Dr. Luis Estrada González, IIF–UNAM

CIUDAD UNIVERSITARIA, CD. MX., JUNIO DE 2018

Dedico este trabajo

a mi madre y a mi padre,

Elena y Armando.

Éste es uno más de los frutos

de su apoyo incondicional.

Agradecimientos

Quiero agradecer no sólo a mi madre y a mi padre, sino también a varias personas más e instituciones que me apoyaron durante el proceso de realización de este trabajo.

Agradezco mucho a Luis Estrada por su apoyo y orientación constantes; la dedicación con la que apoya a sus alumnos es inspiradora. Agradezco a Raymundo Morado, Cristian Gutiérrez y Franciso Hernández por darme bases firmes, tanto intelectuales como éticas. También agradezco a Walter Carnielli por compartirme sus conocimientos, pero aún más por compartirme su pasión por la lógica. Estoy agradecido también con mi gran amigo Carlos César Jiménez, quien me apoyó sobremanera durante el proceso y con quien discuto mis ideas antes y después de compartirlas académicamente. Además, estoy agradecido con los miembros de los seminarios de filosofía de la información y ciencias de la computación, filosofía de la lógica, y el seminario de tesistas en filosofía de las ciencias formales. En particular, agradezco a Manuel Tapia, Elisangela Ramírez, Mauricio Andrade, Francisco Martínez y Claudia Tanús, por valiosos comentarios y sugerencias.

Por otro lado, agradezco profundamente a Armando, Verónica, Valentina, Emiliano, Sofía y Omar, por las pláticas dominicales de sobremesa en familia; sin duda, esas pláticas han fomentado mi creatividad y mi sed de conocimiento desde hace muchos años. También agradezco profundamente a Fernanda por recordarme disfrutar lo que somos y hacemos, y

por las pláticas apasionantes a diario.

En lo que concierne al respaldo institucional, crucial para realizar este trabajo, agradezco la beca nacional de manutención que me fue brindada por el CONACYT durante el periodo comprendido entre el 1 de agosto del 2016 y el 31 de julio del 2018, al estar el programa de posgrado en Filosofía de la Ciencia de la UNAM registrado en el Programa Nacional de Posgrados de Calidad (PNPC). A la Red de Macrouniversidades Públicas de América Latina y el Caribe agradezco el haberme brindado una beca para realizar una estancia de investigación en CLE–UNICAMP (Campinas, Brasil) del 1 de abril al 30 de junio del 2018. Agradezco igualmente al Programa de Apoyo a los Estudios de Posgrado (PAEP) de la UNAM por brindarme apoyo económico para presentar los avances de este trabajo en los siguientes eventos: *Logic Colloquium 2017* (Estocolmo, Suecia) y *PhDs in Logic X* (Praga, República Checa). También agradezco al proyecto PAPIIT IA401117, "Aspectos filosóficos de las lógicas contraclásicas", por brindarme apoyo económico para presentar los avances de este trabajo en los eventos: *Logic Colloquium 2017* y *ALFAn V - PHILOGICA V* (Villa de Leyva, Colombia). Cabe mencionar, que también presenté avances de este trabajo en el evento *Fourth Workshop CLE–Buenos Aires Logic Group* (Guarujá, Brasil). Agradezco a los organizadores e instituciones anfitrionas de cada uno de estos eventos las facilidades brindadas para someter este trabajo a discusión.

Índice general

Capítulo 1

Introducción

Como es bien sabido, el problema P $\stackrel{?}{=}$ NP es uno de los problemas abiertos fundamentales de las Ciencias de la Computación teóricas, cuya solución podría tener implicaciones revolucionarias en muchas y diversas áreas. Por un lado, si P = NP, por mencionar algunos ejemplos de manera informal: la mayoría de la criptografía que actualmente usamos podría ser rota, el esfuerzo cognitivo para encontrar una prueba formal de cualquier enunciado matemático que tenga una prueba de longitud razonable podría ser automatizado, y podríamos encontrar algoritmos eficientes para resolver problemas NP-completos, los cuales abundan en la Ciencia. (Si, además, la prueba de P = NP estuviera acompañada de un algoritmo eficiente en la práctica, se actualizaría la posibilidad correspondiente a cada uno de los ejemplos mencionados). Por otro lado, si P $\neq$ NP, *prima facie* el escenario parecería menos dramático ya que no habría implicaciones revolucionarias evidentes como las recién descritas. Sin embargo, la implicación más importante (también revolucionaria) de una prueba de P $\neq$ NP es precisamente que podría descartar dichas implicaciones revolucionarias evidentes: podríamos seguir comprando de forma segura en Internet y los matemáticos,

científicos e ingenieros podrían seguir recibiendo trabajos indefinidamente. Específicamente, por ejemplo, P $\neq$ NP podría implicar una limitación en toda la computación digital clásica y permitirnos probar que muchos problemas comunes no pueden ser resueltos de manera eficiente (véase [17] y [18]).

Aunque no se ha ofrecido una prueba que resuelva el problema P $\stackrel{?}{=}$ NP, la mayoría de los expertos han conjeturado y argumentado a favor de P $\neq$ NP (véase [19]). Generalmente, sus argumentos se han restringido a "mostrar" que las implicaciones de suponer P = NP van en contra de nuestras intuiciones. Por ejemplo, Wigderson ([33] y [34]) y Aaronson ([5] y [6]) han argumentado que tales implicaciones significan que la creatividad (al menos matemática) podría ser automatizada, y que la existencia de algoritmos eficientes para problemas NP-completos es incompatible con que, después de décadas de investigación, nadie ha sido capaz de encontrar alguno de esos algoritmos. Aaronson ([4]) incluso ha argumentado que P $\neq$ NP eventualmente podría alcanzar el mismo estatus que, por ejemplo, la imposibilidad de la señalización superlumínica y de las máquinas de movimiento perpetuo; es decir, el estatus de un principio (restrictivo) de la física. Dada esta situación, estrategias para argumentar a favor de P = NP han sido escasamente exploradas y gran parte de la investigación que involucra tales clases de complejidad ha estado sesgada en tanto que, implícita o explícitamente, presupone P $\neq$ NP (e.g., impedimentos técnicos para probar P $\neq$ NP han sido estudiados ampliamente, a diferencia de los impedimentos técnicos para probar P = NP). En contraste, en esta tesis exploro un enfoque de traducción entre lógicas (en adelante, TEL) aplicado a la lógica de primer-orden con un operador de punto fijo mínimo (en adelante, LPO(OPFM)) y a la lógica de segundo-orden (LSO); examinando, primero, si está bien motivado como una estrategia para argumentar a favor de P = NP y,

segundo, si hay respaldo para su adecuación como tal.

De acuerdo con la teoría de la complejidad descriptiva (en adelante, TCD), dado un prototipo de máquina de computación limitada en recursos (e.g., el prototipo de máquinas de Turing limitadas en tiempo polinómicamente) y una lógica adecuada $\mathfrak{L}$ (e.g., como se muestra abajo, LPO(OPFM)), para cada maquina M de ese tipo hay una oración ϕ de $\mathfrak{L}$ cuyos modelos son precisamente las estructuras aceptadas por M; y, de manera conversa, para cada ϕ of $\mathfrak{L}$ hay una máquina M de ese tipo que precisamente acepta los modelos de ϕ. Por tanto, las clases de estructuras aceptables por una máquina M de ese tipo corresponden a las clases axiomatizables en $\mathfrak{L}$.[1] De esta manera, TCD analiza la complejidad de todas las *consultas* que son definibles en una lógica dada; siendo su cuestión central la de determinar si, dada una clase de complejidad $\mathcal{C}$, hay una lógica $\mathfrak{L}$ tal que las consultas $\mathfrak{L}$-definibles son exactamente las consultas en $\mathcal{C}$. En el caso afirmativo, la mayoría de las veces hay un procedimiento efectivo para traducir cada fórmula de esa lógica como un programa para la consulta correspondiente.

TCD proporciona descripciones lógicas (independientes de las máquinas) de las clases de complejidad y, de esa manera, también proporciona análogos lógicos de los problemas de inclusión y separación en la teoría de la complejidad (véase [13], VII). En particular, TCD proporciona una manera de entender el problema $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ enteramente desde un punto de vista lógico: P = NP syss cada consulta booleana, sobre estructuras ordenadas, finitas, expresable en el lenguaje de LSO es ya expresable en el lenguaje de LPO(OPFM) (véase [24], 122). En otras palabras, a grandes rasgos, las clases de complejidad P y NP son la misma syss la expresividad (sobre estructuras ordenadas, finitas) de un lenguaje necesario para

[1] Como se explica en el siguiente capítulo, dentro del presente contexto, "axiomatizibilidad en una lógica" significa lo siguiente: Sea K una clase de τ-estructuras y $\mathfrak{L}$ una lógica. K es axiomatizable en $\mathfrak{L}$, si hay una oración ϕ de $\mathfrak{L}$ de vocabulario τ tal que $K = \mathrm{Mod}(\phi)$ (la clase de modelos finitos de ϕ).

describir los problemas contenidos en ellas es la misma. Siendo así, el siguiente argumento muestra cómo se podría probar P = NP si fuera proporcionada una traducción adecuada entre LPO(OPFM) y LSO:

1. P = LPO(OPFM) ([23] y [31])
2. PH = LSO ([28])
3. P = NP $\iff$ LPO(OPFM) = LSO (de 1 y 2; véase [24], 121-122)
4. LPO(OPFM) $\subseteq$ LSO (de P $\subseteq$ NP)
5. LSO $\subseteq$ LPO(OPFM) (si un TEL adecuado fuera proporcionado; véase [25] y [9])
6. LPO(OPFM) = LSO (de 4 y 5)
7. P = NP (de 3 y 6)

Sin embargo, en esta tesis no proporciono tal traducción para el paso 5 (de lo contrario estaría en las puertas del *Clay Mathematics Institute* reclamando algunos dólares), sino únicamente exploro si un enfoque TEL aplicado a LPO(OPFM) y LSO como una estrategia para argumentar a favor de P = NP está bien motivado y cuenta con respaldo. Respecto a esto, aunque mi contribución en esta tesis es modesta, no debe subestimarse ya que, como Terence Tao ha reiterado varias veces, las matemáticas pueden beneficiarse no sólo de pruebas correctas, o de pruebas que requieren de algunos cambios para ser correctas, sino también de esbozos de estrategias para una prueba, ya sea para abrir líneas de investigación o cerrarlas definitivamente.

La estructura de la tesis es como sigue: En el capítulo 2, explico el problema P $\stackrel{?}{=}$ NP en términos meramente lógicos, siguiendo las identidades proporcionadas por TCD. En el

capítulo 3, introduzco un enfoque TEL con el objetivo de comparar los poderes expresivos de LPO(OPFM) y LSO; y, en ese sentido, motivo tal enfoque como una estrategia para argumentar a favor de P = NP. En el capítulo 4, señalo la existencia de respaldo para el enfoque de dos tipos; a saber, matemático-conceptual y filosófico. Además, muestro a muy grandes rasgos cómo las tres barreras conocidas en la teoría de la complejidad representan constreñimientos para dicho enfoque y cómo éste nos permite superarlas. Finalmente, doy algunas conclusiones.

Capítulo 2

El problema en términos lógicos [1]

TCD proporciona una manera de entender la complejidad de todos los problemas de computación a través de la complejidad de sus descripciones lógicas correspondientes. En otras palabras, TCD nos permite entender la complejidad computacional de un problema a través de la expresividad de un lenguaje necesario para describir el problema (a diferencia de como usualmente se entiende: a través del tiempo y espacio nesarios para resolverlo). Dentro del contexto de la complejidad computacional, los problemas a los que nos referimos son problemas de decisión y, de acuerdo con TCD, éstos pueden ser entendidos como *consultas booleanas* expresables por medio de lenguajes de diferentes lógicas. Como veremos abajo, una consulta booleana es una consulta cuya respuesta es un único bit: 1 o 0. En este sentido, dado que las clases de complejidad tradicionales se definen como conjuntos de preguntas sí/no, éstas son exactamente conjuntos de consultas booleanas. Por consiguiente, TCD usa el concepto de *consulta* como el paradigma fundamental de computación: usualmente, un programa de computadora describe precisamente un mapeo de entradas a salidas; TCD

[1] La mayor parte del contendio de este capítulo está basada en los resultados de TCD como los presenta Immerman ([24]).

llama a tal mapeo una consulta de estructuras de entrada a estructuras de salida. Así, para dar la definición formal de consulta (booleana) primero necesitamos dar la definición de *estructura*. Esta última es la estándar:

Una estructura es una tupla,

$$\mathcal{A} = \langle |\mathcal{A}|, R_1^{\mathcal{A}}, ..., R_r^{\mathcal{A}}, c_1^{\mathcal{A}}, ..., c_s^{\mathcal{A}}, f_1^{\mathcal{A}}, ..., f_t^{\mathcal{A}} \rangle,$$

donde $|\mathcal{A}|$, llamado el *universo* (de discurso), es un conjunto no vacío (usamos $\|\mathcal{A}\|$ para denotar su cardinalidad); y la tupla,

$$\tau = \langle R_1^{a_1}, ..., R_r^{a_r}, c_1, ..., c_s, f_1^{r_1}, ..., f_t^{r_t} \rangle,$$

llamada el *vocabulario*, es una tupla de símbolos de relación, símbolos de constante y símbolos de función. R_i es un símbolo relacional de aridad a_i, c_j es un símbolo de constante y f_k es un símbolo de función de aridad r_k. Para cada $R_i \in \tau$, $\mathcal{A}$ tiene una relación $R_i^{\mathcal{A}}$ de aridad a_i definida sobre $|\mathcal{A}|$ de tal manera que $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$. Para cada $c_j \in \tau$, $\mathcal{A}$ tiene un determinado elemento de su universo: $c_j^{\mathcal{A}} \in |\mathcal{A}|$. Para cada $f_i \in \tau$, $\mathcal{A}$ tiene una función total de $|\mathcal{A}|^{r_i}$ a $|\mathcal{A}|$.

Por motivos de simplicidad, en adelante únicamente consideraremos vocabularios relacionales, i.e., vocabularios sin símbolos de función.[2] Para cualquier vocabulario τ, podemos definir el lenguaje de primer-orden $\mathcal{L}(\tau)$ como el conjunto de fórmulas constituido por los símbolos de τ, el símbolo de relación lógica '=', las conectivas booleanas '$\wedge$' y '$\neg$', variables individuales: VAR $= \{x, y, z, ...\}$, cuantificador '$\exists$' y paréntesis '(', ')'.[3] (La sintáxis

[2]Para deshacerse de los símbolos de función en un vocabulario τ, uno debe introducir un nuevo símbolo de relación $(n+1)$-ario F para cada f n-aria $\in \tau$: Sea τ^r el vocabulario (libre de símbolos de función) que consiste en los símbolos de relación y constantes de τ, junto con los nuevos símbolos de relación. Para una τ-estructura $\mathcal{A}$, sea $\mathcal{A}^r$ la τ^r-estructura obtenida de $\mathcal{A}$ reemplazando cada función n-aria $f^{\mathcal{A}}$ por su gráfica $F^{\mathcal{A}}$, $F^{\mathcal{A}} := \{(a_0, ..., a_n, f(a_0, ..., a_n)) | a_0, ..., a_n \in A\}$ (véase [13], 11-12).

[3]'$\vee$', '$\rightarrow$', '$\leftrightarrow$' y '$\forall$' pueden ser definidos de la manera estándar. Por otro lado, en esta tesis tomo a la identidad como un símbolo lógico porque, en la medida en que tomo a la igualdad como una relación primitiva, ésta tiene su significado fijo.

correspondiente es la estándar).

Surge una cuestión matemático-filosófica dentro del contexto de TCD: una computación es inherentemente finita. Los objetos que las computadoras (y, discutiblemente, todas las máquinas reales o idealizadas) tienen, mantienen y manipulan (tales como entradas, programas, bases de datos, etc.), son todos finitos. Esos objetos se pueden modelar convenientemente como estructuras finitas; es decir, estructuras cuyo universo es un conjunto finito. Por un lado, las estructuras finitas se pueden codificar como palabras y, por tanto, pueden ser objetos de computaciones. Por otro lado, tales estructuras se pueden usar para describir ejecuciones finitas de máquinas. De hecho, las fórmulas de un lenguaje lógico a menudo se pueden interpretar como programas tales que, dada una estructura finita como entrada, ejecutan la evaluación correspondiente. Este hecho es muy importante en, por ejemplo, la teoría de bases de datos, donde las bases de datos relacionales se consideran estructuras finitas. Siendo así, definimos ESTRUC$[\tau]$ como el conjunto de estructuras finitas de vocabulario τ.[4]

Por ejemplo, sea $\tau_g = \langle A^2, f, t\rangle$, con '$A^2$' siendo un símbolo de relación binaria, y 'f' y 't' símbolos de constante (cuya denotación pretendida son vértices fuente y terminal especificados, respectivamente). Una gráfica (no-dirigida) es una τ_g-estructura $\mathcal{G} = \langle |\mathcal{G}|, A^{\mathcal{G}}, f^{\mathcal{G}}, t^{\mathcal{G}}\rangle$ que satisface:

1. $\forall x \neg Axx$

2. $\forall x \forall y (Axy \rightarrow Ayx)$

[4]Vale la pena mencionar que, como será evidente abajo, un enfoque "finitista" a los objetos lógico-matemáticos es bastante diferente en muchos aspectos al enfoque "infinitista" correspondiente; se aplican diferentes técnicas y se sostienen diferentes teoremas respectivamente. Por ejemplo, las propiedades que uno puede probar sobre $\mathbb{N}$ a menudo son falsas o irrelevantes si se aplican a los objetos de computación.

Denotamos la clase de gráficas finitas como GRAF. Hablamos de una gráfica dirigida (digráfica) cuando sólo se requiere 1. Así, la fórmula

$$\phi_{nodir}(x, y) \equiv \forall x \forall y(\neg Axx \wedge Axy \rightarrow Ayx)$$

dice que la gráfica en cuestión es no-dirigida y no tine ciclos.[5] (Usualmente, los elementos de $|\mathcal{G}|$ se llaman vértices o nodos, mientras que los elementos de $A^{\mathcal{G}}$ se llaman aristas).

Una estructura del tipo de una gráfica no necesita tener un ordenamiento en sus vértices. Sin embargo, si uno usa una computadora para almacenar o manipular una gráfica, ésta debe estar codificada de alguna manera; lo cual impone un ordenamiento en sus vértices. Cuando uno codifica una entrada en una computadora, lo hace como una cadena de caracteres: el primer carácter, el segundo carácter, y así sucesivamente. Así, siempre hay un ordenamiento involucrado en tal codificación. De hecho, el concepto mismo de ordenamiento está profundamente arraigado en los conceptos de cadena y computación. En este sentido, para discutir computación en general, necesitamos suponer que los universos de las estructuras consideradas están ordenados. Como en el caso particular de las gráficas, esta suposición parece ser natural porque, al codificar una estructura como una entrada a una computadora, inducimos un ordenamiento en su universo.[6] Por estas razones, el símbolo de relación binaria '$\leq$' juega un papel especial in TCD: si '$\leq$' $\in \tau$ y $\mathcal{A} \in$ ESTRUC[τ], $\mathcal{A}$ debe interpretar $\leq$ como un ordenamiento total sobre su universo. En tal caso, también establecemos símbolos de constante '0', '1', 'max' en τ, y los interpretamos respectivamente

[5] Establecemos la siguiente convención para la precedencia de operador: '$\neg$' tiene la precedencia más alta, después '$\wedge$' y '$\vee$', y finalmente '$\rightarrow$' y '$\leftrightarrow$'; mientras que los operadores de igual precedencia se evalúan de izquierda a derecha.

[6] Todas las descripciones lógicas de primer-orden de clases de complejidad suponen que una relación de ordenamiento total sobre el universo está disponible en los lenguajes. De hecho, algunos de los mayores problemas abiertos en TCD conciernen a la pregunta de hasta qué punto los resultados correspondientes pueden ser generalizados a estructuras no ordenadas (véase [24], cap. 12; y [13], cap. 11).

como el mínimo, segundo y máximo elementos bajo el ordenamiento $\leq$.[7] En particular, las fórmulas en lógica de primer-orden necesitan acceso a un ordenamiento total del universo para expresar computación general. Siendo así, sea $\mathcal{A} \in \text{ESTRUC}[\tau]$ una estructura ordenada, con $n = \|\mathcal{A}\|$, y sean los elementos de $|\mathcal{A}|$ en orden creciente $a_0, a_1, ..., a_{n-1}$; entonces hay una correspondencia uno a uno $i \mapsto a_i$, $i = 0, 1, ..., n-1$. (Usualmente uno identifica los elementos del universo con el conjunto de números naturales menores que n).

Por otro lado, para descartar el caso trivial de la estructura con un único elemento que satisfaría la ecuación $0 = 1$, suponemos que todas las estructuras tienen al menos dos elementos; en particular, suponemos que tienen dos constantes desiguales denotadas por 0 y 1. Además, para dar la definición de consulta booleana, también debemos definir qué significa tener una *variable booleana* en una fórmula de primer-orden: tal variable en tal fórmula es una variable que está restringida a ser 0 o 1, donde 0 se identifica con falso y 1 con verdadero.

Ahora podemos dar las definiciones de consulta y de consulta booleana como son presentadas por Immerman:

> Una *consulta* es cualquier mapeo $I : \text{ESTRUC}[\sigma] \to \text{ESTRUC}[\tau]$ de estructuras de un vocabulario a estructuras de otro vocabulario, el cual está limitado polinómicamente. Es decir, hay un polinomio p tal que para toda $\mathcal{A} \in \text{ESTRUC}[\sigma]$, $\|I(\mathcal{A})\| \leq p(\|\mathcal{A}\|)$. Una *consulta booleana* es un mapeo $I_b : \text{ESTRUC}[\sigma] \to \{0, 1\}$. Una consulta booleana también se puede considerar como un subconjunto de ESTRUC[σ]— el conjunto de estructuras $\mathcal{A}$ para las cuales $I(\mathcal{A}) = 1$.

[7] Llamamos respectivamente a los símbolos '$\leq$','0', '1' y '*max*', relación *numérica* y símbolos de constante; mientras que llamamos al resto de τ la relación *de entrada* y símbolos de constante. Los primeros dependen sólo del tamaño del universo; así, éstos no se proporcionan explícitamente en la entrada porque son fácilmente computables como funciones del tamaño de la entrada. Además, siempre que aparezca alguno de ellos, se requiere que tenga su significado estándar.

([24], 17).[8]

Dentro del contexto, una consulta de primer-orden es el tipo más simple de consulta; cada una de ellas es o bien booleana o bien una consulta de primer-orden k-aria. Aunque es fácil dar una definición general de la segunda, aquí sólo nos concierne la primera: cualquier oración de primer-orden $\phi \in \mathcal{L}(\tau)$ define una consulta booleana de primer-orden I_ϕ sobre ESTRUC[τ], donde $I_\phi(\mathcal{A}) = 1$ syss $\mathcal{A} \models \phi$. Siendo así, sea FOL el conjunto de consultas booleanas $\mathcal{L}(\tau)$-expresables. Ejemplos claros abundan en la teoría de bases de datos. TCD es de especial importancia en dicha teoría porque, como se mencionó arriba, una base de datos relacional es exactamente una estructura relacional finita, y los lenguajes de consulta comúnmente utilizados son extensiones simples de la lógica de primer-orden. Siendo así, TCD proporciona un fundamento natural para la teoría de bases de datos, y muchas cuestiones sobre la expresividad de los lenguajes de consulta y la eficiencia de su evaluación han sido establecidas utilizando métodos de TCD.

Por ejemplo, consideremos una base de datos D que contiene los nombres de las principales ciudades en México y los pares $\langle f, t\rangle$ de ciudades tales que una aereolínea determinada ofrece servicio desde f hasta t sin escala. Uno puede ver a D como una estructura de primer-orden, a saber, como una digráfica $\mathcal{G} = \langle |\mathcal{G}|, A^{\mathcal{G}}, f^{\mathcal{G}}, t^{\mathcal{G}}\rangle$, donde $|\mathcal{G}|$ es el conjunto de nombres de las ciudades y Aft significa que hay un vuelo desde f hasta t sin escala. Así, el lenguaje de la lógica de primer-orden, $\mathcal{L}(\tau)$, puede ser interpretado como un lenguaje de consulta. Como ejemplo, consideremos

[8] A *query* is any mapping I : STRUC[σ] $\to$ STRUC[τ] from structures of one vocabulary to structures of another vocabulary, that is polynomially bounded. That is, there is a polynomial p such that for all $\mathcal{A} \in$ STRUC[σ], $\|I(\mathcal{A})\| \leq p(\|\mathcal{A}\|)$. A *boolean query* is a map I_b : STRUC[σ] $\to \{0, 1\}$. A boolean query may also be thought of as a subset of STRUC[σ]—the set of structures $\mathcal{A}$ for which $I(\mathcal{A}) = 1$.
Todas las traducciones en esta tesis fueron realizadas por el autor de la misma.

$$\phi_{par}(x, y) \equiv Axy \vee \exists z(Axz \wedge Azy).$$

Cuando ϕ_{par} se considera como una consulta a D, la respuesta consistirá en el conjunto de pares $\langle f, t\rangle$ de ciudades tales que f se puedan alcanzar desde t haciendo, a lo más, una parada.

Aunque $\mathcal{L}(\tau)$ proporciona una clase abundante de consultas de bases de datos, existen consultas plausibles que no son $\mathcal{L}(\tau)$-expresables. Por ejemplo, la consulta ALCANCE, "¿Es posible volar de x a y (en uno o más vuelos)?", no se puede expresar mediante una $\mathcal{L}(\tau)$-fórmula de modo que obtengamos la respuesta correcta en todas las bases de datos (digráficas) posibles de tipo D. Correspondientemente, desde un punto de vista computacional, las respuestas a ALCANCE son más complejas que las respuestas a ϕ_{par} (véase [13], 118). En realidad, $\mathcal{L}(\tau)$ no es lo suficientemente poderoso como para expresar las computaciones más interesantes ya que carece de la capacidad de expresar adecuadamente procedimientos inductivos o recursivos. Sin embargo, su expresividad se puede aumentar de una manera útil y natural sin saltar (en principio) a la lógica de segundo-orden: añadiendo a $\mathcal{L}(\tau)$ el poder de definir nuevas relaciones por inducción, muchas de las cuales no son $\mathcal{L}(\tau)$-expresables.[9]

Por ejemplo, la relación que formaliza la consulta ALCANCE, conocida como clausura transitiva, no es $\mathcal{L}(\tau)$-expresable, pero puede definirse inductivamente. Con el vocabulario τ_g introducido arriba, se puede definir la clausura reflexiva y transitiva A^* de A de la siguiente manera: Consideremos la fórmula,

$$\phi_{ct}(R, x, y) \equiv x = y \vee \exists z(Axz \wedge Rzy),$$

[9]Por ejemplo, en términos de máquinas de Turing, usando la relación de ordenamiento podemos describir la transición de una configuración a la siguiente por medio de una lógica "simple", en su mayoría FOL. Sin embargo, para averiguar si la computación se detiene y obtener su resultado, necesitamos más poder expresivo (e.g., como se muestra abajo, el poder expresivo proporcionado por el operador OPFM).

donde R es una variable de relación binaria. Esta fórmula formaliza una definición inductiva de A^*: Para cualquier estructura $\mathcal{A}$ con vocabulario τ_g, ϕ_{ct} induce un mapeo de relaciones binarias sobre $|\mathcal{A}|$ a relaciones binarias sobre $|\mathcal{A}|$,

$$\phi_{ct}^{\mathcal{A}} = \{\langle a, b\rangle \mid \mathcal{A} \models (R, a, b)\}.$$

$\phi_{ct}^{\mathcal{A}}$ es llamado monótono si para todas las variables de relación binaria R, S,

$$R \subseteq S \Rightarrow \phi_{ct}^{\mathcal{A}}(R) \subseteq \phi_{ct}^{\mathcal{A}}(S).$$

Como R aparece sólo de manera positiva (i.e., dentro de un número par de símbolos de negación) en ϕ_{ct}, $\phi_{ct}^{\mathcal{A}}$ es monótono. Ahora, sea $(\phi_{ct}^{\mathcal{A}})^r$, $\phi_{ct}^{\mathcal{A}}$ iterado r veces. Por tanto, con $\mathcal{A}$ siendo cualquier gráfica y $r \geq 0$,

$$(\phi_{ct}^{\mathcal{A}})^1(\emptyset) = \{\langle a, b\rangle \in |\mathcal{A}|^2 \mid distancia(a, b) \leq 0\}$$

$$(\phi_{ct}^{\mathcal{A}})^2(\emptyset) = \{\langle a, b\rangle \in |\mathcal{A}|^2 \mid distancia(a, b) \leq 1\},$$

y en general,

$$(\phi_{ct}^{\mathcal{A}})^r(\emptyset) = \{\langle a, b\rangle \in |\mathcal{A}|^2 \mid distancia(a, b) \leq r - 1\}.^{10}$$

Por lo tanto, para $n = \|\mathcal{A}\|$, $(\phi_{ct}^{\mathcal{A}})^n(\emptyset) = A^*$. Es decir, A^* es la relación mínima T tal que $\phi_{ct}^{\mathcal{A}}(T) = T$, i.e., el punto fijo mínimo de $\phi_{ct}^{\mathcal{A}}$. Esta es una situación general, como muestra la versión finita del Teorema de Knaster-Tarski:

[10] $$\phi_{dist1}(x, y) \equiv x = y \vee Axy,$$
$$\phi_{dist2}(x, y) \equiv \exists z(\phi_{dist1}(x, z) \wedge \phi_{dist1}(z, y)),$$
$$\phi_{dist4}(x, y) \equiv \exists z(\phi_{dist2}(x, z) \wedge \phi_{dist2}(z, y)),$$

y así sucesivamente. Donde el significado intuitivo de cada fórmula es que, respectivamente, hay un camino de x a y de longitud máxima 1, 2, 4 y así sucesivamente. Así, para un par a y b de vértices del universo de una gráfica $\mathcal{G}$, $(\mathcal{G}, a/x, b/y) \models \phi_{dist2}$ significa que la distancia de a a b en $\mathcal{G}$ es como máximo 2 (véase [24], 9).

> *Sea R un nuevo símbolo de relación de aridad k, y sea $\phi(R, x_1, ..., x_k)$ una fórmula de primer-orden monótona. Entonces, para cualquier estructura finita $\mathcal{A}$, el punto fíjo mínimo de $\phi^{\mathcal{A}}$ existe. Éste es igual a $(\phi^{\mathcal{A}})^r(\emptyset)$, donde r es mínimo y de modo que $(\phi^{\mathcal{A}})^r(\emptyset) = (\phi^{\mathcal{A}})^{r+1}(\emptyset)$. Además, siendo $n = \|\mathcal{A}\|$, tenemos $r \leq n^k$.* ([24], 58).[11]

El teorema nos asegura que cualquier fórmula de primer-orden R-positiva, $\phi(R, x_1, ..., x_k)$ determina una relación de punto fijo mínimo. Para denotar este punto fijo mínimo usamos $(\text{OPFM}_{R^k x_1 ... x_k}\ \phi)$, donde el subíndice "$R^k x_1 ... x_k$" nos dice explícitamente a cuáles relación y variables individuales corresponde el punto fijo (si la elección de las variables es clara, estos subíndices pueden omitirse). De esta manera, el operador de punto fijo mínimo, OPFM, formaliza el poder de definir nuevas relaciones por inducción. Por consiguiente, definimos $\mathcal{L}_{OPFM}(\tau)$, el lenguaje de las definiciones inductivas de primer-orden, agregando OPFM a $\mathcal{L}(\tau)$: si $\phi(R, x_1, ..., x_k)$ es una fórmua R^k-positiva en $\mathcal{L}_{OPFM}(\tau)$, enotnces $(\text{OPFM}_{R^k x_1 ... x_k}\ \phi)$ se usa como un nuevo símbolo de relación k-aria que denota el punto fijo mínimo de ϕ. (Véase ([24], 59). Por tanto, sea LPO(OPFM) el conjunto de consultas booleanas expresables en la clausura de la lógica de primer-orden bajo el poder de definir inductivamente nuevas relaciones; i.e., el conjunto de consultas booleanas $\mathcal{L}_{OPFM}(\tau)$-expresables. (Veremos los aspectos de la sintaxis y la semántica de LPO(OPFM) relevantes para mis propósitos en el siguiente capítulo).

Por ejemplo, $(\text{OPFM}_{Rxy}\ \phi_{ct})$ denota A^* y, por tanto, la consulta booleana ALCANCE

[11] *Let R be a new relation symbol of arity k, and let $\phi(R, x_1, ..., x_k)$ be a monotone first-order formula. Then for any finite structure $\mathcal{A}$, the least fixed point of $\phi^{\mathcal{A}}$ exists. It is equal to $(\phi^{\mathcal{A}})^r(\emptyset)$ where r is minimal so that $(\phi^{\mathcal{A}})^r(\emptyset) = (\phi^{\mathcal{A}})^{r+1}(\emptyset)$. Furthermore, letting $n = \|\mathcal{A}\|$, we have $r \leq n^k$.*
En el sentido de que el teorema muestra que para cualquier definición inductiva de primer-orden $\phi(R, x_1, ..., x_k)$, el punto fijo mínimo de ϕ equivale a iterar ϕ a lo más n^k veces, OPFM es un operador de iteración polinómica.

se puede expresar como: (OPFM$_{Rxy}$ ϕ_{ct}) (f, t). La complejidad asociada con ALCANCE (consulta definida como el conjunto de digráficas $\mathcal{G}$ tales que hay un camino en $\mathcal{G}$ de f a t) es NL-completa. No obstante, $\mathcal{L}_{OPFM}(\tau)$ puede expresar consultas booleanas cuya complejidad se ha probado que es P-completa. Un ejemplo es la consulta ALCANCE$_a$, definida como el conjunto de gráficas que tienen un camino alterno de f a t:[12] La propiedad de camino alterno, P_a, se puede definir como

$$\phi_{ca}(P, x, y) \equiv x = y \ \vee (\exists z(Axz \wedge Pzy) \ \wedge (Ux \rightarrow \forall z(Axz \rightarrow Pzy))),$$

y, por lo tanto,

$$P_a = (\text{OPFM}_{Pxy}\ \phi_{ca}),$$

$$\text{ALCANCE}_a = (\text{OPFM}_{Pxy}\ \phi_{ca})(a, b)$$

(véase [24], 59).

Basados en el hecho de que $\mathcal{L}_{OPFM}(\tau)$ puede expresar P-completud, Immerman ([23]) y Vardi ([31]) probaron independientemente que LPO(OPFM) describe exactamente el conjunto de todas las consultas booleanas computables en tiempo-polinómico (i.e., P). Específicamente:

[12]Los caminos alternos se definen inductivamente. Una gráfica alterna $\mathcal{G} = \langle\ |\mathcal{G}|, A^{\mathcal{G}}, U^{\mathcal{G}}, f^{\mathcal{G}}, t^{\mathcal{G}}\rangle$ es una digráfica cuyos vértices están etiquetados universal o existencial, cuyo vocabulario es $\tau_{ag} = \langle A^2, U^1, f, t\rangle$ y donde $U \subseteq |\mathcal{G}|$ es el conjunto de vértices universales. Estas gráficas tienen una noción diferente de accesibilidad. Sea $P_a^{\mathcal{G}}(x, y)$ la relación más pequeña sobre los vértices de $\mathcal{G}$ tal que:

1. $P_a^{\mathcal{G}}(x, x)$.
2. Si x es existencial y $P_a^{\mathcal{G}}(z, y)$ se sostiene para alguna arista (x, z), entonces $P_a^{\mathcal{G}}(x, y)$.
3. Si x es universal, hay al menos una arista dejando x, y $P_a^{\mathcal{G}}(z, y)$ se sostiene para todas las aristas (x, z), entonces $P_a^{\mathcal{G}}(x, y)$.

Por lo tanto:

$$\text{ALCANCE}_a = \{\mathcal{G} \mid P_a^{\mathcal{G}}(f, t)\}$$

. (véase [24], 53-54).

Sobre estructuras ordenadas, finitas, LPO(OPFM) = P ([24], 60).[13]

De este manera, contamos con una de las dos identidades necesarias para plantear el problema P $\stackrel{?}{=}$ NP en términos meramente lógicos. Pasemos a la otra.

El lenguaje de lógica de segundo-orden, llamémoslo $\mathcal{L}_2(\tau)$, es (en principio) estrictamente mucho más expresivo que $\mathcal{L}(\tau)$, ya que el el primero consiste en el segundo más el poder de cuantificar sobre nuevas variables de relación en el universo. Como ejemplo, la $\mathcal{L}_2(\tau)$-fórmula $\forall R^r \phi$ significa que para *todas las elecciones* de la relación r-aria R, ϕ se sostiene. Siendo así, sea LSO el conjunto de consultas booleanas $\mathcal{L}_2(\tau)$-expresables. (Veremos los aspectos de la sintaxis y la semántica de LSO relevantes para mis propósitos en el siguiente capítulo).

$\mathcal{L}_2(\tau)$ es tan expresivo que incluso un "fragmento" de él, viz., su (sub)conjunto de $\mathcal{L}_2(\tau)$-fórmulas existenciales (i.e., su conjunto de fórmulas cuya cuantificación de segundo-orden está restringida a ser existencial), puede expresar NP-completud. Por ejemplo, consideremos la fórmula,

$$\phi_{3-color} \equiv \exists R^1 \exists V^1 \exists B^1 \forall x((Rx \vee Vx \vee Bx) \wedge \forall y(Axy \rightarrow$$
$$\neg(Rx \wedge Ry) \wedge \neg(Vx \wedge Vy) \wedge \neg(Bx \wedge By))),$$

donde R, V y B son variables de relación unaria. $\phi_{3-color}$ es satisfecha por una gráfica $\mathcal{G}$ syss $\mathcal{G}$ es 3-coloreable. Por lo tanto, dicha $\mathcal{L}_2(\tau)$-fórmula existencial expresa la consulta de 3-coloreabilidad de gráficas, cuya complejidad asociada es NP-completa. (Llamemos LSO$\exists$ al conjunto de consultas booleanas expresables en el "fragmento" existencial de $\mathcal{L}_2(\tau)$).

[13] *Over finite, ordered structures,* FO(LFP) = P.
Immerman usa FO(LFP) para abreviar *first-order logic plus a least-fixed-point operator*, denotando lo mismo que LPO(OPFM).

Fagin ([15] y [16]) probó que NP = LSO∃ y, más generalmente, Stockmeyer ([28]) probó que las consultas booleanas $\mathcal{L}_2(\tau)$-expresables son exactamente aquellas computables en la jerarquía de tiempo-polinómico. En otras palabras, el resultado de Stockmeyer establece:

> *Una consulta booleana se encuentra en la jerarquía de tiempo-polinómico syss es expresable en segundo-orden*, PH = LSO ([24], 121-122).[14]

La identificación entre PH y LSO probada por Stockmeyer es una extensión natural de la identificación entre NP y LSO∃ probada por Fagin; la cual, a su vez, se basa en la correspondencia entre NP y *espectros generalizados*. Específicamente, para $k \geq 1$, consideremos una k-fórmula de segundo-orden abierta ϕ, que está definida de la siguiente manera: está escrita en $\mathcal{L}_2(\tau)$ y en forma normal prenexa con todos los cuantificadores de segundo-orden precediendo a todos los cuantificadores de primer-orden; además, tiene k-1 alternancias de cuantificadores de segundo-orden encabezadas por cuantificador existencial, y contiene al menos una variable de predicado libre y ninguna variable individual libre. Ahora, para $k \geq 1$, sea GS_k el conjunto de $\mathbb{A}$ tales que, para alguna k-fórmula de segundo-orden abierta ϕ, $\mathbb{A}$ es el conjunto de estructuras finitas (ordenadas o no ordenadas) en las cuales ϕ es verdadera.[15] Fagin ([15] y [16]) describió una codificación e de conjuntos de estructuras finitas a lenguajes, y probó que $\mathbb{A} \in GS_1$ syss $e(\mathbb{A}) \in$ NP (i.e., Σ_1^p). Extendiendo este resultado, Stockmeyer ([28]) mostró que, para cada k, $\mathbb{A} \in GS_k$ syss $e(\mathbb{A}) \in \Sigma_k^p$ (análogamente para las clases correspondientes Π_k^p y Δ_k^p).

Del resultado de Stockmeyer junto con los de Immerman y Vardi, se sigue una caracterización descriptiva del problema P $\stackrel{?}{=}$ NP: P = NP syss cada consulta booleana

[14] *A boolean query is in the polynomial-time hierarchy iff it is second-order expressible*, PH = SO. Immerman usa SO para abreviar *second-order logic*, denotando lo mismo que LSO.

[15] Notablemente, los resultados de Fagin y Stockmeyer se sostienen incluso sin la suposición de ordenamiento correspondiente.

$\mathcal{L}_2(\tau)$-expresable (sobre estructuras ordenadas, finitas) es $\mathcal{L}_{OPFM}(\tau)$-expresable (i.e., LSO $\subseteq$ LPO(OPFM)). En otras palabras:

> *Las siguientes condiciones son equivalentes:*
>
> 1. P = NP
> 2. *Sobre estructuras ordenadas, finitas,* LPO(OPFM) = LSO ([24], 122).[16]

Por lo tanto, si se mostrara que LPO(OPFM) y LSO tienen el mismo poder expresivo, entonces P = NP. En el siguiente capítulo, introduzco y esbozo un enfoque TEL para comparar la expresividad de tales lógicas y, en ese sentido, motivo dicho enfoque como una estrategia para argumentar a favor de P = NP.[17] Además de eso, como se mencionó arriba, en el capítulo 4 discutiré si hay respaldo para la adecuación del enfoque TEL tomado como tal estrategia.

Cierro esta sección señalando una condición fundamental para la plausibilidad de tomar un enfoque TEL como tal estrategia: debe ser el caso que, sobre estructuras ordenadas, finitas, PH = LSO se sostiene cuando LSO se toma con semántica general. Porque, si las semánticas estándar y general de LSO tuvieran diferencias importantes en el caso de tales estructuras, no podríamos usar adecuadamente la segunda para realizar la traducción pertinente. Considero que tal condición se cumple en la medida en que la identidad entre PH y LSO, dentro del contexto de estructuras ordenadas, finitas, exige precisamente *expresabilidad* o *definibilidad* por $\mathcal{L}_2(\tau)$-fórmulas.

[16] *The following conditions are equivalent:*
1. P = NP
2. *Over finite, ordered structures,* FO(LFP) = SO.

[17] Estoy consciente del hecho de que no hay consenso respecto a lo que cuenta como una lógica; sin embargo, en esta tesis tomo LPO(OPFM) y LSO, tal como se definieron, como lógicas que tienen sintaxis y semántica particulares.

Como señalé arriba, de acuerdo con el resultado de Stockmeyer (que extiende el resultado de Fagin), cada propiedad PH de estructuras finitas (ordenadas o no) es *definible* por una $\mathcal{L}_2(\tau)$-oración. Y, conversamente, cualquier $\mathcal{L}_2(\tau)$-oración *define* una propiedad PH. Por lo tanto, una propiedad de estructuras finitas es una propiedad PH syss es *definible* por una $\mathcal{L}_2(\tau)$-oración. Por otro lado, precisamente, la semántica general restringe la clase de estructuras de LSO a aquellas que contienen al menos todos los conjuntos y relaciones *definibles* en la respectiva estructura mediante LSO-fórmulas; i.e., restringe la atención a las estructuras que están *cerradas bajo definibilidad*. Esta situación hace que un enfoque TEL, aplicado a LPO(OPFM) y LSO y, de esa manera, tomado como una estrategia para argumentar a favor de P = NP, sea al menos plausible. Mi objetivo en los próximos dos capítulos es mostrar que dicho enfoque también está bien motivado y cuenta con respaldo como tal estrategia.

Capítulo 3

Motivación para un enfoque TEL

LPO(OPFM) y LSO tendrían el mismo poder expresivo syss para cada LPO(OPFM)-fórmula FOL hubiera una LSO-fórmula con los mismos modelos y viceversa. Sabemos que LPO(OPFM) $\subseteq$ LSO (ya que se sigue de P $\subseteq$ NP); entonces, la cuestión de un millón de dólares (literalmente) es la de determinar si LSO $\subseteq$ LPO(OPFM) (i.e., para cada LSO-fórmula hay una LPO(OPFM)-fórmula con los mismos modelos) también es el caso. Siendo así, si se proporcionara una traducción de LSO a LPO(OPFM), entonces P = NP. Como se ha enfatizado, aquí no proporciono dicha traducción; más bien, en lo que sigue muestro que un enfoque TEL, utilizado para traducir LSO a LPO(OPFM) y, de esa manera, para argumentar a favor de P = NP, en primer lugar, está bien motivado y, en segundo, cuenta con respaldo. Comienzo esta tarea introduciendo y esbozando un enfoque TEL para comparar el poder expresivo de las lógicas correspondientes. Tal enfoque TEL está basado en las técnicas proporcionadas por Manzano en [25].

Para realizar la traducción en cuestión, uno tendría que ajustar o codificar el lenguaje y las estructuras de LSO en el lenguaje y las estructuras de LPO(OPFM). En

otras palabras, la traducción tendría que realizarse en dos niveles: una codificación sintáctica de LSO-expresiones en LPO(OPFM)-expresiones y una conversión semántica de LSO-estructuras en LPO(OPFM)-estructuras. Siendo así, para realizar la traducción, uno debería definir una función recursiva COD para hacer la codificación y una conversión directa CON de estructuras. Específicamente, querríamos que lo siguiente sea verdadero: para cada LSO-estructura, $\mathcal{A} \in$ EST(LSO), hay una LPO(OPFM)-estructura, $CON(\mathcal{A}) \in CON$(EST(LSO)), tal que

$$\mathcal{A} \text{ es un modelo de } \phi \text{ syss } CON(\mathcal{A}) \text{ es un modelo de } COD(\phi)$$

para cada LSO-oración ϕ. Llamemos a este desiderátum D. Antes de esbozar cómo se podrían definir COD y CON con el objetivo de probar D, echemos un vistazo a las similitudes y diferencias entre LSO y LPO(OPFM), ambas extensiones de LPO.

Comencemos con los lenguajes correspondientes: los lenguajes de ambas lógicas incluyen como base al lenguaje de primer-orden $\mathcal{L}(\tau)$ como fue presentado arriba. Sin embargo, por un lado, el lenguaje de LSO tiene tanto variables relacionales como individuales, y ambos tipos de variables pueden ser cuantificadas. De esta manera, el lenguaje de LSO, $\mathcal{L}_2(\tau)$, añade a $\mathcal{L}(\tau)$ variables n-arias de relación de cualquier grado n (para $n \geq 1$, cualquier entero positivo).[1] Es decir, contiene variables individuales (VAR), variables relacionales unarias ($\text{VAR}_1 = \{X^1, Y^1, Z^1, ...\}$), variables relacionales binarias ($\text{VAR}_2 = \{X^2, Y^2, Z^2, ...\}$), y así sucesivamente.[2]

Por otro lado, el LPO(OPFM)-lenguaje, $\mathcal{L}_{OPFM}(\tau)$, amplía las reglas de formación habituales para la $\mathcal{L}(\tau)$-sintáxis con un operador que forma puntos fijos mínimos, viz.,

[1] También agrega '$\imath$', cuya iteración indica la aridad de las variables relacionales y está representada por los números enteros positivos de la manera obvia.

[2] $\mathcal{L}_2(\tau)$ tiene igualdad tanto para los símbolos de individuo como de relación.

OPFM. La $\mathcal{L}_{OPFM}(\tau)$-sintáxis está definida por las reglas habituales para $\mathcal{L}(\tau)$ ampliadas con la siguiente regla de construcción de fórmulas: Si $\phi(R, \bar{x})$ es una fórmula (donde $\bar{x}$ es una k-tupla de variables individuales libres, R es una variable de relación k-aria libre y ϕ es una fórmula en la que R aparece sólo de manera positiva), entonces $(\mathrm{OPFM}_{R,\bar{x}}\phi)(\bar{t})$ es una fórmula (donde $\bar{t}$ es una k-tupla de términos, todas las apariciones de R están ligadas y todas las apariciones de las variables en $\bar{x}$, excepto aquellas que aparecen en $\bar{t}$ y aquellas libres en ϕ, están ligadas).

Ahora, echemos un vistazo a las estructuras de las lógicas en cuestión: como era de esperarse, las estructuras de ambas lógicas se basan en LPO-estructuras. Por un lado, las LSO-estructuras deben contener diferentes universos: el universo de los individuos $|\mathcal{A}|$ (que es un conjunto no vacío), sobre el cual se extienden las variables individuales; el universo de las relaciones unarias $|\mathcal{A}|_1$, como la extensión para las variables relacionales unarias; el universo de las relaciones binarias $|\mathcal{A}|_2$, como la extensión para las variables relacionales binarias, y así sucesivamente. Sin embargo, aparte del universo de individuos, el resto de los universos no son completamente nuevos ya que éstos no contienen más que relaciones entre individuos. En otras palabras, construimos el resto de los universos a partir de elementos en el universo de individuos. En las LSO-estructuras, requerimos que el universo de las relaciones n-arias sea un subconjunto del conjunto potencia del producto cartesiano del universo de idividuos aplicado n-veces. Es decir, si $\mathcal{A}$ es una LSO-estructura, entonces $|\mathcal{A}|_1 \subseteq \mathcal{P}|\mathcal{A}|$, $|\mathcal{A}|_2 \subseteq \mathcal{P}|\mathcal{A}|^2$, y así sucesivamente.

No obstante, hay dos tipos diferentes de LSO-estructuras: estándar y no-estándar. En el caso de las LSO-estructuras estándar, tomamos $|\mathcal{A}|_n = \mathcal{P}|\mathcal{A}|^n$; i.e., por ejemplo, tomamos el significado de la fórmula $\forall X^1\phi$ como: para todos los subconjuntos posibles de $|\mathcal{A}|$, ϕ se

sostiene. Al hacer eso, estamos tomando la noción de subconjunto de la teoría de conjuntos subyacente que estamos usando en el metalenguaje y, así, tal noción está fijada (por tanto, se trata como un concepto primitivo). Esta situación nos obliga, por ejemplo, a incluir en $|\mathcal{A}|_1$ todos los subconjuntos de $|\mathcal{A}|$, incluso todos aquellos que nunca podríamos *describir* o *definir*. Por lo tanto, en las LSO-estructuras estándar, cada $|\mathcal{A}|_n$ contiene todas las posibles relaciones n-arias sobre $|\mathcal{A}|$, donde "posibles" se inscribe en la teoría de conjuntos subyacente utilizada en el metalenguaje. Dicho de otra manera, en tales estructuras, en tanto que conocemos $|\mathcal{A}|$, nos basamos completamente en la teoría de conjuntos para decidir qué se considerará como relación n-aria: no hay indicación en la estructura $\mathcal{A}$ diciendo la propiedad que se ha de cumplir para ser una relación n-aria, sino que dicha propiedad se toma de la (meta)teoría de conjuntos. Por lo tanto, para establecer el conjunto de las oraciones válidas debemos especificar qué teoría de conjuntos será usada en el metalenguaje.[3]

En cambio, aunque en las LSO-estructuras no-estándar todavía se quiere que $|\mathcal{A}|_n \subseteq \mathcal{P}|\mathcal{A}|^n$ para todo n, la decisión de qué será incluido en cada $|\mathcal{A}|_n$ (i.e., qué posibles relaciones estarán contenidas realmente en la estructura $\mathcal{A}$) es hasta cierto punto peculiar para cada estructura $\mathcal{A}$. Se podría considerar que las relaciones a las que nos referimos al cuantificar en LSO no siempre tienen que ser todas las posibles; por ejemplo, a menudo queremos cuantificar sobre ciertos conjuntos de relaciones que son fáciles de controlar. Así, en las estructuras en cuestión, podría ser que para algunos m, $|\mathcal{A}|_m \neq \mathcal{P}|\mathcal{A}|^m$. Específicamente, en las LSO-estructuras no-estándar, la extensión de los cuantificadores de segundo orden debe especificarse directamente; i.e., en estructuras no-estándar, la noción de subconjunto debe darse explícitamente dentro de cada modelo (así, es tratada como un concepto definido en

[3] Esta situación hace que LSO sea una lógica no-absoluta, cuyo concepto de verdad depende de la teoría de conjuntos subyacente.

la estructura). En particular, hay LSO-estructuras no-estándar en las que los universos son extensionales y *cerrados bajo definibilidad*, viz., LSO-estructuras generales. Dicho de otra manera, estas son estructuras donde los universos contienen al menos todos los conjuntos y relaciones que son definibles en la estructura correspondiente por las fórmulas del lenguaje de LSO, i.e., por $\mathcal{L}_2(\tau)$-fórmulas.

Como es bien sabido, tomada con estructuras estándar, LSO es estrictamente mucho más expresivo que LPO y su extensión LPO(OPFM). Por otro lado, como también es bien sabido, tomada con estructuras generales, LSO no es más expresivo que LPO; de hecho, con estructuras generales, LSO es *traducible* a LPO multivariada (véase [25], 277-290). A grandes rasgos, tomada con estructuras estándar, el conjunto de las oraciones válidas de LSO es tan vasto porque la clase de tales estructuras es demasiado pequeña (esa clase incluye sólo estructuras donde cada $|\mathcal{A}|_n = \mathcal{P}|\mathcal{A}|^n$); mientras que, tomado con estructuras generales, dicho conjunto se reduce considerablemente porque la clase de esas estructuras es muy grande (esa clase contiene estructuras donde $|\mathcal{A}|_n \subseteq \mathcal{P}|\mathcal{A}|$, para todo n, pero podría ser que para algún m, $|\mathcal{A}|_m \neq \mathcal{P}|\mathcal{A}|^m$). Siendo así, la adopción de estructuras generales implica una gran reducción en el poder expresivo de LSO. No obstante, como se mencionó arriba y según mis propósitos, considero que, en la medida en que PH = LSO sólo exige definibilidad mediante $\mathcal{L}_2(\tau)$-fórmulas, podemos tomar adecuadamente LSO con estructuras generales.

Además, podemos argumentar que las LSO-estructuras estándar no son lógicamente adecuadas en la medida en que no permiten todas las interpretaciones lógicamente posibles de las $\mathcal{L}_2(\tau)$-fórmulas como modelos. Es decir, si no se exige alguna condición sobre los universos de una estructura, distinta a la de ser un subconjunto del conjunto potencia del producto cartesiano del universo de individuos aplicado n-veces, bien puede suceder

que la estructura falle en contener ciertas relaciones que son definibles en ella mediante $\mathcal{L}_2(\tau)$-fórmulas. Esto, a su vez, significa que el esquema de comprehensión no se sostiene necesariamente. Por esta razón, se vuelve evidente la necesidad de estructuras donde los universos relacionales obedezcan ciertas condiciones de cierre: se necesitan LSO-estructuras generales.[4] Siendo así, procedamos a dar algunas características más específicas de tales estructuras para después compararlas con LPO(OPFM)-estructuras.

Como se mencionó arriba, en las LSO-estructuras generales la extensión de los cuantificadores de segundo-orden debe estar directamente especificada. Específicamente, una forma prototípica de caracterizar a las SOL-estructuras generales es la siguiente: una *pre-estructura general* para $\mathcal{L}_2(\tau)$ es una estructura en el sentido usual (i.e., como se describe al inicio del capítulo 2, un universo de discurso más interpretaciones para los símbolos no-lógicos) junto con un conjunto adicional; a saber, el *universo relacional* n-ario.[5] Éste debe ser una colección de relaciones n-arias sobre el universo de discurso. Como ejemplo, en particular, el universo relacional unario debe ser una colección de subconjuntos del universo; por tanto, es parte (pudiendo ser todo) del conjunto potencia del universo.

Para una pre-estructura general $\mathcal{H}$, hay una manera natural de definir lo que significa que una $\mathcal{L}_2(\tau)$-fórmula ϕ es satisfecha en $\mathcal{H}$ bajo una asignación s de objetos a las variables libres en ϕ; i.e., $\mathcal{H} \models \phi[s]$ (donde los cuantificadores de segundo-orden ahora se definen

[4] Para ver por qué las LSO-estructuras estándar no incluyen todas las interpretaciones lógicamente posibles de las LSO-fórmulas como modelos, tenemos que considerar fórmulas, como la que expresa la hipótesis del continuo generalizada, llamémosla ϕ_{hcg}, que son tanto expresables en LSO como independientes de la teoría de conjuntos de Zermelo-Fraenkel. Por ejemplo, en el caso de ϕ_{hcg}: dentro de las LSO-estructuras estándar, debemos ubicarnos en un universo conjunto-teórico donde la hipótesis del continuo generalizada es o bien verdadera o bien falsa. Supongamos que es verdadera; entonces, en cada modelo estándar de LSO ϕ_{hcg} no sólo es verdadera sino válida. No obstante, dado que la hipótesis generalizada no es derivable de la teoría de conjuntos de Zermelo-Fraenkel, una interpretación $\mathcal{I}$, tal que $\mathcal{I} \nvDash \phi_{hcg}$, no se puede excluir como lógicamente imposible. Por tanto, al menos una $\mathcal{I}$, tal que $\mathcal{I} \vDash \neg\phi_{gch}$, es lógicamente posible; sin embargo, dicho modelo no está permitido en LSO-estructuras estándar.

[5] Como era de esperarse, el lenguaje de LSO, $\mathcal{L}_2(\tau)$, permanece inalterado bajo una u otra semántica.

para que su extensión esté dada sobre el universo correspondiente). Para una variable relacional k-aria X^k: $\mathcal{H} \models \forall X^k\ \phi[s]$ syss para cada k-relación P en el universo relacional k-ario, $\mathcal{H} \models \phi[s']$ es el caso; donde s' difiere de s sólo en asignar la relación P a la variable relacional X^k. (Como era de esperarse, en el caso de oraciones de segundo-orden, la asignación s ya no es relevante y podemos hablar sin ambigüedad de la verdad o falsedad de la oración correspondiente en la pre-estructura general $\mathcal{H}$.

Sin embargo, por ejemplo, realmente no deseamos que el universo relacional unario sea una colección arbitraria de subconjuntos del universo. Hay algunos subconjuntos del universo sobre los que tenemos conocimiento ya que podemos definirlos: supongamos que ϕ es una fórmula donde sólo la variable u aparece libre; entonces, el conjunto que ϕ define en $\mathcal{H}$ consiste en todos los miembros a de $\mathcal{H}$ tales que ϕ es satisfecha en $\mathcal{H}$ cuando a se asigna a u. Obviamente, esta situación puede ampliarse: supongamos que ϕ tiene como únicas variables libres u, v, w, x, Y^m. Además, supongamos que c y d son miembros del universo (de individuos) $|\mathcal{H}|$ y que P está en el universo relacional m-ario de $\mathcal{H}$. Entonces, la relación binaria que ϕ define en $\mathcal{H}$ *a partir de los parámetros* c, d y P es el conjunto de pares $\langle a, b\rangle$ de elementos de $|\mathcal{H}|$ tales que ϕ está satisfecho en $\mathcal{H}$ cuando a sus variables u, v, w, x y Y^m se les asigna a, b, c, d y P, respectivamente. Dicho de otra manera, es la relación binaria

$$\{\langle a, b\rangle \mid \mathcal{H} \models \phi(u, v, w, x, Y^m)[a, b, c, d, P]\}$$

De nuevo, obviamente, esta situación se puede generalizar al caso en que se define una relación k-aria a partir de cualquier número particular de parámetros.

Por lo tanto, es razonable restringir nuestra atención a las pre-estructuras generales que están cerradas bajo definibilidad. Así, particularmente, en la situación recién descrita,

esperamos razonablemente que el universo relacional binario de $\mathcal{H}$ contenga la relación binaria que ϕ define a partir de los parámetros en la pre-estructura. En otras palabras, esperamos que la oración

$$\forall w\ \forall y\ \forall Y^2\ \exists R\ \forall u\ \forall v\ [Ruv \leftrightarrow \phi(u, v, w, x, Y^2)]$$

sea verdadera en $\mathcal{H}$. Llamamos a este tipo de oraciones *axiomas de comrehensión*. Así, una LSO-estructura general es una pre-estructura general en la que todos los axiomas de comprehensión (para todas las fórmulas) son verdaderos. Entre las LSO-estructuras generales están aquellas en las que, por ejemplo, el universo relacional unario es todo el conjunto potencia del universo (de individuos), etc. A una tal LSO-estructura general la llamamos absoluta, pero, como se ha enfatizado, puede haber otras (véase [14]).

Como es bien sabido, la característica más notable de las LSO-estructuras generales es que, tomada con ellas, LSO no es más que LPO (multivariada) junto con los axiomas de comprehensión: usar pre-estructuras generales equivale a tratar $\mathcal{L}_2(\tau)$ como un $\mathcal{L}(\tau)$ multivariado. Por tanto, tomada con estructuras generales, una oración de LSO es válida syss está implicada lógicamente (en LPO) por el conjunto de axiomas de comprehensión. (Con más precisión, tomada con estructuras generales, LSO es reducible a la lógica multivariada, la cual a su vez es reducible a LPO (no-variada). Así, las LSO-estructuras generales se pueden reinterpretar como LPO-estructuras multivariadas (véase [25])). Habiendo dicho esto, procedamos con la exposición de las LPO(OPFM)-estructuras para que las similitudes y diferencias correspondientes sean evidentes.

Las LPO(OPFM)-estructuras son esencialmente las mismas que las LPO-estructuras. Consideremos una fórmula de LPO $\phi(R, \bar{x})$, donde $\bar{x}$ es una k-tupla de variables individuales libres y R es una variable relacional k-aria libre. Si $\mathcal{A}$ es una LPO-estructura, con el universo

$|\mathcal{A}|$, interpretando todos los símbolos en ϕ diferentes a los exhibidos, consideramos a ϕ como definiendo un mapeo Φ de $\mathcal{P}|\mathcal{A}|^k$ a $\mathcal{P}|\mathcal{A}|^k$ dado por

$$\Phi(P) = \{\bar{a}|(\mathcal{A}, P, \bar{a}) \models \phi\},$$

donde $P \subseteq |\mathcal{A}|^k$ y $\bar{a}$ es una k-tupla de elementos de $\mathcal{A}$. Como se mencionó en el capítulo anterior, esta interpretación de una fórmula definiendo un operador sobre el espacio de las relaciones da una formalización natural de definiciones inductivas: si Φ es un mapeo monótono, podemos hablar de la mínima relación R tal que $R(\bar{x})$ syss $\phi(R, \bar{x})$; llamamos a esta relación el punto fijo mínimo del operador definido por ϕ. De esta manera, la semántica pretendida de la regla de construcción de fórmulas mencionada arriba en este capítulo es que, para cualquier LPO-estructura $\mathcal{A}$ (proporcionando una interpretación de las variables libres de ϕ, excepto para $\bar{x}$), $\mathcal{A} \models(\mathrm{LFPO}_{R,\bar{x}}\phi)(\bar{t})$ syss $\bar{t}^{\mathcal{A}}$ (la tupla de elementos de $\mathcal{A}$ definida por los términos $\bar{t}$) está en el punto fijo mínimo del operador monótono definido por $\phi(R, \bar{x})$ sobre $|\mathcal{A}|^k$. Podemos obtener el punto fijo mínimo como el límite de la secuencia de relaciones:

$$R_0 = \emptyset$$

$$R_{\alpha+1} = \Phi(R_\alpha)$$

$$R_\alpha = \bigcup_{\beta<\alpha} R_\beta$$

para ordinales de límite α. El ordinal de clausura de Φ se define como el menor ordinal α tal que $R_\alpha = R_{\alpha+1}$. En otras palabras, cada tal operador $\Phi(P)$ genera una secuencia de etapas que se obtienen al iterar $\Phi(P)$. Definimos las etapas Φ^m, m ≥ 1, de Φ sobre $\mathcal{A}$ mediante la inducción: $\Phi^1 = \Phi(\emptyset)$, $\Phi^{m+1} = \Phi(\Phi^m)$. Intuitivamente, nos gustaría asociar

con un operador $\Phi(P)$ el "límite" de sus etapas: esto solo es posible si la secuencia de etapas Φ^m, m ≥ 1, "converge", i.e., si hay un entero m_0 tal que $\Phi^{m_0} = \Phi^{m_0+1} = \Phi(\Phi^{m_0})$ y, por tanto, $\Phi^{m_0} = \Phi^m$, para todo $m \geq m_0$. Claramente, en este caso Φ^{m_0} es un punto fijo de $\Phi(P)$. (Véase [12], 66-68; [1], 5; y [13], 165-171).

Una vez que dimos un vistazo a las similitudes y diferencias entre LSO y LPO(OPFM), procedamos a esbozar cómo se podrían definir COD y CON introducidas arriba. La idea es, por un lado, codificar las LSO-expresiones en un LPO(OPFM)-lenguaje de una signatura peculiar directamente relacionada con la signatura de LSO. Es decir, nos gustaría pasar de la LSO-signatura a una LPO(OPFM)-signatura que tenga, por así decirlo, una apariencia de segundo-orden. Siendo así, denotemos LPO(OPFM) tomada con tal signatura como LPO(OPFM)*. Sin embargo, debe quedar claro que, aunque por manipulabilidad se utilizaría una LPO(OPFM)-signatura peculiar para realizar tanto COD como CON, los resultados obtenidos serían válidos para cualquier LPO(OPFM)-signatura.[6] Por otro lado, la idea es convertir las estructuras (generales) utilizadas para interpretar LSO (llamemos a su clase EST(SOL)) en estructuras para LPO(OPFM)* (llamemos a su clase EST(LPO(OPFM)*)).

Con más detalle, por un lado, la función

$$COD\text{: EXP(LSO)} \longrightarrow \text{EXP(LPO(OPFM)*)}$$

$$\varepsilon \longmapsto COD(\varepsilon)$$

tendría que definirse mediante recursión sobre la formación de expresiones de LSO. Además, desearíamos que esta función introdujera a lo más un número finito de variables libres en

[6] Una signatura de un lenguaje es un par $\langle V, F\rangle$, donde V es el conjunto que contiene todos los tipos de variables cuantificables y F es una función cuyo dominio es el conjunto de constantes de operación del lenguaje y da tipos como valores; i.e., secuencias finitas de miembros de V (véase [25], 9).

las expresiones cerradas de LSO. Por otro lado, la función

$$CON\text{: EST(LSO)} \longrightarrow \text{EST(LPO(OPFM)*)}$$
$$\mathcal{A} \longmapsto CON(\mathcal{A})$$

tendría que ser una conversión directa de LSO-estructuras (generales) en LPO(OPFM)-estructuras de una signatura particular elegida. El objetivo de esta conversión sería poder probar el desiderátum D introducido arriba; i.e., la equivalencia semántica directa, en el sentido de que la verdad de una oración de LSO en una LSO-estructura (general) $\mathcal{A}$ es equivalente a la verdad de su codificación en su estructura convertida directa $CON(\mathcal{A})$. Específicamente, reformulando D, desearíamos que lo siguiente sea verdadero:

Para cada $\mathcal{A} \in$ EST(LSO) hay una estructura $CON(\mathcal{A}) \in$ EST(LPO(OPFM)*) tal que

1. $\mathcal{A}, [\mathbf{x}_1...\mathbf{x}_n] \models \phi$ en LSO syss $CON(\mathcal{A})[\begin{smallmatrix}\mathbf{x}_1...\mathbf{x}_n\\ x_1...x_n\end{smallmatrix}](COD(\phi)[x_1...x_n]) = 1$ en LPO(OPFM)*

2. $\mathcal{A} \models \phi$ en LSO syss $CON(\mathcal{A})$ es un modelo de $\forall COD(\phi)$ en LPO(OPFM)*,

para cada oración ϕ de LSO. Donde $x_1...x_n$ son las variables libres que aparecen en la codificación (si las hay), y $\mathbf{x}_1...\mathbf{x}_n$ son elementos adecuados tanto en la estructura original como en la convertida. $\forall COD(\phi)$ es la clausura universal de la codificación de ϕ. Finalmente, $\mathcal{A}, [\mathbf{x}_1...\mathbf{x}_n] \models \phi$ representa la verdad de la fórmula ϕ con los parámetros $\mathbf{x}_1...\mathbf{x}_n$, el cual es un concepto de LSO. Si la codificación no produce variables libres, solo 2 sería aplicable; en algunas otras situaciones, 2 se seguiría de 1 (véase [25], 265-266).[7]

[7] En general, lo que se denomina una traducción entre lógicas exige considerablemente más que sólo esta equivalencia o preservación semántica. Por ejemplo, es común que exija equivalencia de consecuencia y, aún más, preservación de metapropiedades (véase [25] y [9]). No obstante, la equivalencia semántica es suficiente para mis propósitos; a saber, una vez más: comparar los poderes expresivos de LSO y LPO(OPFM).

A este respecto, también requeriríamos que el objeto correspondiente que cada expresión ε de LSO define en sus propias estructuras, $\mathcal{A} \in$ EST(LSO), sea "casi el mismo" que el objeto que cada $COD(\varepsilon)$ define en $CON(\mathcal{A})$. Es decir, también exigiríamos $\mathcal{A}(\varepsilon) \approx CON(\mathcal{A})(COD(\varepsilon))$. Esto puede lograrse forzando a las LPO(OPFM)*-estructuras a tener, por así decirlo, apariencia de LSO; en el sentido de que se comporten apropiadamente como si fueran LSO-estructuras. En particular, forzaríamos a los universos de las LPO(OPFM)*-estructuras a ser extensionales y cerrados bajo definabilidad.

Espero que a estas alturas ya no parezca loco cuando afirmo que se podría argumentar a favor de que, sobre estructuras ordenadas, finitas, todas las consultas booleanas expresables en LSO (i.e., cada consulta booleana definible por una $\mathcal{L}_2(\tau)$-oración) es ya expresable como una definición inductiva de primer-orden (i.e., como una consulta booleana definible por una $\mathcal{L}_{OPFM}(\tau)$-oración). En caso de que todavía parezca una locura, en lo que resta de este capítulo ofreceré más motivación para mi afirmación, y en el próximo capítulo también señalaré algo de respaldo para la misma.

Como he mostrado, gran parte de la motivación que tengo para mi afirmación proviene de las técnicas TEL en sí mismas, las cuales han sido meticulosamente estudiadas y validadas (véase [25] y [9]): además de la traducción ampliamente conocida entre LSO (tomada con semántica general) y LPO (multivariada), según la cual esas lógicas tienen el mismo poder expresivo y, aún más, tienen las mismas metapropiedades;[8] hay otras equivalencias de traducción relevantes, como la equivalencia entre LSO$\exists$ (la cual, como mencioné arriba, es igual a NP) y la lógica de la independencia amable (LIA; e.g., [22]), y la lógica de la dependencia (LD; e.g., [30]), las cuales comparten metapropiedades con LPO (e.g.,

[8] Así, la traducción en cuestión es una en el sentido fuerte mencionado en la nota anterior.

compacidad y propiedad de Löwenheim-Skolem) y cuyas semánticas son (discutiblemente) ambas reducibles a la semántica para LPO.[9] De esta manera, todas las propiedades NP-completas, como la 3-colorabilidad de las gráficas mencionada arriba, son expresables en LIA y en LD. A este respecto, además de la motivación para mi afirmación proporcionada directamente por el desarrollo meticuloso de las técnicas TEL, hay más motivación, que está relacionada con TEL pero no proviene directamente de él: hay nociones que *prima facie* son intrínsecamente de LSO, pero que pueden expresarse en (la mayoría de las veces, extendida) LPO.

Para ilustrar, comúnmente, usando el axioma de elección, la noción de infinitud se expresa en LSO$\exists$ como

$$\phi_{inf} \equiv \exists X^2(\forall x\ \forall y\ \forall z(X^2xy \wedge X^2yz \rightarrow X^2xz)\ \wedge\ \forall x\neg X^2xx\ \wedge\ \forall x\ \exists y X^2xy).$$

ϕ_{inf} expresa que el universo de discurso es infinito diciendo que hay una relación transitiva sobre el universo, tal que cada elemento del universo tiene la relación con algo pero no consigo mismo. Mientras tanto, comúnmente, Dedekind-infinitud se expresa también en LSO$\exists$ como

$$\phi_{Di} \equiv \exists f(\forall x\forall y(fx = fy \leftrightarrow x = y) \wedge \forall x(Px \rightarrow Pfx) \wedge \exists t(Pt \wedge \forall x(fx \neq t))).^{10}$$

ϕ_{Di} dice que la extensión de un predicado P es Dedekind-infinita si hay una función inyectiva no-sobreyectiva de P a P.

No obstante, Dedekind-infinitud se puede expresar en LIA como

$$\phi_{di} \equiv \exists t\forall x\forall z(\exists y/\forall z)(\exists w/\forall z)(x = z \leftrightarrow y = w) \wedge (Px \rightarrow Py)(Pt \wedge y \neq t)).$$

[9] Con más precisión, los poderes expresivos de LSO$\exists$, LIA y LD son los mismos con respecto a la definibilidad de clases de modelos. Sin embargo, este no es el caso para fórmulas con variables libres; además, tales lógicas se pueden extender a lo largo de líneas muy diferentes.

[10] Recuérdese la nota 2 del capítulo 2.

ϕ_{di} captura la idea de que un conjunto P es Dedekind-infinito precisamente cuando existe una función inyectiva de P a su subconjunto propio (véase [22] y [29]). Además, como es bien sabido, si asumimos el (altamente controvertido) axioma de elección, infinitud y Dedekind-infinitud son equivalentes.[11]

Controversialmente, en una serie de trabajos (e.g., [21] y [22]), Hintikka afirmó que es posible reconstruir todo el razonamiento matemático normal en el nivel de primer-orden. Con tal objetivo, introdujo, además de LIA, una extensión de la misma: ELIA; la cual, a diferencia de LIA, está cerrada bajo negación contradictoria. Por un lado, LIA puede expresar todos las Σ^1_1-oraciones; por ejemplo, aquellas que expresan la infinitud del dominio, que una relación no es un buen ordenamiento y la negación del principio de inducción aritmética. Por otro lado, ELIA puede expresar todas las Π^1_1-oraciones; por ejemplo, finitud del dominio, que una relación es un buen ordenamiento y el principio de inducción aritmética. Además, Hintikka defendió (e.g., [22]) que, hablando sustancialmente, LIA e incluso ELIA son lógicas de primer-orden. Argumentó que todas las entidades sobre las que se extienden las variables cuantificadas de esas lógicas son individuos, y también lo son todas las entidades con las que operan los jugadores de la semántica juego-teórica (con la que están equipadas tales lógicas).

Como Cook y Shapiro señalaron ([11]), se podría argumentar que aunque LIA y ELIA son sin duda sintácticamente de primer-orden; se ven semánticamente de orden-superior. En el caso de LIA, se puede argumentar que en realidad es de orden-superior ya que su semántica juego-teórica invoca estrategias ganadoras (las cuales son funciones). Pero un defensor de la propuesta de Hintikka respondería que los lenguajes(-objeto) en sí mismos

[11] La equivalencia no requiere toda la fuerza del axioma de elección; de hecho, la equivalencia es estrictamente más débil que el axioma de elección contable (véase [10]).

no invocan funciones u otros elementos de orden-superior. Por el contrario, en el caso de ELIA, es más difícil sostener que no es implícitamente de orden-superior. Dado que, según la semántica juego-teórica de ELIA, una oración en la forma $\neg\phi$ se entiende como la inexistencia de una estrategia ganadora, $\neg\phi$ es una cuantificación casi explícita sobre funciones. Sin embargo, esta última crítica pierde fuerza respecto a la afirmación general de Hintikka mencionada arriba cuando se considera que él también mostró que: 1. Para cualquier oración de n^{o}-orden ϕ, hay una oración de LSO $\phi+$ tal que si ϕ es satisfacible, entonces también lo es $\phi+$, y ϕ es una verdad lógica syss $\phi+$ es una verdad lógica ([20]). 2. $\phi+$ se puede formular como una Σ^1_1-oración ([22], Ch. 9). (Cf. [11]).[12]

Un ejemplo, quizá más adecuado, donde una noción que es *prima facie* de LSO puede expresarse en LPO (no extendida) es el principio de inducción aritmética. Tal principio se expresa comúnmente en LSO como

$$\phi_{IA} \equiv \forall X^1(X^1 0 \wedge \forall y(X^1 y \rightarrow X^1 Sy) \rightarrow \forall y X^1 y).$$

ϕ_{IA} es conocido como el axioma de inducción de segundo-orden y expresa que X^1 es verdadero de todos los elementos de $\mathbb{N}$, si es verdadero de 0 y su verdad en algún número y asegura su verdad en el sucesor de y, sin importar cuál es el conjunto de números del que X^1 podría ser verdadero. Por otro lado, hay un LPO-esquema correspondiente:

$$\phi_{ia} \equiv \psi(0) \wedge \forall y(\psi(y \rightarrow \psi(Sy)) \rightarrow \forall y \psi(y),$$

donde ψ puede ser cualquier fórmula de LPO adecuada con y como su única variable libre.

[12] La cuestión de por qué no estoy comparando el poder expresivo de LIA o LD (las cuales tienen el mismo poder expresivo que LSO$\exists$) con el de LPO(OPFM) surge naturalmente. La razón es meramente pragmática: es más fácil comparar el poder expresivo de LSO (tomada con semántica general) con la de LPO(OPFM). En la medida en la que el objetivo de esta tesis es solamente introducir un enfoque TEL para argumentar a favor de P = NP, mostrando que está bien motivado y cuenta con respaldo, esta situación no representa un problema.

ϕ_{ia} es conocido como el esquema de inducción de primer-orden y asegura que cualquier conjunto *definible* que contenga 0 y esté cerrado bajo sucesor debe contener todo. No obstante, a diferencia de los ejemplos anteriores, hay diferencias significativas entre las dos formulaciones del principio en cuestión. Como sus nombres ya lo indican, el primero no es un esquema, sino un único axioma. Con mayor importantancia, si tomamos LSO con semántica estándar, el poder expresivo de la primera formulación es más fuerte que el correspondiente conjunto infinito de axiomas de inducción de primer-orden.

Es claro que en LPO uno sólo puede aplicar inducción a una clase numerable de conjuntos porque uno sólo tiene una clase numerable de oraciones; mientras que en LSO con semántica estándar, la inducción se aplica a cualquier subconjunto del universo de individuos y ese conjunto es no-numerable. Sin embargo, aquí lo importante no es la cantidad de conjuntos a los que se puede aplicar inducción, sino la "calidad" de los mismos. Como es bien sabido, dentro de la LPO-aritmética uno tiene modelos no-estándar; es decir, modelos no-isomorfos a la estructura pretendida de $\mathbb{N}$, cuyos universos correspondientes contienen números no-estándar (i.e., números que no son sucesores de 0). Si la cadena (o conjunto) de los números estándar fuera definible en cualquier modelo de la LPO-aritmética, entonces uno no tendría modelos no-estándar en dicha aritmética; sin embargo, como también es bien sabido, esa cadena no se puede definir en LPO.

Siendo así, la principal diferencia entre las formulaciones en LSO y LPO del principio de inducción aritmética es que con la segunda no se puede aplicar inducción sobre los números estándar en un modelo que tiene números no-estándar. Por lo tanto, la formulación en LPO no es capaz de detener la aparición de números no-estándar; mientras que, como también es bien sabido, si LSO se toma con semántica estándar, la formulación en LSO detiene

tal aparición dado que la LSO-aritmética, bajo tal supuesto, es categórica (i.e., cualquiera dos modelos de la LSO-aritmética son isomorfos). Pero, ¿qué pasa si tomamos LSO con semántica general?

Aunque los significados de no-estándar en lógica de orden-superior y en aritmética no son los mismos, están estrechamente relacionados: un modelo de la LSO-aritmética que tiene números no-estándar en el universo de individuos también debe ser no-estándar respecto a LSO. La LSO-aritmética es categórica sólo cuando "conjunto", tal y como aparece en ϕ_{IA}, se interpreta con su significado estándar; i.e., cuando se toma su significado de la metateoría usando semántica estándar. En contraste, uno puede construir un modelo no-estándar de la LSO-aritmética, que es tal de las dos maneras descritas. Siendo así, uno debe abandonar el punto de vista estándar y tomar el concepto de conjunto como adjunto al modelo, y en consecuencia cambiar la semántica.

Cuando uno permite interpretaciones no-estándar, LSO pierde parte significativa de su poder expresivo y la LSO-aritmética deja de ser categórica. Sin embargo, sigue siendo más fuerte que la LPO-aritmética ya que la consistencia de la LPO-aritmética puede ser probada en la LSO-aritmética. Como se mencionó arriba, la LPO-aritmética no es categórica porque el conjunto de números estándar no es definible por una fórmula de LPO en una estructura donde hay números no-estándar y, por tanto, uno carece de inducción para tal conjunto. Por otro lado, tomando LSO con semántica estándar, la LSO-aritmética es categórica porque uno tiene inducción para todos los conjuntos posibles y una estructura con números no-estándar nunca sería un modelo de ϕ_{IA}.

Si en LSO uno permite estructuras con universos relacionales no totales, la cuantificación se restringe a los conjuntos y relaciones que están presentes en la estructura. Por

consiguiente, muy probablemente el universo de las relaciones unarias no tendría al conjunto de números estándar como uno de sus miembros. Como se explicó arriba, los universos de las estructuras generales incluyen todos los conjuntos y las relaciones que son paramétricamente definibles en la estructura correspondiente mediante fórmulas de LSO. De manera adecuada, como en el caso de LPO, el conjunto de números estándar tampoco es definible por una fórmula de LSO en una estructura que tiene números no-estándar. (Véase [25], Ch. V).

Cierro esta sección señalando un tipo más de motivación para la afirmación principal de esta tesis, el cual proviene directamente de TCD y apoya la idea de que LSO $\subseteq$ LPO(OPFM) es posible: Existen lógicas de primer-orden equipadas con ciertos operadores que, sobre estructuras ordenadas, finitas, son al menos tan expresivas como LSO y probablemente incluso más expresivas. Por ejemplo, es un resultado bien establecido de TCD que, sobre estructuras ordenadas, finitas, LSO $\subseteq$ LPO(OPFP) (véase [13], 211); donde LPO(OPFP) denota al conjunto de consultas booleanas expresables en $\mathcal{L}(\tau)$ más un operador de punto fijo parcial (OPFP). Veamos algunas de las características de LPO(OPFP) a grandes rasgos.

Consideremos $\phi(R, \bar{x})$, una fórmula arbitraria definiendo un operador (no necesariamente monótono) Φ y la secuencia de relaciones (para α finito)

$$R_0 = \emptyset$$

$$R_{\alpha+1} = \Phi(R_\alpha)$$

Así, esta secuencia no está creciendo necesariamente. Sin embargo, sobre estructuras finitas, tal secuencia o bien converge a un punto fijo, o se asienta en un ciclo de período mayor que

1. El punto fijo parcial de Φ se define como el punto fijo alcanzado en el primer caso, y la relación vacía en el segundo caso. Por consiguiente, obtenemos LPO(OPFP) cerrando la lógica de primer-orden simultáneamente bajo las reglas usuales de formación de fórmulas para la $\mathcal{L}(\tau)$-sintáxis, junto con la regla que nos permite formar la fórmula $(\mathrm{OPFP}_{R,\bar{x}}\phi)(\bar{t})$ a partir de la fórmula ϕ. Como era de esperarse, esto se usa para indicar que $\bar{t}$ es una tupla que se encuentra en el punto fijo parcial de $\phi(R,\bar{x})$.

La importancia de LPO(OPFP) radica en el hecho (mostrado por Abiteboul y Vianu en [2]) de que, sobre estructuras ordenadas, finitas, exactamente las consultas computables en el espacio polinómico son expresables o definibles en LPO(OPFP); es decir, sobre tales estructuras, LPO(OPFP) = PSPACE. Siendo así, las relaciones de LPO(OPFP) y LSO con las clases de complejidad correspondientes muestran que, sobre estructuras ordenadas, finitas, LPO(OPFP) es al menos tan expresivo como LSO. Además, bajo el supuesto ampliamante aceptado de que PH $\subsetneq$ PSPACE, LPO(OPFP) es estrictamente más expresivo. Aunque no estoy dispuesto a abogar por que LPO(OPFM) sea más expresivo que LSO ya que, después de todo, en esta tesis precisamente estoy desafiando supuestos ampliamente aceptados; como se enfatizó, es un resultado bien establecido de DCT que LPO(OPFP) es al menos tan expresivo como LSO (véase [12], 74-76; y [13], 121-122, 191-198).[13]

Después de haber mostrado que un enfoque TEL aplicado a LSO y LPO(OPFM) está

[13]Como en todas las descripciones lógicas de primer-orden de clases de complejidad, aquí el supuesto de orden es indispensable. Sin tal supuesto, se puede probar que hay propiedades que son definibles en LSO pero no en LPO(OPFP) (e.g., no existe una oración de LPO(OPFP) que sea verdadera en exactamente las estructuras finitas de tamaño par). Sin embargo, como se explicó brevemente arriba, el supuesto de orden parece muy natural dentro del contexto de computación. Por otro lado, vale la pena mencionar que existe otra lógica de primer-orden de punto fijo, cuya expresividad, sobre estructuras ordenadas, finitas, está probado que es igual a la de LSO: LPO(OPFN) (la lógica de primer-orden con un operador de punto fijo no-determinista). Así, sobre tales estructuras, LPO(OPFN) = PH. Además, está probado que, sobre tales estructuras, la expresividad del fragmento positivo libre de alternancia de esta lógica es igual a la de LSO$\exists$; i.e., ese fragmento es igual a NP (véase [12], 76-81). Sin embargo, recordando la nota anterior, dejo este asunto aquí.

bien motivado como una estrategia para argumentar a favor de P = NP, procedo a mostrar que también cuenta con respaldo para su adecuación como tal.

Capítulo 4

Respaldo para un enfoque TEL

El tipo fundamental de respaldo con el que cuenta el enfoque TEL es su exactitud matemático-conceptual: ni TCD ni las técnicas de traducción correspondientes distorsionan los fenómenos de complejidad. Para empezar, como se mostró en el capítulo 2, TCD "imita" adecuadamente a la teoría (estándar) de la complejidad. Con más precisión, las descripciones lógicas de las clases de complejidad que proporciona TCD son confiables y, por tanto, también lo son los análogos lógicos de los problemas de inclusión y separación en la teoría de la complejidad que también proporciona.

En TCD y en la teoría de la complejidad lidiamos sólo con diferentes presentaciones de las clases de complejidad. TCD proporciona un puente sólido entre la lógica y la teoría de la complejidad al relacionar caracterizaciones orientadas puramente por prototipos de máquinas de computación con caracterizaciones por medio de definibilidad lógica. La adecuación de tal relación se basa en los siguientes hechos: Por un lado, existen ciertas lógicas que pueden servir para describir las computaciones de una clase de complejidad dada. Este hecho, a su vez, se basa en lo siguiente: en primer lugar, las estructuras finitas

se pueden utilizar para describir ejecuciones finitas de máquinas. Además, la relación de ordenamiento sobre una estructura se puede usar para describir la transición de una configuración a la siguiente mediante una lógica "simple" (la mayoría de veces, LPO). Más aún, el poder expresivo adicional (e.g., el operador OPFM) se puede utilizar para saber si la computación se detiene y para obtener su resultado. Por otro lado, la complejidad de la relación de satisfacción ($\mathcal{A} \models_{\mathfrak{L}} \phi$) de esas lógicas es exactamente la de las computaciones que describen. Este hecho, a su vez, se basa en lo siguiente: Primero, las estructuras finitas se pueden codificar como palabras y, por tanto, pueden ser objetos de computación; en particular, entradas. En segundo lugar, las fórmulas de un lenguaje lógico se pueden interpretar a menudo como programas que, dada una estructura como entrada, realizan la evaluación correspondiente.

Al reunir ambos hechos, se obtienen las caracterizaciones descriptivas de las clases de complejidad y, así, una nueva medida de complejidad: la complejidad de las descripciones lógicas. La interacción entre la teoría de la complejidad y la lógica es fructífera para ambos lados. Específicamente, las caracterizaciones de TCD son importantes en, al menos, los siguientes aspectos: Pueden ayudar a reconocer que un cierto problema se encuentra en una clase de complejidad dada mediante la identificación del lenguaje necesario para expresarlo. Además, permiten interpretar las lógicas involucradas como lenguajes de programación superiores para los problemas de la clase de complejidad correspondiente. (En el sentido de que permiten transformar una oración ϕ en un algoritmo que acepta la clase de modelos de ϕ y que satisface las restricciones de recursos requeridas). Más aún, también permiten que los rasgos característicos de la lógica puedan ser vistos como rasgos característicos de la clase de complejidad descrita por ella, y pueden contribuir a una mejor comprensión. Todavía

más, permiten transformar problemas, métodos y resultados de la teoría de la complejidad en correlatos correspondientes de la lógica y viceversa; ampliando así las posibilidades metodológicas para ambos lados. (Véase [13], Chs. 7 y 8).

El último aspecto es particularmente importante dentro del contexto específico de esta tesis: Por un lado, las caracterizaciones de TCD permiten equiparar los problemas de separación e inclusión sobre clases de complejidad con problemas de separación e inclusión sobre el poder expresivo entre las lógicas correspondientes (sobre estructuras ordenadas, finitas);[1] los segundos siendo de naturaleza puramente modelo-teórica y, por tanto, susceptibles de ser enfrentados con técnicas TEL. Esto significa que podemos enfrentar de manera segura problemas en la teoría de la complejidad enfrentando sus correlatos de TCD; y, lo que es más, dada la naturaleza de esos correlatos, también podemos aplicar de manera segura técnicas TEL al hacerlo (donde "de manera segura" significa sin distorsionar los fenómenos de complejidad). Por otro lado, TCD es consistente con los resultados (bien establecidos) sobre la separación e inclusión entre clases proporcionados por la teoría de la complejidad; la mayoría de los cuales, como las caracterizaciones estándar de tales clases, fueron obtenidos independientemente de TCD. Aún más, análogamente al caso de los problemas, las caracterizaciones de TCD permiten obtener correlatos lógicos de esos resultados; en los que la relación entre los poderes expresivos de lógicas es lo que justifica inclusiones y separaciones. Por tanto, en la medida en que dichos correlatos están justificados por la relación entre el poder expresivo de lógicas, las técnicas TEL que se pueden

[1] En algunos casos, la equivalencia entre un problema de separación o inclusión sobre clases de complejidad y el correspondiente problema de inclusión o separación sobre poderes expresivos de lógicas, exige solamente finitud de las estructuras; i.e., se sostiene incluso sin suposición alguna de orden. Por ejemplo, Abiteboul y Vianu mostraron ([3]) que los poderes expresivos de LPO(OPFI) (la lógica de primer-orden más un operador de punto fijo inflacionario) y de LPO (OPFP) son equivalentes sobre estructuras finitas (en ausencia de orden) syss las clases de complejidad P y PSPACE coinciden.

usar para abordarlos también son consistentes con ellos.

Correspondientemente, uno puede de manera segura, no sólo aplicar técnicas TEL para enfrentar problemas de inclusión y separación en la teoría de la complejidad, sino que también los resultados posiblemente obtenidos utilizando esas técnicas estarían justificados de manera adecuada. En particular, un enfoque TEL aplicado a LSO y LPO(OPFM) cuenta con este tipo de respaldo como una estrategia para argumentar a favor de P = NP. Para empezar, recordando el final del capítulo 2, la plausibilidad de un enfoque TEL como una estrategia para argumentar a favor de P = NP está garantizada: dado que el resultado de TCD según el cual PH = LSO, dentro del contexto de estructuras ordenadas, finitas, sólo exige definibilidad mediante fórmulas de LSO, éste se sostiene tomando LSO con semántica general. Siendo así, tomar LSO con tal semántica no distorsiona los fenómenos de complejidad correspondientes dentro del caso de las estructuras ordenadas, finitas.

Además de eso, como se señaló al final del capítulo 3, el hecho de que haya lógicas como LIA y LD, cuyo poder expresivo es igual al de LSO$\exists$, asegura que los rasgos específicos de los problemas NP no se pierden necesariamente cuando son transferidos a alguna versión de LPO. Es decir, los rasgos característicos de esos problemas se pueden expresar adecuadamente en versiones de LPO. Además, como también se mencionó en ese punto, en la medida en que existen lógicas como LPO(OPFP), las cuales, sobre estructuras ordenadas, finitas, son al menos tan expresivas como LSO, una situación análoga se cumple respecto a problemas cuya complejidad identificada es PH.

Más aún, el hecho de que la desigualdad de poderes expresivos entre LPO(OCTD) (la lógica de primer-orden más un operador de clausura transitiva determinista) y LPO(OPFP) está probada (por medios puramente modelo-teóricos) sobre estructuras ordenadas, finitas

(i.e., sobre tales estructuras, LPO(OCDT) $\neq$ LPO(OPFP). Véase [13], 155 y 272), señala que la traducción potencial de LSO a LPO(OPFM) sería consistente con el resultado bien establecido de la teoría de la complejidad según el cual LOGSPACE $\neq$ PSPACE (véase [13], 155). Por tanto, no colapsaría la jerarquía de complejidad completa.

El otro tipo de respaldo con el que cuenta el enfoque TEL es más bien filosófico y, por tanto, de la misma naturaleza del tipo de argumentos que la mayoría de los expertos que defienden P $\neq$ NP ha ofrecido (e.g., [4], [5], [6], [33] y [34]). En primer lugar, una identificación de LPO(OPFM) y LSO (i.e., de P y NP) mediante TEL no implica necesariamente la automatización de la creatividad (ya sea matemática o de otro tipo): Una prueba potencial de P = NP a través del TEL en cuestión casi seguramente sería no-constructiva. Es decir, aunque tal prueba aseguraría que hay algoritmos de tiempo polinomial para problemas NP-completos (i.e., oraciones de LPO(OPFM) que expresan adecuadamente tales problemas), es casi seguro que no los produciría. Además, incluso si esa prueba fuera constructiva, es plausible que estaría acompañada de algoritmos no eficientes en la práctica ya que los límites polinómicos correspondientes podrían ser altos o incluso desconocidos (i.e., oraciones de LPO(OPFM) cuya complejidad de satisfacibilidad es, aunque polinómicamente limitada, impráctica). En otras palabras, es plausible que tal prueba produciría algoritmos de tiempo polinómico que son completamente imprácticos, debido a un grado muy grande del polinomio o una constante multiplicativa muy grande; e.g., $(10n)^{1000}$ (véase [32]).[2] Por lo tanto, incluso si resulta ser el caso que LSO = LPO(OPFM), además de que tal resultado no implicaría necesariamente la automatización de la creatividad, y aunque abriría la puerta a las implicaciones revolucionarias evidentes mencionadas al inicio del

[2]De manera dual, no sólo en el caso de una prueba a través de TEL sino en general, P $\neq$ NP no significa necesariamente que los problemas NP-completos sean necesariamente intratables en la práctica. Después de todo, $n^{log\ log\ log\ n}$ no es un límite polinómico pero se comporta increíblemente bien (véase [32]).

capítulo 1, casi seguramente no las actualizaría. Eso significa que podríamos, de acuerdo a la situación, continuar confiando en nuestra criptografía, contratando matemáticos, científicos e ingenieros, y todavía habría problemas comunes que no seríamos capaces de resolver eficientemente (cf., [5] y [6], y [33] y [34]).

En segundo lugar, la plausibilidad del resultado de que P = NP (a través de un enfoque TEL o cualquier otro), no parece en peligro por el hecho de que, después de décadas de investigación, nadie ha sido capaz de encontrar un algoritmo eficiente para algún problema NP-completo. Como dice Vardi: El espacio de los algoritmos es muy grande y estamos sólo al comienzo de su exploración ([19], 74).[3] Por ejemplo, recordemos que el problema P $\stackrel{?}{=}$ NP ha estado abierto desde principios de la década de 1970; sin embargo, tomó 40 años probar que la Programación Lineal está en P. Además, como también señala Vardi ([32], hay fenómenos de complejidad como SAT, donde hay claras brechas entre la teoría y la práctica: Por un lado, SAT es el problema NP-completo canónico. Por otro lado, en la actualidad los solucionadores de SAT resuelven de forma rutinaria casos con más de un millón de variables y, no obstante, hay casos con algunos cientos de variables que no pueden ser resueltos por ningún solucionador de SAT existente. Considero que las indicaciones de Vardi significan, dentro del presente contexto, que, al menos en el estado actual de las teorías disponibles sobre los fenómenos de complejidad, debemos ser muy cuidadosos al apelar a hechos prácticos actuales para discutir cuestiones teóricas (cf., [5] y [6], y [33] y [34]).

Finalmente, de acuerdo con el respaldo filosófico ya señalado, el argumento, contra P = NP, de que P $\neq$ NP eventualmente podría alcanzar el mismo estatus que un principio

[3] The space of algorithms is very large and we are only at the beginning of its exploration.

(restrictivo) de la física, es más controversial que la misma plausibilidad de P = NP. Hay un viejo cliché que dice que los principios de la física y los principios de las matemáticas tienen un estatus diferente. Según este cliché, P ≠ NP sería diferente a cualquier otro principio físico que conocemos: a diferencia de los principios físicos que conocemos, P ≠ NP podría ser falsificado por descubrimientos puramente (lógico-)matemáticos tales como LPO(OPFM) = LSO. Si suponemos que P ≠ NP es un principio de la física, no está claro en qué medida éste depende de la realidad física y de las matemáticas.[4] El cliché es mucho más antiguo que el mismo problema P $\stackrel{?}{=}$ NP, y siendo lo que dice el cliché verdadero o no, en la medida en que éste aún prevalece, el argumento en cuestión debe confrontarlo seriamente (cf., [4]).

Por último, pero no por ello menos importante, el enfoque TEL también cuenta con respaldo que proviene del hecho de que nos permite superar (quizá no de la manera esperada) las tres barreras conocidas en la teoría de la complejidad: relativización, pruebas naturales y algebrización. Por tanto, en el resto de este capítulo, primero presento muy a grandes rasgos cómo estas barreras representan restricciones para cualquier enfoque al problema P $\stackrel{?}{=}$ NP, para luego mostrar cómo el enfoque TEL nos permite superarlas.

El primer resultado sobre una barrera fue publicado por Baker, Gill y Solovay ([8]) en 1975: relativización. Ellos mostraron que las técnicas de la lógica y la teoría de la computabilidad, tales como diagonalización y simulación, no pueden ser lo suficientemente poderosas como para resolver el problema P $\stackrel{?}{=}$ NP. Específicamente, mostraron que esas técnicas funcionarían igual de bien en un "mundo relativizado", donde tanto las máquinas P como NP podrían computar alguna función f en un solo paso; sin embargo, hay algunos

[4] Aaronson ([4], 48) mismo señala esta cuestión, pero la descarta inmediatamente mediante un bosquejo de argumento no convincente.

mundos relativizados donde P = NP y otros donde P ≠ NP. En otras palabras, mostraron que hay oráculos A y B tales que $\mathrm{P}^A = \mathrm{NP}^A$ y $\mathrm{P}^B \neq \mathrm{NP}^B$.[5] Por tanto, si una prueba candidata de P = NP sigue siendo válida cuando las máquinas correspondientes tienen acceso a un oráculo, entonces en particular sigue siendo válida cuando tales máquinas tienen acceso a B. Entonces la misma prueba de P = NP también da $\mathrm{P}^B = \mathrm{NP}^B$, pero sabemos que esto es falso. (El mismo razonamiento se sostiene con una prueba candidata de P ≠ NP). Siendo así, las técnicas en una prueba candidata pretendiendo resolver el problema $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ deben verse afectadas por la introducción de oráculos en el prototipo de máquina. Sin embargo, en ese momento, la mayoría de las técnicas de la lógica y la teoría de la computabilidad, si no todas, eran insensibles a dicha introducción. Por lo tanto, según Baker, Gill y Solovay, cualquier solución potencial del problema $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ requerirá técnicas no-relativizantes; así, el lema advino como: necesitamos técnicas más poderosas que nos permitan analizar computaciones en lugar de simplemente simularlas.

El segundo resultado constituye una barrera para probar límites inferiores de circuitos: pruebas naturales. Con el objetivo de analizar computaciones más de cerca y, de ese modo, evitar la relativización, el prototipo de circuito fue introducido. Sin embargo, después de muchos años de investigación, no se encontraron límites inferiores significativos para los lenguajes en las clases tales como NP, y los pocos encontrados fueron para prototipos restringidos de circuitos. Como una explicación potencial de por qué se había hecho tan poco progreso, Razborov y Rudich ([26]) mostraron en 1997 que si estas nuevas técnicas funcionaran para probar separaciones como P ≠ NP, podríamos darles la vuelta para obtener formas más rápidas de distinguir funciones aleatorias de funciones pseudoaleatorias.

[5] Un oráculo A es una colección de funciones booleanas: $\{A_n : \{0,1\}^n \longrightarrow \{0,1\} | n \in \mathbb{N}\}$. Para una clase de complejidad $\mathcal{C}$, denotamos mediante $\mathcal{C}^A$ la clase de todos los lenguajes decidibles por una máquina $\mathcal{C}$ que puede consultar A_n para cualquier n.

Sin embargo, en ese caso, estaríamos obteniendo paralelamente algoritmos rápidos para algunos de los problemas que queríamos probar que fueran difíciles (e.g., invertir funciones unidireccionales). Siendo así, ellos mostraron que el enfoque en cuestión también tiene su propia barrera intrínseca.

El tercer resultado fue publicado por Aaronson y Wigderson ([7]) en 2009. Esta barrera fue motivada principalmente por algunos resultados con respecto a los protocolos de prueba interactiva que son no-relativizantes (e.g., el resultado IP = PSPACE proporcionado por Shamir ([27]) en 1992). Estos resultados fueron basados en la idea de tratar una fórmula booleana como una expresión aritmética y afirmaciones probadas que se sabe que tienen una relativización contraria precisamente como P $\stackrel{?}{=}$ NP. Además, se descubrieron límites inferiores de circuitos que evitan tanto las barreras de relativización como de pruebas naturales simultáneamente (e.g., que PP no tiene circuitos de tamaño lineal). Así, ambos tipos de resultados dieron lugar a la cuestión de si las ideas sobre las que se establecieron serían suficientes para resolver P $\stackrel{?}{=}$ NP. Aaronson y Wingderson mostraron que existe una tercera barrera para resolver dicho problema y los otros problemas centrales de la teoría de la complejidad: algebrización. Ellos descubrieron esta barrera mientras exploraban el alcance de la idea de tratar una fórmula booleana como una expresión aritmética. Tal idea fue que, en lugar de tratar una fórmula booleana ϕ como sólo una caja negra que mapea las entradas a las salidas, podemos aprovechar la estructura de ϕ "ascendiendo" sus puertas Y, O, o NO a las operaciones aritméticas sobre algún campo o anillo más grande $\mathbb{F}$. Así, podemos extender ϕ a un polinomio de grado bajo $\tilde{\phi} : \mathbb{F}^n \longrightarrow \mathbb{F}$, el cual tiene propiedades de corrección de errores que el caso booleano no proporciona. Para modelar esa idea, Aaronson y Wigderson consideraron lo que llamaron "oráculos algebraicos": oráculos capaces de

evaluar no sólo una función booleana f, sino también una extensión de grado bajo $\tilde{f}$ de f sobre un campo finito o los enteros. De esta manera, ellos definieron algebrización: decimos que una inclusión entre clases de complejidad $\mathcal{C} \subseteq \mathcal{D}$ *algebriza* si $\mathcal{C}^A \subseteq \mathcal{D}^{\tilde{A}}$ para todos los oráculos A y todas las extensiones de grado bajo $\tilde{A}$ de A. Del mismo modo, una separación $\mathcal{C} \not\subset \mathcal{D}$ algebriza si $\mathcal{C}^A \not\subset \mathcal{D}^{\tilde{A}}$ para todo A, $\tilde{A}$ ([7], 3).[6]

Como ya se mencionó al presentar el resplado filosófico, actualmente estamos conscientes del hecho de que es muy probable que una prueba potencial de P = NP (en particular, una obtenida a través de un enfoque TEL) sería no-constructiva y, por tanto, tendría poco (si algo) que ver con los detalles internos de computaciones. Por consiguiente, por un lado, la naturaleza altamente abstracta de las herramientas modelo-teóricas del enfoque TEL lo hacen susceptible a las tres barreras. En otras palabras, el enfoque TEL relativiza, algebraiza y ofrece pruebas naturales; i.e., produciría traducciones bajo las cuales P = NP y otras bajo las cuales P $\neq$ NP. Sin embargo, por otra parte, esa misma naturaleza del enfoque TEL también nos permite argumentar, precisamente desde el nivel abstracto correspondiente y a diferencia de otros enfoques también susceptibles a las barreras, que tenemos fuertes razones para preferir las traducciones bajo las cuales P = NP sobre las otras. Estas razones provienen del respaldo matemático-conceptual que podemos proporcionar para las primeras pero no para las segundas: Aunque el respaldo filosófico presentado arriba también se puede proporcionar para las traducciones bajo las que P $\neq$ NP (véase la nota 2 del presente capítulo y, en general, [32]), el respaldo matemático-conceptual también presentado arriba sólo se puede proporcionar para las traducciones bajo las cuales P = NP. Esto,

[6] [W]e say that a complexity class inclusion $\mathcal{C} \subseteq \mathcal{D}$ *algebrizes* if $\mathcal{C}^A \subseteq \mathcal{D}^{\tilde{A}}$ for all oracles A and all low-degree extensions $\tilde{A}$ of A. Likewise, a separation $\mathcal{C} \not\subset \mathcal{D}$ algebrizes if $\mathcal{C}^A \not\subset \mathcal{D}^{\tilde{A}}$ for all A, $\tilde{A}$.

La idea es que, al relativizar cierta inclusión o separación entre clases de complejidad, la máquina correspondiente debería tener acceso no sólo a un oráculo A, sino también a una extensión de grado bajo de A sobre un campo o anillo finito.

a su vez, es el caso porque el resultado de que PH = LSO (así como el resultado de que NP = LSO$\exists$), dentro del contexto de estructuras ordenadas, finitas, exige definibilidad por LSO-(LSO$\exists$-)fórmulas; es decir, exige semántica general en lugar de semántica estándar. Por lo tanto, no podemos proporcionar respaldo matemático-conceptual (al menos en la forma presentada arriba) para las traducciones bajo las cuales P $\neq$ NP, que toman SOL con semántica estándar.

Capítulo 5

Conclusiones

Después de un breve resumen de los principios y métodos de TCD, los cuales producen una caracterización puramente lógica del problema P $\stackrel{?}{=}$ NP, introduje un enfoque TEL aplicado a LPO(OPFM) y LSO como una estrategia para argumentar a favor de P = NP. Las conclusiones obtenidas a través del análisis de dicho enfoque son las siguientes.

Tomar un enfoque TEL aplicado a LPO(OPFM) y LSO como una estrategia para argumentar a favor de P = NP es plausible: sobre estructuras ordenadas, finitas, PH = LSO se sostiene cuando LSO se toma con semántica general; por lo que la traducción relevante podría ser realizada. Además, el enfoque en cuestión está bien motivado como tal estrategia. Esta motivación no sólo proviene del desarrollo meticuloso de técnicas TEL, sino también de al menos otras dos fuentes: 1. Hay nociones que parecen intrínsecamente de LSO y, sin embargo, se pueden expresar en algunas versiones de LPO. 2. Existen lógicas de primer-orden equipadas con ciertos operadores que, sobre estructuras ordenadas, finitas, son al menos tan expresivas como LSO. Más aún, el enfoque TEL cuenta con respaldos matemático-conceptual y filosófico. Por último, pero no menos importante, el enfoque nos

permite superar las tres barreras conocidas en la teoría de la complejidad: aunque produce traducciones bajo las que $P = NP$ y otras bajo las que $P \neq NP$, tenemos fuertes razones para preferir a las primeras sobre las segundas; razones que provienen del respaldo matemático-conceptual que podemos dar para algunas traducciones pero no para otras.

Finalmente, debo enfatizar que mi objetivo aquí no era solamente señalar que la evidencia a favor de $P \neq NP$ no es tan fuerte como se cree y que no es una locura argumentar a favor de $P = NP$, sino empezar a poner a prueba lo que considero que podría ser un enfoque novedoso para el problema $P \stackrel{?}{=} NP$.

Bibliografía

[1] Abiteboul, S., Vardi, M. Y., y Vianu, V. (1992). Fixpoint logics, relational machines, and computational complexity. *Journal of the ACM (JACM)*, 44(1), 30-56.

[2] Abiteboul, S. y Vianu, V. (1991). Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43(1), 62-124.

[3] Abiteboul, S. y Vianu, V. (1995). Computing with first-order logic. *Journal of Computer and Systems Sciences*, 50(2), 309-335.

[4] Aaronson, S. (2005). Guest column: NP-complete problems and physical reality. *ACM SIGACT News*, 36(1), 30-52.

[5] Aaronson, S. (2013). P, NP, and friends. En *Quantum Computing since Democritus*. Cambridge University Press, 54-70.

[6] Aaronson, S. (2016). P $\stackrel{?}{=}$ NP. En J. F. Nash Jr. and M. Th. Rassias (eds.). *Open Problems in Mathematics*. Springer, 1-121.

[7] Aaronson, S. y Wigderson, A. (2009). Algebrization: A New Barrier in Complexity Theory. *ACM Transactions on Computation Theory (TOCT)*, 1(1), 1-54.

[8] Baker, T. Gill, J. y Solovay, R. (1975). Relativizations of the P =? NP question. *SIAM Journal on computing*, 4(4), 431-442.

[9] Carnielli, W. A., M. E. Coniglio and I. M. L. D'Ottaviano (2009). New Dimensions on Translations Between Logics. *Logica Universalis*, 3(1), 1–18.

[10] Chailos, G. (2016). The notion of Infinity within the Zermelo system and its relation to the Axiom of Countable Choice. *Theoretical Mathematics and Applications*, 6 (1), 39-66.

[11] Cook, R. y Shapiro, S. (1998). Hintikka's Revolution: The Principles of Mathematics Revisited. *British Journal of Philosophy of Science*, 49, 309-316.

[12] Dawar, A. y Guverich, Y. (2002). Fixed Point Logics. *The Bulletin of Symbolic Logic*, 8(1), 65-88.

[13] Ebbinghaus, H.-D. y Flum, J. (1999). *Finite Model Theory*. Springer, 2a. edición.

[14] Enderton, H. B. (2007). Second-order and Higher-order Logic. En Edward N. Zalta (ed.). *The Stanford Encyclopedia of Philosophy* (Edición de Otoño del 2015). URL=<https://plato.stanford.edu/archives/fall2015/entries/logic-higher-order/>.

[15] Fagin, R. (1973). Contributions to the Model Theory of Finite Structures. Tesis doctoral, U. C. Berkeley.

[16] Fagin, R. (1974). Generalized First-Order Spectra and Polynomial-Time Recognizable Sets. En R Karp (ed.). *Complexity of Computation. SIAM-AMS Proccedings*, 7, 43-73.

[17] Fortnow, L. (2009). The Status of the P versus NP Problem. *Communications of the ACM*, 52(9), 78-86.

[18] Fortnow, L. (2013). *The Golden Ticket: P, NP and the search of the impossible*. Princenton University Press.

[19] Gasarch, W. I. (2012). Guest column: The second P=? NP poll. *ACM SIGACT News*, 43(2), 53-77.

[20] Hintikka, K., J. (1955). Reductions in the Theory of Types. En *Two Papers on Symbolic Logic*, Acta Philosophica Fennica, 8. Helsinki, 57–115.

[21] Hintikka, K., J. (1993). New Foundations for Mathematical Theories. En J. Hintikka, 1998, *Language, Truth and Logic in Mathematics* (Jaakko Hintikka: Selected Papers, Volume 3). Kluwer, 225–247.

[22] Hintikka, K., J. (1996). *The Principles of Mathematics Revisited*. Cambridge University Press.

[23] Immerman, N. (1982). Upper and lower bounds for first-order expressibility. *Journal of Computer and System Sciences*, 25(1), 76-98.

[24] Immerman, N. (1999). *Descriptive Complexity*. Springer.

[25] Manzano, M. (1996). *Extensions of First Order Logic*. Cambridge University Press.

[26] Razborov, A. A. y Rudich, S. (1997). Natural Proofs. *Journal of Computer and System Sciences*, 55(1), 24-35.

[27] Shamir, A. (1992). IP = PSPACE. *Journal of the ACM (JACM)*, 39(4), 869-877.

[28] Stockmeyer, L. (1977). The Polynomial-Time Hierarchy. *Theoretical Computer Science*, 3(1), 1-22.

[29] Tulenheimo, T. (2009). Independence Friendly Logic. En Edward N. Zalta (ed.). *The Stanford Encyclopedia of Philosophy* (Edición de Primavera del 2017). URL=<https://plato.stanford.edu/archives/spr2017/entries/logic-if/>.

[30] Väänänen, J. (2007). *Dependence Logic: A New Approach to Independence Friendly Logic*. (London Mathematical Society student texts, 70). Cambridge University Press.

[31] Vardi, M. Y. (1982). The complexity of relational query languages. En *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, 137-146.

[32] Vardi, M. Y. (2010). On P, NP, and Computational Complexity. *Communications of the ACM*, 53(11), 5-5.

[33] Wigderson, A. (2006). P, NP and mathematics-a computational complexity perspective. En *Proceedings of the International Congress of Mathematicians (Vol. 3)*. EMS Publishing House, 665-713.

[34] Wigderson, A. (2009). Knowledge, Creativity and P versus NP. (A very informal draft), 1-25.