ISSN 1870-4069

$\mathsf{TFL}^{\mathsf{PL}}$: Programación con lógica de términos

J. Martín Castro-Manzano, L. Ignacio Lozano-Cobos

Facultad de Filosofía y Humanidades, UPAEP, Puebla, México

josemartin.castro@upaep.mx

Resumen. A partir del sistema lógico *Term Functor Logic* (TFL) y del concepto de base de datos aristotélica, en esta contribución presentamos los avances de un lenguaje de programación lógica que hemos denominado *Term Functor Logic Programming Language* ($\mathsf{TFL}^{\mathsf{PL}}$).

Palabras clave: Lógica de términos, silogística, base de datos aristotélica.

$\mathsf{TFL}^{\mathsf{PL}}$: Programming with Term Logic

Abstract. Given the system *Term Functor Logic* (TFL) and the concept of Aristotelian database, in this paper we present the development of a programming language we call *Term Functor Logic Programming Language* ($\mathsf{TFL}^{\mathsf{PL}}$).

Keywords: Term logic, syllogistic, Aristotelian database.

1. Introducción

Bajo la influencia directa de [26,28,29,32,16], en otro lugar hemos ofrecido una versión de una silogística (relacional con cuantificadores no-clásicos) que trata con una amplia gama de patrones inferenciales de sentido común con las ventajas de un enfoque algebraico (el sistema TFL^{+}); y bajo la influencia directa de [5,8,6,7] y del mismo sistema TFL^{+}, en otro lugar hemos presentado un sistema de diagramas lineales para razonar visualmente (el sistema $\mathsf{TFL}^{\oplus}$) (cf. [19]). Ahora, bajo la influencia de estos dos últimos sistemas y la noción de base de datos aristotélica [17], en este trabajo presentamos los avances de un lenguaje de programación lógica que hemos denominado *Term Functor Logic Programming Language* ($\mathsf{TFL}^{\mathsf{PL}}$).

Así pues, en esta contribución tratamos de alcanzar dos metas: *i)* introducir los elementos básicos de un sistema lógico de términos capaz de lidiar con una amplia gama de patrones inferenciales de razonamiento ordinario (TFL^{+}) y *ii)*

presentar los avances de un lenguaje de programación diseñado a partir de tal sistema ($\mathsf{TFL}^{\mathsf{PL}}$). Para alcanzar estos resultados presentamos el sistema de términos TFL [26,28,29,5,6,7] a la luz de lo que hemos llamado "la visión heredada de la lógica" (§2) y el concepto de base de datos aristotélica [17] (§3): esto permitirá apreciar la relevancia del lenguaje que proponemos; posteriormente exponemos los avances del lenguaje $\mathsf{TFL}^{\mathsf{PL}}$ mediante una presentación de su sintaxis y su funcionamiento global (§4); al final, cerramos con algunas observaciones sobre trabajo futuro y problemas por resolver (§5).

2. La visión heredada de la lógica

La lógica estudia la relación de inferencia y para llevar a cabo tal estudio es costumbre hacer uso de lenguajes de primer orden. Así, por ejemplo, la lógica proposicional, la lógica de primer orden y la lógica de primer orden con identidad son sistemas lógicos definidos mediante lenguajes de primer orden:

$\{p, q, r, \ldots, \neg, \Rightarrow\}$, $\{a, b, c, \ldots, x, y, z, \ldots, f, g, h, \ldots, A, B, C, \ldots, \neg, \Rightarrow, \forall, \exists\}$
$\{a, b, c, \ldots, x, y, z, \ldots, f, g, h, \ldots, A, B, C, \ldots, \neg, \Rightarrow, \forall, \exists, =\}$, respectivamente.

El origen de esta costumbre está relacionado con las ventajas de orden representativo que los lenguajes de primer orden ofrecen frente a sistemas más tradicionales. Russell, por ejemplo, popularizó la idea de que las limitaciones del programa lógico tradicional, i.e., silogístico (*vide* Apéndice A), se debían al análisis de las proposiciones en clave terminista, como triadas de términos sujeto y predicado unidos por una cópula [24]. Carnap generalizó esta consideración a toda la lógica tradicional al sostener que la única sintaxis disponible en este tipo de lógica es predicativa [4]. Ciertamente, las limitaciones de la sintaxis de términos ternaria (sujeto-cópula-predicado) generan dificultades para representar proposiciones singulares, relacionales y proposicionales, y si estos impedimentos parecen menores, consideremos un problema con consecuencias graves: la homogeneidad de términos. Geach argumenta:

> *Our distinction between names and predicables enables us to clear up the confusion, going right back to Aristotle, as to whether there are genuine negative terms: predicables come in contradictory pairs, but names do not, and if names and predicables are both called "terms" there will be a natural hesitation over the question "Are there negative terms?" [10, p. 64]*

De acuerdo a este argumento, la homogeneidad de términos no permite preservar la distinción fundamental entre nombre y predicado. Esta incapacidad de la sintaxis de términos resulta problemática porque las funciones de un nombre y las propiedades de un predicado no son intercambiables: mientras la función de un nombre es nombrar, la de un predicado es predicar; pero predicar no es nombrar y nombrar no es predicar. Por tanto, como argumenta Geach, es lógicamente imposible un intercambio sintáctico entre los términos sujeto y

predicado sin un cambio en el sentido de las proposiciones, pues sólo un nombre puede ser un sujeto lógico, pero un nombre no puede mantener su rol de nombre si se convierte en predicado. Por ello, esta dificultad sintáctica es también una dificultad semántica que se antoja insalvable: entre una lógica de términos y la lógica genuina, dice Geach, sólo puede haber guerra [9, p. 54].

En contraste, la lógica genuina, caracterizada por la lógica de primer orden (LPO), sigue la sintaxis del paradigma Fregeano que resulta de abandonar el uso de términos para favorecer una gramática binaria de pares función-argumento. Estos pares promueven una sintaxis que incluye constantes $(a, b, c, \ldots)$ o variables $(x, y, z, \ldots)$ como argumentos para referir a objetos individuales como sujetos lógicos, además de relaciones $(A, B, C, \ldots)$ como funciones para referir a conceptos, no a objetos, como predicados lógicos. Así, por ejemplo, una proposición como "Sócrates es mortal" no podría ser entendida como una relación entre términos sujeto y predicado, sino como un par función-argumento donde una constante, un elemento saturado y completo, digamos s, denota a un objeto de nombre "Sócrates" que funge como argumento de la función incompleta y no-saturada "... *es mortal*", digamos Mx, de tal modo que Ms representa la proposición *Sócrates es mortal*: claramente, esta representación sintáctica no permite el intercambio de términos. Más todavía, dada esta sintaxis binaria, una proposición como "Todos los hombres son mortales" no puede ser entendida como una relación de términos sino como una relación entre variables y cuantificadores, digamos $\forall x(Hx \Rightarrow Mx)$, de tal modo que existe una diferencia sintáctica clara entre una proposición singular (como "Sócrates es mortal") y una proposición universal (como "Todos los hombres son mortales"): esto será importante más adelante.

Así pues, de acuerdo a estas consideraciones, la lógica genuina sigue la sintaxis del paradigma Fregeano que resulta de abandonar el uso de una sintaxis ternaria (sujeto-cópula-predicado) para favorecer una sintaxis binaria (función-argumento). Esta sintaxis binaria promueve lenguajes de primer orden. Esta elección sintáctica es la que nos es familiar actualmente porque es la que nos acompaña en la docencia, la investigación y la aplicación de la lógica contemporánea: esta es la visión heredada de la lógica. Sin embargo, no hace falta mucho ojo crítico para notar que esta visión nos puede ser familiar, pero no por ello nos resulta natural. Woods comenta (el énfasis es nuestro):

> *It is no secret that classical logic and its mainstream variants aren't much good for human inference as it actually plays out in the conditions of real life*—in life on the ground, so to speak. It isn't surprising. *Human reasoning is not what the modern orthodox logics were meant for.* The logics of Frege and Whitehead & Russell were purpose-built for the pacification of philosophical perturbation in the foundations of mathematics, notably but not limited to the troubles occasioned by the paradox of sets in their application to transfinite arithmetic. [35, p. 404].

Así pues, si bien la lógica genuina (clásica, según Woods) ha sido fundamental para el estudio de la inferencia en ciencias cognitivas e inteligencia artificial, no

deja de extrañarnos que, a pesar de su finalidad original, sea constantemente utilizada para modelar razonamiento en lenguaje natural. Consideremos, a este efecto, lo que hemos denominado "el reto de Bar-Hillel":

> *I challenge anybody here to show me a serious piece of argumentation in natural languages that has been successfully evaluated as to its validity with the help of formal logic. I regard this fact as one of the greatest scandals of human existence. Why has this happened? How did it come to be that logic which, at least in the views of some people 2,300 years ago, was supposed to deal with evaluation of argumentation in natural languages, has done a lot of extremely interesting and important things, but not this? [30, p. 256]*

Esto, ciertamente, es escandaloso. Sin embargo, desde finales de la década de los 60's Fred Sommers defendió una revisión y una revitalización de la sintaxis ternaria tradicional a la luz del reto de Bar-Hillel. Sommers, cercano a un proyecto de naturalización de la lógica, estaba preocupado por cómo es que razonamos. Así, por ejemplo, Sommers se preguntaba por las razones por las cuales un agente racional se da cuenta de que el siguiente par de creencias es inconsistente:

(C_1) Todo perro es animal pero (C_2) alguien que quiere a un perro no quiere a un animal.

Por supuesto que LPO es capaz de ofrecer una respuesta al problema anterior haciendo uso de lenguajes de primer orden (sea C_1, $\forall x(Px \Rightarrow Ax)$; y C_2, $\exists x \exists y((Py \wedge Qxy) \wedge \forall z(Az \Rightarrow \neg Qxz))$) y métodos de demostración adecuados (Ecuación 1), pero como veremos más adelante, esta capacidad no necesariamente es relevante para comprender el razonamiento en lenguaje natural.

1	$\forall x(Px \Rightarrow Ax)$	C_1
2	$\exists x \exists y((Py \wedge Qxy) \wedge \forall z(Az \Rightarrow \neg Qxz))$	C_2
3	$\quad \exists y(Py \wedge Qay) \wedge \forall z(Az \Rightarrow \neg Qaz)$	**E∃ 2, a/x**
4	$\quad\quad (Pb \wedge Qab) \wedge \forall z(Az \Rightarrow \neg Qaz)$	**E∃ 3, b/y**
5	$\quad\quad Pb \wedge Qab$	**E∧ 4**
6	$\quad\quad \forall z(Az \Rightarrow \neg Qaz)$	**E∧ 4**
7	$\quad\quad Ab \Rightarrow \neg Qab$	**E∀ 6, b/z**
8	$\quad\quad Pb \Rightarrow \neg Ab$	**E∀ 1, b/x**
9	$\quad\quad Pb$	**E∧ 5**
10	$\quad\quad Qab$	**E∧ 5**
11	$\quad\quad Ab$	**E⇒ 8 y 9**
12	$\quad\quad \neg Qab$	**E⇒ 7 y 11**
13	$\quad\quad Qab \wedge \neg Qab$	**I∧ 10 y 12**
14	$\quad X$	**EFSQ 4 a 13**
15	X	**EFSQ 3 a 14**

(1)

2.1. El sistema TFL

En este contexto, Sommers [26,28,29] y Englebretsen [5,6,7] desarrollaron una lógica más cercana a nuestro lenguaje natural. El resultado fue el sistema algebraico *Term Functor Logic* (TFL), el cual asume una sintaxis ternaria[1] y ofrece la siguiente gramática para representar proposiciones categóricas:[2]

- $\mathsf{SaP} := -\mathsf{S} + \mathsf{P} = -\mathsf{S} - (-\mathsf{P}) = -(-\mathsf{P}) - \mathsf{S} = -(-\mathsf{P}) - (+\mathsf{S})$
- $\mathsf{SeP} := -\mathsf{S} - \mathsf{P} = -\mathsf{S} - (+\mathsf{P}) = -\mathsf{P} - \mathsf{S} = -\mathsf{P} - (+\mathsf{S})$
- $\mathsf{SiP} := +\mathsf{S} + \mathsf{P} = +\mathsf{S} - (-\mathsf{P}) = +\mathsf{P} + \mathsf{S} = +\mathsf{P} - (-\mathsf{S})$
- $\mathsf{SoP} := +\mathsf{S} - \mathsf{P} = +\mathsf{S} - (+\mathsf{P}) = +(-\mathsf{P}) + \mathsf{S} = +(-\mathsf{P}) - (-\mathsf{S})$

Dada esta representación, TFL ofrece un método de decisión correcto, completo y simple para la silogística: una conclusión se sigue válidamente de un conjunto de premisas syss *i)* la suma de las premisas es algebraicamente igual a la conclusión y *ii)* el número de conclusiones con cantidad particular (*viz.*, cero o uno) es igual al número de premisas con cantidad particular [6, p.167]. Así, por ejemplo, si consideramos un silogismo válido, digamos un silogismo tipo aaa-1, podemos ver cómo la aplicación de este método produce la conclusión correcta (Tabla 1).

Tabla 1. Un razonamiento válido: aaa-1.

Proposición	TFL
1. Todo mamífero es animal.	$-\mathsf{M} + \mathsf{A}$
2. Todo perro es mamífero.	$-\mathsf{P} + \mathsf{M}$
3. Todo perro es animal.	$-\mathsf{P} + \mathsf{A}$

En el ejemplo de arriba podemos apreciar claramente cómo funciona el método: *i)* si sumamos las premisas obtenemos la expresión algebraica $(-\mathsf{M} + \mathsf{A}) + (-\mathsf{P} + \mathsf{M}) = -\mathsf{M} + \mathsf{A} - \mathsf{P} + \mathsf{M} = -\mathsf{P} + \mathsf{A}$, de tal suerte que la suma de las premisas es igual a la conclusión y la conclusión es $-\mathsf{P} + \mathsf{A}$, en lugar de $+\mathsf{A} - \mathsf{P}$, porque *ii)* el número de conclusiones con cantidad particular (cero en este caso) es igual al número de premisas con cantidad particular (cero en este caso).

Pero además, como Leibniz habría deseado [14], este sistema algebraico no sólo es capaz de modelar inferencias silogísticas, sino que es capaz de representar,

[1] Que podemos razonar sin elementos lingüísticos de primer orden—como variables individuales o cuantificadores—no es una idea novedosa (cf. [23,20,13]), pero el proyecto lógico de Sommers tiene un alcance más amplio: que sea posible usar una lógica de términos en lugar de un sistema de primer orden no tiene nada que ver con el hecho sintáctico, por decirlo de algún modo, de que podemos hacer inferencia sin cuantificadores o variables, sino con la visión más general de que el lenguaje natural es una fuente genuina de una lógica natural (cf. [27,15]).

[2] Aquí seguimos la presentación de [6].

en respuesta a las críticas mencionadas en la sección anterior, inferencias con proposiciones relacionales (Tabla 2), singulares[3] (Tabla 3) y proposicionales[4] (Tabla 4) con facilidad y claridad preservando la idea nuclear de TFL, a saber, que la inferencia es un proceso lógico que ocurre usando términos.

Tabla 2. Argumento relacional.

Proposición	TFL
1. Algunos caballos son más rápidos que algunos perros.	$+\mathsf{C}+(+\mathsf{R}+\mathsf{P})$
2. Los perros son más rápidos que algunos hombres.	$-\mathsf{P}+(+\mathsf{R}+\mathsf{H})$
3. La relación *más rápido que* es transitiva.	$-(+\mathsf{R}+(+\mathsf{R}+\mathsf{H}))+(+\mathsf{R}+\mathsf{H})$
$\vdash$ Algunos caballos son más rápidos que algunos hombres.	$+\mathsf{C}+(+\mathsf{R}+\mathsf{H})$

Tabla 3. Argumento con singulares

Proposición	TFL
1. Todo hombre es mortal.	$-\mathsf{H}+\mathsf{M}$
2. Sócrates es hombre.	$-\mathsf{s}+\mathsf{H}$
$\vdash$ Sócrates es mortal.	$-\mathsf{s}+\mathsf{M}$

Tabla 4. Argumento proposicional

Proposición	TFL
1. Si P entonces Q.	$-[\mathsf{p}]+[\mathsf{q}]$
2. P.	$[\mathsf{p}]$
$\vdash$ Q.	$[\mathsf{q}]$

Estos elementos básicos de TFL son suficientes para ofrecer una solución a los problemas de representación de las lógicas de términos que mencionábamos renglones arriba. Pero más todavía, estos elementos ofrecen razones para explicar, por ejemplo, por qué juzgamos instantáneamente que (C_1) *Todo perro es animal* y (C_2) *Alguien que quiere a un perro no quiere a un animal* son mutuamente inconsistentes. Consideremos, a este efecto, la derivación de la contradicción entre C_1 y C_2 en TFL, con fines comparativos (Tabla 5).

Tabla 5. Derivación de la contradicción entre C_1 y C_2 en TFL.

Proposición	TFL
C_1. Todo perro es animal.	$-\mathsf{P}_2+\mathsf{A}_2$
C_2. Alguien que quiere a un perro no quiere a un animal.	$+(+\mathsf{Q}_{12}+\mathsf{P}_2)-(+\mathsf{Q}_{12}+\mathsf{A}_2)$
$\vdash$ Alguien que quiere a un animal no quiere a un animal.	$+(+\mathsf{Q}_{12}+\mathsf{A}_2)-(+\mathsf{Q}_{12}+\mathsf{A}_2)$

[3] Provisto que los términos singulares como *Sócrates* se represetan con minúsculas.

[4] Dado que el razonamiento proposicional se puede representar de la siguiente manera, $P := [\mathsf{p}]$, $Q := [\mathsf{q}]$, $\neg P := -[\mathsf{p}]$, $P \Rightarrow Q := -[\mathsf{p}]+[\mathsf{q}]$, $P \wedge Q := +[\mathsf{p}]+[\mathsf{q}]$ y $P \vee Q := --[\mathsf{p}]--[\mathsf{q}]$, el método de decisión se comporta como resolución (cf. [21]).

Si comparamos la prueba de la contradicción de C_1 y C_2 en LPO (Ecuación 1) con la prueba de la contradicción en TFL (Tabla 5), podemos observar que en LPO no sólo necesitamos una traducción más compleja haciendo uso de variables y cuantificadores, sino también una demostración más larga para justificar que existe una contradicción; y sin embargo, a diferencia de TFL, tal demostración no ofrece luces sobre por qué *vemos* instantáneamente una inconsistencia entre C_1 y C_2.

En contraste, la explicación en TFL tiene una naturalidad sintáctica y un conjunto de reglas simples para razonar. Y como LPO, por su origen, no posee estas características, no puede ofrecer información cognitivamente relevante sobre el modo en como razonamos. Y por ello, aunque LPO tiene mérito en la fundamentación de las matemáticas, no puede ser una lógica del razonamiento en lenguaje natural. Y como puede sospecharse en este punto, esta narrativa tiene ciertos nexos con la historia de los lenguajes de programación lógica [2].

3. Bases de datos aristotélicas

En efecto, de acuerdo con Mozes [17], el abandono de la lógica aristotélica, por parte de los computólogos, para favorecer LPO no está justificado (Cf. [12]). La lógica silogística es una lógica de términos que fue creada con el fin de comprender y guiar el razonamiento humano; pero como hemos comentado renglones arriba, esta no es la intención de LPO, y como resultado de esto, una lógica de términos no sólo tiene importancia cognitiva, sino también computacional.

En consecuencia, Mozes desarrolló el concepto de *base de datos aristotélica*. Una base de datos es aristotélica cuando posee las siguientes características:

- La habilidad de proveer explicaciones sobre las deducciones utilizadas en lenguaje natural.
- La habilidad de ofrecer información, en respuesta a preguntas dicotómicas ("sí/no"), si una versión más fuerte o débil de un "sí" puede ser probada.
- La habilidad de señalar resultados que no pueden ser probados pero que son posibles.
- La habilidad de sugerir reglas implícitas que si se añaden a la base de datos, podrían proveer respuestas afirmativas.
- La habilidad de indicar instancias en las cuales patrones no-deductivos, como analogías, pueden ser útiles.

Para obtener estas características, la estructura de una base de datos *à la* Mozes consiste de un conjunto de constantes para representar objetos y un conjunto de relaciones (no hay funciones) para representar propiedades de los objetos (volveremos a este punto en la siguiente sección pues, como puede notarse, Mozes asume una sintaxis Fregeana). La información sobre los objetos se expresa mediante hechos que consisten de relaciones aplicadas a objetos, por ejemplo, `hombre`(`Socrates`). En este sentido, estas bases siguen una sintaxis similar a la de Prolog [31,3]. Una regla, por otro lado, consiste de un sujeto, que es la conjunción de una o más relaciones aplicadas a variables y constantes, y un

predicado, que es una relación única; más un tipo de regla que indica la conexión entre el sujeto y el predicado. Cuando se escribe una regla se escribe primero el predicado, luego el tipo de la regla y finalmente el sujeto. Hay cuatro posibles tipos de reglas que corresponden a las cuatro proposiciones categóricas (*vide* Apéndice A). Así, por ejemplo, `mortal(X) A hombre(X)` significa *Todo hombre es mortal* (en Prolog, `mortal(X) :- hombre(X).`).

Como en Prolog, en estas bases de datos existen dos tipos de consultas: consultas de respuesta, que especifican un hecho o una regla y tratan de probarlas; y consultas de recuperación, que regresan una respuesta dado el cumplimiento de una conjunción de relaciones. Después de obtener respuesta a una consulta, el usuario puede preguntar por una explicación de la respuesta. Si la consulta es de recuperación, el usuario puede especificar una constante y preguntar por qué dicha constante no fue regresada como respuesta. Si la consulta es dicotómica y no pudo ser probada como verdadera o falsa, el usuario puede preguntar por reglas implícitas. Además de probar conocimiento negativo de manera explícita, la base de datos también hace uso de la negación por fallo. Finalmente, los procesos deductivos de este tipo de base de datos se basan en la silogística. Consideremos, a modo de ejemplo, el siguiente fragmento adaptado de [17]:

```
\\Hechos
hombre(Socrates)
sabio(Socrates)
hombre(Joe)
edad(Joe,1)
hombre(John)
edad(John,15)
hombre(Peter)
edad(Peter,40)
perro(Fido)
obra_de_arte(MonaLisa)

\\Reglas
animal(X) A hombre(X)
animal(X) A perro(X)
mortal(X) A hombre(X)
mortal(X) E obra_de_arte(X)
hombre(X) E perro(X)
responsable(X) E ~hombre(X)
responsable(X) A sabio(X)
responsable(X) E bebe(X)
responsable(X) I adulto(X)
responsable(X) E idiota(X)
edad(X,0-120) A hombre(X)
bebe(X) A (hombre(X)^edad(X,0-2))
adulto(X) A (hombre(X)^edad(X,21-120))
```

```
idiota(X) I (hombre(X)^edad(X,30-50))

\\Ejemplo
> mortal(X) ^ ~responsible(X)
Joe
Por negacion por fallo: John
Posibilidad: Fido
> Explica Joe
Porque Joe es un hombre, es mortal.
Porque Joe es un hombre con edad 1, es bebe.
Porque Joe es un bebe, no es responsable.
> Explica Fido
Porque Fido es un perro, es animal.
Algun animal es mortal; por ejemplo, hombre.
Porque Fido es animal, tal vez es mortal.
Porque Fido es un perro, no es hombre.
Porque Fido no es hombre, no es responsable.
> Explica Socrates
Porque Socrates es sabio, es responsable.
> Explica Peter
Porque Peter es hombre con 40, es adulto.
Porque Peter es adulto, tal vez es responsable.
```

Las bases de datos aristotélicas, pues, usan reglas silogísticas como modelos adecuados de inferencia. La principal ventaja de este modo de hacer bases de datos es su cercanía cognitiva con el razonamiento en lenguaje natural. Esto sugiere, en opinión de Mozes, dos áreas en las que la lógica tradicional podría ser aplicada: aplicaciones en las cuales la interacción en lenguaje natural con humanos es ubicua; y aplicaciones que tienen como objetivo simular razonamiento humano cuando es preciso sugerir posibilidades o inducciones.

4. El lenguaje TFL^{PL}

Ahora bien, si asumimos la meta de interacción en lenguaje natural con humanos, parece que es conveniente hacer uso de una base de datos aristotélica más poderosa en la medida en que sea capaz de representar un rango más amplio de inferencias de sentido común. Para lograr esto, a continuación hacemos una breve introducción del sistema TFL^{+}.

4.1. El sistema TFL^{+}

Peterson [22] y Thompson [32] desarrollaron extensiones para la silogística añadiendo cuantificadores adicionales: "la mayoría"(para proposiciones mayoritarias), "muchos"(para proposiciones comunes) y "poco"(para proposiciones

predominantes)[5]. Así, esta silogística intermedia añade las siguientes proposiciones de sentido común: p es la predominante afirmativa (*Pocos* S *no son* P), b es la predominante negativa (*Pocos* S *son* P), t es la mayoritaria afirmativa (*La mayoría de* S *es* P), d es la mayoritaria negativa (*La mayoría de* S *no es* P), k es la común afirmativa (*Muchos* S *son* P) y g es la común negativa (*Muchos* S *no son* P).

Ahora bien, como hemos visto, TFL provee un enfoque algebraico simple y correcto para el razonamiento silogístico que, desafortunadamente, no cubre casos de razonamiento de sentido común con cuantificadores no-clásicos como "mayoría", "muchos" o "pocos"; por otro lado, la silogística extendida de Peterson y Thompson, SYLL$^+$, incluye un rango más amplio de inferencias de sentido común en lenguaje natural, pero carece de un procedimiento algebraico. Así, dado este estado de cosas, en esta sección exponemos brevemente el sistema TFL$^+$ como una extensión de TFL que es capaz de lidiar con una amplia gama de inferencias de sentido común en lenguaje natural pero con las ventajas de un enfoque algebraico. Para exponer este sistema procedemos en tres pasos, primero proponemos una modificación de la sintaxis de TFL con el objetivo de representar los cuantificadores adicionales de SYLL$^+$, posteriormente mostramos el método de decisión de TFL$^+$ y por último mencionamos en qué sentido este sistema es confiable.

Pues bien, primero, para representar las proposiciones p, t, k, b, d y g dentro del marco algebraico de TFL, consideremos la propuesta desplegada en el Cuadro 6.

Tabla 6. Representación de las proposiciones básicas.

SYLL$^+$	TFL$^+$	SYLL$^+$	TFL$^+$
SaP	$:= -\mathsf{S}^0 + \mathsf{P}^0$	SeP	$:= -\mathsf{S}^0 - \mathsf{P}^0$
SpP	$:= +\mathsf{S}^3 + \mathsf{P}^0$	SbP	$:= +\mathsf{S}^3 - \mathsf{P}^0$
StP	$:= +\mathsf{S}^2 + \mathsf{P}^0$	SdP	$:= +\mathsf{S}^2 - \mathsf{P}^0$
SkP	$:= +\mathsf{S}^1 + \mathsf{P}^0$	SgP	$:= +\mathsf{S}^1 - \mathsf{P}^0$
SiP	$:= +\mathsf{S}^0 + \mathsf{P}^0$	SoP	$:= +\mathsf{S}^0 - \mathsf{P}^0$

La razón detrás de esta propuesta es simple: de acuerdo con el marco lógico de SYLL$^+$, las proposiciones intermedias no-universales, i.e., p (b), t (d) y k (g), son particulares hasta cierto punto, tal como lo son las proposiciones tipo i (o), lo cual nos obliga a elegir, siguiendo la sintaxis de TFL, una combinación $+/+$ de términos para las proposiciones afirmativas; y una combinación $+/-$ para las negativas. Sin embargo, esto no es suficiente porque, de acuerdo con SYLL$^+$, las proposiciones p (b), t (d) y k (g) no son convertibles,[6] y por tanto, no son

[5] Aquí seguimos la presentación de [32].

[6] Así, por ejemplo, t := *La mayoría de mexicanos hablan español* es particular, tal y como i := *Algunos mexicanos hablan español*, pero claramente t no es convertible y

equivalentes a proposiciones de tipo i (o), lo cual nos obliga a usar algún tipo de bandera para denotar explícitamente este hecho: nosotros proponemos el uso de superíndices.

Ahora, de acuerdo con SYLL$^+$, los nuevos cuantificadores implican un cierto orden (p (b) implica t (d), t (d) implica k (g) y k (g) implica i (o)) y por ende los superíndices se usan no sólo como banderas, sino como niveles ordenados de cuantificación. Esta elección sintáctica tiene las siguientes características: las proposiciones tipo a, e, i y o tienen nivel 0 para denotar el hecho de que se comportan de manera usual, como si no se hubieran hecho modificaciones; los superíndices se añaden a cada término con la finalidad de especificar el detalle de que las proposiciones tipo p, t, k, b, d y g no son convertibles; y además, estos índices nos permiten inducir un orden ($3 \geq 2 \geq 1 \geq 0$) que indica que a (e) no entraña p (b), t (d), k (g), i (o); pero p (b), t (d), k (g) sí entrañan i (o).[7].

Ahora, dada esta representación, la modificación del método de decisión es como sigue: una conclusión se sigue válidamente de un conjunto de premisas syss *i)* la suma de las premisas es algebraicamente igual a la conclusión, *ii)* el número de conclusiones con cantidad particular (*viz.*, cero o uno) es igual al número de premisas con cantidad particular; y *iii)* el nivel de cuantificación de la conclusión es menor o igual que el máximo nivel de cuantificación de las premisas.

Para ejemplificar este procedimiento consideremos un par de ejemplos, uno válido (Tabla 7), uno inválido (Tabla 8: denotamos el hecho de que una conclusión no se sigue mediante "$\nvdash$"). Como es de esperarse, la adición de p, t, k, b, d y g incrementa el número de patrones o modos silogísticos correctos (*vide* Apéndice B).

Tabla 7. att-1

Proposición	TFL$^+$
1. Todo H es M.	$-\mathsf{H}^0 + \mathsf{M}^0$
2. La mayoría de G son H.	$+\mathsf{G}^2 + \mathsf{H}^0$
$\vdash$ La mayoría de G son M.	$+\mathsf{G}^2 + \mathsf{M}^0$

Tabla 8. tta-1

Proposición	TFL$^+$
1. La mayoría de H son M.	$+\mathsf{H}^2 + \mathsf{M}^0$
2. La mayoría de G son M.	$+\mathsf{G}^2 + \mathsf{M}^0$
$\nvdash$ Todo G es M.	$-\mathsf{G}^0 + \mathsf{M}^0$

Como podemos ver, el sistema TFL$^+$ tiene las ventajas de un sistema algebraico (la reducción de un conjunto complejo de inferencias a un lenguaje formal

por tanto no es equivalente a i: notemos que si *Algunos mexicanos hablan español* entonces seguramente *Algunos hispanohablantes son mexicanos*, pero *La mayoría de mexicanos hablan español* no entraña que *La mayoría de hispanohablantes son mexicanos*. Contraejemplos similares pueden ser expuestos para mostrar que las proposiciones p (b), t (d) y k (g) no colapsan en proposiciones tipo i (o).

[7] Esto es diferente de la versión original de [32]: Thompson permite que las proposiciones universales entrañen proposiciones particulares, pero nuestra versión sigue la propuesta de Sommers y Englebretsen, por lo que nosotros tenemos que añadir otra regla al marco SYLL$^+$: si dos premisas son universales, la conclusión no puede ser particular (*vide* Apéndice B)

simple y unificado) y, al mismo tiempo, tiene las ventajas de una teoría inferencial con cuantificadores no-clásicos (la inclusión de un modelo de inferencia de sentido común en lenguaje natural), con la adición de que es confiable en el sentido de que el proceso inferencial es correcto:

Proposición 1 (Confiabilidad) *Una inferencia es* $\mathsf{SYLL}^{+}_{\text{válida}}$ *syss es* $\mathsf{TFL}^{+}_{\text{válida}}$.

La relevancia de este sistema puede apreciarse mejor al considerar el compromiso entre las limitaciones expresivas de TFL y las limitaciones algebraicas de SYLL^{+} frente a la confiabilidad de TFL^{+} con respecto a inferencias de sentido común en lenguaje natural. A modo de ilustración, consideremos algunos ejemplos (Tabla 9- 12).

Tabla 9. kaa-1.

Proposición	TFL^{+}
1. Muchos H son I.	$+\mathsf{H}^1 + \mathsf{I}^0$
2. Este g es H.	$-\mathsf{g}^0 + \mathsf{H}^0$
$\nvdash$ Este g es I.	$-\mathsf{g}^0 + \mathsf{I}^0$

Tabla 10. akt-4.

Proposición	TFL^{+}
1. Todo C es F.	$-\mathsf{C}^0 + \mathsf{F}^0$
2. Muchos M son C.	$+\mathsf{M}^1 + \mathsf{C}^0$
$\nvdash$ La mayoría de M son F.	$+\mathsf{M}^2 + \mathsf{F}^0$

Tabla 11. bao-3.

Proposición	TFL^{+}
1. Pocos M son B.	$+\mathsf{M}^3 - \mathsf{B}^0$
2. Todo M es R.	$-\mathsf{M}^0 + \mathsf{R}^0$
$\vdash$ Algunos R no son B.	$+\mathsf{R}^0 - \mathsf{B}^0$

Tabla 12. etg-2.

Proposición	TFL^{+}
1. Ningún F es C.	$-\mathsf{F}^0 - \mathsf{C}^0$
2. La mayoría de V es C.	$+\mathsf{V}^2 + \mathsf{C}^0$
$\vdash$ Muchos V no son F.	$+\mathsf{V}^1 - \mathsf{F}^0$

4.2. Una introducción a $\mathsf{TFL}^{\mathsf{PL}}$

Pues bien, si la relevancia de la confiabilidad de TFL^{+} tiene que ver con las limitaciones expresivas y algebraicas de TFL y SYLL^{+}—sin mencionar las limitaciones de LPO—con respecto a inferencias de sentido común en lenguaje natural, la utilidad de *Term Functor Logic Programming Language* ($\mathsf{TFL}^{\mathsf{PL}}$) resulta de su potencial aplicación a sistemas de recuperación de información en los que la interacción en lenguaje natural con humanos es fundamental. Por estas consideraciones, $\mathsf{TFL}^{\mathsf{PL}}$ es un lenguaje que, como Prolog, tiene una gramática especial. El siguiente fragmento es un ejemplo de programa en $\mathsf{TFL}^{\mathsf{PL}}$:

```
-s0+H0
-H0+M0
```

La primera línea puede leerse como "Sócrates es hombre" (i.e. $-\mathtt{s}^0 + \mathsf{M}^0$ en TFL^+) mientras que la segunda línea representa "Todo hombre es mortal" (i.e. $-\mathsf{H}^0 + \mathsf{M}^0$ en TFL^+). Esto permite ver que, así como en TFL la distinción entre proposición singular y universal desaparece, en $\mathsf{TFL}^{\mathsf{PL}}$ desaparece la distinción entre hechos y reglas que es usual, por ejemplo, en Prolog; esto también es diferente de la noción original de base de datos aristotélica de Mozes.

Ahora, por ejemplo, dado este programa, podemos llevar a cabo la siguiente consulta inferencial:

```
> s

-H0+M0
-s0+H0
-------
-s0+M0
```

esto es, "¿Qué es sócrates?" (s), y el programa responde $-\mathtt{s0} + \mathtt{M0}$, es decir, "Sócrates es mortal". A partir de este pequeño ejemplo podemos notar que, a diferencia de Prolog, la sintaxis de $\mathsf{TFL}^{\mathsf{PL}}$ es ternaria, aristotélica, y en ese sentido, $\mathsf{TFL}^{\mathsf{PL}}$ induce una base de datos *à la* Mozes; sin embargo, a diferencia de una base de datos aristotélica original, $\mathsf{TFL}^{\mathsf{PL}}$ rehuye la elección de una sintaxis binaria *à la* Frege (cf. §3) para favorecer una sintaxis más cercana al proyecto de Sommers. Todo esto resulta en la generación de un lenguaje de programación lógica basado en un sistema de lógica de términos y no en un sistema de primer orden, lo cual nos acerca a un lenguaje de programación lógica basado en una proyecto cognitivo de lógica natural.

4.3. Sintaxis y semántica de $\mathsf{TFL}^{\mathsf{PL}}$

Como resultado de estar basado en TFL^+, la sintaxis de $\mathsf{TFL}^{\mathsf{PL}}$ es la misma de TFL^+ con el evidente cambio en la notación: el superíndice se escribe inmediatamente después de cada término. La sintaxis, entonces, se puede resumir mediante la siguiente BNF:

- $\langle \mathtt{programa} \rangle ::= \langle \mathtt{proposición} \rangle | \langle \mathtt{proposición} \rangle \langle \mathtt{programa} \rangle$
- $\langle \mathtt{proposición} \rangle ::= \langle \mathtt{término}^{\mathtt{n}} \rangle \langle \pm \mathtt{T0} \rangle$
- $\langle \mathtt{término}^{\mathtt{n}} \rangle ::= \langle \pm \mathtt{T0} \rangle | \langle + \mathtt{T1} \rangle | \langle + \mathtt{T2} \rangle | \langle + \mathtt{T3} \rangle$

Un programa en $\mathsf{TFL}^{\mathsf{PL}}$ consiste de una o más proposiciones y cada proposición se define mediante dos términos definidos como en TFL^+. Esta sintaxis formal nos permite definir, de modo riguroso y sin ambigüedad, los constructos de nuestro lenguaje, además de que permite apreciar la conexión directa entre la motivación filosófica y la motivación cognitiva.

Claramente, dada esta sintaxis, la semántica formal de $\mathsf{TFL}^{\mathsf{PL}}$ es directamente la semántica formal de TFL^+, y dada la Proposición 1, es posible inferir que $\mathsf{TFL}^{\mathsf{PL}}$ preserva los resultados técnicos que pretendemos capturar, a saber, el proceso inferencial usando una lógica de términos.

4.4. Implementación

La implementación de $\mathsf{TFL}^{\mathsf{PL}}$ se ha realizado en C. Al día de hoy, el motor inferencial recibe un programa $\mathsf{TFL}^{\mathsf{PL}}$ y hace un uso recursivo del procedimiento inferencial definido en la sección 4.1, a saber, que una conclusión se sigue válidamente de un conjunto de premisas syss *i)* la suma de las premisas es algebraicamente igual a la conclusión, *ii)* el número de conclusiones con cantidad particular (viz., cero o uno) es igual al número de premisas con cantidad particular; y *iii)* el nivel de cuantificación de la conclusión es menor o igual que el máximo nivel de cuantificación de las premisas.

4.5. Ejemplo

```
-s0+M0 // Sócrates es hombre.
-f0+D0 // Fido es un perro.
-M0+A0 // Todo hombre es animal.
-D0+A0 // Todo perro es animal.
-M0+O0 // Todo hombre es mortal.
-A0+O0 // Todo animal es mortal.
+A3+O0 // Pocos animales no son mortales.

> s // Qué es Sócrates.
-M0+A0 // Todo hombre es animal.
-s0+M0 // Sócrates es hombre.
-------
-s0+A0 // Luego, Sócrates es animal.

-M0+O0 // Todo hombre es mortal.
-s0+M0 // Sócrates es hombre.
-------
-s0+O0 // Luego, Sócrates es mortal.
```

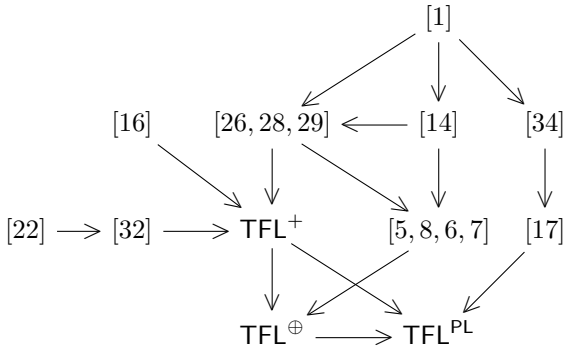
Algunas consideraciones que podemos extraer de la sintaxis, la implementación y el ejemplo anterior son las siguientes: *a)* como la sintaxis de TFL conlleva el abandono de la sintaxis Fregeana, $\mathsf{TFL}^{\mathsf{PL}}$ no requiere del uso de variables y constantes individuales. *b)* Como la sintaxis de TFL conlleva la eliminación de la distinción entre proposición singular y proposición universal, la distinción entre hecho y regla desaparece en $\mathsf{TFL}^{\mathsf{PL}}$. *c)* Dadas las consideraciones anteriores, TFL no necesita hacer uso del predicado de igualdad, $=$, por lo que en $\mathsf{TFL}^{\mathsf{PL}}$ no necesitamos hacer uso de algoritmos de sustitución y unificación como en Prolog. *d)* $\mathsf{TFL}^{\mathsf{PL}}$ permite definir bases de datos aristotélicas en tanto que usa una base inferencial silogística y tiene una motivación más cercana a una lógica natural, si bien, a diferencia de una base de datos aristotélica original, su sintaxis no es binaria. *e)* Hasta el momento, $\mathsf{TFL}^{\mathsf{PL}}$ sólo hace uso de proposiciones construidas mediante pares de términos, pero como TFL modela relaciones más complejas, es necesario añadir a $\mathsf{TFL}^{\mathsf{PL}}$ un módulo relacional, un módulo para lidiar con

razonamiento numérico y, por supuesto, un módulo que regrese las respuestas en lenguaje natural.

5. Conclusiones

En esta contribución hemos alcanzado dos metas: *i)* la introducción de los elementos básicos del sistema lógico de términos TFL^{+} y *ii)* la presentación de los avances de un lenguaje de programación diseñado a partir de tal sistema, $\mathsf{TFL}^{\mathsf{PL}}$. [25] ha sugerido, por un lado, que la creación de un lenguaje de programación puede ocurrir por varias razones, como cuando una nueva área de aplicación demanda un nuevo lenguaje, o cuando las deficiencias de un lenguaje son notables, o cuando es más sencillo crear un nuevo lenguaje en lugar de modificar uno existente; y por otro lado, ha propuesto que un lenguaje es significativo cuando tal lenguaje es práctico y técnicamente novedoso. Pues bien, $\mathsf{TFL}^{\mathsf{PL}}$ surgió porque nos parece que, si bien hay lenguajes de programación lógica capaces de procesar lenguaje natural y realizar inferencia, no existe un lenguaje de programación basado en un proyecto de lógica natural, como se puede ver por la adopción de LPO por parte de los lenguajes de programación lógica (Cf. §2). Además, si bien hasta este momento $\mathsf{TFL}^{\mathsf{PL}}$ no parece más práctico que otros lenguajes de programación, ciertamente es técnicamente novedoso, tanto por su sintaxis como por su motivación. Y por ello, si bien $\mathsf{TFL}^{\mathsf{PL}}$ está en desarrollo, creemos que tiene potencial para el manejo de inferencias en lenguaje natural, lo cual es relevante dentro del paradigma clásico o fundacional de la inteligencia artificial: comprender y construir sistemas inteligentes.

Por último, nos gustaría cerrar este trabajo comentando que $\mathsf{TFL}^{\mathsf{PL}}$ forma parte de un proyecto cognitivo más general que incluye aspectos lógicos, lingüísticos, didácticos y filosóficos, y que se puede apreciar con mayor claridad de manera gráfica:



Consideramos, pues, que $\mathsf{TFL}^{\mathsf{PL}}$ es un sistema que promete no sólo como una herramienta computacional, sino como un dispositivo de investigación asociado a un proyecto cognitivo general en el que se podrán incluir módulos de lógica relacional (dado que TFL^{+} es capaz de lidiar con inferencias relacionales),

razonamiento probabilístico (en tanto que TFL^+ puede usarse para representar medidas probabilísticas (Cf. [33])) y razonamiento numérico (en la medida en que sea posible añadir módulos de inferencia silogística numérica (Cf. [18])); además de que podría adaptarse a estudios psicológicos (en la medida en que podría enriquecer las explicaciones psicológicas del razonamiento de sentido común (Cf. [11])) y filosóficos (en tanto sea capaz de promover la revisión de las lógicas de términos (Cf. [34,28,6,7]) como herramientas que podrían ser más interesantes y útiles de lo que originalmente habíamos creído (Cf.[4,10])).

Apéndice A. Aspectos generales de la silogística

La *silogística* (SYLL) es una lógica de términos que tiene sus orígenes en *Primeros analíticos* [1] y estudia la relación de inferencia entre proposiciones categóricas. Una *proposición categórica* es una proposición compuesta por dos términos, una cantidad y una cualidad. El sujeto y el predicado de la proposición se llaman *términos*: el término-esquema S denota el término sujeto de la proposición y el término-esquema P denota el predicado. La *cantidad* puede ser universal (*Todo*) o particular (*Algún*) y la *cualidad* puede ser afirmativa (*es*) o negativa (*no es*).

Estas proposiciones categóricas se denotan mediante una *etiqueta* (a (para la universal afirmativa, SaP), e (para la universal negativa, SeP), i (para la particular afirmativa, SiP), y o (para la particular negativa, SoP)) que nos permite determinar una secuencia de tres proposiciones que se conoce como *modo*. Un *silogismo categórico*, entonces, es un modo ordenado de tal manera que dos proposiciones fungen como premisas y la última como conclusión. Al interior de las premisas existe un término que ocurre en ambas premisas pero no en la conclusión: este término especial, usualmente denotado con el término-esquema M, funciona como un enlace entre los términos restantes y es conocido como término medio. De acuerdo a la posición del término medio se pueden definir cuatro arreglos o *figuras* que codifican los modos o patrones silogísticos válidos (Tabla 13)[8].

Tabla 13. Modos silogísticos válidos

Figura 1	Figura 2	Figura 3	Figura 4
aaa	eae	iai	aee
eae	aee	aii	iai
aii	eio	oao	eio
eio	aoo	eio	

[8] Por mor de brevedad, pero sin pérdida de generalidad, omitimos los silogismos que requieren carga existencial.

Apéndice B. Modos de la silogística intermedia

Tabla 14. Extensión de los modos silogísticos válidos (adaptado de [32])

	Figura 1	Figura 2	Figura 3	Figura 4
Con "mayoría"	aat	aed	ati	aed
	att	add	eto	eto
	ati	ado	tai	tai
	ead	ead	dao	
	etd	etd		
	eto	eto		
Con "muchos"	aak	aeg	aki	aeg
	atk	adg	eko	eko
	aki	ago	kai	kai
	akk	agg	gao	
	eag	eag		
	etg	etg		
	eko	eko		
	ekg	ehg		
Con "pocos"	aap	aeb	pai	aeb
	app	abb	epo	pai
	apt	abd	bao	epo
	apk	abg	api	
	api	abo		
	eab	eab		
	epb	epb		
	epd	epd		
	epg	epg		
	epo	epo		

Para los modos que necesitan carga existencial, como aat-1 o aak-1, el único requisito que se necesita para producir una inferencia válida en TFL^+ es añadir la premisa implícita que enuncia la existencia del término sujeto, es decir, algo como $+\mathsf{S} + \mathsf{S}$.

Referencias

1. Aristotle, Smith, R.: Prior Analytics. Hackett Classics Series, Hackett (1989)
2. Bergin, T., Gibson, R.: History of Programming Languages II. ACM Press books, ACM Press (1996)
3. Bratko, I.: Prolog Programming for Artificial Intelligence. International computer science series, Addison Wesley (2001)
4. Carnap, R.: Die alte und die neue logik. Erkenntnis 1, 12–26 (1930), `http://www.jstor.org/stable/20011586`

5. Englebretsen, G.: The New Syllogistic. 05, P. Lang (1987)
6. Englebretsen, G.: Something to Reckon with: The Logic of Terms. Canadian electronic library: Books collection, University of Ottawa Press (1996)
7. Englebretsen, G., Sayward, C.: Philosophical Logic: An Introduction to Advanced Topics. Bloomsbury Academic (2011)
8. Englebretsen, G.: Linear diagrams for syllogisms (with relationals). Notre Dame J. Formal Logic 33(1), 37–69 (12 1991), https://doi.org/10.1305/ndjfl/1093636009
9. Geach, P.: Reference and Generality: An Examination of Some Medieval and Modern Theories. Contemporary Philosophy / Cornell University, Cornell University Press (1962)
10. Geach, P.: Logic Matters. Campus (University of California Press), University of California Press (1980)
11. Khemlani, S., Johnson-laird, P.N.: Theories of the syllogism: a meta-analysis. Psychological Bulletin pp. 427–457 (2012)
12. Kowalski, R.A.: The early years of logic programming. Commun. ACM 31(1), 38–43 (Jan 1988), http://doi.acm.org/10.1145/35043.35046
13. Kuhn, S.T.: An axiomatization of predicate functor logic. Notre Dame J. Formal Logic 24(2), 233–241 (04 1983), https://doi.org/10.1305/ndjfl/1093870313
14. von Leibniz, G., Couturat, L.: Opuscules et fragments inédits de Leibniz: extraits des manuscrits de la Bibliothèque royale de Hanovre. Olms paperback, Olms (1961)
15. Moss, L.: Natural logic. In: Lappin, S., Fox, C. (eds.) The Handbook of Contemporary Semantic Theory. John Wiley & Sons (2015)
16. Mostowski, A.: On a generalization of quantifiers. Fundamenta Mathematicae 44(2), 12–36 (1957)
17. Mozes, E.: A deductive database based on aristotelian logic. Journal of Symbolic Computation 7(5), 487–507 (1989), http://www.sciencedirect.com/science/article/pii/S0747717189800306
18. Murphree, W.A.: Numerical term logic. Notre Dame J. Formal Logic 39(3), 346–362 (07 1998), https://doi.org/10.1305/ndjfl/1039182251
19. Nakatsu, R.T.: Diagrammatic Reasoning in AI. Wiley (2009)
20. Noah, A.: Predicate-functors and the limits of decidability in logic. Notre Dame J. Formal Logic 21(4), 701–707 (10 1980), https://doi.org/10.1305/ndjfl/1093883255
21. Noah, A.: Sommers's cancellation technique and the method of resolution. In: Oderberg, D. (ed.) The Old New Logic: Essays on the Philosophy of Fred Sommers, pp. 169–182. a Bradford book (2005)
22. Peterson, P.L.: On the logic of "few", "many", and "most". Notre Dame J. Formal Logic 20(1), 155–179 (01 1979), https://doi.org/10.1305/ndjfl/1093882414
23. Quine, W.V.O.: Predicate functor logic. In: Fenstad, J.E. (ed.) Proceedings of the Second Scandinavian Logic Symposium. North-Holland
24. Russell, B.: A critical exposition of the philosophy of Leibniz: with an appendix of leading passages. G. Allen & Unwin (1937)
25. Sammet, J.E.: Programming languages: History and future. Commun. ACM 15(7), 601–610 (Jul 1972), http://doi.acm.org/10.1145/361454.361485
26. Sommers, F.: On a fregean dogma. In: Lakatos, I. (ed.) Problems in the Philosophy of Mathematics, Studies in Logic and the Foundations of Mathematics, vol. 47, pp. 47–81. Elsevier (1967), http://www.sciencedirect.com/science/article/pii/S0049237X08715210
27. Sommers, F.: Intelectual autobiography. In: Oderberg, D. (ed.) The Old New Logic: Essays on the Philosophy of Fred Sommers, pp. 1–24. A Bradford book (2005)

28. Sommers, F.: The Logic of Natural Language. Clarendon Library of Logic and Philosophy, Clarendon Press; Oxford: New York: Oxford University Press (1982)
29. Sommers, F., Englebretsen, G.: An Invitation to Formal Reasoning: The Logic of Terms. Ashgate (2000)
30. Staal, J.F.: Formal logic and natural languages (a symposium). Foundations of Language 5(2), 256–284 (1969), http://www.jstor.org/stable/25000379
31. Sterling, L., Shapiro, E.: The Art of Prolog: Advanced Programming Techniques. Logic programming, MIT Press (1994)
32. Thompson, B.: Syllogisms using "few", "many", and "most". Notre Dame J. Formal Logic 23(1), 75–84 (01 1982), https://doi.org/10.1305/ndjfl/1093883568
33. Thompson, B.: Syllogisms with statistical quantifiers. Notre Dame J. Formal Logic 27(1), 93–103 (01 1986), https://doi.org/10.1305/ndjfl/1093636527
34. Veatch, H.B.: Intentional logic: a logic based on philosophical realism. Archon Books (1970)
35. Woods, J.: Logic Naturalized, pp. 403–432. Springer International Publishing, Cham (2016), https://doi.org/10.1007/978-3-319-26506-3_18