Uma Lógica Paraconsistente das Teorias de Quase-Verdade e Algumas Metapropriedades Via Traduções entre Lógicas

Luiz Henrique da Cruz Silvestrini
Hércules de Araújo Feitosa

Como citar: SILVESTRINI, L. H. da C.; FEITOSA, H. de A. Uma lógica paraconsistente das teorias de quase-verdade e algumas metapropriedades via traduções entre lógicas. *In*: ALVES, M. A.; GRÁCIO, M. C. C.; MARTÍNEZ-ÁVILA, D. (org.). **Informação, conhecimento e modelos**. Marília: Oficina Universitária; São Paulo: Cultura Acadêmica, 2017. p. 79-98. DOI: https://doi.org/10.36311/2017.978-85-86497-29-2.p79-98

Uma Lógica Paraconsistente das Teorias de Quase-Verdade e Algumas Metapropriedades Via Traduções entre Lógicas

Luiz Henrique da Cruz Silvestrini
silvestrini@fc.unesp.br;
Hércules de Araújo Feitosa
haf@fc.unesp.br

1. Introdução

Interessados na investigação da teoria de quase-verdade e sua formalização, Silvestrini e Coniglio (2011; 2014) mostraram que uma lógica subjacente da teoria da quase-verdade, segundo a abordagem proposta, é uma lógica paraconsistente denominada lógica da verdade pragmática (doravante, LPT).

Por outro lado, Feitosa e D'Ottaviano (2001) desenvolveram uma teoria de traduções entre lógicas, evidenciaram alguns metarresultados gerais sobre traduções entre lógicas, introduziram o conceito de tradução conservativa e construíram alguns exemplos de traduções conservativas.

Para compor um contexto de entendimento dessas noções, iniciamos com uma seção sobre noções gerais acerca das lógicas paraconsistentes, em particular, sobre as Lógicas de Inconsistência Formal (LFIs), pois a lógica LPT pode ser vista como um caso particular de LFI, conforme indicado no final da Seção 2. Em seguida, introduzimos o sistema LPT, em versão semântica, dado por tabelas de verdade trivalentes, e também em versão sintática, num sistema axiomático com dezesseis axiomas.

Na quarta seção, abordamos alguns elementos dessa teoria de traduções e, na seção seguinte, introduzimos duas traduções conservativas que envolvem a LPT. A primeira é uma tradução conservativa da lógica proposicional clássica em LPT; a segunda é uma tradução conservativa da LPT na lógica trivalente de Łukasiewicz, a qual é bastante conhecida e tem

Marcos Antonio Alves, Maria Claudia Cabrini Grácio, Daniel Martínez-Ávila (Org.). *Informação, conhecimento e modelos.* Coleção CLE, v. 78, pp. 79-98, 2017.

https://doi.org/10.36311/2020.978-85-86497-29-2.p78-98

modelos algébricos interessantes.

Demonstramos, nas considerações finais, que, com o uso das traduções, podemos demonstrar algumas metapropriedades da lógica LPT, não apresentadas na versão original de Coniglio e Silvestrini (2014), a saber, a consistência e a decidibilidade de LPT.

2. Sobre as lógicas paraconsistentes

Na tradição das lógicas paraconsistentes, podemos distinguir os conceitos de contradição e trivialidade, que, no caso da lógica clássica e muitas outras conhecidas lógicas, são conceitos equivalentes.

Assim, no caso clássico, adota-se o preceito básico de que contradições, em uma teoria, equivalem à trivialização dedutiva da teoria. Esse fato também é conhecido como *Princípio de Explosão*. A seguir, procuramos explicitar esses conceitos usuais, em conformidade com Carnielli et al. (2000).

Definição 2.1: Um *sistema lógico* ***L*** é um par $(For, \vdash_L)$, determinado por um conjunto de fórmulas *For* e por uma relação de consequência $\vdash_L$ sobre *For*.

Definição 2.2: Uma *teoria* em $(For, \vdash_L)$ é um conjunto de fórmulas $\Gamma \subseteq For$ que é fechado para a relação de consequência $\vdash_L$, isto é, $\Gamma \vdash_L \alpha$ se, e somente se, $\alpha \in \Gamma$.

Definição 2.3: Uma teoria Γ é *contraditória* se, para alguma fórmula $\alpha \in For$, se tem que $\Gamma \vdash_L \alpha$ e $\Gamma \vdash_L \neg\alpha$.

Num contexto contraditório, algo pode ser e não ser ao mesmo tempo.

Definição 2.4: Uma teoria Γ é *consistente*, se não é contraditória.

Uma teoria consistente não admite contradições.

Definição 2.5: Uma teoria Γ é *trivial* se, para toda fórmula $\beta \in For$, se tem que $\Gamma \vdash_L \beta$.

Uma teoria trivial faz com que todas as suas fórmulas sejam teoremas e, assim, não distingue uma classe de fórmulas que devem ser aceitas naquela teoria de uma outra classe de fórmulas não aceitas.

Definição 2.6: Um sistema lógico ***L*** é paraconsistente, quando permite a distinção entre teorias contraditórias e triviais.

De modo equivalente, podemos afirmar que um sistema lógico ***L*** é paraconsistente, se ele é *não-explosivo*, isto é, não vale em ***L*** o princípio de explosão, i.e., $\alpha, \neg\alpha \vdash_L \beta$, para toda fórmula $\beta \in For$.

Nos sistemas lógicos clássicos e outros bastante conhecidos, não há distinção entre contradição e trivialização. Assim, adota-se o preceito básico de que contradições, em uma teoria, equivalem à trivialização dedutiva (o princípio de explosão).

Por outro lado, as LFIs determinam uma classe ampla de lógicas paraconsistentes, lógicas tolerantes às contradições, no sentido de que o princípio de explosão não vale em geral. Além disso, a não-trivialidade não pode ser definida apenas como ausência de contradição, pois, nessa relação, está pressuposto o conceito de consistência. Nesse sentido, a trivialidade não mais equivale à contradição e pode ser representada por meio da seguinte equação: Contradição + Consistência = Trivialização.

Ademais, numa LFI, tal sistema lógico mantém sua capacidade de realizar inferências razoáveis, ou seja, não trivializa, mesmo na presença de contradições.

Por exemplo, podemos conceber uma LFI subjacente a algum banco de dados relacionais, e esta é considerada uma ferramenta conveniente para a manipulação de informação em um ambiente em que dados inconsistentes podem ser representados e manipulados, mas também que novas restrições de integridade sejam adicionadas, as quais poderiam mudar o estado dos dados já armazenados.

Assim, as LFIs permitem raciocinar sob contradição, porém, não inferem contradições como teoremas: são lógicas que podem servir de base para teorias inconsistentes, contudo, não triviais.

Um aspecto das LFIs que as distingue da versão original dos cálculos paraconsistentes de da Costa consiste em tomar a consistência (e/ou inconsistência) de alguma fórmula como uma noção primitiva e descrita através de específicos conectivos da linguagem.

Cada LFI apresenta um símbolo para o conectivo primitivo de negação "$\neg$" e admite os seguintes princípios:

(i) para algum subconjunto $\Gamma\cup\{\alpha, \beta\} \subseteq For$, temos que $\Gamma, \alpha, \neg\alpha \nVdash_L \beta$ (não explosão);

(ii) há um conjunto de fórmulas $\circ(p)$, que depende exatamente da variável proposicional p e que, para fórmulas α e β quaisquer, satisfaz:

1. $\circ(\alpha), \alpha \nVdash_L \beta$;
2. $\circ(\alpha), \neg\alpha \nVdash_L \beta$.

(iii) para todo subconjunto $\Gamma\cup\{\alpha, \beta\} \subseteq For$, temos que $\Gamma, \circ(\alpha), \alpha, \neg\alpha \vdash_L \beta$ (princípio de explosão fraca).

Silvestrini (2011) verificou que a lógica LPT é uma lógica não explosiva e que adota o princípio de explosão fraca. Portanto, LPT é uma LFI.

A seguir, descreveremos as origens da LPT, a lógica da verdade pragmática, a qual surge como alternativa e generalização da proposta original de da Costa e colaboradores, em que algo de incerteza pode ser preservado em qualquer fórmula do sistema lógico e não apenas nas fórmulas atômicas, como na versão inicial.

3. A lógica da verdade pragmática

Newton da Costa e colaboradores desenvolveram uma teoria da verdade, denominada teoria da *quase-verdade* ou teoria da *verdade pragmática*, por receberem influência de textos dos filósofos pragmáticos, particularmente, de Charles Sanders Peirce (cf. HIFUME, 2003).

Nessa teoria, a quase-verdade é empregada como a concepção de verdade inerente às ciências empíricas, i.e., em domínios do conhecimento em que há conhecimento parcial, ou até mesmo conflitante, como no caso das teorias da mecânica clássica, da relativista e da quântica (cf. DA COSTA; FRENCH, 2003).

Uma das originalidades da concepção de quase-verdade reside no fato de que as estruturas ou modelos, nas quais uma determinada linguagem é interpretada, deixam de ser estruturas totais (ou usuais da Teoria de Modelos), como no caso da teoria de Tarski, e tornam-se *estruturas parciais*, em que as relações envolvidas são parciais (cf. MIKENBERG et al., 1986).

Ao formalizar o conceito de quase-verdade, Silvestrini (2011) assinala que a lógica proposicional subjacente à noção de quase-verdade de da Costa é uma lógica paraconsistente, denotada por LPT, a qual será descrita a seguir. Como apontado no final da seção anterior, a LPT é uma LFI.

A LPT foi motivada por uma semântica trivalente que permite formalizar os seguintes aspectos da verdade pragmática: as sentenças verdadeiras, as sentenças falsas e as sentenças indeterminadas, no sentido da concepção de relações parciais de da Costa (ver CONIGLIO; SILVESTRINI, 2014).

A linguagem de LPT coincide com a linguagem proposicional clássica seguinte $L = (\neg, \wedge, \rightarrow)$, na qual os operadores proposicionais $\neg$, $\wedge$, $\rightarrow$ denotam, respectivamente, as noções de negação, conjunção e implicação.

Os significados desses operadores são dados pelas seguintes tabelas de verdade:

$\rightarrow$	0	½	1
0	1	1	1
½	0	1	1
1	0	1	1

$\wedge$	0	½	1
0	0	0	0
½	0	½	½
1	0	½	1

	$\neg$
0	1
½	½
1	0

Além desses operadores básicos, definem-se os seguintes operadores e constantes de LPT:

Disjunção: $\varphi \vee \psi \stackrel{\text{def}}{=} \neg(\neg\varphi \wedge \neg\psi)$

Bicondicional: $\varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Top: $\top \stackrel{\text{def}}{=} \varphi \rightarrow \varphi$

Botton: $\bot \stackrel{\text{def}}{=} \neg\top$

Negação clássica: $\sim\varphi \stackrel{\text{def}}{=} \varphi \rightarrow \bot$

Consistência: $\circ\varphi \stackrel{\text{def}}{=} \sim(\varphi \wedge \neg\varphi)$

Os significados desses novos entes são dados pelas seguintes tabelas de verdade:

$\vee$	0	½	1
0	0	½	1
½	½	½	1
1	1	1	1

	$\sim$
0	1
½	0
1	0

	$\circ$
0	1
½	0
1	1

$\leftrightarrow$	0	½	1
0	1	0	0
½	0	1	1
1	0	1	1

	$\top$
0	1
½	1
1	1

	$\perp$
0	0
½	0
1	0

A semântica matricial de LPT é dada pela matriz:

$$\mathcal{M}_{\text{LPT}} = (\{0, ½, 1\}, \neg, \wedge, \rightarrow, \{½, 1\}),$$

com o conjunto de valores designados $D = \{½, 1\}$ e, dessa maneira, a relação de consequência semântica é dada como segue.

Seja $Var(\text{LPT}) = \{s_1, s_2, s_3, ...\}$ o conjunto das variáveis proposicionais de LPT. Uma valoração para LPT é qualquer função v: $Var(\text{LPT}) \rightarrow \{0, ½, 1\}$, a qual é estendida de modo único para o conjunto $For(\text{LPT})$, segundo os operadores introduzidos acima. Para $\Gamma \subseteq For(\text{LPT})$, tem-se que $v(\Gamma) = \{v(\gamma) : \gamma \in \Gamma\}$.

Assim, se $\Gamma\cup\{\varphi\} \subseteq For(\text{LPT})$, então, $\Gamma \vDash \varphi$, quando para toda valoração v tem-se: $v(\Gamma) \subseteq D \Rightarrow v(\varphi) \in D$.

Decorre dessa definição de valoração que toda fórmula de LPT, válida segundo uma valoração v: $Var(\text{LPT}) \rightarrow \{0, ½, 1\}$, é igualmente válida segundo a restrição booleana de v, isto é, segundo $v : Var(\text{LPT}) \rightarrow \{0, 1\}$ com os significados booleanos dos operadores $\neg$, $\wedge$ e $\rightarrow$, em que é apagado o valor ½ (ver SILVESTRINI, 2011). Assim, toda fórmula LPT-válida é uma tautologia.

Baseados na linguagem e matrizes expostas acima, Coniglio e Silvestrini (2014) propuseram os seguintes axiomas e regra de dedução, construindo o sistema axiomático, ou a lógica, que denominamos LPT:

Esquemas de Axiomas:

(A1) $\varphi \to (\psi \to \varphi)$

(A2) $(\varphi \to (\psi \to \sigma)) \to ((\varphi \to \psi) \to (\varphi \to \sigma))$

(A3) $\varphi \to (\psi \to (\varphi \wedge \psi))$

(A4) $(\varphi \wedge \psi) \to \varphi$

(A5) $(\varphi \wedge \psi) \to \psi$

(A6) $\varphi \to (\varphi \vee \psi)$

(A7) $\psi \to (\varphi \vee \psi)$

(A8) $(\varphi \to \sigma) \to ((\psi \to \sigma) \to ((\varphi \vee \psi) \to \sigma))$

(A9) $\varphi \vee (\varphi \to \psi)$

(A10) $\varphi \vee \neg\varphi$

(A11) $\neg\neg\varphi \to \varphi$

(A12) $\circ\varphi \to (\varphi \to (\neg\varphi \to \psi))$

(A13) $\neg\circ\varphi \to (\varphi \wedge \neg\varphi)$

(A14) $\circ(\varphi \to \psi)$

(A15) $(\circ\varphi \wedge \circ\psi) \to \circ(\varphi \wedge \psi)$

(A16) $(\varphi \wedge \neg\varphi \wedge \psi) \to (\neg(\varphi \wedge \psi) \wedge \neg(\psi \wedge \varphi))$.

Regra de Dedução:

(MP) $\varphi, \varphi \to \psi \vdash \psi$.

Uma demonstração de que o sistema dedutivo acima é correto e completo segundo a semântica matricial $\mathcal{M}_{LPT}$ pode ser obtida em Coniglio e Silvestrini (2014).

4. Sobre as traduções entre lógicas

Para desenvolver uma teoria de traduções entre lógicas, da Silva, D'Ottaviano e Sette (1999) precisaram de um conceito bastante geral de lógica, dado por um conjunto não vazio e um operador de consequência sobre esse conjunto e, então, definiram tradução como funções que preservam as relações de consequência. Vejamos alguns detalhes.

Definição 4.1: Lógica abstrata é um par $\boldsymbol{L} = \langle L, C\rangle$, em que L é um conjunto não vazio, o domínio ou universo de $\boldsymbol{L}$, e C é um operador de consequência sobre L, isto é, C é uma função $C : \mathcal{P}(L) \to \mathcal{P}(L)$ tal que, para todos $A, B \in \mathcal{P}(L)$:

(i) $A \subseteq C(A)$

(ii) $A \subseteq B \Rightarrow C(A) \subseteq C(B)$

(iii) $C(C(A)) \subseteq C(A)$.

Esta é uma versão de lógica abstrata introduzida por Alfred Tarski, na década de 1930. Assim, cada lógica abstrata é uma lógica de Tarski. Os sistemas lógicos contemporâneos portam outros aspectos, e alguns serão evidenciados na sequência, contudo, esse caráter universal pode auxiliar num olhar geral sobre lógicas.

Definição 4.2: Seja $\boldsymbol{L}$ uma lógica abstrata. O conjunto A é fechado em $\boldsymbol{L}$ se $C(A) = A$, e A é aberto, quando o seu complemento relativo à L, denotado por A^C, é fechado em $\boldsymbol{L}$.

Definição 4.3: O operator de consequência C é finitário se, para todo $A \subseteq L$, tem-se:

$$C(A) = \cup\{C(A_f) : A_f \subseteq A \text{ e } A_f \text{ é finito}\}.$$

Agora, podemos dar a definição de tradução entre lógicas, como em da Silva, D'Ottaviano e Sette (1999).

Definição 4.4: Tradução da lógica abstrata $\boldsymbol{L}_1$ na lógica $\boldsymbol{L}_2$ é uma função t: $\boldsymbol{L}_1 \to \boldsymbol{L}_2$ tal que, para todo $\mathrm{A}\cup\{x\} \subseteq \mathrm{L}_1$:

$$x \in C_1(A) \Rightarrow t(x) \in C_2\,(t(A)).$$

A seguir, estão algumas definições e resultados sobre traduções entre lógicas. Detalhes e demonstrações podem ser encontrados em Feitosa e D'Ottaviano (2001).

Proposição 4.5: Uma função t: $\boldsymbol{L}_1 \to \boldsymbol{L}_2$ é tradução se, e somente se, para todo $A \subseteq L_1$, se tem: $t(C_1(A)) \subseteq C_2(t(A))$.

Proposição 4.6: A composição de traduções é uma tradução; a função identidade entre lógicas abstratas é tradução; a composição de traduções é associativa; a função identidade é o elemento neutro para a composição de traduções.

Proposição 4.7: Se t: $\boldsymbol{L}_1 \rightarrow \boldsymbol{L}_2$ é uma função entre lógicas, então as seguintes condições são equivalentes:

(i) t é tradução;

(ii) a imagem inversa de cada conjunto fechado de $\boldsymbol{L}_2$ é um conjunto fechado de $\boldsymbol{L}_1$;

(iii) a imagem inversa de cada conjunto aberto de $\boldsymbol{L}_2$ é um conjunto aberto de $\boldsymbol{L}_1$

(iv) para todo $B \subseteq L_2$, tem-se: $C_1(t^{-1}(B)) \subseteq t^{-1}(C_2(B))$.

Definição 4.8: Duas lógicas $\boldsymbol{L}_1$ e $\boldsymbol{L}_2$ são L-isomorfas, quando existe uma função bijetiva t: $L_1 \rightarrow L_2$, tal que t e t^{-1} são traduções. A função t é um L-homeomorfismo.

Proposição 4.9: Se t: $\boldsymbol{L}_1 \rightarrow \boldsymbol{L}_2$ é uma função bijetiva, então, a função t é um L-homeomorfismo se, e somente se, para todo $A \subseteq L_1$, se tem: $t(C_1(A)) = C_2(t(A))$.

Essa definição de lógica abstrata é bastante geral. Se desejarmos torná-la um pouco mais semelhante aos sistemas lógicos usuais, podemos impor condições adicionais sobre ela. É mais usual considerar que os sistemas lógicos são pares ordenados $\mathcal{L} = \langle L, C \rangle$, em que L é uma linguagem formal e C é um operador sobre o conjunto das fórmulas de L, denotado por $For(L)$, como na Definição 2.1.

Definição 4.10: Sistema lógico sobre a linguagem L é um par $\mathcal{L} = \langle L, C \rangle$, no qual L é uma linguagem formal e C é um operador-padrão sobre a álgebra das fórmulas de L.

Definição 4.11: Sejam L uma linguagem formal, C um operador de consequência sobre $For(L)$ e s um endomorfismo (homomorfismo de um conjunto sobre ele mesmo) sobre $For(L)$, isto é, $s \in Hom(For(L), For(L))$. O operador C é *estrutural* se, para todo $\Gamma \subseteq For(L)$ e todo endomorfismo s sobre $For(L)$, $s(C(\Gamma)) \subseteq C(s(\Gamma))$. O operador C é *padrão* (*standard*), quando C é estrutural e finitário.

O endomorfismo s deve ser entendido como uma substituição sobre fórmulas de L (cf. MENDELSON, 1964).

Se $\mathcal{L}_1$ e $\mathcal{L}_2$ são dois sistemas lógicos com as respectivas relações de consequência sintáticas $\vdash_1$ e $\vdash_2$, então, uma tradução entre lógicas pode ser descrita do seguinte modo: t é uma tradução de $\mathcal{L}_1$ em $\mathcal{L}_2$ se, para $\Gamma \cup \{\varphi\} \subseteq For(\mathcal{L}_1)$, tem-se:

$$\Gamma \vdash_1 \varphi \Rightarrow t(\Gamma) \vdash_2 t(\varphi).$$

Trataremos, a seguir, com três lógicas com apresentação formal constituídas sobre específicas linguagens artificiais e conhecidas na literatura, a LPT, a $Ł_3$ de Łukasiewicz e a lógica proposicional clássica (LPC).

Agora, estendemos o conceito de tradução para tradução conservativa. Esse conceito foi introduzido em Feitosa (1997); detalhes e demonstrações dos resultados indicados podem ser encontrados em Feitosa e D'Ottaviano (2001).

Definição 4.12: Se $\boldsymbol{L}_1$ e $\boldsymbol{L}_2$ são lógicas abstratas, então, uma tradução conservativa de $\boldsymbol{L}_1$ em $\boldsymbol{L}_2$ é qualquer função t: $L_1 \rightarrow L_2$ tal que, para todo conjunto $A \cup \{x\} \subseteq L_1$, se tem:

$$x \in C_1(A) \Leftrightarrow t(x) \in C_2(t(A)).$$

A expressão conservativa procura destacar a implicação da direita para a esquerda da expressão acima, enquanto a outra direção está dada por ser tradução.

Definição 4.13: *Aplicação conservativa* da lógica $\boldsymbol{L}_1$ na lógica $\boldsymbol{L}_2$ é uma função t: $L_1 \rightarrow L_2$ tal que, para todo $x \in L_1$:

$$x \in C_1(\varnothing) \Leftrightarrow t(x) \in C_2(\varnothing).$$

Uma aplicação conservativa preserva e conserva pontos do espaço, mas não garante preservar e conservar a dedutibilidade das lógicas envolvidas.

Para os sistemas lógicos $\mathcal{L}_1$ e $\mathcal{L}_2$, uma tradução conservativa é uma função t: $For(\mathcal{L}_1) \rightarrow For(\mathcal{L}_2)$ de maneira que, para todo $\Gamma \cup \{\varphi\} \subseteq \mathrm{For}(\mathcal{L}_1)$:

$$\Gamma \vdash_1 \varphi \Leftrightarrow t(\Gamma) \vdash_2 t(\varphi).$$

Uma aplicação conservativa é tal que, para todo $\varphi \in For(\mathcal{L}_1)$:

$$\vdash_1 \varphi \Leftrightarrow \; \vdash_2 t(\varphi).$$

Seguem mais alguns resultados sobre traduções conservativas.

Proposição 4.14: Seja t: $\boldsymbol{L}_1 \rightarrow \boldsymbol{L}_2$ uma função tal que, para todo $A \subseteq L_1$, se tem que $C_2(t(A)) \subseteq Im(t)$. Se t é uma tradução conservativa, então, $t(C_1(A)) = C_2(t(A))$.

Proposição 4.15: Seja t: $\boldsymbol{L}_1 \rightarrow \boldsymbol{L}_2$ uma função sobrejetiva. Se t é tradução conservativa, então, para todo $A \subseteq L_1$, vale $t(C_1(A)) = C_2(t(A))$.

Proposição 4.16: Seja t: $\boldsymbol{L}_1 \rightarrow \boldsymbol{L}_2$ uma função bijetiva. Então, t é tradução conservativa se, e somente se, para todo $A \subseteq L_1$, vale $t(C_1(A)) = C_2(t(A))$.

Como cada L-homeomorfismo é uma tradução bijetiva, tal que sua inversa também é tradução, cada L-homeomorfismo é uma tradução conservativa. Contudo, muitas traduções conservativas não são exemplos de L-homeomorfismos.

Este último resultado indica condição necessária e suficiente para uma função entre lógicas abstratas ser uma tradução conservativa.

Teorema 4.17: Uma tradução t: $\boldsymbol{L}_1 \rightarrow \boldsymbol{L}_2$ é conservativa se, e somente se, para todo $A \subseteq L_1$, se tem $t^{-1}(C_2(t(A))) \subseteq C_1(A)$.

Proposição 4.18: A composição de traduções conservativas é uma tradução conservativa; a função identidade entre lógicas abstratas é tradução conservativa; a composição de traduções conservativas é associativa; a função identidade é o elemento neutro para a composição de traduções conservativas.

5. Traduções conservativas com LPT

Nesta seção, introduzimos duas traduções conservativas envolvendo LPT. Primeiro, definimos uma tradução conservativa da lógica proposicional clássica (LPC) em LPT. Depois, apresentamos uma tradução conservativa de LPT em $Ł_3$, o sistema lógico trivalente de Łukasiewicz.

Essas traduções, além de caracterizarem exemplos de traduções conservativas, as quais, por não serem fáceis de serem encontradas, já teriam algum valor intrínseco nelas mesmas, também permitem a obtenção de metapropriedades da lógica LPT, o que aponta para possíveis aplicações das traduções entre lógicas.

5.1 Uma tradução da LPC na LPT

Para simplificar e separar a notação, indicaremos o conjunto das variáveis proposicionais da LPC por *Var*(LPC) = $\{p_1, p_2, p_3, ...\}$ e o conjunto das variáveis proposicionais de LPT por *Var*(LPT) = $\{s_1, s_2, s_3, ...\}$.

Agora, podemos definir a função pretendida tradução τ:

$$\tau\colon For(\text{LPC}) \to For(\text{LPT})$$

$$\tau(p_i) \stackrel{\text{def}}{=} s_i$$

$$\tau(\neg\varphi) \stackrel{\text{def}}{=} \sim \tau(\varphi)$$

$$\tau(\varphi \to \psi) \stackrel{\text{def}}{=} \tau(\varphi) \to \tau(\psi)$$

Lema 5.1.1: Dada uma valoração booleana *e* para a LPC, a função τ induz uma LPT-valoração ω_e tal que:

$$e(\sigma) = 0 \Leftrightarrow \omega_e(\tau(\sigma)) = 0.$$

Demonstração: Se *e* é uma valoração booleana para *For*(LPC), então, define-se uma valoração ω_e para LPT do seguinte modo:

Para cada variável $p_i \in Var(\text{LPC})$, a função τ é tal que $\tau(p_i) = s_i$, com $s_i \in Var(\text{LPT})$. Então, seja:

(i) $\omega_e(\tau(p_i)) = \omega_e(s_i) = 0 \Leftrightarrow e(p_i) = 0$;

(ii) $\omega_e(\sim \tau(\varphi)) = 0 \Leftrightarrow \omega_e(\tau(\varphi)) \neq 0$, isto é, $\tau(\varphi) \in \{½, 1\}$;

(iii) $\omega_e(\tau(\varphi) \rightarrow \tau(\psi)) = 0 \Leftrightarrow \omega_e(\tau(\varphi)) \neq 0$ e $\omega_e(\tau(\psi)) = 0$.

A demonstração segue por indução sobre a complexidade de σ.

Se σ é uma variável p_i, então, por (i), $\omega_e(\tau(\sigma)) = \omega_e(\tau(p_i)) = \omega_e(s_i) = 0 \Leftrightarrow e(p_i) = e(\sigma) = 0$.

Se σ é do tipo $\neg\varphi$, então $\tau(\sigma) = \tau(\neg\varphi) = \sim \tau(\varphi)$. Pela hipótese de indução, $\omega_e(\tau(\varphi)) = 0 \Leftrightarrow e(\varphi) = 0$ e, por (ii), $\omega_e(\tau(\varphi)) \neq 0 \Leftrightarrow \omega_e(\sim\tau(\gamma)) = 0$. Daí, $\omega_e(\tau(\sigma)) = 0 \Leftrightarrow \omega_e(\tau(\neg\varphi)) = 0 \Leftrightarrow \omega_e(\sim\tau(\varphi)) = 0 \Leftrightarrow \omega_e(\tau(\varphi)) \neq 0 \Leftrightarrow e(\varphi) = 1 \Leftrightarrow e(\neg\varphi) = 0 \Leftrightarrow \mathrm{e}(\sigma) = 0$.

Se σ é do tipo $\varphi \rightarrow \phi$, então $\tau(\sigma) = \tau(\varphi \rightarrow \psi) = \tau(\varphi) \rightarrow \tau(\psi)$. Pela hipótese de indução, $\omega_e(\tau(\varphi)) = 0 \Leftrightarrow e(\varphi) = 0$ e $\omega_e(\tau(\psi)) = 0 \Leftrightarrow e(\psi) = 0$. Assim, $\omega_e(\tau(\sigma)) = 0 \Leftrightarrow \omega_e(\tau(\varphi) \rightarrow \tau(\psi)) = 0 \Leftrightarrow$, por (iii), $\omega_e(\tau(\varphi)) \neq 0$ e $\omega_e(\tau(\psi)) = 0 \Leftrightarrow e(\varphi) = 1$ e $e(\psi) = 0 \Leftrightarrow e(\varphi \rightarrow \psi) = 0 \Leftrightarrow e(\sigma) = 0$.

Lema 5.1.2: Dada uma valoração ω para a lógica LPT, a função τ induz uma valoração e_ω para a LPC tal que:

$$e_\omega(\sigma) = 0 \Leftrightarrow \omega(\tau(\sigma)) = 0.$$

Demonstração: Se ω é uma valoração para LPT, então, ω: *For*(LPT) $\rightarrow \{0, ½, 1\}$. A partir de ω, define-se uma valoração booleana e_ω do seguinte modo:

$$e_\omega(p_i) = 0 \Leftrightarrow \omega(s_i) = 0.$$

A demonstração segue por indução sobre a complexidade de σ.

Se σ é uma variável p_i, então, $\omega(\tau(\sigma)) = \omega(\tau(p_i)) = \omega(s_i) = 0 \Leftrightarrow e_\omega(p_i) = e_\omega(\sigma) = 0$.

Se σ é do tipo $\neg\varphi$, então, $\tau(\sigma) = \tau(\neg\varphi) = \sim \tau(\varphi)$. Pela hipótese de indução, $\omega(\tau(\varphi)) = 0 \Leftrightarrow e_\omega(\varphi) = 0$. Agora, $\omega(\tau(\sigma)) = 0 \Leftrightarrow \omega(\tau(\neg\varphi)) = 0 \Leftrightarrow \omega(\sim \tau(\varphi)) = 0 \Leftrightarrow \omega(\tau(\varphi)) \neq 0 \Leftrightarrow e_\omega(\varphi) = 1 \Leftrightarrow e_\omega(\neg\varphi) = 0 \Leftrightarrow e_\omega(\sigma) = 0$.

Se σ é do tipo $\varphi \rightarrow \psi$, então, $\tau(\sigma) = \tau(\varphi \rightarrow \psi) = \tau(\varphi) \rightarrow \tau(\psi)$. Pela hipótese de indução, $\omega(\tau(\varphi)) = 0 \Leftrightarrow e_\omega(\varphi) = 0$ e $\omega(\tau(\psi)) = 0 \Leftrightarrow e_\omega(\psi) = 0$. Dessa maneira, $\omega(\tau(\sigma)) = 0 \Leftrightarrow \omega(\tau(\varphi) \rightarrow \tau(\psi)) = 0 \Leftrightarrow \omega(\tau(\varphi)) \neq 0$ e $\omega(\tau(\psi)) = 0 \Leftrightarrow e_\omega(\varphi) = 1$ e $e_\omega(\psi) = 0 \Leftrightarrow e_\omega(\sigma) = 0$.

Uma vez que cada valoração booleana e cada valoração ω: *For*(LPT) $\to \{0, \frac{1}{2}, 1\}$ são modelos fortemente adequados para as lógicas LPC e LPT, podemos enunciar e demonstrar o próximo teorema.

Teorema 5.1.3: A função τ é uma tradução conservativa da LPC em LPT.

Demonstração: Precisamos mostrar que, para todo $\Gamma \cup \{\varphi\} \subseteq$ *For*(LPC), se tem:

$$\Gamma \vDash \varphi \Leftrightarrow \tau(\Gamma) \vDash \tau(\varphi).$$

($\Rightarrow$) Se $\tau(\Gamma) \nvDash \tau(\varphi)$, então, existe uma valoração ω para LPT tal que, para toda fórmula $\gamma \in \Gamma$, se tem $\omega(\tau(\gamma)) \neq 0$, mas $\omega(\tau(\varphi)) = 0$. Pelo Lema 5.1.2, existe uma valoração booleana e_ω, tal que $e_\omega(\gamma) = 1$ e $e_\omega(\varphi) = 0$, ou seja, $\Gamma \nvDash \varphi$.

($\Leftarrow$) Se $\Gamma \nvDash \varphi$, então, existe uma valoração booleana e tal que $e(\gamma) = 1$, para toda $\gamma \in \Gamma$, mas $e(\varphi) = 0$. Pelo Lema 5.1.1, existe uma valoração ω_e para LPT tal que, para toda $\gamma \in \Gamma$, se tem $\omega_e(\tau(\gamma)) \neq 0$ e $\omega_e(\tau(\varphi)) = 0$, isto é, $\omega_e \vDash \tau(\Gamma)$ e $\omega_e \nvDash \tau(\varphi)$. Finalmente, $\tau(\Gamma) \nvDash \tau(\varphi)$.

O conjunto de teoremas ou fórmulas válidas de LPT é menor que o conjunto de teoremas (tautologias) do CPC. Contudo, há uma parte booleana dentro do conjunto de fórmulas válidas da LPT, em que podemos mergulhar o conjunto das tautologias clássicas.

5.2 Uma tradução de LPT em Ł$_3$

Não é fácil definir uma tradução conservativa de LPT ou outras lógicas não clássicas em LPC. Conseguimos definir uma de LPT em outra lógica trivalente, qual seja, a lógica Ł$_3$ de Łukasiewicz. Para isso, precisamos apresentar essa lógica.

Łukasiewicz introduziu seus sistemas lógicos proposicionais multivalorados sobre a linguagem proposicional $L(\neg, \to)$ com dois operadores básicos, '$\neg$' para negação, e '$\to$' para a implicação.

Se $x, y \in [0, 1]$, o intervalo real unitário, então, esses operadores são definidos por:

$$\neg x \stackrel{\text{def}}{=} 1 - x \ \text{ e}$$

$$x \to y \stackrel{\text{def}}{=} min(1, 1 - x + y).$$

Se considerarmos o modelo usual de matrizes da LPC, o qual toma exatamente o subconjunto $\{0, 1\} \subseteq [0, 1]$, os dois operadores acima coincidem com os seus correlatos clássicos da LPC.

As matrizes para a lógica Ł$_3$ de Łukasiewicz têm seus valores dados no conjunto $\{0, ½, 1\} \subseteq [0, 1]$.

De acordo com os operadores acima, as tabelas dos operadores de negação e implicação de Ł$_3$ são as seguintes:

$\to$	0	½	1
0	1	1	1
½	½	1	1
1	0	½	1

	$\neg$
0	1
½	½
1	0

A semântica matricial de Ł$_3$ é dada pela matriz:

$$\mathcal{M}_{Ł3} = (\{0, ½, 1\}, \neg, \to, \{1\}),$$

com o conjunto de valores designados $D = \{1\}$.

Conforme Cignoli, D'Ottaviano e Mundici (1994), nas lógicas de Łukasiewicz, podem ser definidas as seguintes operações: disjunção $\vee$, conjunção $\wedge$, soma $\oplus$, produto $\odot$ e a negação clássica $\sim$ do seguinte modo:

$$x \vee y \stackrel{\text{def}}{=} (x \to y) \to \text{y} = max(x, y)$$

$$x \wedge y \stackrel{\text{def}}{=} \neg(\neg x \vee \neg y) = min(x, y)$$

$$x \oplus y \stackrel{\text{def}}{=} \neg x \to y = min(1, x{+}y)$$

$$x \odot y \stackrel{\text{def}}{=} \neg(\neg x \oplus \neg y) = max(0, x + y - 1)$$

$\sim x \overset{\text{def}}{=} \neg x \oplus \neg x$

$x \rightsquigarrow y \overset{\text{def}}{=} x \to (x \to y)$.

As tabelas da disjunção e conjunção coincidem com as de LPT, enquanto as tabelas dos demais operadores são as seguintes:

$\oplus$	0	½	1
0	0	½	1
½	½	1	1
1	1	1	1

$\odot$	0	½	1
0	0	0	0
½	0	½	½
1	0	½	1

	$\sim$
0	1
½	1
1	0

$\rightsquigarrow$	0	½	1
0	0	½	1
½	1	1	1
1	1	1	1

Considerando $Var(Ł_3) = \{q_1, q_2, q_3, ...\}$, a nossa tradução é a função:

$$\rho: \text{LPT} \to Ł_3$$
$$\rho(s_i) = \sim \neg\, q_i$$
$$\rho(\neg\, \varphi) = \sim \rho(\varphi)$$
$$\rho(\varphi \to \psi) = \rho\,(\varphi) \to \rho(\psi).$$

Assim, perante essa função, segue que $w(\rho(\Delta)) \subseteq \{0, 1\}$, para todo $\Delta \subseteq For(\text{LPT})$.

Lema 5.2.1: Para cada valoração v: $Var(\text{LPT}) \to \{0, ½, 1\}$, existe uma valoração w: $Var(Ł_3) \to \{0, ½, 1\}$, tal que:

$$v(\sigma) = 0 \Leftrightarrow w(\rho(\sigma)) = 0.$$

Demonstração: Tomamos $w(q_i) = v(s_i)$. A demonstração segue por indução sobre a complexidade de σ.

Se σ é s_i, então, $v(\sigma) = 0 \Leftrightarrow v(s_i) = 0 \Leftrightarrow w(q_i) = 0 \Leftrightarrow w(\neg q_i) = 1 \Leftrightarrow w(\sim \neg q_i) = 0 \Leftrightarrow w(\rho(\sigma)) = 0$.

Se σ é $\neg\psi$, então, $v(\sigma) = 0 \Leftrightarrow v(\psi) = 1 \Leftrightarrow w(\rho(\psi)) = 1 \Leftrightarrow w(\sim \rho(\psi)) = 0 \Leftrightarrow w(\rho(\neg\psi)) = 0 \Leftrightarrow w(\rho(\sigma)) = 0$.

Se σ é $\psi \to \gamma$, então, $v(\sigma) = 0 \Leftrightarrow v(\psi) \neq 0$ e $v(\gamma) = 0$ see $w(\rho(\psi)) = 1$ e $w(\rho(\gamma)) = 0$ see $w(\rho(\psi) \to \rho(\gamma)) = 0$ see $w(\rho(\sigma)) = 0$. ■

Proposição 5.2.2: A função ρ é uma tradução conservativa.

Demonstração: Dada a completude dos modelos matriciais, $\Gamma \nvdash \varphi$ see $\Gamma \nvDash \varphi$ see existe alguma valoração v: $Var(\text{LPT}) \to \{0, ½, 1\}$ tal que $v(\gamma) \neq 0$, para toda $\gamma \in \Gamma$, e $v(\varphi) = 0$ see existe alguma valoração w: $Var(\text{Ł}_3) \to \{0, ½, 1\}$ tal que $w(\rho(\gamma)) = 1$, para toda $\gamma \in \Gamma$, e $w(\rho(\varphi)) \neq 1$ see $\rho(\Gamma) \nvdash \rho(\varphi)$.

6. Considerações finais

Com as traduções conservativas, podemos verificar a validade de certas propriedades de uma lógica em função de outra, pois a definição de tradução conservativa pressupõe uma relação de reciprocidade entre as lógicas envolvidas na respectiva tradução. As primeiras traduções, encontradas na literatura (cf. FEITOSA; D'OTTAVIANO, 2001), foram usadas para a demonstração da consistência relativa de um sistema segundo o outro.

Faremos o mesmo agora.

Por exemplo, como sabemos que Ł_3 é consistente, obtemos a consistência de LPT do seguinte modo.

Suponhamos que LPT não seja consistente. Então, $\vdash_{LPT} \bot \Leftrightarrow \vdash_{LPT} \neg(\varphi \to \varphi)$. Pela tradução ρ, segue que $\vdash_{\text{Ł}3} \rho(\neg(\varphi \to \varphi)) \Leftrightarrow \vdash_{\text{Ł}3} {\sim}(\rho(\varphi) \to \rho(\varphi)) \Leftrightarrow \vdash_{\text{Ł}3} \bot$. Todavia, Ł_3 seria inconsistente.

Outra aplicação de uma tradução pode ser considerada para a demonstração da decidibilidade de LPT. Como Ł_3 é decidível, então também LPT é decidível, porque, dada uma fórmula φ de LPT, pela tradução ρ, toma-se $\rho(\varphi)$ e se testa essa fórmula em Ł_3. Se $\rho(\varphi)$ é um teorema de Ł_3, então φ é teorema de LPT. Se $\rho(\varphi)$ não é teorema de $\rho(\varphi)$, então também φ não é teorema de LPT.

Desse modo, com o uso das traduções, conseguimos demonstrar novas e relevantes metapropriedades de LPT. Mais inter-relações podem ser verificadas através das funções de tradução.

Agradecimentos

Agradecemos ao apoio da FAPESP e do CNPq.

Referências

BUENO, O.; DA COSTA, N. C. A. Quasi-truth, paraconsistency, and the foundations of science. *Synthese*, v. 154, p. 383-399, 2007.

CARNIELLI, W. A.; MARCOS, J.; DE AMO, S. Formal inconsistency and evolutionary databases. *Logic and Logical Philosophy*, v. 8, p. 115-152, 2000.

CIGNOLI, R. L. O.; D'OTTAVIANO, I. M. L.; MUNDICI, D. Álgebras das lógicas de Łukasiewicz. Campinas: UNICAMP, Centro de Lógica, Epistemologia e História da Ciência, 1994. (Coleção CLE, v. 12).

CONIGLIO, M. E.; SILVESTRINI, L. H. C. An alternative approach for quasi-truth. *Logic Journal of IGPL*, v. 22, p. 387-410, 2014.

DA COSTA, N. C. A.; FRENCH, S. *Science and partial truth:* a unitary approach to models and scientific reasoning. Oxford: Oxford University Press, 2003.

DA SILVA, J. J.; D'OTTAVIANO, I. M. L.; SETTE, A. M. Translations between logics. In: CAICEDO, X.; MONTENEGRO, C. H. (Ed.). *Models, Algebras and Proofs*, v. 203. New York: Marcel Dekker, 1999. p. 435-448. (Lectures Notes in Pure and Applied Mathematics).

ENDERTON, H. B. *A mathematical introduction to logic*. San Diego: Academic, 1972.

EPSTEIN, R. L. *The semantic foundations of logic*. Volume 1: propositional logics. Dordrecht: Kluwer Academic, 1990.

FEITOSA, H. A. *Traduções conservativas*. 1997. Tese (Doutorado) – IFCH, UNICAMP, Campinas,1997.

FEITOSA, H. A.; D'OTTAVIANO, I. M. L. Conservative translations. *Annals of Pure and Applied Logic*, v. 108, n. 1-3, p. 205-227, 2001.

FEITOSA, H. A.; PAULOVICH, L. *Um prelúdio à lógica*. São Paulo: Editora UNESP, 2005.

HIFUME, C. *Uma teoria da verdade pragmática:* a quase-verdade de Newton C. A. da Costa. 2003. Dissertação (Mestrado) – IFCH, UNICAMP, Campinas, 2003.

KRAUSE, D. *Introdução aos fundamentos axiomáticos da ciência*. São Paulo: EPU, 2002.

MALINOWSKI, G. *Many-valued logics*. Oxford: Clarendon, 1993.

MENDELSON, E. *Introduction to mathematical logic*. Princeton: D. Van Nostrand, 1964.

MIKENBERG, I.; DA COSTA, N. C. A.; CHUAQUI, R. Pragmatic truth and approximation to truth. *The Journal of Symbolic Logic*, v. 51, n. 1, p. 201-221, 1986.

SILVESTRINI, L. H. *Uma nova abordagem para a noção de quase-verdade*. 2011. Tese (Doutorado) – IFCH, UNICAMP, Campinas, 2011.