Monadic Predicate Łukasiewicz Logic. Standard versus General Tautologies

Félix Bou

University of Barcelona (UB)
bou@ub.edu

May 20th
ASUV 2011 (Salerno)

Classical Predicate Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge$) + vocabulary ϑ

Classical Predicate Logic

- Syntax: Primitive symbols $(\forall, \exists, \neg, \vee, \wedge)$ + vocabulary ϑ
- Examples of Sentences: $\forall x P x$, $\forall x \exists y R x y$, etc.

Classical Predicate Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge$) + vocabulary ϑ
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.
- Semantics: 2-valued (for every structure, we get a mapping from sentences into $\{0, 1\}$).

Classical Predicate Logic

- Syntax: Primitive symbols $(\forall, \exists, \neg, \vee, \wedge)$ + vocabulary ϑ
- Examples of Sentences: $\forall x P x$, $\forall x \exists y R x y$, etc.
- Semantics: 2-valued (for every structure, we get a mapping from sentences into $\{0, 1\}$).

Full vocabulary

- Valid sentences are recursively enumerable (Gödel)
- Undecidability of valid sentences (Church, ...)
- FO^2 is decidable: "effective fmp" holds (Scott, Mortimer)
- FO^3 is undecidable (Surányi, ...)

Classical Predicate Logic

- Syntax: Primitive symbols $(\forall, \exists, \neg, \vee, \wedge)$ + vocabulary ϑ
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.
- Semantics: 2-valued (for every structure, we get a mapping from sentences into $\{0, 1\}$).

Full vocabulary

- Valid sentences are recursively enumerable (Gödel)
- Undecidability of valid sentences (Church, ...)
- FO^2 is decidable: "effective fmp" holds (Scott, Mortimer)
- FO^3 is undecidable (Surányi, ...)

Monadic vocabulary: $P_1, P_2, P_3, \ldots$

- Decidability of valid sentences: filtration method provides an "effective fmp" (Löwenheim).

Predicate Łukasiewicz (standard) Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ

Predicate Łukasiewicz (standard) Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.

Predicate Łukasiewicz (standard) Logic

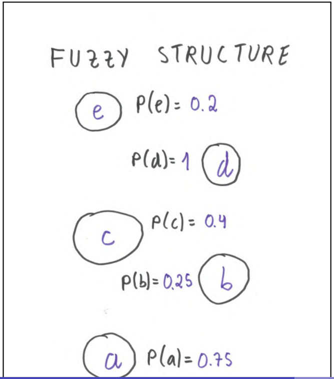
- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.

Predicate Łukasiewicz (standard) Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.
- Examples of Sentences: $\forall x P x$, $\forall x \exists y R x y$, etc.
- Semantics: $[0, 1]$-valued (for every structure, we get a mapping from sentences into $[0, 1]$).

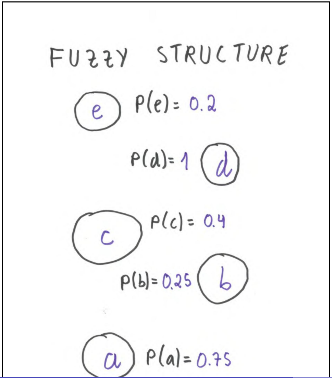
Predicate Łukasiewicz (standard) Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.
- Semantics: $[0,1]$-valued (for every structure, we get a mapping from sentences into $[0,1]$).



Predicate Łukasiewicz (standard) Logic

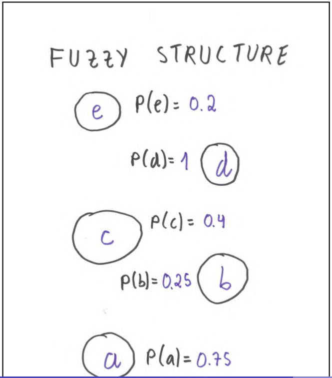
- Syntax: Primitive symbols $(\forall, \exists, \neg, \vee, \wedge, \rightarrow)$ + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.
- Semantics: $[0, 1]$-valued (for every structure, we get a mapping from sentences into $[0, 1]$).



$\forall x Px =$ $\qquad$ $\exists x(Px \vee \neg Px) =$

Predicate Łukasiewicz (standard) Logic

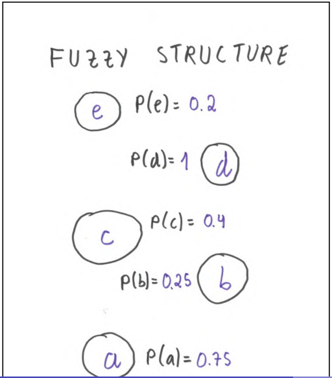
- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.
- Semantics: $[0, 1]$-valued (for every structure, we get a mapping from sentences into $[0, 1]$).



$\forall x Px = 0.2 \qquad \exists x (Px \vee \neg Px) =$

Predicate Łukasiewicz (standard) Logic

- Syntax: Primitive symbols ($\forall, \exists, \neg, \vee, \wedge, \rightarrow$) + vocabulary ϑ
- Definable symbols: $\odot, \oplus, \ominus$, *etc*.
- Examples of Sentences: $\forall x Px$, $\forall x \exists y Rxy$, etc.
- Semantics: $[0,1]$-valued (for every structure, we get a mapping from sentences into $[0,1]$).



$\forall x Px = 0.2 \qquad \exists x (Px \vee \neg Px) = 1$

Three semantics using MV-chains

- Standard (*stL*$\forall$): $[0, 1]$-valued
- General (*genL*$\forall$): **A**-valued (where **A** is an arbitrary MV-chain) structures requiring "safeness" condition (all formulas in ϑ have a truth value).
- Supersound (*spsL*$\forall$): **A**-valued (where **A** is an arbitrary MV-chain) structures only requiring the existence of the value of your formula.

Some Trivial Remarks

Three semantics using MV-chains

- Standard ($stL\forall$): $[0,1]$-valued
- General ($genL\forall$): $\mathbf{A}$-valued (where $\mathbf{A}$ is an arbitrary MV-chain) structures requiring "safeness" condition (all formulas in ϑ have a truth value).
- Supersound ($spsL\forall$): $\mathbf{A}$-valued (where $\mathbf{A}$ is an arbitrary MV-chain) structures only requiring the existence of the value of your formula.

Some Trivial Remarks

- $spsL\forall \subseteq genL\forall \subseteq stL\forall$

Three semantics using MV-chains

- Standard (*stL*∀): $[0,1]$-valued
- General (*genL*∀): **A**-valued (where **A** is an arbitrary MV-chain) structures requiring "safeness" condition (all formulas in ϑ have a truth value).
- Supersound (*spsL*∀): **A**-valued (where **A** is an arbitrary MV-chain) structures only requiring the existence of the value of your formula.

Some Trivial Remarks

- *spsL*∀ $\subseteq$ *genL*∀ $\subseteq$ *stL*∀
- Safeness holds when **A** is complete (e.g., **A** finite).

Three semantics using MV-chains

- Standard ($stL\forall$): $[0,1]$-valued
- General ($genL\forall$): **A**-valued (where **A** is an arbitrary MV-chain) structures requiring "safeness" condition (all formulas in ϑ have a truth value).
- Supersound ($spsL\forall$): **A**-valued (where **A** is an arbitrary MV-chain) structures only requiring the existence of the value of your formula.

Some Trivial Remarks

- $spsL\forall \subseteq genL\forall \subseteq stL\forall$
- Safeness holds when **A** is complete (e.g., **A** finite).
- Safeness holds in the following cases: finite structures, structures where the range of vocabulary symbols is finite ("secure"), witnessed structures.

Full vocabulary

- *genL*$\forall$ is Σ_1-complete (Chang, Belluce)
- *stL*$\forall$ is not in Σ_1 (Scarpellini)
- *stL*$\forall$ is Π_2-complete (Ragaz)
- *stL*$\forall$ $= \bigcap_{n\in\omega} \mathit{Taut}(L_n) = \mathit{Taut}([0,1]\cap\mathbb{Q})$. (Rutledge)
- General and standard semantics are complete for witnessed structures (Hájek)

Full vocabulary

- *genL*$\forall$ is Σ_1-complete (Chang, Belluce)
- *stL*$\forall$ is not in Σ_1 (Scarpellini)
- *stL*$\forall$ is Π_2-complete (Ragaz)
- $\mathit{stL}\forall = \bigcap_{n\in\omega} \mathit{Taut}(L_n) = \mathit{Taut}([0,1]\cap\mathbb{Q})$. (Rutledge)
- General and standard semantics are complete for witnessed structures (Hájek)

Monadic vocabulary: $P_1, P_2, P_3, \ldots$

- *stL*$\forall$ is in Π_1. Filtration method shows fmp (i.e., if $\varphi \notin \mathit{stL}\forall$ then it is not valid in some finite $[0,1]$-structure) (Hájek)
- standard and general semantics coincide for FO^1, and it is decidable (Rutledge)
- standard and general semantics coincide for “classical formulas” (i.e., $\forall, \exists, \neg, \wedge, \vee$), and this fragment is decidable.

- From Scarpellini's (and Ragaz) result it follows that there are sentences which are standard tautologies while not general tautologies, but his proof do not provide us any explicit example.

- From Scarpellini's (and Ragaz) result it follows that there are sentences which are standard tautologies while not general tautologies, but his proof do not provide us any explicit example.
- Only very recently an explicit formula with this property has been given (by Hájek), but it requires quite a lot of machinery.

- From Scarpellini's (and Ragaz) result it follows that there are sentences which are standard tautologies while not general tautologies, but his proof do not provide us any explicit example.
- Only very recently an explicit formula with this property has been given (by Hájek), but it requires quite a lot of machinery.

A problem for this talk

Is there some "simple" sentence which is a standard Łukasiewicz tautology but not a general Łukasiewicz tautology? Can we write one?

- From Scarpellini's (and Ragaz) result it follows that there are sentences which are standard tautologies while not general tautologies, but his proof do not provide us any explicit example.
- Only very recently an explicit formula with this property has been given (by Hájek), but it requires quite a lot of machinery.

A problem for this talk

Is there some "simple" sentence which is a standard Łukasiewicz tautology but not a general Łukasiewicz tautology? Can we write one?

For the *BL* case, $\forall x(Px \odot Px) \rightarrow (\forall x Px \odot \forall x Px)$ does it.

- From Scarpellini's (and Ragaz) result it follows that there are sentences which are standard tautologies while not general tautologies, but his proof do not provide us any explicit example.
- Only very recently an explicit formula with this property has been given (by Hájek), but it requires quite a lot of machinery.

A problem for this talk

Is there some "simple" sentence which is a standard Łukasiewicz tautology but not a general Łukasiewicz tautology? Can we write one?

For the *BL* case, $\forall x(Px \odot Px) \to (\forall x Px \odot \forall x Px)$ does it.

Intuition: In the monadic case, general Łukasiewicz semantics behaves like arbitrary classical models, while standard Łukasiewicz semantics behaves like finite classical models.

A related problem

Is the monadic fragment of $stL\forall$ decidable?

A related problem

Is the monadic fragment of $stL\forall$ decidable?

Next, we will sketch a proof that this fragment is Π_1-complete. We stress that we will not need a countable number of monadic predicate symbols.

A related problem

Is the monadic fragment of $stL\forall$ decidable?

Next, we will sketch a proof that this fragment is Π_1-complete. We stress that we will not need a countable number of monadic predicate symbols.

But first of all, let me make some historical remarks about this problem.

6.2. Läßt sich in Satz 6.1 die Anzahl der Prädikate auf drei, zwei oder eins reduzieren? Falls nein, sind die entsprechenden Mengen entscheidbar?
Hierzu sei noch bemerkt, daß es bereits in $\mathscr{S}_1$ Sätze gibt, die nur unendliche Modelle haben, zum Beispiel $\exists x Px \wedge \forall x \exists y Py \lessdot Px$.
Über die Menge der allgemeingültigen Formeln der einstelligen uendlichwertigen Prädikatenlogik findet sich in [14] die falsche Behauptung, daß Rutledge in seiner Dissertation [13] ihre Entscheidbarkeit bewiesen habe. In Wirklichkeit handelt es sich dort um den „Monadischen Prädikatenkalkül", welcher einer Prädikatenlogik entspricht, in welcher nur über eine Variable quantifiziert werden kann. Die Frage ist meines Wissens weiterhin offen:

6.3. Ist die Menge der allgemeingültigen Sätze der einstelligen unendlichwertigen Prädikatenlogik entscheidbar? Wenn nein, ist sie Π_1-vollständig?

Zum Schluß sei noch bemerkt, daß die *Folgerungsrelation*, also die Menge aller Paare (α, β), wo jedes Modell von α auch ein Modell von β ist, schon in der unendlichwertigen Logik mit fünf einstelligen Prädikaten Π_2-vollständig ist. Der Beweis (s. [11, 4.9, S. 63ff.]) erfolgt durch eine Erweiterung der hier angewendeten Methode.

M. Ragaz, Archiv für Mathematische Logik, 1983 (23), 129-139

40 SL, 1998 (61), 35-47 *M. Baaz, P. Hájek, J. Krajíček, D. Švejda*

Let us mention in passing some false statements in the literature. As Ragaz mentions in [12], Scarpellini's claim saying that Rutledge has shown decidability of the set of 1-tautologies of the monadic Łukasiewicz predicate logic is false (since Rutledge's system allows only quantification of a single variable). Gottwald claims ([5], p. 232) that Rutledge [13] has shown the axiomatizability of the set of 1-tautologies of the monadic Łukasiewicz predicate calculus, and (p. 237) that Ragaz has shown its undecidability. It follows from the quotations above that both claims are false.

40 SL, 1998 (61), 35-47 *M. Baaz, P. Hájek, J. Krajíček, D. Švejda*

Let us mention in passing some false statements in the literature. As Ragaz mentions in [12], Scarpellini's claim saying that Rutledge has shown decidability of the set of 1-tautologies of the monadic Łukasiewicz predicate logic is false (since Rutledge's system allows only quantification of a single variable). Gottwald claims ([5], p. 232) that Rutledge [13] has shown the axiomatizability of the set of 1-tautologies of the monadic Łukasiewicz predicate calculus, and (p. 237) that Ragaz has shown its undecidability. It follows from the quotations above that both claims are false.

5. Monadic predicate t-norm logics F. Montagna, ARCH, 2005(44), 97-114

It is well-known that monadic classical predicate logic is decidable. In [BCF] it is shown that the monadic predicate Gödel logic over [0, 1] (i.e., the set of monadic predicate formulas which are valid in $[0,1]_G$) is undecidable. More generally, it makes sense to ask for which sets $\mathbf{C}$ of t-norm BL-algebras the set $Taut_M(\mathbf{C}\forall)$ of monadic predicate formulas valid in all algebras in $\mathbf{C}$ is decidable.

Theorem 5.1. *If* $\mathbf{C}$ *is not included in* $\{[0,1]_L, [0,1]_\Pi\}$, *then* $Taut_M(\mathbf{C}\forall)$ *is undecidable.*

Interpretability Method

- $\beta(x, y)$ is an arbitrary formula with just two free variables and which only involves unary predicate symbols (perhaps several)
- $Bival(\beta)$ is the sentence $\forall x \forall y(\beta(x, y) \vee \neg\beta(x, y))$.
- φ is a classical sentence (i.e., it only involves $\forall, \exists, \neg, \wedge, \vee$) which only uses a binary predicate symbol R.
- $\varphi(R|\beta)$ is the result of replacing all Rxy with $\beta(x, y)$.

Interpretability Method

- $\beta(x, y)$ is an arbitrary formula with just two free variables and which only involves unary predicate symbols (perhaps several)
- $Bival(\beta)$ is the sentence $\forall x \forall y(\beta(x, y) \vee \neg\beta(x, y))$.
- φ is a classical sentence (i.e., it only involves $\forall, \exists, \neg, \wedge, \vee$) which only uses a binary predicate symbol R.
- $\varphi(R|\beta)$ is the result of replacing all Rxy with $\beta(x, y)$.

Lemma (A Uniform Statement on β's)

1. If $\emptyset \models_{fin} \varphi$, then $2.(\neg Bival(\beta) \vee \varphi(R|\beta)) \in stL\forall$.
2. If $\emptyset \models \varphi$, then $2.(\neg Bival(\beta) \vee \varphi(R|\beta)) \in spsL\forall$.

Interpretability Method

- $\beta(x, y)$ is an arbitrary formula with just two free variables and which only involves unary predicate symbols (perhaps several)
- $Bival(\beta)$ is the sentence $\forall x \forall y(\beta(x, y) \vee \neg\beta(x, y))$.
- φ is a classical sentence (i.e., it only involves $\forall, \exists, \neg, \wedge, \vee$) which only uses a binary predicate symbol R.
- $\varphi(R|\beta)$ is the result of replacing all Rxy with $\beta(x, y)$.

Lemma (A Uniform Statement on β's)

1. If $\emptyset \models_{fin} \varphi$, then $2.(\neg Bival(\beta) \vee \varphi(R|\beta)) \in stL\forall$.
2. If $\emptyset \models \varphi$, then $2.(\neg Bival(\beta) \vee \varphi(R|\beta)) \in spsL\forall$.

Sketch of the Proof: Given any fuzzy **A**-structure M such that $[\![\beta(a, b)]\!]^M$ is never 0.5, then take the classical structure $M/2$ by

- $R(a, b) = 1$, if $[\![\beta(a, b)]\!]^M > 0.5$
- $R(a, b) = 0$, if $[\![\beta(a, b)]\!]^M < 0.5$

Interpretability Method

- $\beta(x, y)$ is an arbitrary formula with just two free variables and which only involves unary predicate symbols (perhaps several)
- $Bival(\beta)$ is the sentence $\forall x \forall y(\beta(x, y) \vee \neg\beta(x, y))$.
- φ is a classical sentence (i.e., it only involves $\forall, \exists, \neg, \wedge, \vee$) which only uses a binary predicate symbol R.
- $\varphi(R|\beta)$ is the result of replacing all Rxy with $\beta(x, y)$.

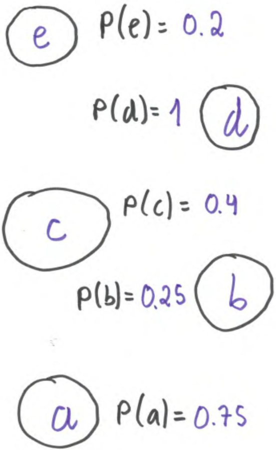
Lemma (A Uniform Statement on β's)

1. If $\emptyset \models_{fin} \varphi$, then $2.(\neg Bival(\beta) \vee \varphi(R|\beta)) \in stL\forall$.
2. If $\emptyset \models \varphi$, then $2.(\neg Bival(\beta) \vee \varphi(R|\beta)) \in spsL\forall$. $[\in spsMTL\forall]$

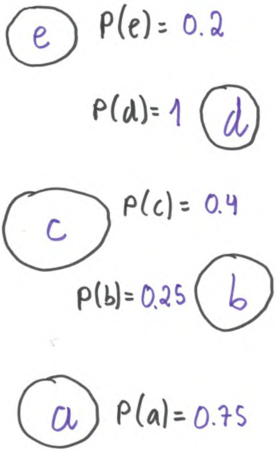
Sketch of the Proof: Given any fuzzy **A**-structure M such that $[\![\beta(a, b)]\!]^M$ is never 0.5, then take the classical structure $M/2$ by

- $R(a, b) = 1$, if $[\![\beta(a, b)]\!]^M > 0.5$
- $R(a, b) = 0$, if $[\![\beta(a, b)]\!]^M < 0.5$

FUZZY STRUCTURE



FUZZY STRUCTURE



TWO LINEAR ORDERS

$R_1 x y \iff P(x) \leq P(y)$

Hence, $e < b < c < a < d$

FUZZY STRUCTURE

e: $P(e) = 0.2$

d: $P(d) = 1$

c: $P(c) = 0.4$

b: $P(b) = 0.25$

a: $P(a) = 0.75$

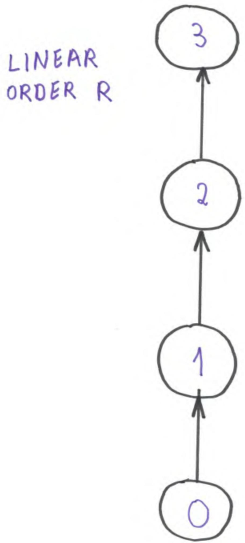
TWO LINEAR ORDERS

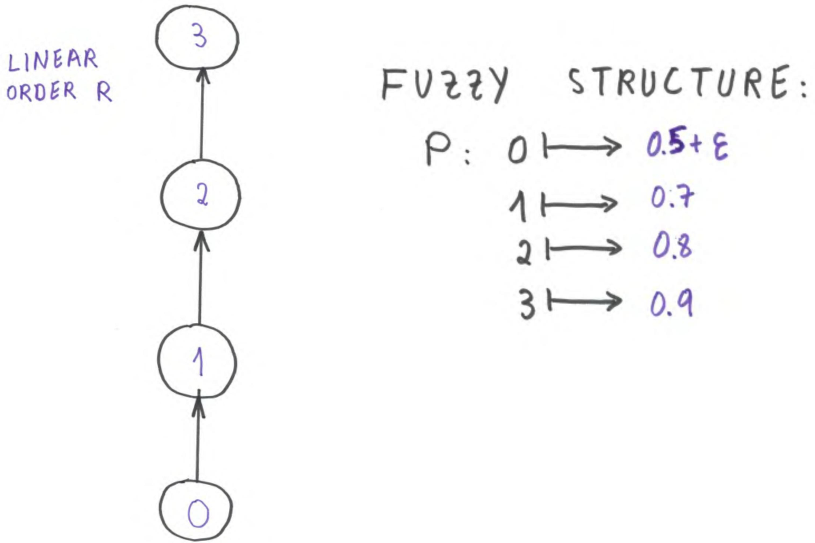
$R_1 x y \iff P(x) \leq P(y)$

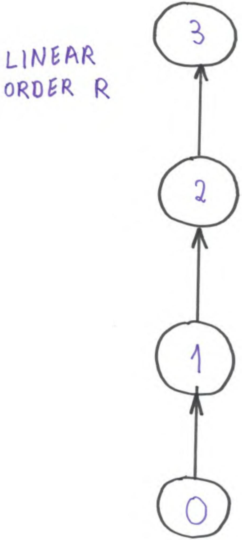
Hence, $e < b < c < a < d$

$$R_2 x y \iff a \begin{cases} x = min_1 \\ P(x) - P(med_1 x) \leq P(y) - P(med_1 y) \end{cases}$$

Hence, $e < b < c < d < a$


LINEAR
ORDER R
3
2
1
0


LINEAR
ORDER R
3
2
1
0
FUZZY STRUCTURE:
P: 0 ⟼ 0.5+ε
1 ⟼ 0.7
2 ⟼ 0.8
3 ⟼ 0.9



FUZZY STRUCTURE:

P: $0 \longmapsto 0.5+\varepsilon$

$1 \longmapsto 0.7$

$2 \longmapsto 0.8$

$3 \longmapsto 0.9$

$$Rxy \iff (\forall x Px) \odot (Px \to Py) > 0.5$$

$$\neg Rxy \iff (\forall x Px) \odot (Px \to Py) < 0.5$$

- Take now $\beta(x, y) := (\forall x Px) \odot (Px \to Py)$.

- Take now $\beta(x, y) := (\forall x Px) \odot (Px \to Py)$.

Theorem

Let Φ_{lo} be the first-order sentence (with a binary predicate symbol R) axiomatizing the theory of (classical) linear orders. Then,

$$\Phi_{lo} \models_{fin} \varphi \qquad \text{iff} \qquad 2.(\neg Bival(\beta) \vee (\neg\Phi_{lo} \vee \varphi)(R|\beta)) \in stL\forall.$$

- Take now $\beta(x,y) := (\forall x Px) \odot (Px \to Py)$.

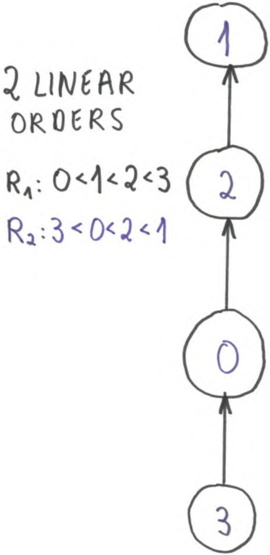
Theorem

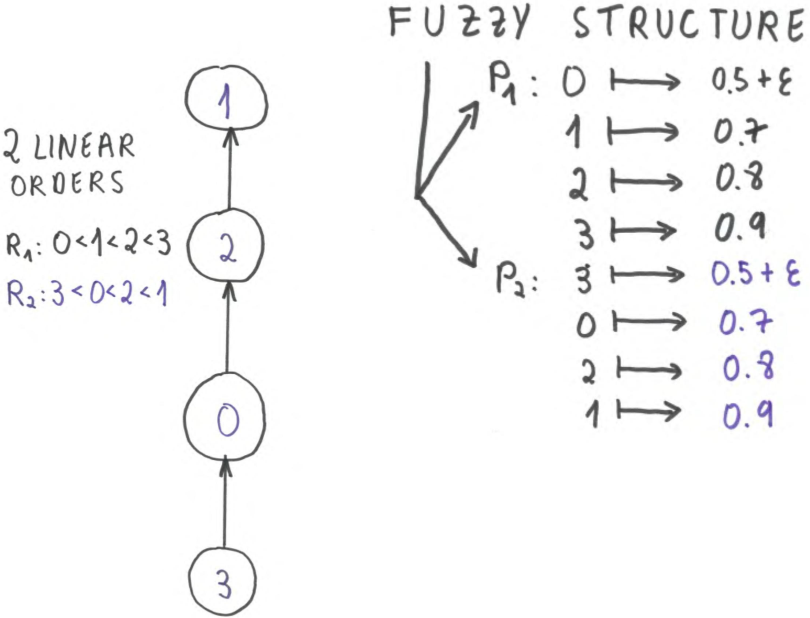
Let Φ_{lo} be the first-order sentence (with a binary predicate symbol R) axiomatizing the theory of (classical) linear orders. Then,

$$\Phi_{lo} \models_{fin} \varphi \qquad \textit{iff} \qquad 2.(\neg Bival(\beta) \vee (\neg\Phi_{lo} \vee \varphi)(R|\beta)) \in stL\forall.$$

The same theorem holds if we use

- $\beta'(x,y) := (\forall x Px) \odot (Px \leftrightarrow Py)$, and
- the sentence Φ_{eq} axiomatizing the theory of one equivalence relation.


2 LINEAR ORDERS
R1: 0<1<2<3
R2: 3<0<2<1
1
2
0
3


FUZZY STRUCTURE
2 LINEAR ORDERS
R1: 0<1<2<3
R2: 3<0<2<1
1
2
0
3
P1: 0 ⟼ 0.5+ε
1 ⟼ 0.7
2 ⟼ 0.8
3 ⟼ 0.9
P2: 3 ⟼ 0.5+ε
0 ⟼ 0.7
2 ⟼ 0.8
1 ⟼ 0.9

FUZZY STRUCTURE

2 LINEAR ORDERS

$R_1: 0 < 1 < 2 < 3$

$R_2: 3 < 0 < 2 < 1$

P_1:

$0 \longmapsto 0.5 + \varepsilon$

$1 \longmapsto 0.7$

$2 \longmapsto 0.8$

$3 \longmapsto 0.9$

P_2:

$3 \longmapsto 0.5 + \varepsilon$

$0 \longmapsto 0.7$

$2 \longmapsto 0.8$

$1 \longmapsto 0.9$

$$R_1 x y \iff (\forall x P_1 x) \odot (P_1 x \to P_1 y) > 0.5$$

$$\neg R_1 x y \iff (\forall x P_1 x) \odot (P_1 x \to P_1 y) < 0.5$$

$$R_2 x y \iff (\forall x P_2 x) \odot (P_2 x \to P_2 y) > 0.5$$

$$\neg R_2 x y \iff (\forall x P_2 x) \odot (P_2 x \to P_2 y) < 0.5$$

- $\beta_1(x, y) := (\forall x P_1 x) \odot (P_1 x \rightarrow P_1 y)$.

- $\beta_1(x, y) := (\forall x P_1 x) \odot (P_1 x \to P_1 y)$.
- $\beta_2(x, y) := (\forall x P_2 x) \odot (P_2 x \to P_2 y)$.

- $\beta_1(x, y) := (\forall x P_1 x) \odot (P_1 x \to P_1 y)$.
- $\beta_2(x, y) := (\forall x P_2 x) \odot (P_2 x \to P_2 y)$.

Theorem

Let Φ_{2lo} be the first-order sentence (with binary predicate symbols R_1 and R_2) axiomatizing the theory of two (classical) linear orders. Then,

- $\Phi_{2lo} \models_{fin} \varphi$, *iff*
- $2.(\neg Bival(\beta_1) \vee \neg Bival(\beta_2) \vee (\neg \Phi_{2lo} \vee \varphi)(R_1|\beta_1, R_2|\beta_2)) \in stL\forall$.

- $\beta_1(x, y) := (\forall x P_1 x) \odot (P_1 x \to P_1 y)$.
- $\beta_2(x, y) := (\forall x P_2 x) \odot (P_2 x \to P_2 y)$.

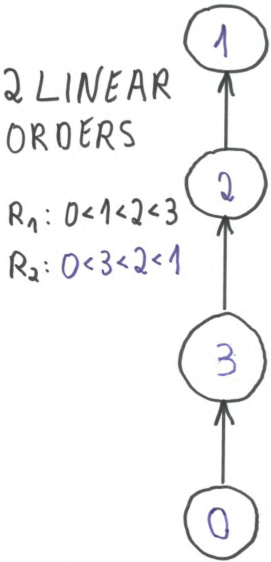
Theorem

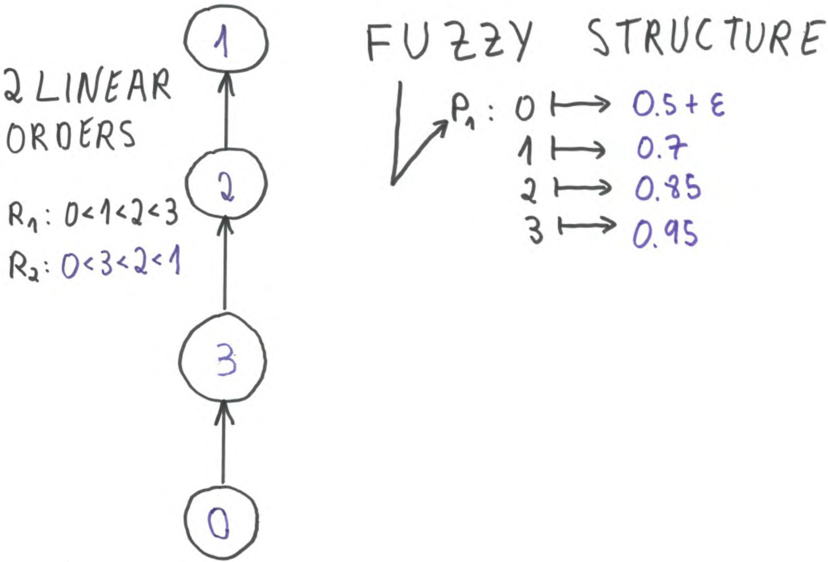
Let Φ_{2lo} be the first-order sentence (with binary predicate symbols R_1 and R_2) axiomatizing the theory of two (classical) linear orders. Then,

- $\Phi_{2lo} \models_{fin} \varphi$, *iff*
- $2.(\neg Bival(\beta_1) \vee \neg Bival(\beta_2) \vee (\neg \Phi_{2lo} \vee \varphi)(R_1|\beta_1, R_2|\beta_2)) \in stL\forall$.

Corollary

Monadic Predicate standard Lukasiewicz Logic with two unary predicate symbols is undecidable.


2 LINEAR
ORDERS
R1: 0<1<2<3
R2: 0<3<2<1
1
2
3
0


2 LINEAR ORDERS
R1 : 0<1<2<3
R2 : 0<3<2<1
1
2
3
0
FUZZY STRUCTURE
P1 : 0 ⟼ 0.5 + ε
1 ⟼ 0.7
2 ⟼ 0.85
3 ⟼ 0.95

FUZZY STRUCTURE

P_1:

$0 \longmapsto 0.5 + \varepsilon$

$1 \longmapsto 0.7$

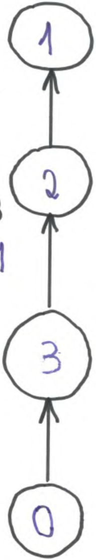
$2 \longmapsto 0.85$

$3 \longmapsto 0.95$

2 LINEAR ORDERS

$R_1 : 0 < 1 < 2 < 3$

$R_2 : 0 < 3 < 2 < 1$



$$R_1 x y \Leftrightarrow (\forall x P_1 x) \odot (P_1 x \to P_1 y) > 0.5$$

$$\neg R_1 x y \Leftrightarrow (\forall x P_1 x) \odot (P_1 x \to P_1 y) < 0.5$$

LET $B_2(x,y) := (x \approx min_1) \vee$

$$\vee \left[(\forall x P_1 x) \odot \left(P_1 x \ominus P_1(med_1, x) \to P_1 y \ominus P_1(med_1, y) \right) \right]$$

$$R_2 x y \Leftrightarrow B_2(x,y) > 0.5$$

$$\neg R_2 x y \Leftrightarrow B_2(x,y) < 0.5$$

- $\beta_1(x,y) := (\forall x Px) \odot (Px \to Py)$
- $pred_1(x,y) :=$
  $\big(R_1 yx \ \wedge \ \forall z R_1 xz\big) \vee \big(R_1 yx \ \wedge \ \neg R_1 xy \ \wedge \ \forall z(R_1 zy \vee R_1 xz)\big)$
  [this defines a total function on finite linear orders]
- $pred_{\beta_1}(x,y) :=$
  $\big(\beta_1 yx \ \wedge \ \forall z \beta_1 xz\big) \vee \big(\beta_1 yx \ \wedge \ \neg\beta_1 xy \ \wedge \ \forall z(\beta_1 zy \vee \beta_1 xz)\big)$
- $\delta(x,x',y,y') := (\forall x Px) \odot ((Px \ominus Px') \to (Py \ominus Py'))$
- $\beta_2(x,y) := \exists x' \exists y' (pred_{\beta_1}(x,x') \wedge pred_{\beta_1}(y,y') \wedge \delta(x,x',y,y'))$

- $\beta_1(x, y) := (\forall x Px) \odot (Px \rightarrow Py)$
- $pred_1(x, y) :=$
  $\big(R_1 yx \;\wedge\; \forall z R_1 xz\big) \vee \big(R_1 yx \;\wedge\; \neg R_1 xy \;\wedge\; \forall z(R_1 zy \vee R_1 xz)\big)$
  [this defines a total function on finite linear orders]
- $pred_{\beta_1}(x, y) :=$
  $\big(\beta_1 yx \;\wedge\; \forall z \beta_1 xz\big) \vee \big(\beta_1 yx \;\wedge\; \neg\beta_1 xy \;\wedge\; \forall z(\beta_1 zy \vee \beta_1 xz)\big)$
- $\delta(x, x', y, y') := (\forall x Px) \odot ((Px \ominus Px') \rightarrow (Py \ominus Py'))$
- $\beta_2(x, y) := \exists x' \exists y' (pred_{\beta_1}(x, x') \wedge pred_{\beta_1}(y, y') \wedge \delta(x, x', y, y'))$

Theorem

Let Φ_{2lo*} be the first-order sentence (with binary predicate symbols R_1 and R_2) axiomatizing the theory of two (classical) linear orders with the same minimum element. Then,

- $\Phi_{2lo*} \models_{fin} \varphi$, *iff*
- $2.(\neg Bival(\beta_1) \vee \neg Bival(\beta_2) \vee (\neg\Phi_{2lo} \vee \varphi)(R_1|\beta_1, R_2|\beta_2)) \in stL\forall$.

Corollary

Monadic Predicate standard Łukasiewicz Logic with just one unary predicate symbol is undecidable. Indeed, four variables are enough for undecidability.

Corollary

Monadic Predicate standard Łukasiewicz Logic with just one unary predicate symbol is undecidable. Indeed, four variables are enough for undecidability.

Corollary

There is a sentence with only one unary predicate symbol P and at most four variables that is a standard Łukasiewicz tautology but not a general Łukasiewicz tautology.

Undecidability of General semantics

- Providing a characterization for the general semantics seems (due the safeness condition) much more difficult.

Undecidability of General semantics

- Providing a characterization for the general semantics seems (due the safeness condition) much more difficult.
- Fortunately, finitely inseparability of the theory of two linear orders (a particular case of "recursive inseparability") helps to avoid this difficulty.

Undecidability of General semantics

- Providing a characterization for the general semantics seems (due the safeness condition) much more difficult.
- Fortunately, finitely inseparability of the theory of two linear orders (a particular case of “recursive inseparability”) helps to avoid this difficulty.

Theorem

Let ϑ be the vocabulary with one one unary predicate symbol. Then,

- *spsL$\forall$ is recursively inseparable from stL$\forall$. This means there is no recursive set X such that spsL$\forall \subseteq X \subseteq$ stL$\forall$.*
- *spsMTL$\forall$ is recursively inseparable from stL$\forall$.*

LET US CONSIDER THE ORDER ω^+

LET US CONSIDER THE ORDER ω^+

- $Th(\omega) \neq Th(\text{FIN L.O.})$

  [e.g. $\exists x$ ("x is a limit")

LET US CONSIDER THE ORDER ω^+

· $Th(\omega) \neq Th(\text{FIN L.O.})$

$\llcorner$ e.g. $\exists x$ ("x is a limit")

· IS THERE SOME $P : \omega^+ \to [0,1]^*$ SUCH THAT:

1) $\langle \omega^+, P \rangle$ is SAFE

2) $x \leq y \iff (\forall x P x) \odot (P x \to P y) > 0.5$

$y < x \iff (\forall x P x) \odot (P x \to P y) < 0.5$

Let us suppose there is a positive answer to the previous question, and let us consider the sentence

$$\psi := \neg Bival(\beta) \vee \neg\varphi(R|\beta),$$

where

- φ is the sentence using only the binary predicate symbol R which says “it is a linear order with at least one limit point”
- $\beta(x, y) := (\forall x Px) \odot (Px \to Py)$,
- $Bival(\beta) := \forall x \forall y (\beta(x, y) \vee \neg\beta(x, y))$.

Then, $\psi \oplus \psi$ is a standard Łukasiewicz tautology that is not a general tautology.

Remark: We already know that $\psi \oplus \psi$ is a standard Łukasiewicz tautology that is not a supersound tautology.

A FUZZY STRUCTURE

$P(w) = 1$

$P(2) = 3/4$

$P(1) = 2/3$

$P(0) = 1/2 + \mu$

μ INFINITESIMAL

A FUZZY STRUCTURE

$P(w) = 1$

$P(2) = 3/4$

$P(1) = 2/3$

$P(0) = 1/2 + \mu$

μ INFINITESIMAL

IT HOLDS

$x \leq y \Leftrightarrow (\forall x Px) \odot (Px \to Py) > 0.5$

$y < x \Leftrightarrow (\forall x Px) \odot (Px \to Py) < 0.5$

A FUZZY STRUCTURE

$P(w) = 1$

$P(2) = 3/4$

$P(1) = 2/3$

$P(0) = 1/2 + \mu$

μ INFINITESIMAL

IT HOLDS

$$x \leq y \iff (\forall x Px) \odot (Px \to Py) > 0.5$$

$$y < x \iff (\forall x Px) \odot (Px \to Py) < 0.5$$

- IS IT SAFE?
- IS IT WITNESSED?

Open Question

Can we give some sentence φ (using only a binary predicate symbol) such that

$$2.(\neg \mathit{Bival}(\beta) \vee \varphi(R|\beta)) \in \mathit{stL}\forall \setminus \mathit{genL}\forall \ ?$$