Technische Universität Dresden

Faculty of Computer Science • Institute of Theoretical Computer Science • Chair of Automata Theory

Reasoning in OWL 2 EL with Hierarchical Concrete Domains

Francesco Kriegel

Theoretical Computer Science, Technische Universität Dresden, Germany
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Germany

15th International Symposium on Frontiers of Combining Systems,
29 September – 1 October 2025

OWL 2 EL and State Of The Art

OWL 2 EL

- OWL 2 EL is a profile of the Web Ontology Language.
- The $\mathcal{EL}$ family of description logics underpins OWL 2 EL.
- We here use “$\mathcal{EL}$” and “OWL 2 EL” as synonyms despite minor technical differences.

OWL 2 EL

- OWL 2 EL is a profile of the Web Ontology Language.
- The $\mathcal{EL}$ family of description logics underpins OWL 2 EL.
- We here use “$\mathcal{EL}$” and “OWL 2 EL” as synonyms despite minor technical differences.
- The semantics is based on models.
- Models $\mathcal{I}$ of an application domain use:
  - *atomic concepts* A to classify objects
  - *roles* r to interlink objects
  - *individuals* i to name objects
  - and *features* f to assign concrete values to objects.
- The concrete domain $\mathcal{D}$ uses *predicates* P to interlink concrete values.

Knowledge Bases

- An application domain can be described by a *knowledge base (KB)* consisting of:
  - *Concept inclusions* $C \sqsubseteq D$ involving compound concepts C, D built by $C ::= \bot \mid \top \mid \{i\} \mid A \mid \exists f_1, \ldots, f_k.P \mid C \sqcap C \mid \exists r.C$
  - *Role inclusions* $R \sqsubseteq s$ involving role chains $R ::= \varepsilon \mid r \mid R \circ R$ and roles s
  - *Range inclusions* $\mathrm{Ran}(r) \sqsubseteq C$ involving roles r and compound concepts C

Knowledge Bases

- An application domain can be described by a *knowledge base (KB)* consisting of:
  - *Concept inclusions* $C \sqsubseteq D$ involving compound concepts C, D built by $C ::= \bot \mid \top \mid \{i\} \mid A \mid \exists f_1, \ldots, f_k.P \mid C \sqcap C \mid \exists r.C$
  - *Role inclusions* $R \sqsubseteq s$ involving role chains $R ::= \varepsilon \mid r \mid R \circ R$ and roles s
  - *Range inclusions* $\mathrm{Ran}(r) \sqsubseteq C$ involving roles r and compound concepts C
- Syntactic sugar:
  - *Concept assertions* $\{i\} \sqsubseteq C$, also written $i : C$
  - *Role assertions* $\{i\} \sqsubseteq \exists r.\{j\}$, also written $(i, j) : r$
  - *Domain inclusions* $\exists r.\top \sqsubseteq C$, also written $\mathrm{Dom}(r) \sqsubseteq C$

Knowledge Bases

- An application domain can be described by a *knowledge base (KB)* consisting of:
  - *Concept inclusions* $C \sqsubseteq D$ involving compound concepts C, D built by $C ::= \bot \mid \top \mid \{i\} \mid A \mid \exists f_1, \dots, f_k.P \mid C \sqcap C \mid \exists r.C$
  - *Role inclusions* $R \sqsubseteq s$ involving role chains $R ::= \varepsilon \mid r \mid R \circ R$ and roles s
  - *Range inclusions* $\mathrm{Ran}(r) \sqsubseteq C$ involving roles r and compound concepts C
- Syntactic sugar:
  - *Concept assertions* $\{i\} \sqsubseteq C$, also written $i : C$
  - *Role assertions* $\{i\} \sqsubseteq \exists r.\{j\}$, also written $(i, j) : r$
  - *Domain inclusions* $\exists r.\top \sqsubseteq C$, also written $\mathrm{Dom}(r) \sqsubseteq C$
- If the predicate P in a constraint $\exists f_1, \dots, f_k.P$ is defined through a mathematical expression or a logical formula with k free variables, then we may represent the constraint also through this expression or formula but with the free variables replaced by the features $f_1, \dots, f_k$.
  Example: $f + g - h = 0$ represents $\exists f, g, h.P$ where $P^{\mathcal{D}} := \{ (x, y, z) \mid x + y - z = 0 \}$

Reasoning

- Reasoning in $\mathcal{EL}$ (i.e. deciding entailment between knowledge bases) can be done in polynomial time.
- Fastest reasoner implementation: **ELK**

Reasoning

- Reasoning in $\mathcal{EL}$ (i.e. deciding entailment between knowledge bases) can be done in polynomial time.
- Fastest reasoner implementation: **ELK**
- Various application domains of knowledge bases in general:

  web search, virtual assistants,
  health care, energy sector,
  finance, e-commerce,
  Industry 4.0, etc.
- Two prominent $\mathcal{EL}$ knowledge bases:
  - Systematized Nomenclature of Medicine – Clinical Terms (abbrv. **SNOMED CT**)
  - **Gene Ontology** (exception: one pair of inverse roles)

Concrete Domains

The concrete domain $\mathcal{D}_{\mathbb{Q},\mathrm{diff}}$ supports the constraints $f = b$, $f > b$, and $f - g = b$ for all features f, g and for all rational numbers b.

Example. Hypertension is expressed by the following concept inclusions:

$$(\mathrm{sys} = 140) \sqsubseteq \mathrm{Hypertension}, (\mathrm{sys} > 140) \sqsubseteq \mathrm{Hypertension},$$
$$(\mathrm{dia} = 90) \sqsubseteq \mathrm{Hypertension}, (\mathrm{dia} > 90) \sqsubseteq \mathrm{Hypertension}.$$

Specific values of a patient can be expressed as follows: $\mathrm{bob} : (\mathrm{sys} = 114) \sqcap (\mathrm{dia} = 69)$.

Concrete Domains

The concrete domain $\mathcal{D}_{\mathbb{Q},\mathrm{diff}}$ supports the constraints $f = b$, $f > b$, and $f - g = b$ for all features f, g and for all rational numbers b.

Example. Hypertension is expressed by the following concept inclusions:

$$(\mathrm{sys} = 140) \sqsubseteq \mathrm{Hypertension},\ (\mathrm{sys} > 140) \sqsubseteq \mathrm{Hypertension},$$
$$(\mathrm{dia} = 90) \sqsubseteq \mathrm{Hypertension},\ (\mathrm{dia} > 90) \sqsubseteq \mathrm{Hypertension}.$$

Specific values of a patient can be expressed as follows: $\mathrm{bob} : (\mathrm{sys} = 114) \sqcap (\mathrm{dia} = 69)$.

However, neither non-elevated blood pressure (dia. below 120 and sys. below 70) nor elevated blood pressure (dia. between 120 and 140, and sys. between 70 and 90) are expressible since the other relations $\geq, \leq, <$ are unavailable.

Otherwise, $\{\top \sqsubseteq (f > 0),\ (f = 3) \sqsubseteq C,\ (f > 3) \sqsubseteq C,\ (f < 3) \sqsubseteq A,\ C \sqcap A \sqsubseteq \bot\}$ could enforce that the atomic concept A is the complement of the concept C, thus enabling emulation of the description logic $\mathcal{ALC}$ (which has exponential-time reasoning complexity).

Concrete Domains

The concrete domain $\mathcal{D}_{\mathbb{Q},\mathrm{lin}}$ provides the constraints $a_1 \cdot f_1 + \ldots + a_k \cdot f_k = b$ for all features $f_1, \ldots, f_k$ and for all rational numbers $a_1, \ldots, a_k, b$.

Example. The concept inclusion $\mathsf{Human} \sqsubseteq (\mathsf{sys} - \mathsf{dia} - \mathsf{pp} = 0)$ expresses that, for every human, the pulse pressure is the difference between the systolic and the diastolic blood pressure.

Concrete Domains

The concrete domain $\mathcal{D}_{\mathbb{Q},\mathrm{lin}}$ provides the constraints $a_1 \cdot f_1 + \ldots + a_k \cdot f_k = b$ for all features $f_1, \ldots, f_k$ and for all rational numbers $a_1, \ldots, a_k, b$.

Example. The concept inclusion $\mathsf{Human} \sqsubseteq (\mathsf{sys} - \mathsf{dia} - \mathsf{pp} = 0)$ expresses that, for every human, the pulse pressure is the difference between the systolic and the diastolic blood pressure.

The combined expressivity of $\mathcal{D}_{\mathbb{Q},\mathrm{diff}}$ and $\mathcal{D}_{\mathbb{Q},\mathrm{lin}}$ would be useful since then with the concept inclusion $\mathsf{ICUPatient} \sqcap (\mathsf{pp} > 50) \sqsubseteq \mathsf{NeedsAttention}$ it could be expressed that intensive-care patients with a pulse pressure exceeding 50 need attention.

However, the TBox $\{\top \sqsubseteq (f + g = 0),\ (f = 0) \sqsubseteq C,\ (f > 0) \sqsubseteq C,\ (g > 0) \sqsubseteq A,\ C \sqcap A \sqsubseteq \bot\}$ could declare A as the complement of C.

Concrete Domains

There is another concrete domain involving strings, but it is also too restricted to be of practical use.

Concrete Domains

There is another concrete domain involving strings, but it is also too restricted to be of practical use.

In order to ensure polynomial-time reasoning, every concrete domain for $\mathcal{EL}$ should be P-*admissible*, i.e. fulfill the following conditions:

- it is *convex*, i.e. every finite disjunction of constraints and negated constraints is already equivalent to one disjunct,
- satisfiability of constraint conjunctions is decidable in polynomial time (i.e. is in P),
- and validity of constraint inclusions is decidable in polynomial time (i.e. is in P).

Concrete Domains

There is another concrete domain involving strings, but it is also too restricted to be of practical use.

In order to ensure polynomial-time reasoning, every concrete domain for $\mathcal{EL}$ should be P-*admissible*, i.e. fulfill the following conditions:

- it is *convex*, i.e. every finite disjunction of constraints and negated constraints is already equivalent to one disjunct,
- satisfiability of constraint conjunctions is decidable in polynomial time (i.e. is in P),
- and validity of constraint inclusions is decidable in polynomial time (i.e. is in P).

Similarly, we can define NP-*admissibility* and EXP-*admissibility*. Reasoning within the logical domain is then still in P, but more expensive within the concrete domain.

NP- or EXP-admissible concrete domains are thus more appropriate for Horn logics more expressive than $\mathcal{EL}$, such as $\mathcal{ELI}$, Horn-$\mathcal{ALC}$, Horn-$\mathcal{SROIQ}$, and existential rules.

Hierarchical Concrete Domains

Semi-Lattices

We introduce a novel form of concrete domains based on semi-lattices.

A *semi-lattice* $\mathbf{L} := (L, \leq, \wedge)$ consists of

- a set L,
- a partial order $\leq$,
- and a binary meet operation $\wedge$.

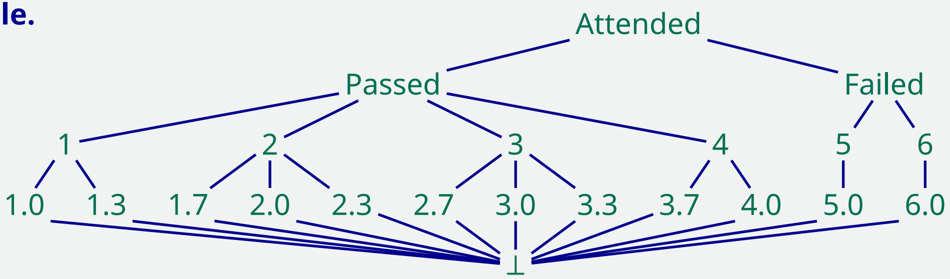
Semi-Lattices

We introduce a novel form of concrete domains based on semi-lattices.

A *semi-lattice* $\mathbf{L} := (L, \leq, \wedge)$ consists of

- a set L,
- a partial order $\leq$,
- and a binary meet operation $\wedge$.

Example.



Hierarchical Concrete Domains

- The *hierarchical concrete domain* $\mathcal{D}_{\mathbf{L}}$ has values in L and supports only constraints of the form $f \leq p$ involving a feature f and a value p.
- $\leq$ is understood as an “information order,” i.e. a constraint $f \leq p$ means that f’s value is equal to or more specific than p.
- $\wedge$ is used to combine the values p and q of two constraints $f \leq p$ and $f \leq q$.

Hierarchical Concrete Domains

- The *hierarchical concrete domain* $\mathcal{D}_{\mathsf{L}}$ has values in L and supports only constraints of the form $f \leq p$ involving a feature f and a value p.
- $\leq$ is understood as an “information order,” i.e. a constraint $f \leq p$ means that f’s value is equal to or more specific than p.
- $\wedge$ is used to combine the values p and q of two constraints $f \leq p$ and $f \leq q$.

Example. Real intervals form a semi-lattice with subset inclusion $\subseteq$ as partial order and intersection $\cap$ as meet operation. Non-elevated blood pressure could be expressed with the concept inclusion $\mathsf{NonElevatedBP} \equiv (\mathsf{sys} \subseteq [0,120)) \sqcap (\mathsf{dia} \subseteq [0,70))$.

Feature Inclusions

- Dependencies between features as well as aggregations of features can be expressed by feature inclusions.
- A *feature inclusion* $f \leq H(g_1, \dots, g_n)$ consists of features $f, g_1, \dots, g_n$ and a computable n-ary operation $H : L^n \to L$ that is monotonic (i.e. $H(p_1, \dots, p_n) \leq H(q_1, \dots, q_n)$ whenever $p_1 \leq q_1, \dots,$ and $p_n \leq q_n$).
- An FBox is a finite set of feature inclusions.
- Reasoning in the concrete domain takes only models of the FBox into account.
- Knowledge bases can now contain feature inclusions.

Feature Inclusions

- Dependencies between features as well as aggregations of features can be expressed by feature inclusions.
- A *feature inclusion* $f \leq H(g_1, \dots, g_n)$ consists of features $f, g_1, \dots, g_n$ and a computable n-ary operation $H \colon L^n \to L$ that is monotonic (i.e. $H(p_1, \dots, p_n) \leq H(q_1, \dots, q_n)$ whenever $p_1 \leq q_1, \dots,$ and $p_n \leq q_n$).
- An FBox is a finite set of feature inclusions.
- Reasoning in the concrete domain takes only models of the FBox into account.
- Knowledge bases can now contain feature inclusions.

Example. Through the feature inclusion $\mathsf{pp} \subseteq \mathsf{sys} - \mathsf{dia}$ we can obtain an interval value of the pulse pressure given intervals of the systolic and the diastolic blood pressure. With the concept inclusion $\mathsf{ICUPatient} \sqcap (\mathsf{pp} \subseteq (50, \infty)) \sqsubseteq \mathsf{NeedsAttention}$ we can now express that intensive-care patients having a pulse pressure above 50 need attention.

Convex by Design

Hierarchical concrete domains are **convex by design** since the semi-lattice semantics effectively expels disjunction.

Example. A model of the constraint $\mathsf{sys} \subseteq [110, 120)$ need not assign a particular number between 110 and 120 to the feature sys but can instead assign the whole interval $[110, 120)$.

Convex by Design

Hierarchical concrete domains are **convex by design** since the semi-lattice semantics effectively expels disjunction.

Example. A model of the constraint $\mathsf{sys} \subseteq [110, 120)$ need not assign a particular number between 110 and 120 to the feature sys but can instead assign the whole interval $[110, 120)$.

In particular, $\mathcal{D}_{\mathbf{L}}$ is convex w.r.t. an FBox $\mathcal{F}$ in each of the following cases:

- $\mathbf{L}$ is complete
- $\mathbf{L}$ is computable and well-founded
- $\mathbf{L}$ is computable and $\mathcal{F}$ is acyclic.

For the remaining case we have an answer in a particular semi-lattice, but otherwise this is left for future research.

Examples

Examples

$\mathbf{L}$	L	$\leq$	$\wedge$
$\mathrm{Int}(\mathbb{R})$	real intervals	subset inclusion $\subseteq$	set intersection $\cap$

Examples

$\mathbf{L}$	L	$\leq$	$\wedge$
$\mathrm{Int}(\mathbb{R})$	real intervals	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{CGon}(\mathbb{R}^2)$	convex polygons	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{UGon}(\mathbb{R}^2)$	unions of polygons	subset inclusion $\subseteq$	set intersection $\cap$

Examples

$\mathbf{L}$	L	$\leq$	$\wedge$
$\mathrm{Int}(\mathbb{R})$	real intervals	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{CGon}(\mathbb{R}^2)$	convex polygons	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{UGon}(\mathbb{R}^2)$	unions of polygons	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{Reg}(\Sigma)$	regular languages	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{FA}(\Sigma)$	finite automata	automata inclusion $\preceq$	direct product $\times$
$\mathrm{DFA}(\Sigma)$	deterministic finite automata	automata inclusion $\preceq$	direct product $\times$

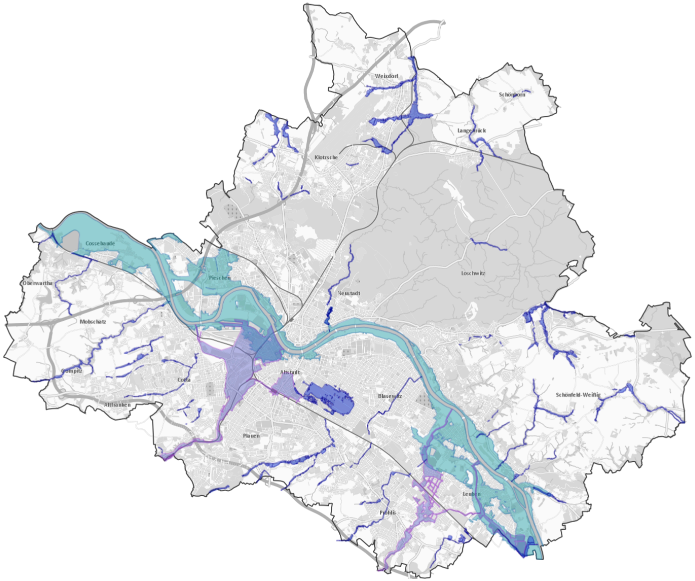
Examples

L	L	$\leq$	$\wedge$
$\mathrm{Int}(\mathbb{R})$	real intervals	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{CGon}(\mathbb{R}^2)$	convex polygons	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{UGon}(\mathbb{R}^2)$	unions of polygons	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{Reg}(\Sigma)$	regular languages	subset inclusion $\subseteq$	set intersection $\cap$
$\mathrm{FA}(\Sigma)$	finite automata	automata inclusion $\preceq$	direct product $\times$
$\mathrm{DFA}(\Sigma)$	deterministic finite automata	automata inclusion $\preceq$	direct product $\times$
Graph	finite, directed graphs with labels	existence of homomorphism (in converse direction)	disjoint union $\uplus$

Polygons

Weixdorf
Schönborn
Langebrück
Klotzsche
Cossebaude
Pieschen
Oberwartha
Neustadt
Loschwitz
Mobschatz
Altstadt
Gompitz
Cotta
Blasewitz
Schönfeld-Weißig
Altfranken
Plauen
Leuben
Prohlis

Polygons


Weixdorf
Schönborn
Langebrück
Klotzsche
Cossebaude
Pieschen
Oberwartha
Neustadt
Loschwitz
Mobschatz
Altstadt
Gompitz
Cotta
Blasewitz
Schönfeld-Weißig
Altfranken
Plauen
Leuben
Prohlis

Regular Languages

Let $\mathfrak{A}$ be a finite automaton that recognizes $\Sigma^* \circ \{\text{description logic}\} \circ \Sigma^*$.

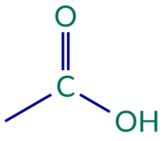
The concept inclusion

$$\text{ScientificArticle} \sqcap (\text{hasTitle} \preceq \mathfrak{A}) \sqsubseteq \text{DLPaper}$$

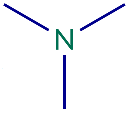
expresses that a scientific article is a DL paper if its title contains “description logic” as substring.

Graphs

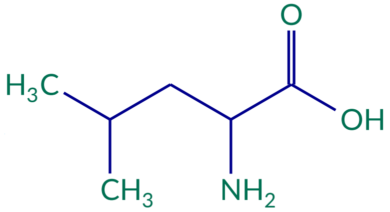
Three graphs representing chemical compounds:



$\mathcal{G}_{\text{carboxylic acid group}}$



$\mathcal{G}_{\text{amino group}}$



$\mathcal{G}_{\text{L-leucine}}$

If the knowledge base $\mathcal{K}$ contains

$$\begin{aligned} &\text{L-Leucine} \equiv (\text{hasMolecularStructure} \leq \mathcal{G}_{\text{L-leucine}}) \\ \text{and } &\text{AminoAcid} \equiv (\text{hasMolecularStructure} \leq \mathcal{G}_{\text{carboxylic acid group}}) \\ &\qquad \sqcap (\text{hasMolecularStructure} \leq \mathcal{G}_{\text{amino group}}), \end{aligned}$$

then $\mathcal{K}$ entails L-Leucine $\sqsubseteq$ AminoAcid since $\mathcal{G}_{\text{L-leucine}} \leq \mathcal{G}_{\text{carboxylic acid group}} \wedge \mathcal{G}_{\text{amino group}}$.

Computational Complexity

L	Deciding L	Deciding $\leq$	Computing $\wedge$
$\mathrm{Int}(\mathbb{R})$	in P	in P	poly. time
$\mathrm{CGon}(\mathbb{R}^2)$	in P (arithmetic)	in P (arithmetic)	arithmetic poly. time
$\mathrm{UGon}(\mathbb{R}^2)$	in P (arithmetic)	in P (arithmetic)	arithmetic poly. time
$\mathrm{Reg}(\Sigma)$	—	—	—
$\mathrm{FA}(\Sigma)$	in P	in PSpace	poly. time
$\mathrm{DFA}(\Sigma)$	in P	in P	poly. time
Graph	in P	in NP	poly. time

Non-Duplicating Operations

In general, reasoning in a hierarchical concrete domain $\mathcal{D}_{\mathbf{L}}$ can be done as follows:

1. We maintain a valuation v, which is a function from features to values.
2. For every feature f, collect all constraints $f \leq p_1, f \leq p_2, \ldots, f \leq p_n$, then compute the meet $p := p_1 \wedge p_2 \wedge \cdots \wedge p_n$ and set $v(f) := p$.
3. Update the valuation by means of the feature inclusions until no further changes occur, i.e. replace the value $v(f)$ by the meet $v(f) \wedge H(v(g_1), \ldots, v(g_n))$ whenever there is a feature inclusion $f \leq H(g_1, \ldots, g_n)$ such that these two values differ.

Non-Duplicating Operations

In general, reasoning in a hierarchical concrete domain $\mathcal{D}_\mathsf{L}$ can be done as follows:

1. We maintain a valuation v, which is a function from features to values.
2. For every feature f, collect all constraints $f \leq p_1, f \leq p_2, \ldots, f \leq p_n$, then compute the meet $p \coloneqq p_1 \wedge p_2 \wedge \cdots \wedge p_n$ and set $v(f) \coloneqq p$.
3. Update the valuation by means of the feature inclusions until no further changes occur, i.e. replace the value $v(f)$ by the meet $v(f) \wedge H(v(g_1), \ldots, v(g_n))$ whenever there is a feature inclusion $f \leq H(g_1, \ldots, g_n)$ such that these two values differ.

During the updates, the valuation could grow to an exponential size (or even larger).

The valuation is bounded by the input size (and thus polynomial) if the meet operation $\wedge$ and all operations H in the FBox are non-duplicating:

An operation $H : L^n \to L$ is *non-duplicating* if, for all $(p_1, \ldots, p_n) \in L^n$, the size of $H(p_1, \ldots, p_n)$ is no larger than the size of $(p_1, \ldots, p_n)$.

Computational Complexity

L	$\wedge$ is non-duplicating	Acyclic FBoxes with non-duplicating, poly.-time-comp. operations	Acyclic FBoxes with poly.-time-comp. operations	Linear FBoxes
$\mathrm{Int}(\mathbb{R})$	Yes	P-admissible	EXP-admissible	P-admissible
$\mathrm{CGon}(\mathbb{R}^2)$	Yes	P-admissible	EXP-admissible	?

Computational Complexity

L	$\wedge$ is non-duplicating	Acyclic FBoxes with non-duplicating, poly.-time-comp. operations	Acyclic FBoxes with poly.-time-comp. operations	Linear FBoxes
$\mathrm{Int}(\mathbb{R})$	Yes	P-admissible	EXP-admissible	P-admissible
$\mathrm{CGon}(\mathbb{R}^2)$	Yes	P-admissible	EXP-admissible	?
$\mathrm{UGon}(\mathbb{R}^2)$	No	EXP-admissible	EXP-admissible	?
$\mathrm{FA}(\Sigma)$	No	EXP-admissible	EXP-admissible	EXP-admissible
$\mathrm{DFA}(\Sigma)$	No	EXP-admissible	EXP-admissible	EXP-admissible
Graph	Yes	NP-admissible	EXP-admissible	?

Future Prospects

Future Prospects

- Non-local feature inclusions $f \leq H(R_1 \circ g_1, \dots, R_n \circ g_n)$ where the R_i are role chains. Decidable? Complexity?

Example. $\mathsf{combinedWealth} \sqsubseteq \sum(\mathsf{hasAccount} \circ \mathsf{balance}) + \sum(\mathsf{holdsAsset} \circ \mathsf{value})$

- More results on cyclic FBoxes?
- Integration into existential rules, extension of the chase procedure.
- Empirical evaluation (with separation of logical and concrete reasoning).
- More hierarchical concrete domains of practical relevance?

Do you have questions or comments?