The Concrete Evonne: Visualization Meets Concrete Domain Reasoning (Extended Abstract)*

Christian Alrabbaa[1], Franz Baader[1], Raimund Dachselt[2], Alisa Kovtunova[1] and Julián Méndez[2]

[1]Institute of Theoretical Computer Science, TU Dresden, Germany
[2]Interactive Media Lab Dresden, TU Dresden, Germany

Keywords
Explainable AI, Concrete Domains, Visualization, Linear Equations, Difference Constraints

1. Description Logics, Concrete Domains and Combined Proofs

Symbolic AI approaches based on logic are inherently explainable—conclusions derived via automated reasoning can be traced through proofs showing step-by-step inference from axioms. Every inference step in symbolic reasoning carries explicit semantic meaning, enabling full transparency. However, to fully benefit from this, proofs must serve as effective explanations while remaining understandable. In ongoing work, we address these issues in the context of Description Logics (DLs) [1], proofs [2, 3] and our interactive visualization tool Evonne[1] [4, 5]. In this paper, we concentrate on the extension of the proof visualization facilities of Evonne to DLs with so-called concrete domains (CDs) [6, 7]. In particular, we consider extensions of tractable DLs of the $\mathcal{EL}$ family [8] with two p-admissible concrete domains based on rational numbers, one ($\mathcal{D}_{\mathbb{Q},lin}$) that can use linear equations to formulate constraints [9] and another ($\mathcal{D}_{\mathbb{Q},diff}$) based on difference constraints [10]. Such numerical constraints are useful for describing concepts whose definition involves quantitative information, as in the following examples.

Example 1. *For a delivery drone, its current* <u>b</u>attery <u>p</u>ercentage *is measured at checkpoints, denoted as* bp_1 *and* bp_2*. Additionally, the following safety constraints are imposed:* $bp_1 - 0.2 = bp_2$, $bp_1 > 0.3$ *and* $bp_2 > 0.25$*. If the initial percentage (*bp_1*) equals* 0.4*, then not all the constraints hold, and the drone is not permitted to fly—see Figure 1a.*

Example 2. *Assume* nr *and* hr *represent the average* <u>n</u>ormal *and* <u>h</u>igh *battery* discharge <u>r</u>ates*. Under normal conditions, the* <u>d</u>elivery <u>d</u>rone *(*DD*) can fly for 8 hours on a single charge with a* $30Ah$ *battery, i.e.,* $8nr = 30$*. In cold conditions, one hour of flight increases the battery consumption such that* $4nr + hr = 30$*. Next, if a system consumes* $30Ah$ *in two hours at a high discharge rate, it qualifies as a* <u>l</u>arge <u>b</u>attery <u>d</u>rone *(*LBD*), i.e.,* $[2hr = 30] \sqsubseteq \mathsf{LBD}$*. Given these constraints, it follows that the delivery drone is a large battery drone, i.e.,* $\mathsf{DD} \sqsubseteq \mathsf{LBD}$*—see the combined proof in Figure 1b, where the proof in Figure 1c shows the* $\mathcal{D}_{\mathbb{Q},lin}$ *inferences.*

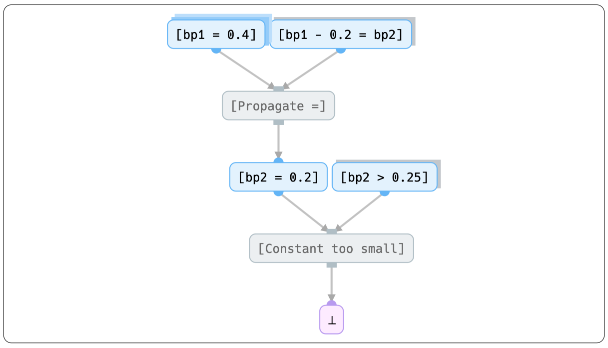
Essentially, the CD $\mathcal{D}_{\mathbb{Q},diff}$ contains predicates $x = q$, $x > q$, and $x + q = y$, and the $\mathcal{D}_{\mathbb{Q},lin}$ predicates are given by linear equations $\sum_{i=1}^{n} a_i x_i = b$, for constants $q, a_i, b \in \mathbb{Q}$, with their natural

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

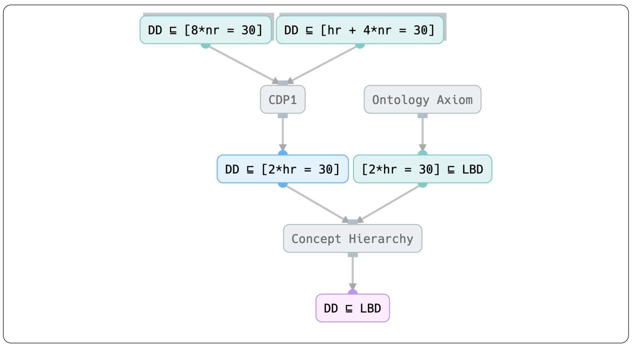
christian.alrabbaa@tu-dresden.de (Christian Alrabbaa); franz.baader@tu-dresden.de (Franz Baader); raimund.dachselt@tu-dresden.de (Raimund Dachselt); alisa.kovtunova@tu-dresden.de (Alisa Kovtunova); julian.mendez2@tu-dresden.de (Julián Méndez)
0000-0002-2925-1765 (Christian Alrabbaa); 0000-0002-4049-221X (Franz Baader); 0000-0002-2176-876X (Raimund Dachselt); 0000-0001-9936-0943 (Alisa Kovtunova); 0000-0003-1029-7656 (Julián Méndez)

*This is an extended abstract of a paper accepted at the 15th International Symposium on Frontiers of Combining Systems (FroCoS 2025).

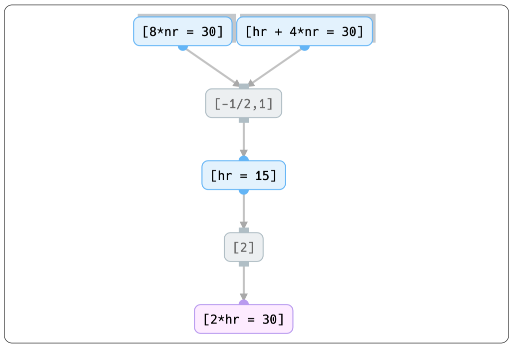
[1]Evonne's source code, documentation, and evaluation material (user studies, benchmark) are available at: https://imld.de/evonne.



(a) $\mathcal{D}_{\mathbb{Q},\mathit{diff}}$ proof from Example 1



(b) Combined proof from Example 2



(c) Expanding CDP1 in (b)

Figure 1: Examples of proofs in Evonne.

semantics [8, 9]. An *implication* is of the form $\mathfrak{C} \to \beta$, where β is a constraint, i.e., a predicate with variables as arguments, and $\mathfrak{C}$ is a set of constraints. The implication is *valid* if all variable assignments satisfying all constraints in $\mathfrak{C}$ also satisfy β. A set $\mathfrak{C}$ is *unsatisfiable* iff $\mathfrak{C} \to \bot$ is valid.

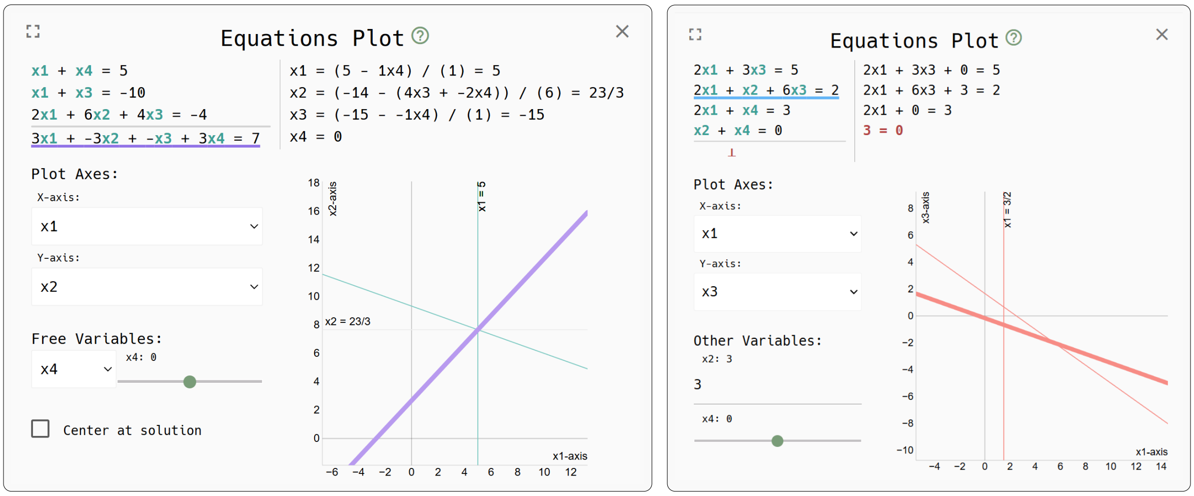
In [10], we have *theoretically* addressed the problem of generating proofs for consequences derived from knowledge bases formulated in such DLs. Here we contribute the *practical* proof visualization tool supporting DLs with linear equations and difference constraints.

2. Domain-Specific Visualizations

This work introduces novel visual explanations for unsatisfiability and entailment in the considered CDs, designed to reflect their unique properties and offer more intuitive insight into the underlying numerical reasoning.

Case $\mathcal{D}_{\mathbb{Q},\mathbf{lin}}$. If we consider systems of linear equations with at most two variables, which are shared across all equations, then we can use *lines* in the 2D *Euclidean Space* to achieve a compact representation of all solutions to the equations, as each solution corresponds to a point in the 2D space. Using *dimensionality reduction*, we can apply the same idea to equations with more than two variables. Let $\mathfrak{C}$ be a set of equations where $\mathfrak{C} \to \beta$ holds, let s be a solution to $\mathfrak{C}$, and let x and y be any two variables appearing in β. By replacing all the variables, except x and y, in all equations with their corresponding values in s, we obtain equations involving at most two variables. Therefore, in the xy-plane all the lines must intersect in the same point. To explain unsatisfiability, we can use a similar approach.

Figure 2 shows examples of $\mathcal{D}_{\mathbb{Q},\mathit{lin}}$ explanations in Evonne. Users can toggle between planes and assign values to free variables. The equation system appears top-left, with current variable assignments shown top-right. Since the system in Figure 2a has one degree of freedom, setting $x4 = 0$ uniquely determines the remaining variables. In contrast, in Figure 2b, the top right shows the system of



(a) Implication of $3x_1 - 3x_2 - x_3 + 3x_4 = 7$ (b) Implication of $\bot$

Figure 2: Examples of explanations for $\mathcal{D}_{\mathbb{Q},lin}$ implications in Evonne

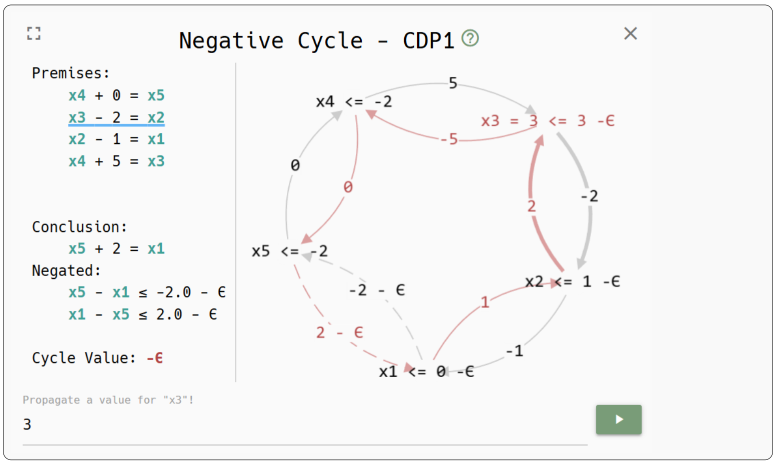


Figure 3: Example of an explanation for $\mathcal{D}_{\mathbb{Q},diff}$ implications in Evonne.

equations with respect to the currently chosen variable assignment. In this case the visualization makes contradictions immediately apparent—as demonstrated by the highlighted contradiction $3 = 0$.

Case $\mathcal{D}_{\mathbb{Q},\mathbf{diff}}$. It is well established [11] that *difference constraints* (i.e., $x - y \leq q$) can be represented as weighted directed graphs such that every variable corresponds to a vertex and every constraint to a weighted edge between x and y labelled by q. Given a set of difference constraints, deciding whether it is *unsatisfiable* can be reduced to finding a *simple cycle* in the corresponding graph with a *negative weight* [11].

We show how to rewrite all types of constraints in $\mathcal{D}_{\mathbb{Q},diff}$ into difference constraints. Thus, we can represent any set $\mathfrak{C}$ of $\mathcal{D}_{\mathbb{Q},diff}$ constraints as a difference graph, and if $\mathfrak{C} \to \bot$, we can explain the contradiction in $\mathfrak{C}$ by identifying the negative cycle in the graph. In the case when $\mathfrak{C}$ is satisfiable and $\mathfrak{C} \to \beta$, we can explain the implication by showing that $\mathfrak{C} \cup \{\neg\beta\} \to \bot$, which allows us to effectively use the notion of negative cycles.

Figure 3 shows a screenshot of a negative cycle in Evonne. In the implementation, the cycle is animated, allowing users to visually follow its progression. Additionally, users can assign concrete values to variables. These values are then automatically propagated along the negative cycle, allowing users to observe how the set of constraints behaves under such assignments. In particular, this makes it possible to see exactly how the cycle leads to logical inconsistencies, which are highlighted in red, e.g., $x_3 = 3 \leq 3 - \epsilon$.

Empirical Evaluation. We conducted an empirical validation through user studies and benchmarks, to demonstrate the effectiveness of our approach. The creation of these visualizations remains performant (under 200ms) with the largest proofs in our experiment, and the studies revealed areas of improvement for the visualization of proofs and our alternative CD explanations. The current version of Evonne is accessible through https://imld.de/evonne, further details of our evaluation results are also available there.

3. Conclusion

The latest extension of Evonne enables interactive visualization of proofs for DLs with concrete domains—crucial for modeling concepts involving quantitative constraints. This extension contributes: (1) the *first* proof visualization tool supporting DLs with linear equations and difference constraints, (2) novel domain-specific visual explanations tailored to enhance comprehension of numerical reasoning, and (3) empirical validation through user studies and benchmarks. Our assessments showed that the proposed explanations support users in understanding conclusions. When comparing proofs with domain-specific visual explanations, participants' opinions varied for $\mathcal{D}_{\mathbb{Q},lin}$, though most alluded to a trust factor favoring proofs over visual explanations, suggesting that, in this case, such visual explanations might not be necessary. In contrast, for $\mathcal{D}_{\mathbb{Q},diff}$, participants highly valued the clarity and ease of understanding provided by the animated cycles, making them a more preferred form of explanation. As future work, we plan to address issues raised by the participants' feedback on both proofs and the visual CD explanations.

Acknowledgments

This work is funded by DFG under Germany's Excellence Strategy: EXC 2050/1, 390696704 – "Centre for Tactile Internet" (CeTI); by DFG grant 389792660 as part of TRR 248 – CPEC; and by BMBF and Saxon State Ministry for Science, Culture and Tourism (SMWK) in Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI, SCADS22B).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4 [https://chat.openai.com/] and DeepSeek-V3 [https://chat.deepseek.com/] in order to: Grammar and spelling check, Paraphrase and reword. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge Univ. Press, 2017. doi:10.1017/9781139025355.

[2] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding small proofs for description logic entailments: Theory and practice, in: LPAR 2020: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Alicante, Spain, May 22-27, 2020, volume 73 of *EPiC Series in Computing*, EasyChair, 2020, pp. 32–67. doi:10.29007/NHPP.

[3] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding good proofs for description logic entailments using recursive quality measures, in: Automated Deduction - CADE 28 - 28th

International Conference on Automated Deduction 2021, Proceedings, volume 12699 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 291–308. doi:10.1007/978-3-030-79876-5_17.
[4] C. Alrabbaa, F. Baader, S. Borgwardt, R. Dachselt, P. Koopmann, J. Méndez, Evonne: Interactive proof visualization for description logics (system description), in: Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Proceedings, volume 13385 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 271–280. doi:10.1007/978-3-031-10769-6_16.
[5] J. Méndez, C. Alrabbaa, P. Koopmann, R. Langner, F. Baader, R. Dachselt, Evonne: A visual tool for explaining reasoning with OWL ontologies and supporting interactive debugging, Comput. Graph. Forum 42 (2023). doi:10.1111/CGF.14730.
[6] F. Baader, P. Hanschke, A scheme for integrating concrete domains into concept languages, in: Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991, Morgan Kaufmann, 1991, pp. 452–457. URL: http://ijcai.org/Proceedings/91-1/Papers/070.pdf.
[7] C. Lutz, Description logics with concrete domains-a survey, in: Advances in Modal Logic 4, papers from the fourth conference on "Advances in Modal logic," held in Toulouse, France, 30 September - 2 October 2002, King's College Publications, 2002, pp. 265–296. URL: http://www.aiml.net/volumes/volume4/Lutz.ps.
[8] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005, Professional Book Center, 2005, pp. 364–369. URL: http://ijcai.org/Proceedings/05/Papers/0372.pdf.
[9] F. Baader, J. Rydval, Using model theory to find decidable and tractable description logics with concrete domains, J. Autom. Reason. 66 (2022) 357–407. doi:10.1007/s10817-022-09626-2.
[10] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Combining proofs for description logic and concrete domain reasoning, in: Rules and Reasoning - 7th International Joint Conference, RuleML+RR 2023, Proceedings, volume 14244 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 54–69. doi:10.1007/978-3-031-45072-3_4.
[11] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, third edition, Computer science, MIT Press, 2009. URL: https://books.google.de/books?id=i-bUBQAAQBAJ.