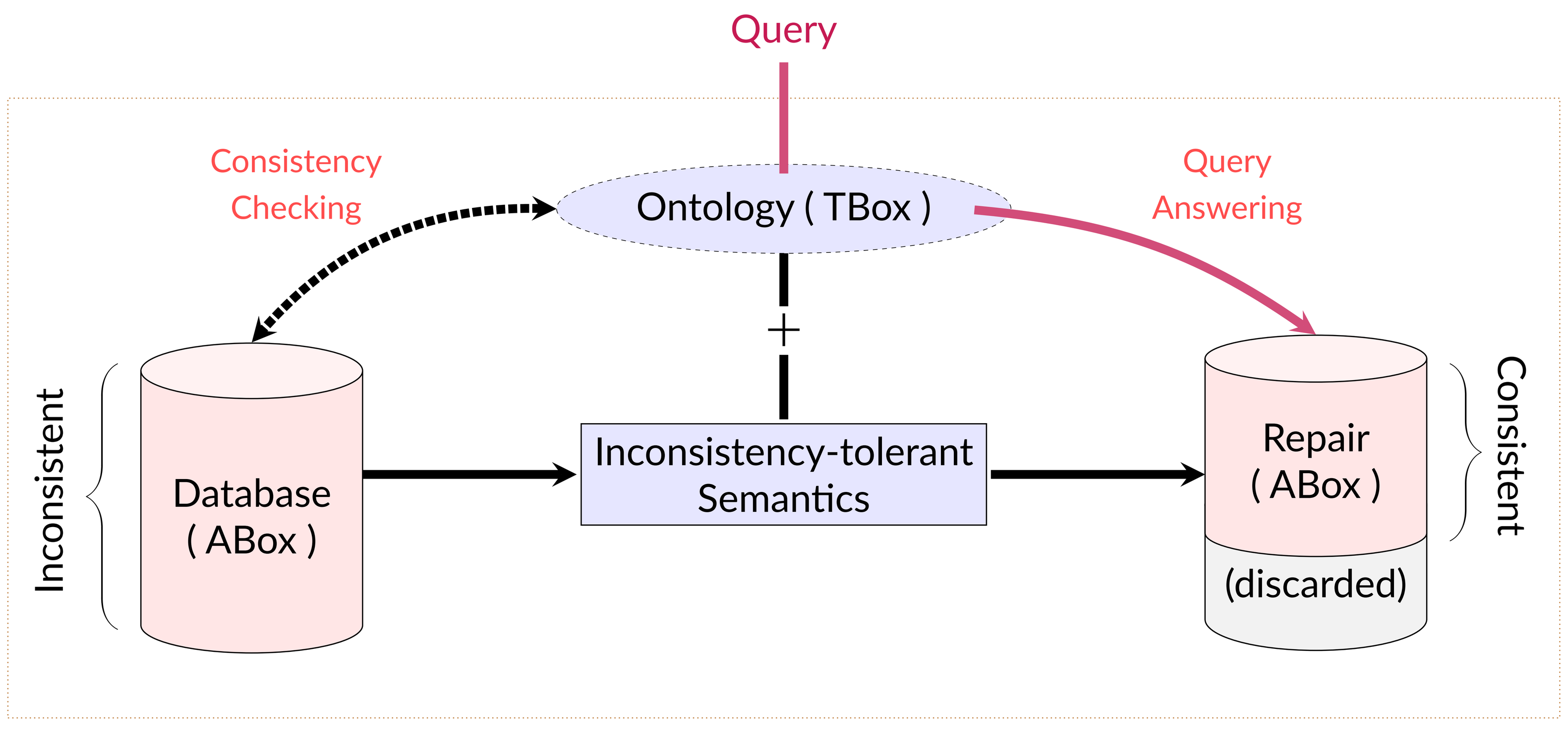
DL 2024: 37th International Workshop on Description Logics

On the Computation of a Productive Partially Ordered Possibilistic Repair

Ahmed Laouar[1] Sihem Belabbes[2] Salem Benferhat[1]

[1] CRIL, Univ. Artois & CNRS, Lens, France, [2] LIASD, IUT de Montreuil, Univ. Paris 8, Saint-Denis, France

Context: Inconsistent DL-Lite Knowledge Bases

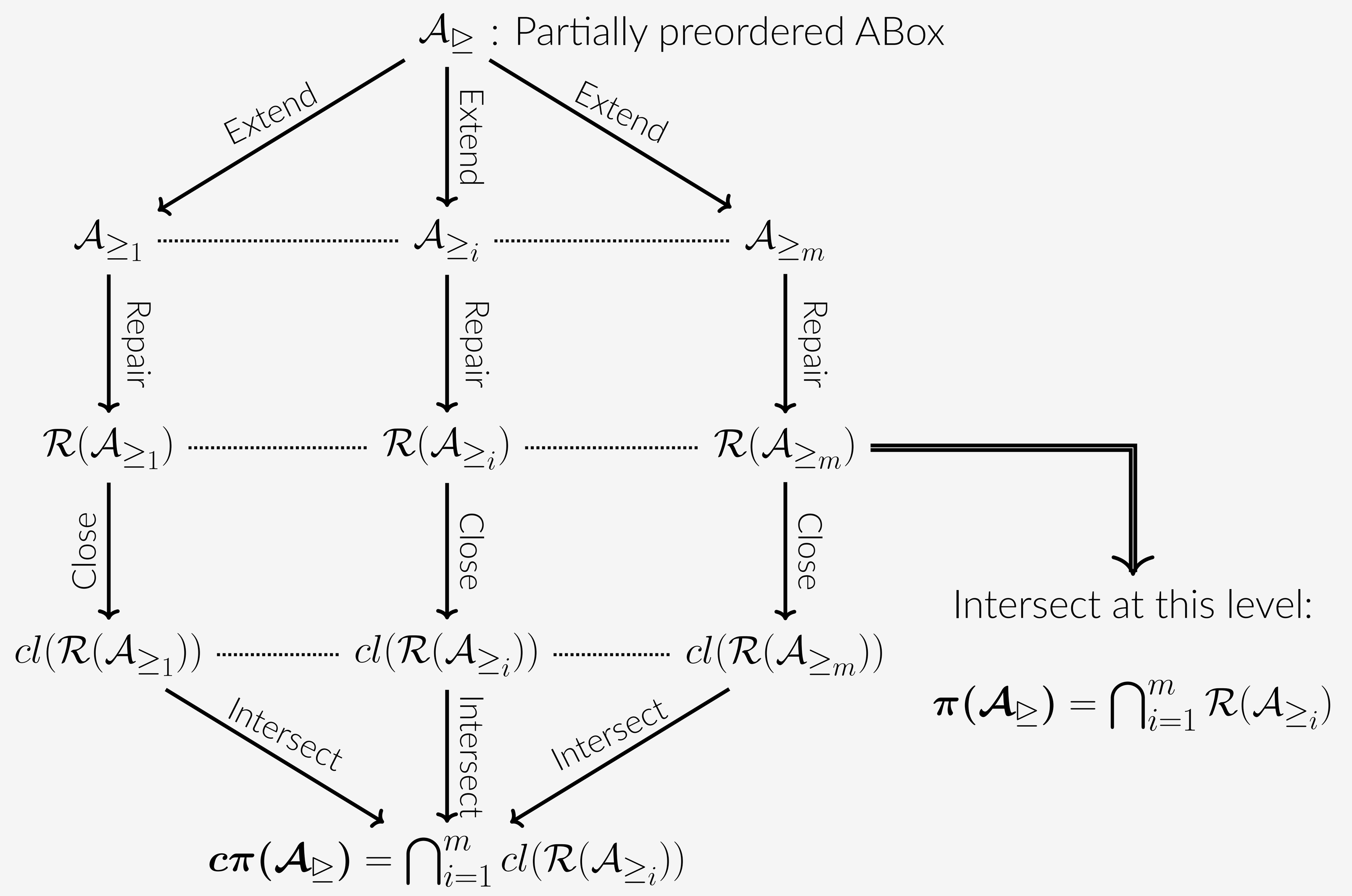


Repair ≡ Compute a Consistent Subset

Which repairing semantics for partially ordered data?

- Should an assertion that is less preferred to a conflict be considered in a repair?
- Possiblistic setting: NO! $\Longrightarrow$ a **safe** and **cautious** approach.

Semantics of the Closure-Based Partially Ordered Possibilistic Repair

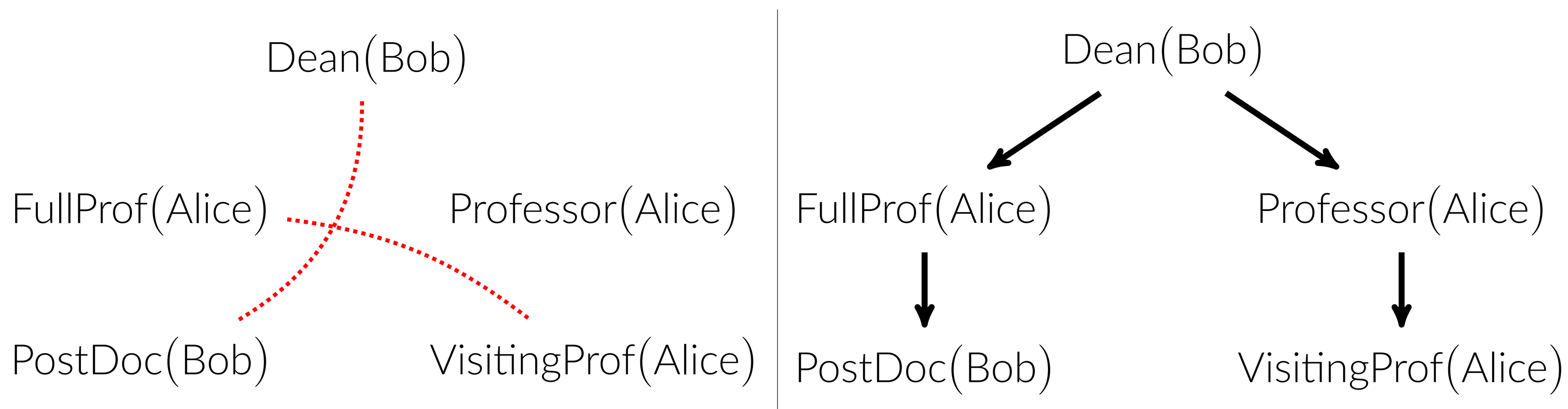


- $\mathcal{R}(\mathcal{A}_{\geq i})$: standard possibilistic repair.
- Checking if $\varphi \in \pi(\mathcal{A}_{\trianglerighteq}) \Longrightarrow$ consistency checking of a subset of $\mathcal{A}_{\trianglerighteq}$.

$$\pi(\mathcal{A}_{\trianglerighteq}) \subseteq cl(\pi(\mathcal{A}_{\trianglerighteq})) \subseteq c\pi(\mathcal{A}_{\trianglerighteq})$$

Example

$$\mathcal{T} = \left\{ \begin{array}{l} \text{FullProf} \sqsubseteq \neg\,\text{VisitingProf},\ \text{PostDoc} \sqsubseteq \neg\,\text{Dean}, \\ \text{FullProf} \sqsubseteq \text{Professor},\ \text{Professor} \sqsubseteq \text{FacMember},\ \text{Dean} \sqsubseteq \text{FacMember} \end{array} \right\}$$



(a) ABox Conflicts (b) Partially Ordered ABox

$\longrightarrow$: strict preference, – – – : conflict.

$$c\pi(\mathcal{A}_{\trianglerighteq}) = \{\text{Dean(Bob)}, \text{Professor(Alice)}, \text{FacMember(Bob)}, \text{FacMember(Alice)}\}$$

$$\pi(\mathcal{A}_{\trianglerighteq}) = \{\text{Dean(Bob)}, \text{FacMember(Bob)}\}$$

Support

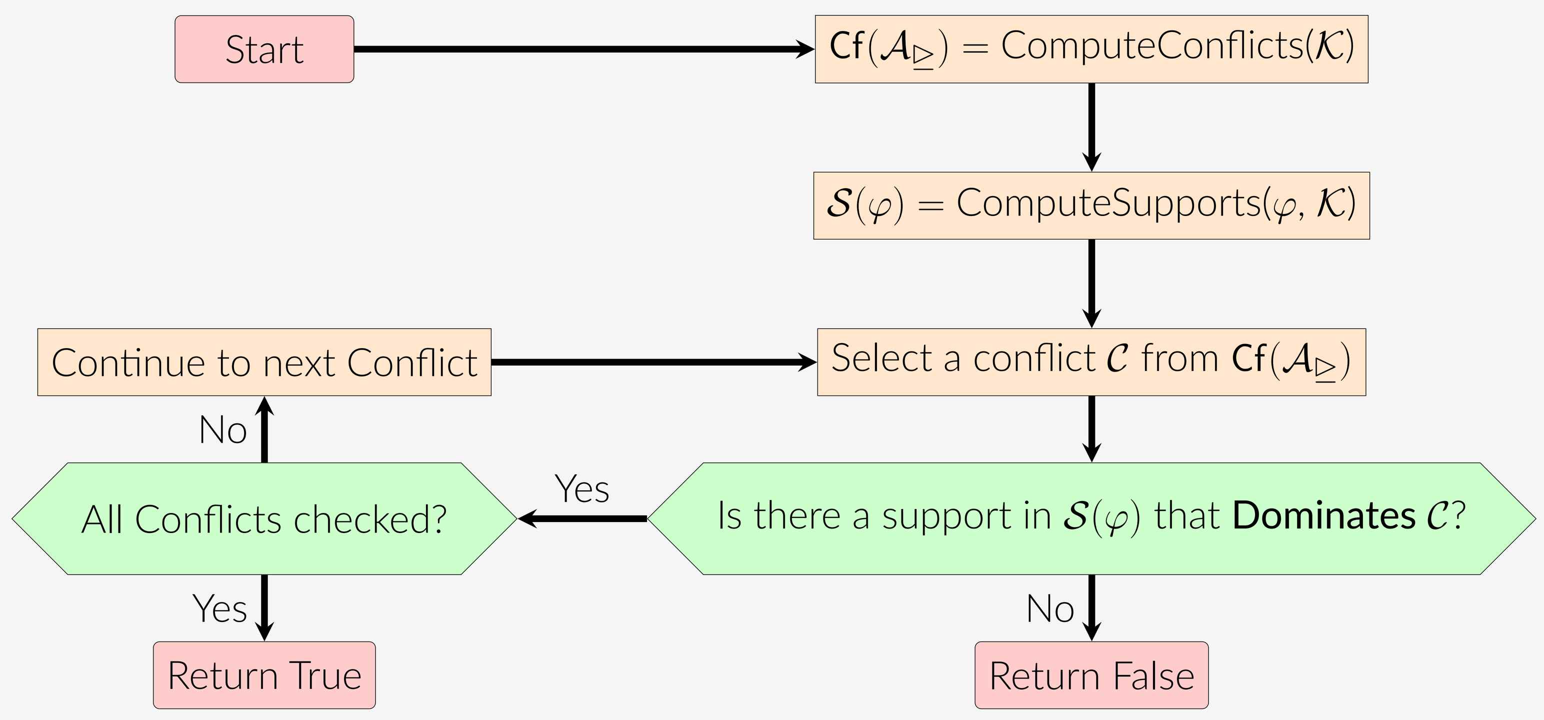
- The **support** of an assertion is a **minimal consistent subset** of the ABox that allows to derive it.
- {Dean(Bob)} **supports** {FacMember(Bob)} because: $\langle \mathcal{T}, \{\text{Dean(Bob)}\} \rangle \models$ FacMember(Bob).

Dominance

- $\mathcal{B}_1$ **dominates** $\mathcal{B}_2$ if for all $\varphi_j \in \mathcal{B}_1$, there is $\varphi_k \in \mathcal{B}_2$ such that $\varphi_j \rhd \varphi_k$.
- {Dean(Bob), FullProf(Alice)} **dominates** {PostDoc(Bob), VisitingProf(Alice)}.
- {Dean(Bob), Professor(Alice)} does not **dominate** {FullProf(Alice), PostDoc(Bob)}.

Sound and Complete Tractable Characterization

Check-in-Cπ-repair($\varphi \in cl(\mathcal{A}_{\trianglerighteq})$: assertion, $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$: KB)

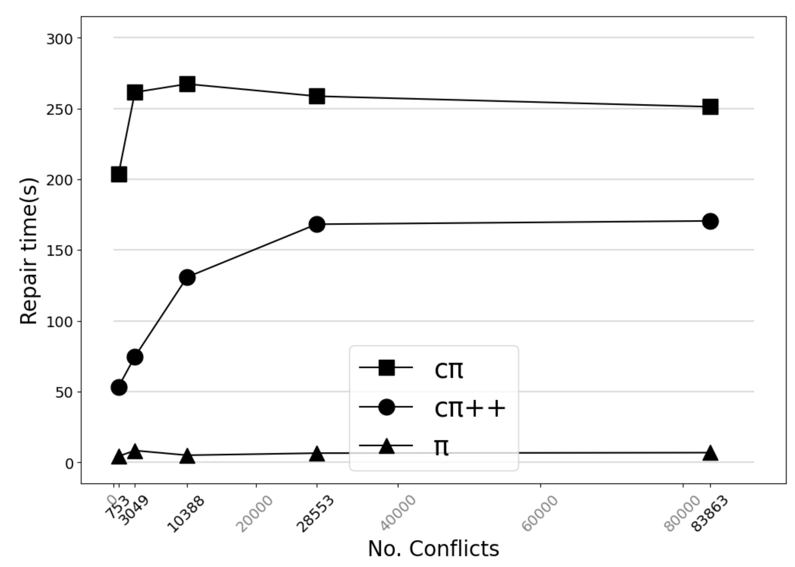


Compute $c\pi(\mathcal{A}_{\trianglerighteq})$: apply Check-in-C$\pi$-repair() on the Closed ABox $cl(\mathcal{A}_{\trianglerighteq})$

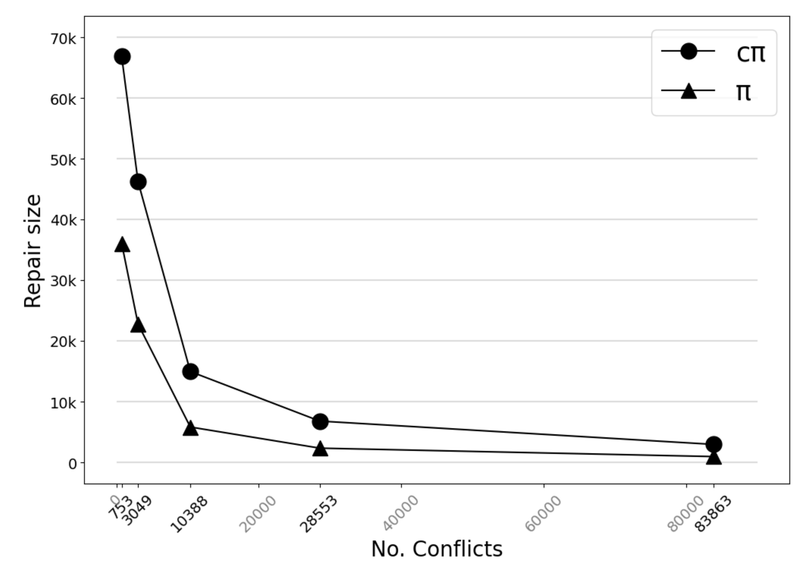
Two Main Optimisations

- Focusing on **dominant** conflicts and supports:
  1. If $\mathcal{C}_i$ **dominates** $\mathcal{C}_j$ then it is enough to check $\mathcal{C}_i$.
  2. If $\mathcal{S}_i(\varphi)$ **dominates** $\mathcal{S}_j(\varphi)$ then it is enough to check $\mathcal{S}_i(\varphi)$.
- Incrementally computing Cπ-repair ($c\pi$++):
  1. Compute $\pi(\mathcal{A}_{\trianglerighteq})$ and $cl(\pi(\mathcal{A}_{\trianglerighteq}))$.
  2. For all $\varphi \notin cl(\pi(\mathcal{A}_{\trianglerighteq}))$: if $|\mathcal{S}(\varphi)| = 1$ then $\varphi \notin c\pi(\mathcal{A}_{\trianglerighteq})$.
  3. Check the remaining assertions with the tractable characterization.

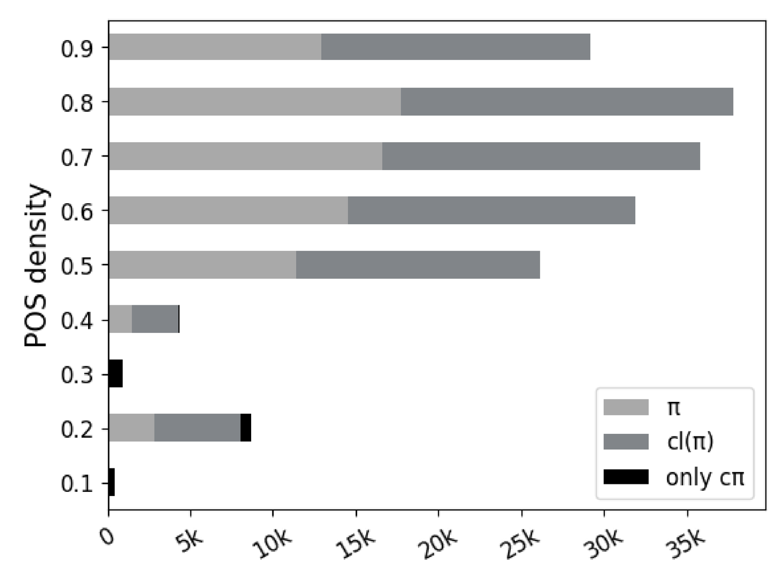
Experimental Studies



(a) Evolution of repair time w.r.t. No. conflicts



(b) Evolution of repair size w.r.t. No. conflicts



(c) Proportion of assertions in each repair vs partial order density

- High density approximates a total order.
- Low density $\Longrightarrow$ more incomparabilities.
- A higher chance to have a larger Cπ-repair.

Important Results and Conclusion

- **Productive** repair: using the Deductive Closure.
- **Tractable** repair: in polynomial time in ABox's size.
- A more **efficient** algorithm in terms of execution time.
- Cπ-repair benefits vs. density of the partial order.
- How to avoid computing the conflicts in Cπ-repair?

https://github.com/ahmedlaouar/py_reasoner