Fixpoint Semantics for Recursive SHACL

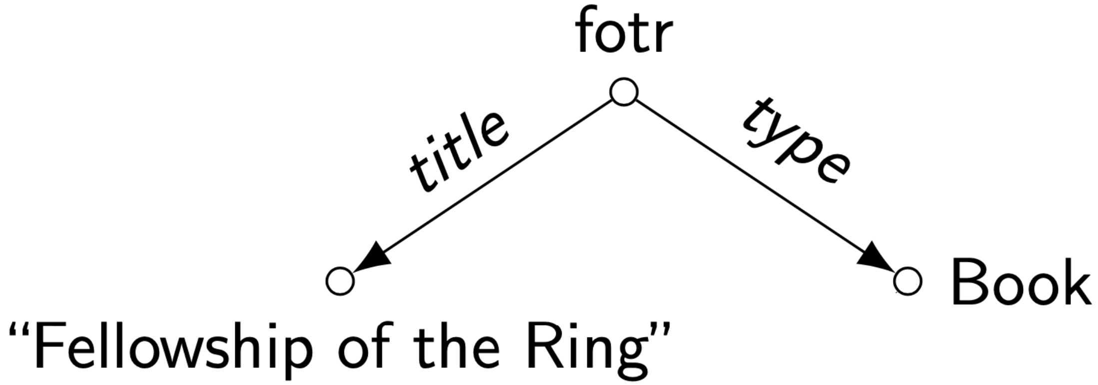
Presented at ICLP 2021

Bart Bogaerts &

Maxime Jakubowski

SHACL

- **Sha**pes **C**onstraint **L**anguage
- Constraint language for RDF graphs
- Conformance checking



:BookShape
  a sh:PropertyShape;
  sh:path :title;
  sh:minCount 1.

:BookShape sh:targetClass :Book.

$\exists \text{type. hasValue(Book)} \subseteq \exists \text{title.} \top$

Shapes

Let N, P and S be disjoint universes of node names, property names and shape names.

The language L

$$\phi \coloneqq \top \mid \text{hasValue}(c) \mid \text{hasShape}(s) \mid \text{eq}(E, p) \mid \text{disj}(E, p) \mid \text{closed}(Q)$$

$$\mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \forall E.\phi \mid \ \geq_n E.\phi \mid \ \leq_n E.\phi$$

$$E \ \coloneqq p \mid p^- \mid E \cup E \mid E/E \mid E^* \mid E?$$

where $c \in N, p \in P, s \in S$ and $Q \subseteq P$

E are regular path expressions

We will use the symbol $\exists$ to abbreviate $\geq_1$

Interpretations for shapes

… have the following components:

- An (infinite) domain: Δ^I
- For each constant c, an element $c^I \in \Delta^I$
- For each shape name s, a set $s^I \subseteq \Delta^I$
- For each property name p, a set $p^I \subseteq \Delta^I \times \Delta^I$

An RDF graph G is a specific interpretation where:

- $\Delta^I = N$ (the universe of all nodes)
- $c^I = c$ for every node name $c \in N$
- $p^I = p^G$, for every property name $p \in P$

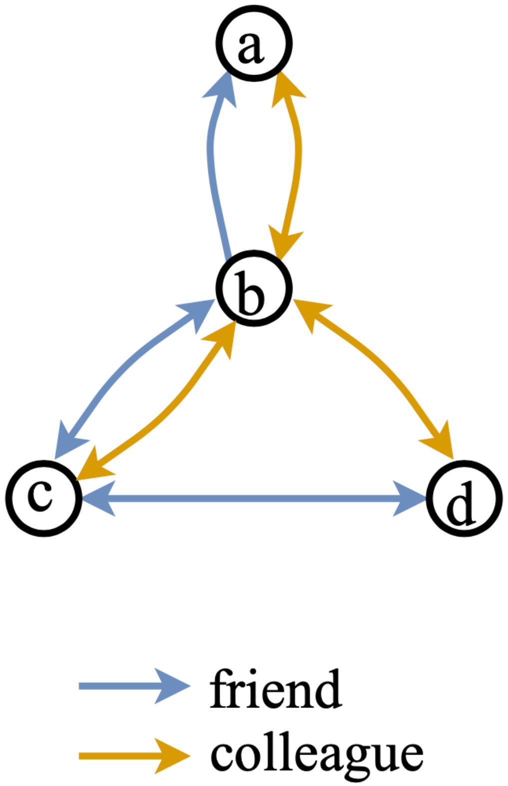
$\boldsymbol{\phi}$	$\boldsymbol{I, a \vDash \phi}$ **if:**
$\mathrm{hasValue(a)}$	$a = [\![c]\!]^I$
$\geq_n E.\psi$	$\#\ \{b \in [\![E]\!]^I(a) \mid I, b \vDash \psi\} \geq n$
$eq(F, p)$	The sets $[\![F]\!]^I(a)$ and $[\![p]\!]^I(a)$ are equal
$disj(F, p)$	The sets $[\![F]\!]^I(a)$ and $[\![p]\!]^I(a)$ are disjoint
$closed(R)$	$[\![p]\!]^I(a)$ is empty for each $p \in \Sigma - R$

Example shapes

- *"Through a path of **friend** edges, the node can reach node d"*
  - **FriendOfD** $\leftarrow \quad \exists \text{friend}^*.\,\text{hasValue}(d)$
  - b, c, and d satisfy **FriendOfD** in G

- *"Nodes where **friend**ship is mutual"*
  - **MutualFriends** $\leftarrow \text{eq}(\text{friend}, \text{friend}^-)$
  - c and d satisfy **MutualFriends** in G

- *"Nodes who have at least one **colleague** who is also a **friend**"*
  - **ColleagueFriend** $\leftarrow \neg\text{disj}(\text{friend}, \text{colleague})$
  - b and c satisfy **ColleagueFriend** in G



Shape schemas

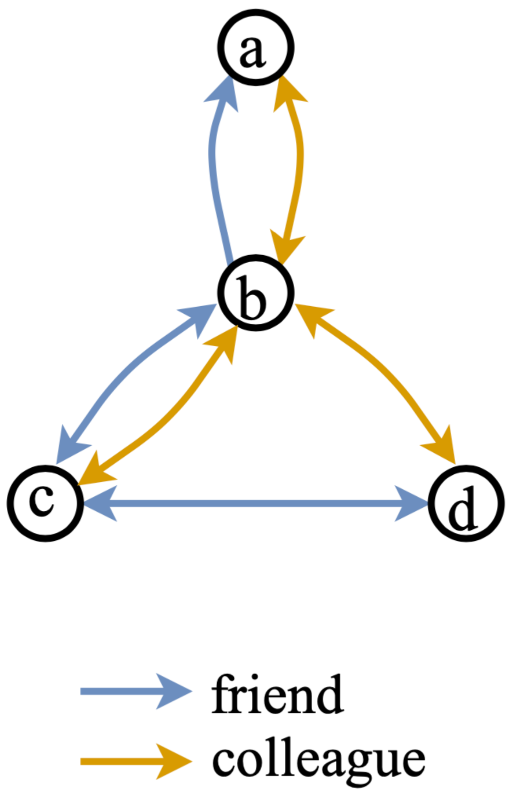
The main task is to check whether a **graph** conforms to some constraints, not single nodes.

Shape definition: $s \leftarrow \phi$

Target statement: $\phi_t \subseteq \phi_s$

Example schema (Def, T):

- Def: $\mathbf{FriendOfD} \leftarrow \exists \mathrm{friend}^*.\mathrm{hasValue}(d)$
- T: $\exists \mathrm{friend}.\top \subseteq \mathbf{FriendOfD}$



Recursion

Given an interpretation (associated with a graph) G, and a schema (Def, T)

⇨ AFT as a tool to define our recursive semantics

Two-valued lattice L:

- the set of interpretations that *expand* G (N and P are fixed, so they expand S)
- truth order $I_1 \leq_t I_2$ if $s^{I_1} \subseteq s^{I_2}$ for all $s \in S$
- semantic operator: $T_{Def}(I)(s) ::= \phi^I$ for each defining rule $s \leftarrow \phi \in \mathrm{Def}$

Three-valued lattice L^c:

- the set of pairs of interpretations $\mathrm{J} = (J_1, J_2)$ such that $J_1 \leq_t J_2$
- three-valued interpretations associate with every $s \in \mathrm{S}$ a mapping $\Delta \mapsto \{\boldsymbol{t}, \boldsymbol{f}, \boldsymbol{u}\}$:
  - a maps to $\boldsymbol{t}$ if a in s^{J_1} to **f** if a not in s^{J_2}, and to $\boldsymbol{u}$ otherwise
  - we extend this evaluations to shapes ϕ (straightforward extension of Kleene's truth tables)
- semantic operator: $\Phi_{Def}(J)(s) ::= \phi^J$

Is that all?

We only needed to:

- Choose an order on our two-valued lattice
- Define the three-valued evaluation

We get:

- Well defined semantics for recursive SHACL
- Theorethical body of results coming from AFT, now applicable to SHACL

Brave vs Cautious validation

There may be multiple (expanded) models for a given graph G and schema $(\mathrm{Def}, \mathrm{T})$.

- **Brave** validation: *one* such model must satisfy T
- **Cautious** validation: *every* model must satisfy T

$\ldots$ makes a difference for stable and supported model semantics

Existing Semantics

[Corman 2018] defined supported model semantics (CRS-supported)

- Already defined the three-valued semantic operator Φ_{Def}
- …but only characterized supported models for (brave) validation

Theorem CRS-supported models coincide with the AFT-supported models

⇨ we agree with the literature

Existing Semantics

[Andreşel 2020] defined stable model semantics (ACORSS-stable)

- Defined in terms of ‘level-mappings’
- Focus on translation to ASP

Theorem Every AFT-stable model is a ACORSS-stable model. If Def is in *negation normal form*, the converse also holds.

… where does it go wrong?

ACORSS-stable ≠ AFT-stable

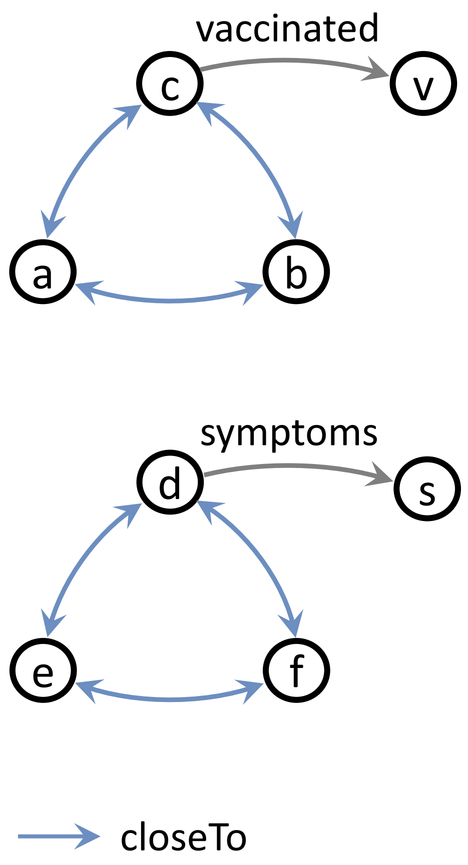
*“You are **safe** if you are vaccinated or you are close to at most one person who is not safe.”*

$\mathbf{Safe} \leftarrow \exists\text{vaccinated}.\top \vee\ \leq_1\text{closeTo}.\neg\text{hasShape}(\mathbf{Safe})$

AFT-stable has only one model: $\{\ \mathbf{Safe}(a),\ \mathbf{Safe}(b),\ \mathbf{Safe}(c)\ \}$

ACORSS-stable has an additional model where everyone is safe.

⇨ AFT gives us a more intuitive semantics here



Concluding remarks

- Our semantics often agrees with the proposed semantics from the literature
- Where the semantics differ, we argue our semantics is more intuitive
- The application of AFT in this context was natural
- We define new recursive semantics for SHACL (like the well-founded semantics)
- We supply a strong formal foundation for the study of recursive SHACL