The Expressive Power of Description Logics with Numerical Constraints over Restricted Classes of Models

Franz Baader Filippo De Bortoli

15th International Symposium on Frontiers of Combining Systems (FroCoS 2025)
Reykjavik, Iceland, September 29, 2025

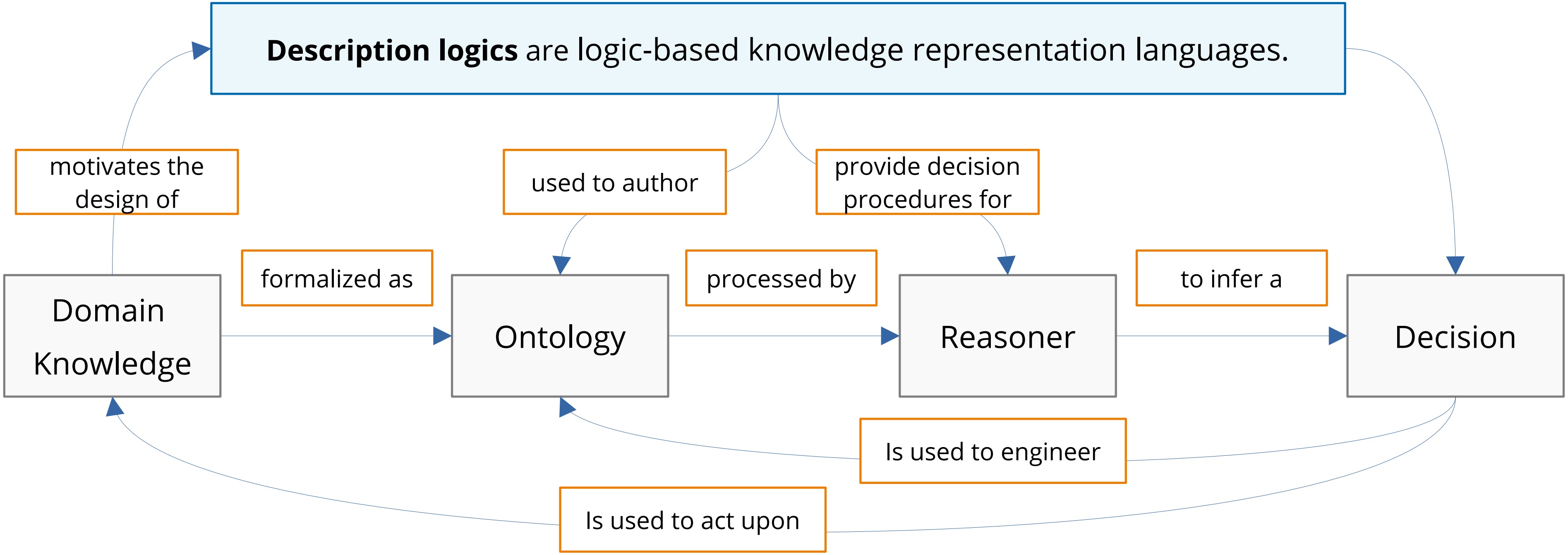
Technische Universität Dresden
ScaDS.AI DRESDEN LEIPZIG
CPEC CENTER FOR PERSPICUOUS COMPUTING

Description Logics: what are they?

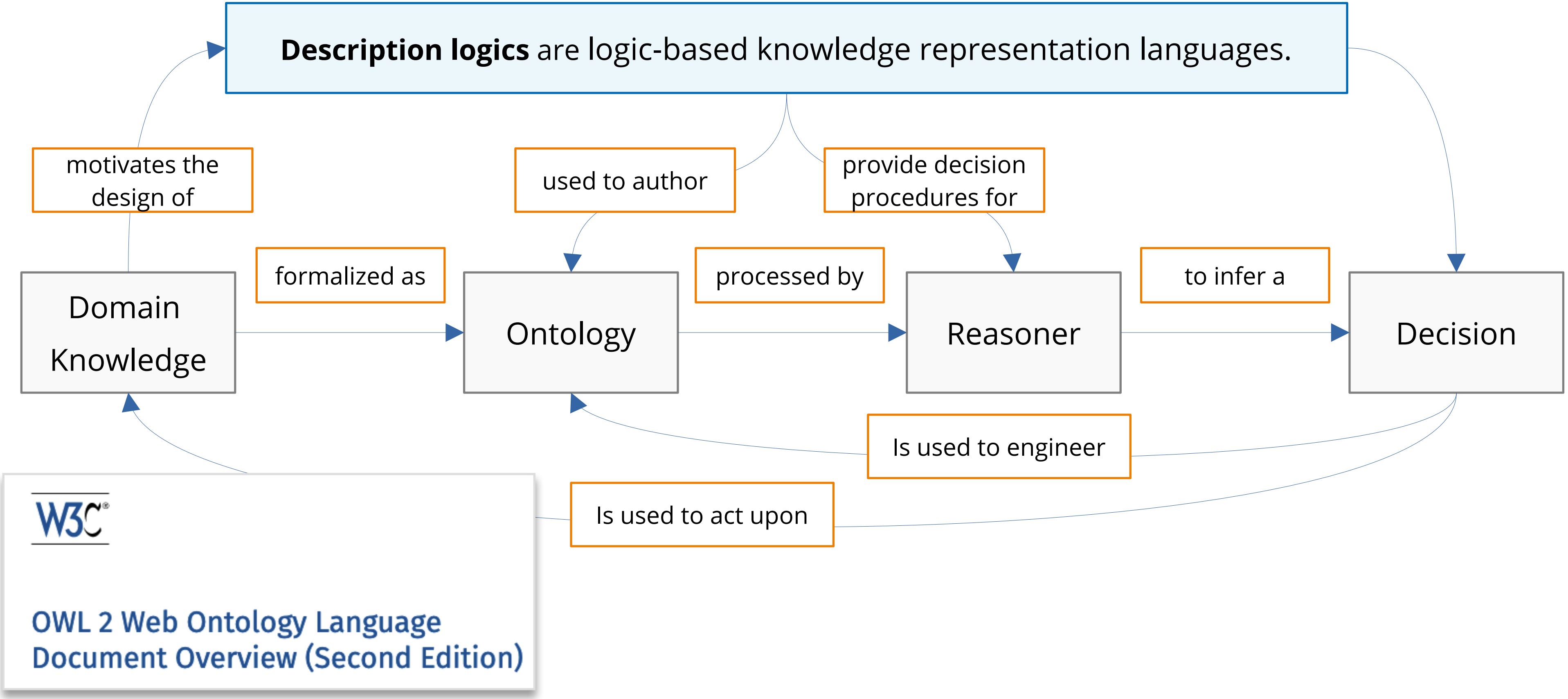
Description Logics: what are they?

Description logics are logic-based knowledge representation languages.

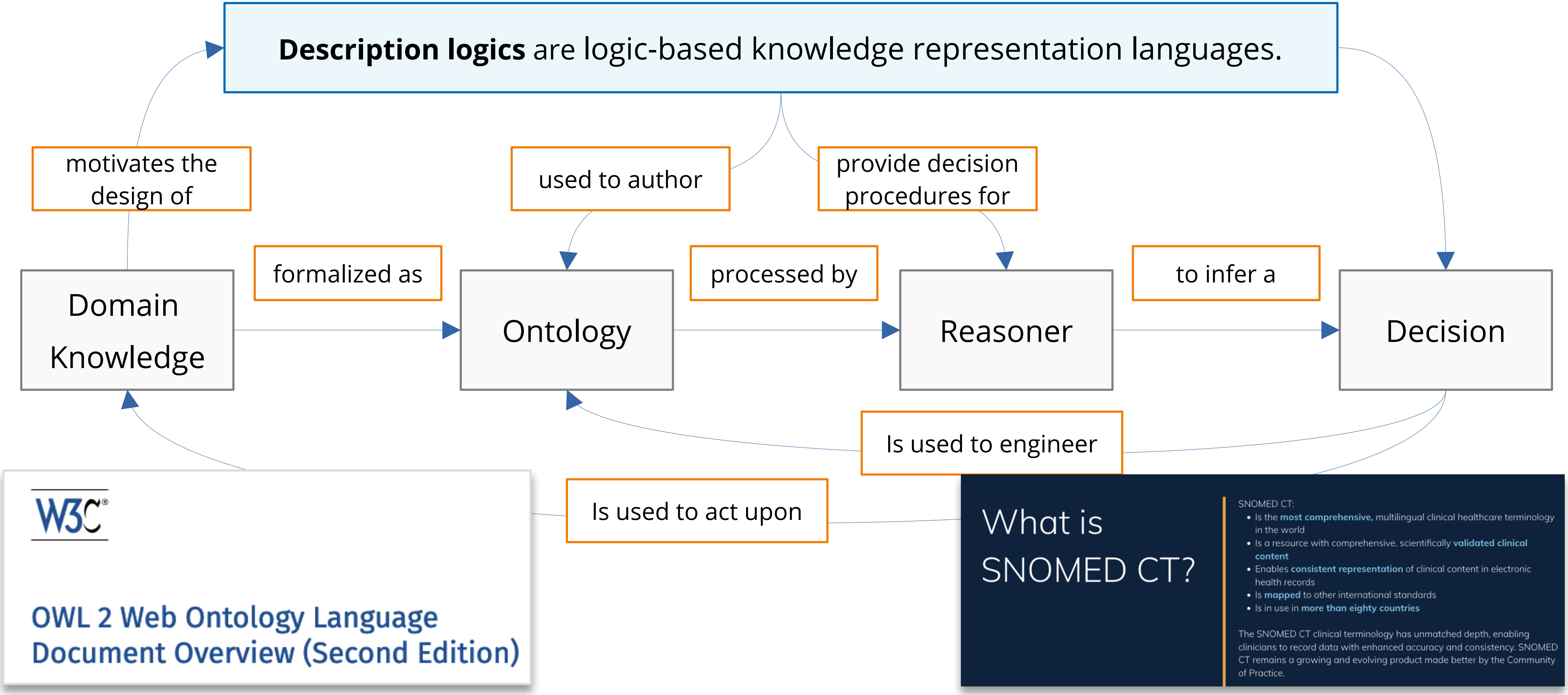
Description Logics: what are they?



Description Logics: what are they?

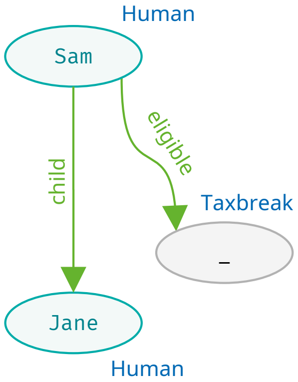


Description Logics: what are they?



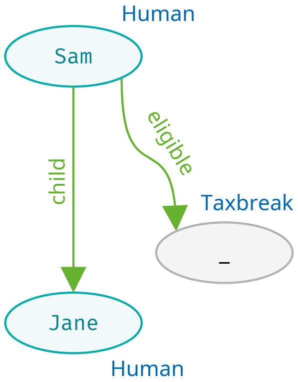
Basics of Description Logics

Basics of Description Logics



concept names (Human, TaxBreak), role names (child, eligible)

Basics of Description Logics



Concept descriptions

$C ::= A \mid \neg C \mid C \sqcap C \mid \exists r.C$ ← ALC

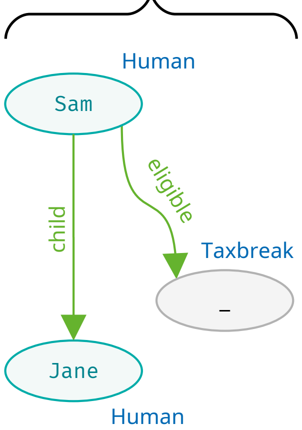
$$\overbrace{\text{Human} \sqcap \exists \text{child}.\text{Human}} \sqsubseteq \overbrace{\exists \text{eligible}.\text{TaxBreak}}$$

← **TBox** (*concept inclusions*)

concept names (Human, TaxBreak), role names (child, eligible)

Basics of Description Logics

interpretation



$\models$

Concept descriptions

$$\underbrace{\text{Human} \sqcap \exists \text{child}.\text{Human}} \sqsubseteq \underbrace{\exists \text{eligible}.\text{TaxBreak}}$$

$C ::= A \mid \neg C \mid C \sqcap C \mid \exists r.C$ ← ALC

TBox (*concept inclusions*)

TBox consistency is ExpTime-complete for ALC ontologies.

concept names (Human, TaxBreak), role names (child, eligible)

Cardinality constraints in description logics

Cardinality constraints in description logics

Qualified number restriction

ALC + $(\geqslant n\ r.\ C)$ and $(\leqslant n\ r.\ C)$

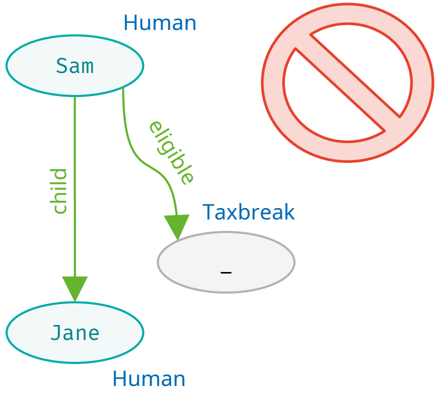
ALCQ

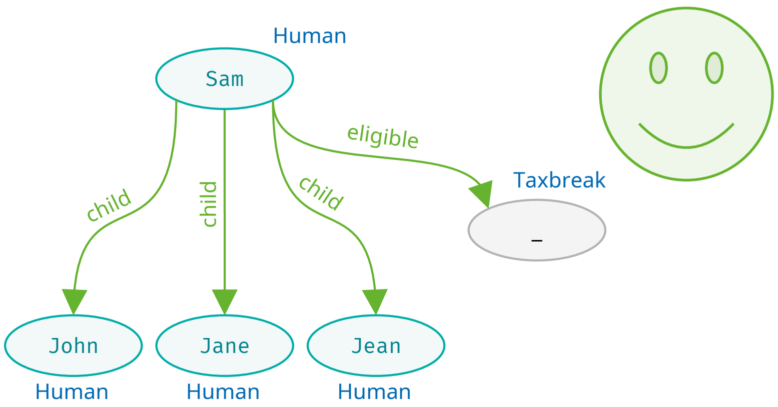
$$\mathsf{Human} \sqcap \underbrace{(\geqslant 3\ \mathsf{child}.\mathsf{Human})}_{\text{Qualified number restriction}} \sqsubseteq \exists \mathsf{eligible}.\mathsf{TaxBreak}$$

Cardinality constraints in description logics

$\mathrm{ALC} + (\geqslant n\ r.\ C) \text{ and } (\leqslant n\ r.\ C) \longleftarrow \boxed{\mathrm{ALCQ}}$

$$\mathrm{Human} \sqcap \overbrace{(\geqslant 3\ \mathrm{child}.\mathrm{Human})}^{\textit{Qualified number restriction}} \sqsubseteq \exists \mathrm{eligible}.\mathrm{TaxBreak}$$

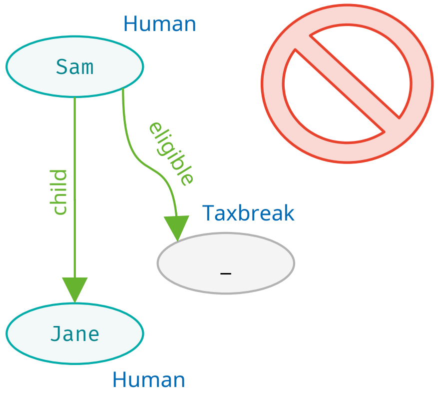


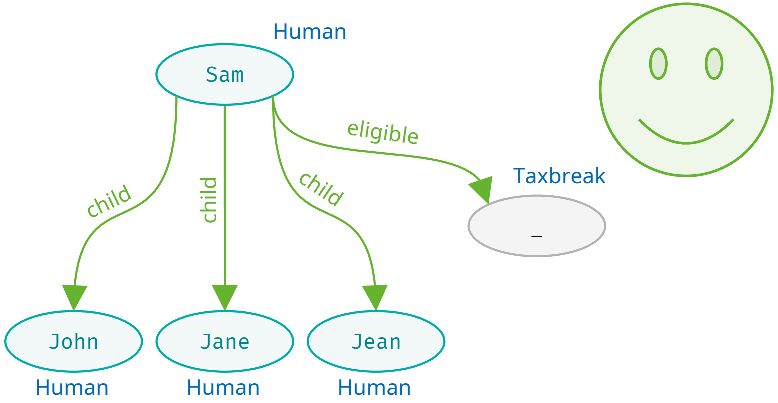


Cardinality constraints in description logics

$\mathcal{ALC}$ + $(\geqslant n\ r.\ C)$ and $(\leqslant n\ r.\ C)$ ← $\mathcal{ALCQ}$

$$\mathsf{Human} \sqcap \overbrace{(\geqslant 3\ \mathsf{child}.\mathsf{Human})}^{\textit{Qualified number restriction}} \sqsubseteq \exists \mathsf{eligible}.\mathsf{TaxBreak}$$





TBox consistency is ExpTime-complete in $\mathcal{ALCQ}$ (Tobies '00,'01) for unary and binary coding of numbers

Concrete domains

Concrete domains

$\text{Human} \sqcap (\geqslant 3\ \text{child}.\text{Human}) \sqcap \exists \text{salary}.{<_{100{,}000}} \sqsubseteq \exists \text{eligible}.\text{TaxBreak}$

$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.{>}$

Feature names

Concrete domains

Concrete domain restrictions

ALC + concrete domain restrictions ← $\mathcal{ALC}(\mathfrak{D})$

$\text{Human} \sqcap (\geqslant 3\ \text{child}.\text{Human}) \sqcap \exists \text{salary}.{<_{100,000}} \sqsubseteq \exists \text{eligible}.\text{TaxBreak}$

$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.{>}$

Feature names

Concrete domains

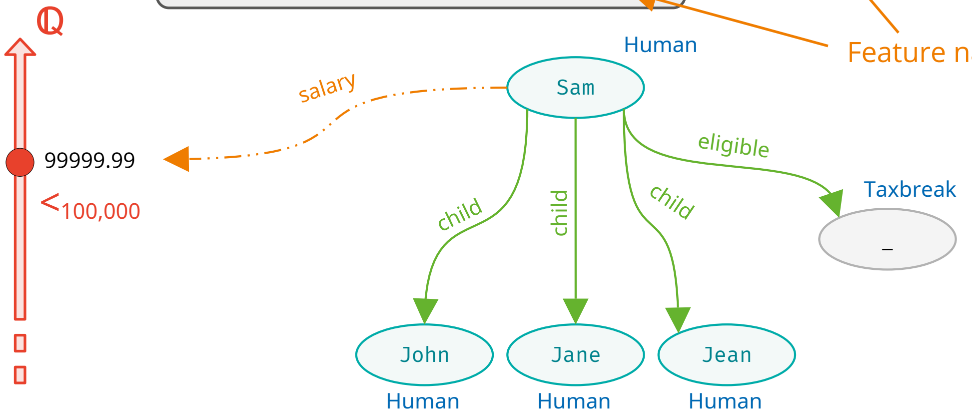
Concrete domain restrictions

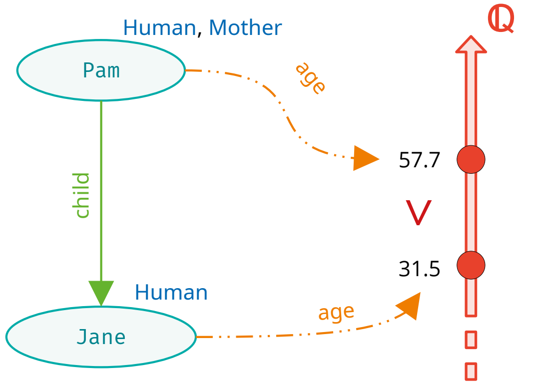
ALC + concrete domain restrictions ← $\mathcal{ALC}(\mathfrak{D})$

$\text{Human} \sqcap (\geqslant 3\ \text{child}.\text{Human}) \sqcap \exists \text{salary}.<_{100,000} \sqsubseteq \exists \text{eligible}.\text{TaxBreak}$

$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.>$

Feature names





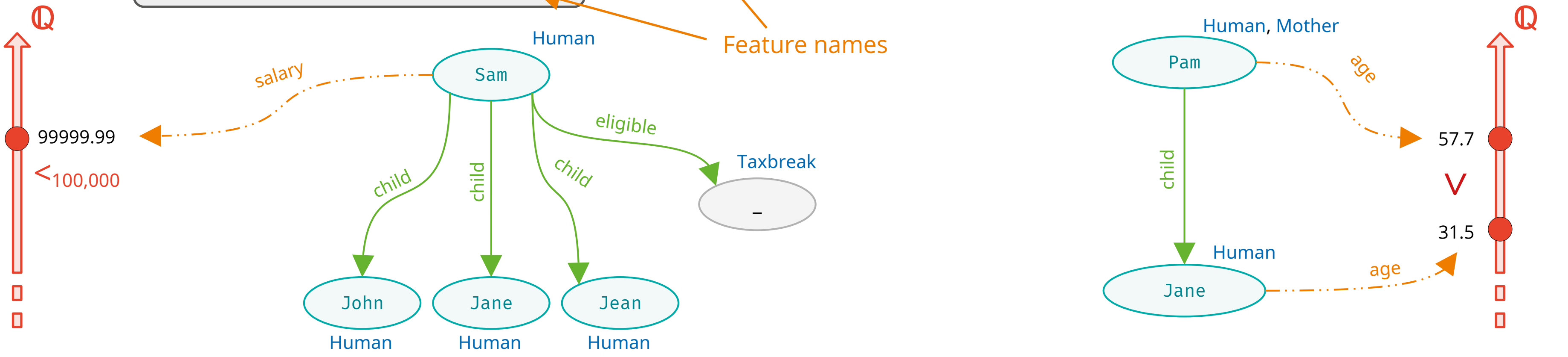
Concrete domains

Concrete domain restrictions

ALC + concrete domain restrictions ← $\mathcal{ALC}(\mathfrak{D})$

$\mathsf{Human} \sqcap (\geqslant 3\ \mathsf{child}.\mathsf{Human}) \sqcap \exists \mathsf{salary}.<_{100{,}000} \sqsubseteq \exists \mathsf{eligible}.\mathsf{TaxBreak}$

$\mathsf{Mother} \sqsubseteq \exists \mathsf{age}, \mathsf{child}\ \mathsf{age}.>$



TBox consistency ExpTime-complete for ALC extended by integers with comparison (Labai, Ortiz & Šimkus '20) or rationals with comparisons (Borgwardt, D., Koopmann '24)

Expressive power of concept languages

$$\exists \text{child.Human} \equiv (\geqslant 1\ \text{child.Human}) \equiv \varphi(x) := \exists y.\ (\text{child}(x,y) \land \text{Human}(y))$$

Expressive power of concept languages

$\exists \text{child.Human} \equiv (\geqslant 1 \text{ child.Human}) \equiv \varphi(x) := \exists y.\,(\text{child}(x,y) \land \text{Human}(y))$

equivalent

FO-definable

Expressive power of concept languages

$\exists$child.Human $\equiv$ ($\geqslant$ 1 child.Human) $\equiv$ $\varphi(x) := \exists y.\ (\text{child}(x,y) \wedge \text{Human}(y))$

equivalent

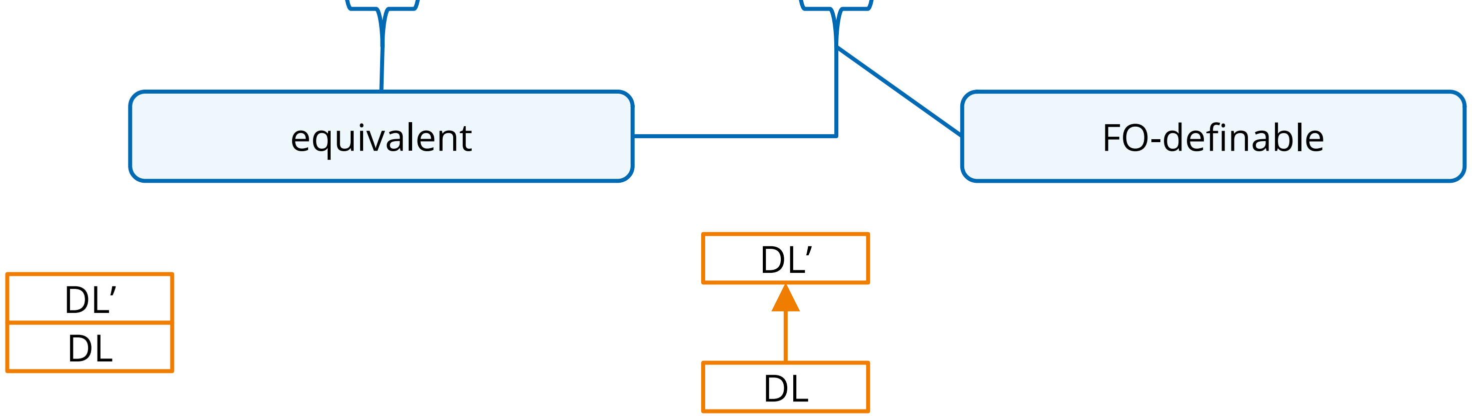
FO-definable

DL’
DL

“DL and DL’ are equivalent”

Expressive power of concept languages

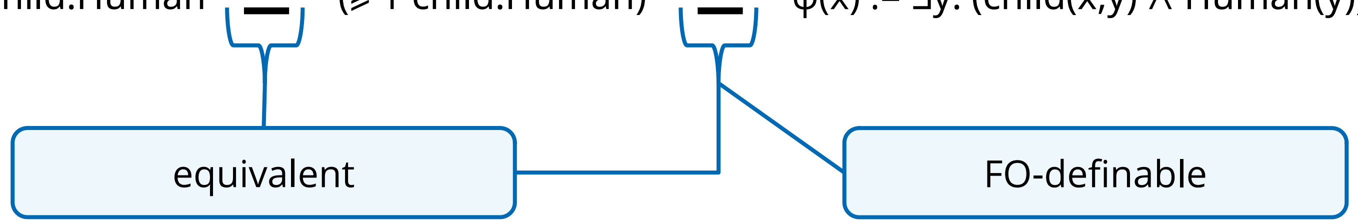
$\exists$child.Human $\equiv$ ($\geqslant$ 1 child.Human) $\equiv$ $\varphi(x) := \exists y.\,(\mathrm{child}(x,y) \wedge \mathrm{Human}(y))$

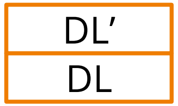


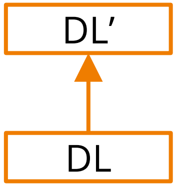
"DL and DL' are equivalent" "DL' strictly more expressive than DL"

Expressive power of concept languages

$\exists$child.Human $\equiv$ ($\geqslant$ 1 child.Human) $\equiv$ $\varphi(x) := \exists y.\ (\mathrm{child}(x,y) \wedge \mathrm{Human}(y))$







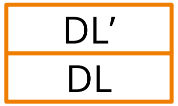
“DL and DL’ are equivalent” “DL’ strictly more expressive than DL” “DL is a first-order fragment”

Expressive power of concept languages

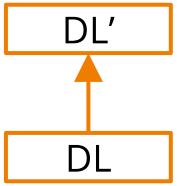
$\exists$child.Human $\equiv$ ($\geqslant$ 1 child.Human) $\equiv$ $\varphi(x) := \exists y.\ (\text{child}(x,y) \land \text{Human}(y))$

equivalent

FO-definable



“DL and DL’ are equivalent”



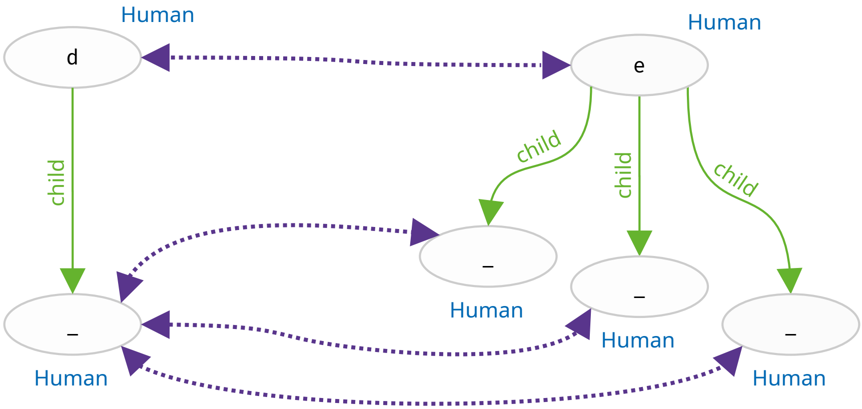
“DL’ strictly more expressive than DL”

“DL is a first-order fragment”

We assume that DL, DL’, FOL use *the same sets* of names.

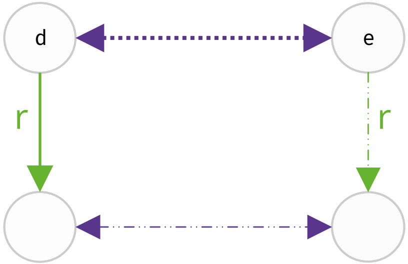
Bisimulations, characterization and non-expressivity

Bisimulations, characterization and non-expressivity



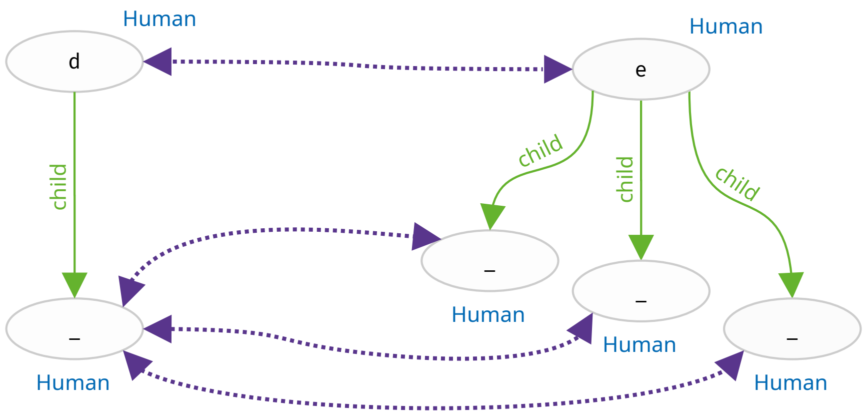
Bisimulation

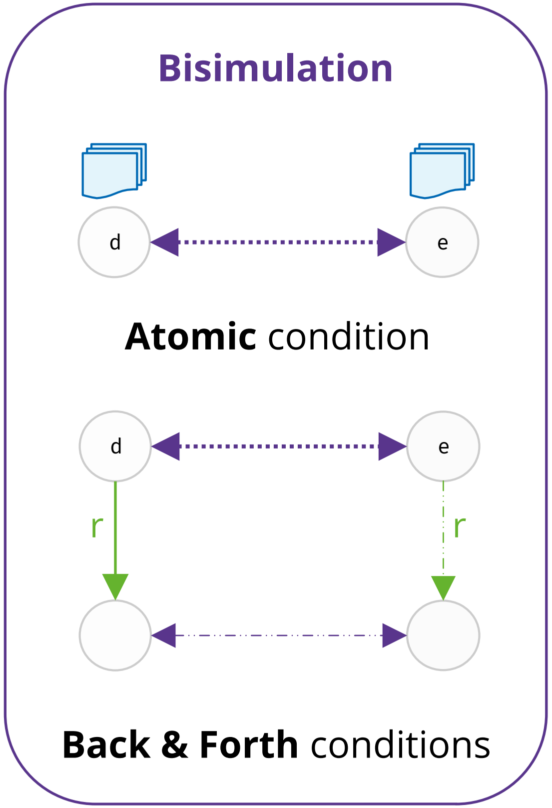
Atomic condition



Back & Forth conditions

Bisimulations, characterization and non-expressivity

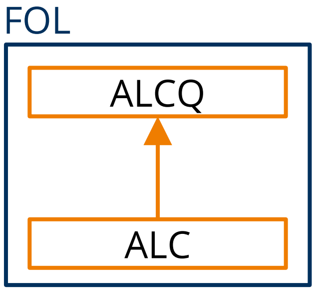


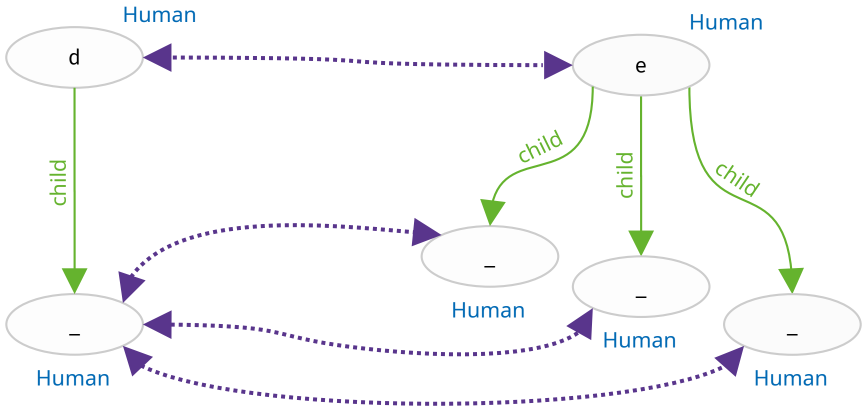


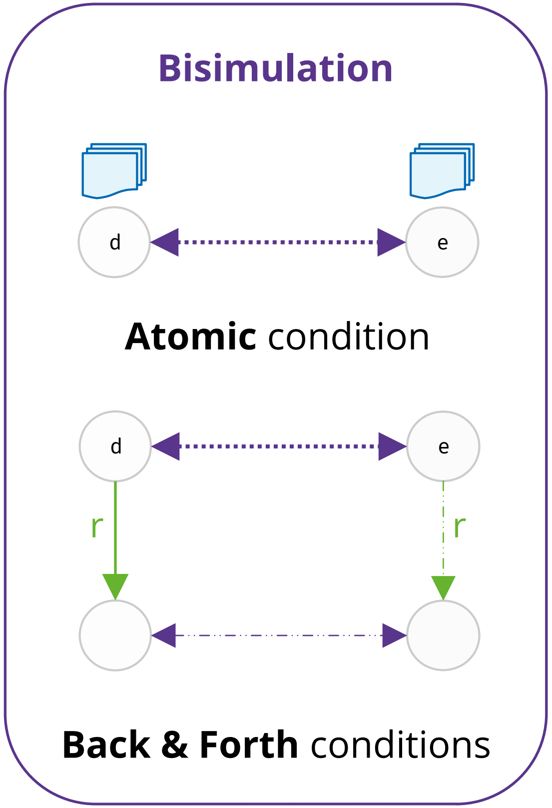
φ(x) is invariant under bisimulation ↔ φ(x) is equivalent to ALC concept

- Van Benthem '76: proof w.r.t. all interpretations
- Rosen '97: proof also w.r.t. finitely branching or finite interpretations

Bisimulations, characterization and non-expressivity







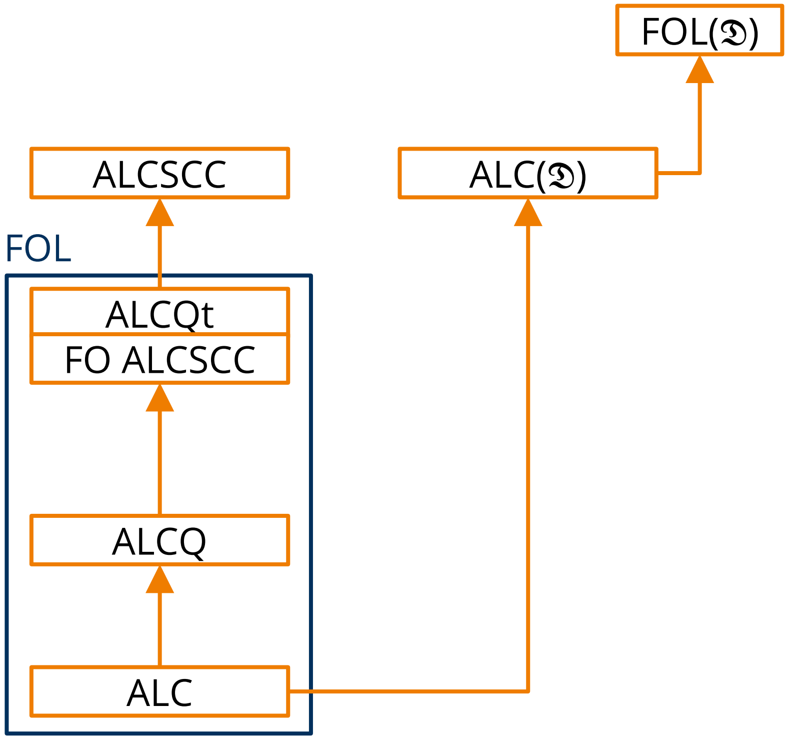
φ(x) is invariant under bisimulation ↔ φ(x) is equivalent to ALC concept

- Van Benthem '76: proof w.r.t. all interpretations
- Rosen '97: proof also w.r.t. finitely branching or finite interpretations

(⩽ 1 child. Human) is *not* invariant under bisimulation!

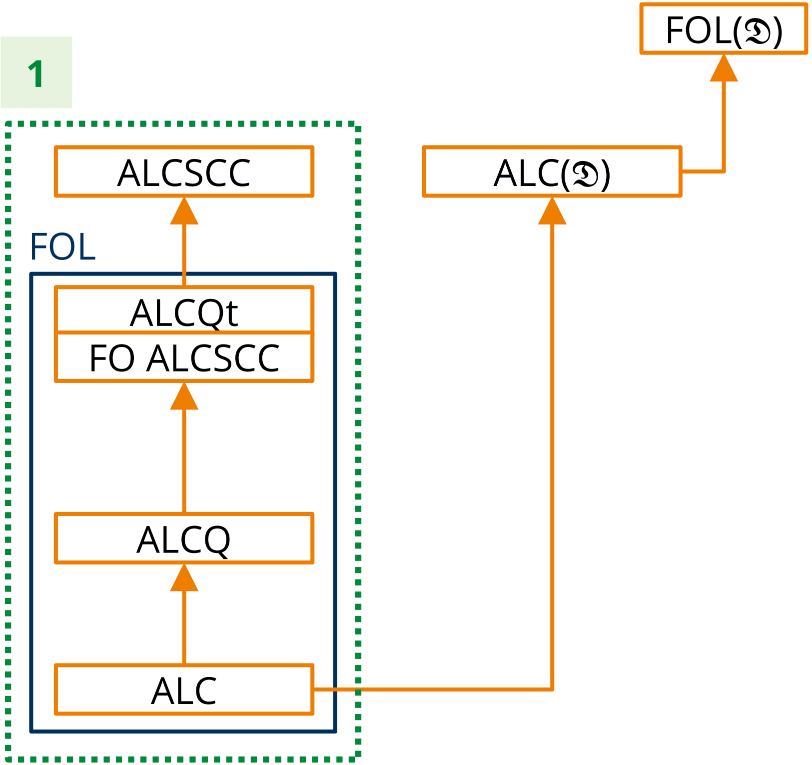
Outline

1) *Cardinality constraints*
2) *Concrete domains*



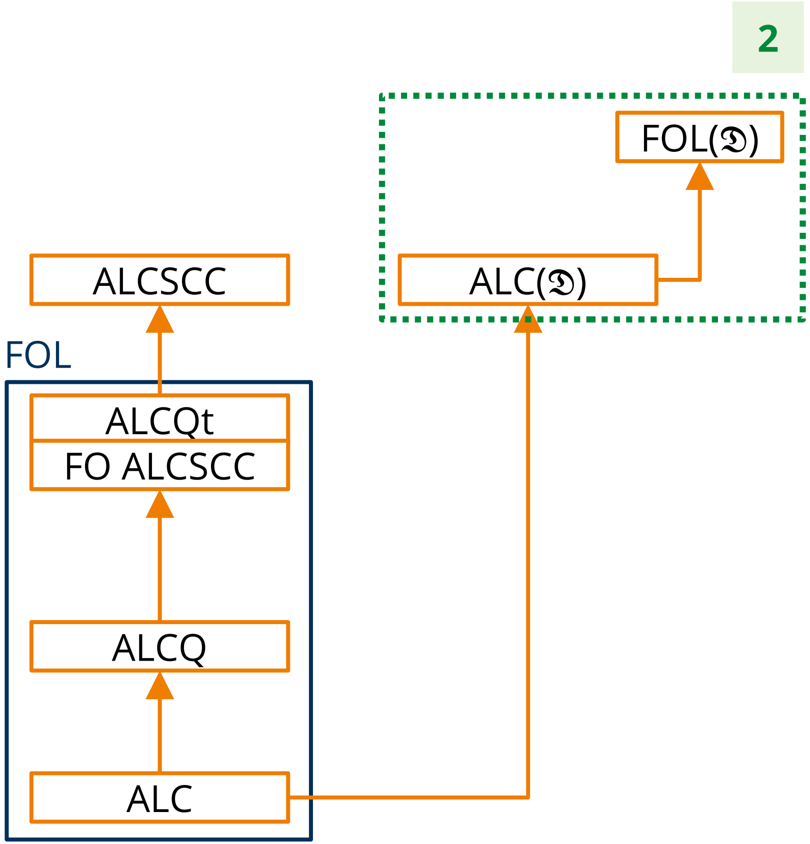
Outline

1) *Cardinality constraints*
2) *Concrete domains*

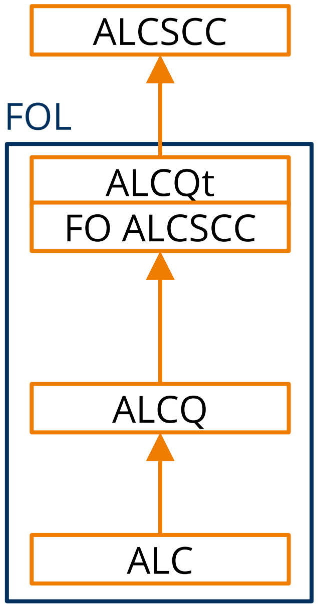


Outline

1) *Cardinality constraints*
2) *Concrete domains*



Cardinality constraints



The description logic ALCSCC

The description logic ALCSCC

ALC + successor restrictions based on QFBAPA

ALCSCC

The description logic ALCSCC

ALC + successor restrictions based on QFBAPA ← ALCSCC

$$\text{Human} \sqcap \underbrace{\text{succ}(|\text{pet} \cap \text{Dog}| = |\text{child} \cap \text{Human}|)}_{\textit{successor restriction}}$$

Successor restrictions evaluated w.r.t. all role successors of an individual

The description logic ALCSCC

ALC + successor restrictions based on QFBAPA ← ALCSCC

$$\text{Human} \sqcap \overbrace{\text{succ}(|\text{pet} \cap \text{Dog}| = |\text{child} \cap \text{Human}|)}^{\textit{successor restriction}}$$

Successor restrictions evaluated w.r.t. all role successors of an individual

The description logic ALCSCC

ALC + successor restrictions based on QFBAPA ← ALCSCC

$$\mathrm{Human} \sqcap \overbrace{\mathrm{succ}(|\mathrm{pet} \cap \mathrm{Dog}| = |\mathrm{child} \cap \mathrm{Human}|)}^{\textit{successor restriction}}$$

Successor restrictions evaluated w.r.t. all role successors of an individual

- QFBAPA: set and cardinality constraints over finite sets
- Satisfiability for QFBAPA is NP-complete
- **Baader & D. '19:** QFBAPA$^{\infty}$, like QFBAPA but with infinite sets, we show that it is NP-complete

The description logic ALCSCC

ALC + successor restrictions based on QFBAPA ← ALCSCC

$$\text{Human} \sqcap \overbrace{\text{succ}(|\text{pet} \cap \text{Dog}| = |\text{child} \cap \text{Human}|)}^{\textit{successor restriction}}$$

Successor restrictions evaluated w.r.t. all role successors of an individual

- QFBAPA: set and cardinality constraints over finite sets
- Satisfiability for QFBAPA is NP-complete
- **Baader & D. '19:** QFBAPA$^\infty$, like QFBAPA but with infinite sets, we show that it is NP-complete

Baader '17: TBox consistency is ExpTime complete; defined only over finitely branching interpretations!

Baader & D. '19

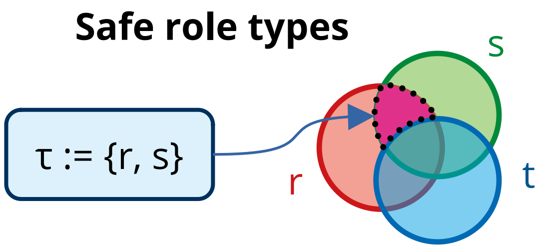
- ALCSCC$^\infty$:= ALCSCC defined over all interpretations
- TBox consistency remains ExpTime-complete

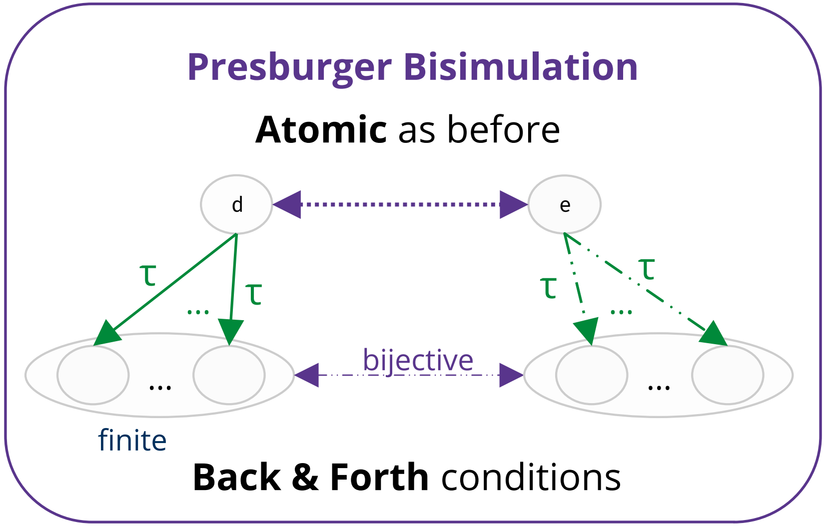
Separation using *counting* bisimulation → ALCSCC$^\infty$

ALCQ → ALCSCC$^\infty$

Presburger bisimulation

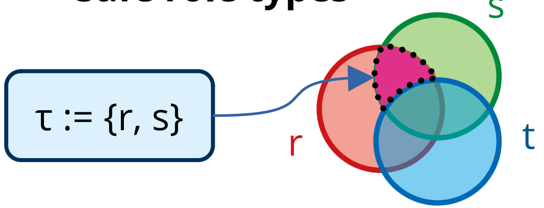
Presburger bisimulation





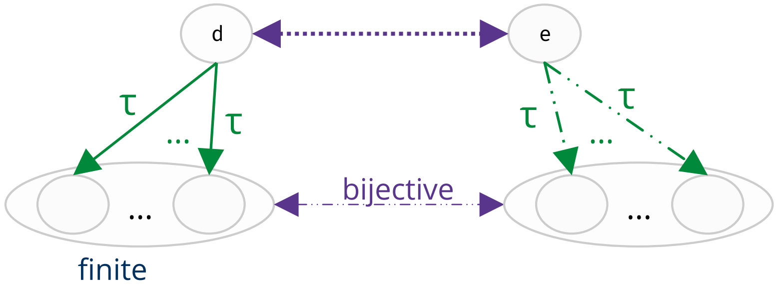
Presburger bisimulation

Safe role types



Presburger Bisimulation

Atomic as before



Back & Forth conditions

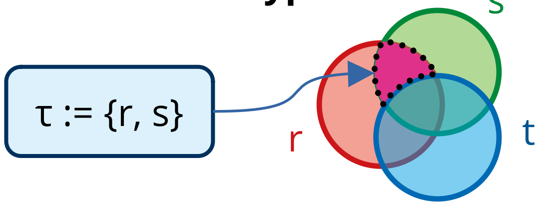
For all FOL formulae φ(x), the following are equivalent:

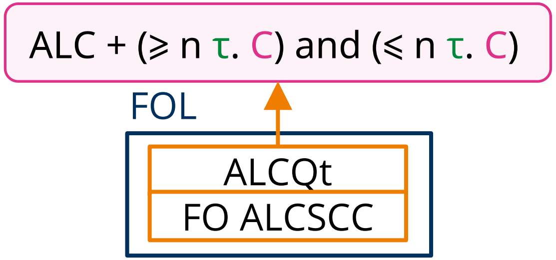
1) φ(x) is equivalent to some ALCSCC concept.
2) φ(x) is invariant under Presburger bisimulation.
3) φ(x) is equivalent to some ALCQt concept.

- **Baader & D. '19:** showed for all interpretations
- **FroCoS '25:** extended to finitely branching/finite

Presburger bisimulation

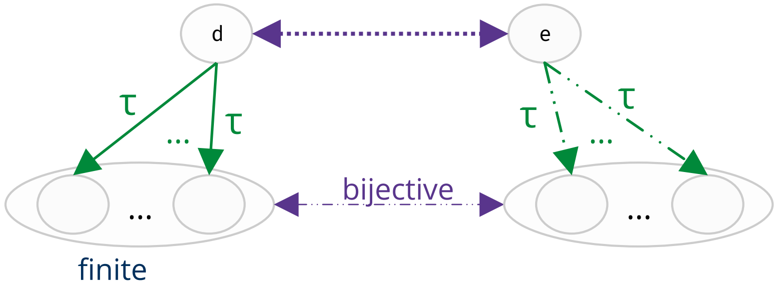
Safe role types





Presburger Bisimulation

Atomic as before



Back & Forth conditions

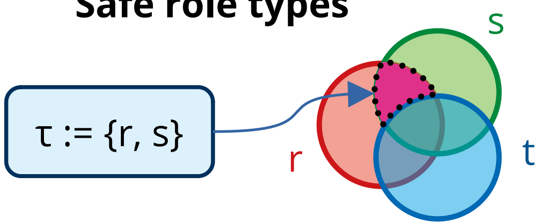
For all FOL formulae φ(x), the following are equivalent:

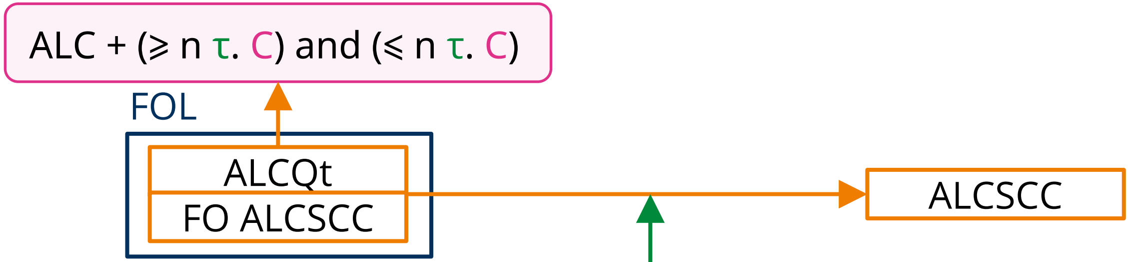
1) φ(x) is equivalent to some ALCSCC concept.
2) φ(x) is invariant under Presburger bisimulation.
3) φ(x) is equivalent to some ALCQt concept.

- **Baader & D. ‘19:** showed for all interpretations
- **FroCoS ‘25:** extended to finitely branching/finite

Presburger bisimulation

Safe role types

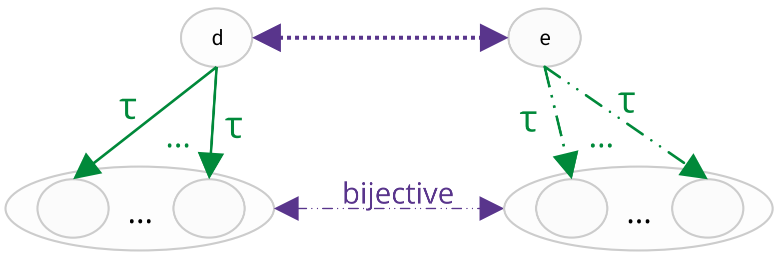




succ(|r ∩ A| = |r ∩ ¬A|) is not first-order definable!

Presburger Bisimulation

Atomic as before



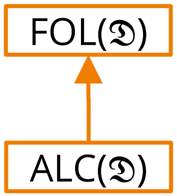
Back & Forth conditions

For all FOL formulae φ(x), the following are equivalent:

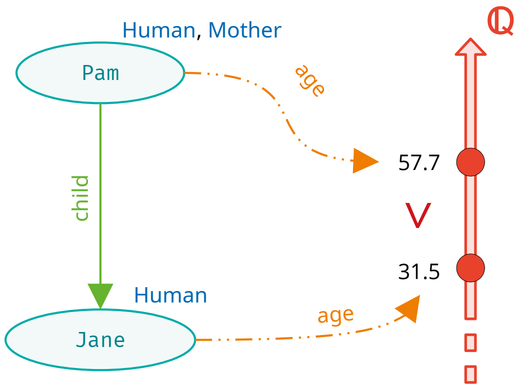
1) φ(x) is equivalent to some ALCSCC concept.
2) φ(x) is invariant under Presburger bisimulation.
3) φ(x) is equivalent to some ALCQt concept.

- **Baader & D. '19:** showed for all interpretations
- **FroCoS '25:** extended to finitely branching/finite

Concrete Domains



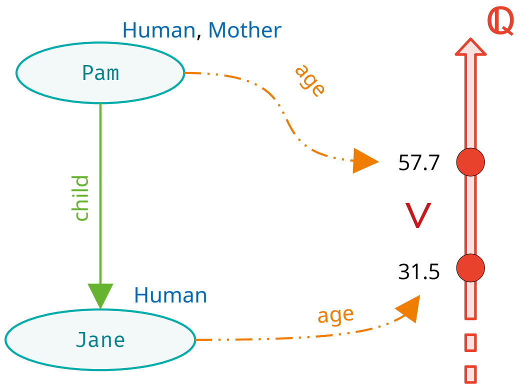
First-order logics with concrete domains (SAC '24)



First-order logics with concrete domains (SAC '24)

$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.<$

ALC + concrete domain restrictions ← $\mathcal{ALC}(\mathfrak{D})$

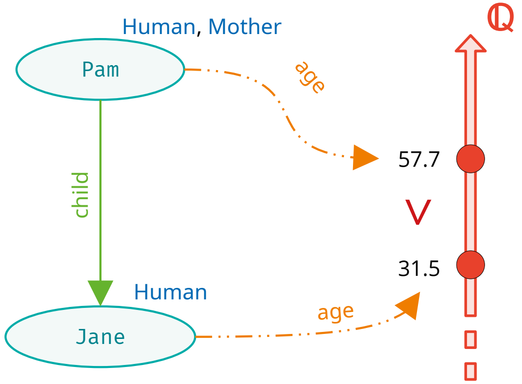


First-order logics with concrete domains (SAC ‘24)

$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.<$

FOL + definedness predicates + concrete domain predicates ← FOL($\mathfrak{D}$)

ALC + concrete domain restrictions ← ALC($\mathfrak{D}$)



First-order logics with concrete domains (SAC '24)

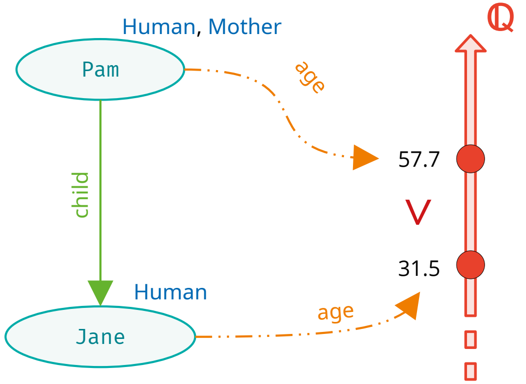
$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.<$

FOL + definedness predicates + concrete domain predicates ← FOL($\mathfrak{D}$)

ALC + concrete domain restrictions ← ALC($\mathfrak{D}$)

$\forall x,y.\ \mathrm{Def}(id)(x) \wedge (x \neq y \rightarrow <(id,id)(x,y) \vee <(id,id)(y,x))$

"id acts as an injective function" (satisfiable)



$\exists x,y,z.\ (<(age,age)(x,y) \wedge <(age,age)(y,z) \wedge <(age,age)(z,x))$

"there is a <-cicle of age-values" (unsatisfiable)

First-order logics with concrete domains (SAC '24)

$\text{Mother} \sqsubseteq \exists \text{age}, \text{child age}.<$

FOL + definedness predicates + concrete domain predicates ← FOL($\mathfrak{D}$)

ALC + concrete domain restrictions ← ALC($\mathfrak{D}$)

Definedness predicate

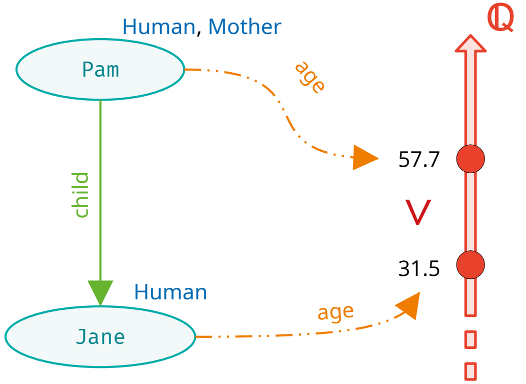
$\forall x,y.\ \text{Def}(\text{id})(x) \land (x \neq y \rightarrow <(\text{id},\text{id})(x,y) \lor <(\text{id},\text{id})(y,x))$

"id acts as an injective function" (satisfiable)

Concrete domain predicates

$\exists x,y,z.\ (<(\text{age},\text{age})(x,y) \land <(\text{age},\text{age})(y,z) \land <(\text{age},\text{age})(z,x))$

"there is a <-cicle of age-values" (unsatisfiable)



Concrete domains and negation

Concrete domains and negation

Conditions on $\mathfrak{D}$ that enable the expression of negated predicates in ALC($\mathfrak{D}$) or FOL($\mathfrak{D}$):

- **Weakly Closed Under Negation (WCUN):** complement of a k-ary predicate is a union of k-ary predicates

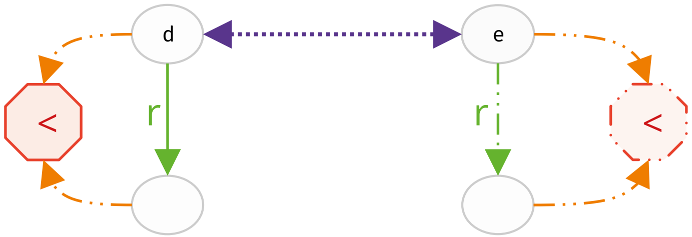
$$\neg(x = y) \text{ iff } (x < y) \vee (y < x)$$

Bisimulations for concrete domains

Bisimulations for concrete domains

$\mathfrak{D}$ **Bisimulation**

Atomic as before
Back & Forth as before



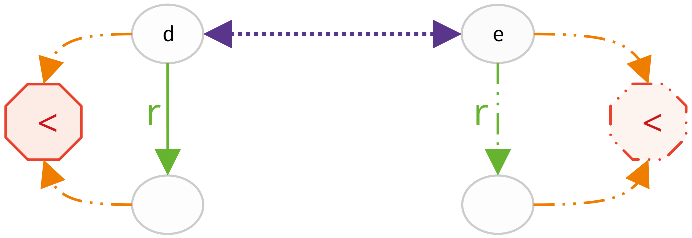
Feature conditions

if $(d,e) \in \rho$, then there is $(v_1,\dots,v_k) \in P^D$ with $v_1 \in p_1^{\mathcal{I}}(d), \dots, v_k \in p_k^{\mathcal{I}}(d)$ iff there is $(w_1,\dots,w_k) \in P^D$ with $w_1 \in p_1^{\mathcal{I}}(e), \dots, w_k \in p_k^{\mathcal{I}}(e)$.

Bisimulations for concrete domains

$\mathfrak{D}$ Bisimulation

Atomic as before
Back & Forth as before



Feature conditions

if $(d,e) \in \rho$, then there is $(v_1,\ldots,v_k) \in P^D$ with $v_1 \in p_1^{\mathcal{I}}(d), \ldots, v_k \in p_k^{\mathcal{I}}(d)$ iff there is $(w_1,\ldots,w_k) \in P^D$ with $w_1 \in p_1^{\mathcal{I}}(e), \ldots, w_k \in p_k^{\mathcal{I}}(e)$.

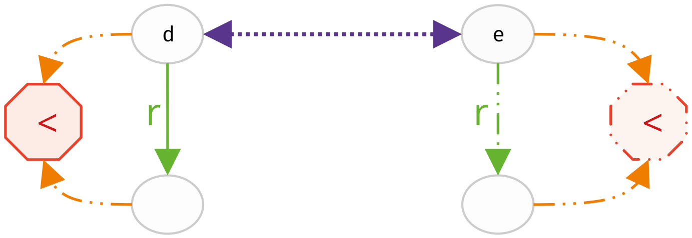
Bisimulation and non-expressivity:

- extensions of ALC with different concrete domains
  - e.g. ALC($\mathbb{Q}$, $+_1$) and ALC($\mathbb{Q}$, $+_2$) "orthogonal"
- different extensions of ALC w/ same concrete domain
  - e.g. ALC($\mathbb{Q}$, <) cannot express restriction with constraint systems e.g. $\exists r\ f, r\ f, r\ f.(x < y \land y < z)$

Bisimulations for concrete domains

$\mathfrak{D}$ Bisimulation

Atomic as before
Back & Forth as before



Feature conditions

if $(d,e) \in \rho$, then there is $(v_1,\ldots,v_k) \in P^D$ with $v_1 \in p_1^{\mathcal{I}}(d), \ldots, v_k \in p_k^{\mathcal{I}}(d)$ iff there is $(w_1,\ldots,w_k) \in P^D$ with $w_1 \in p_1^{\mathcal{I}}(e), \ldots, w_k \in p_k^{\mathcal{I}}(e)$.

Bisimulation and non-expressivity:

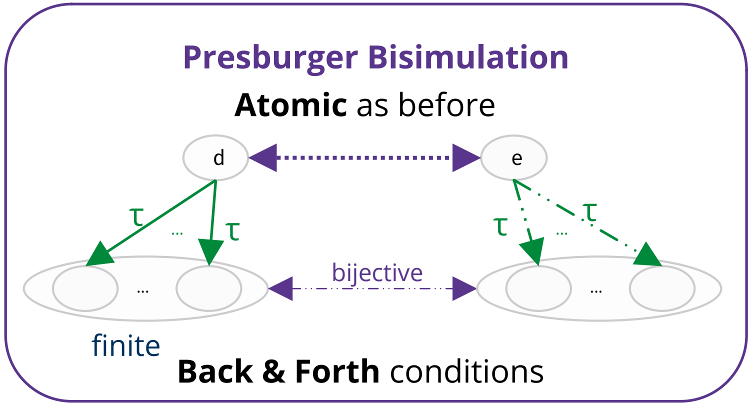
- extensions of ALC with different concrete domains
  - e.g. ALC($\mathbb{Q}$, $+_1$) and ALC($\mathbb{Q}$, $+_2$) "orthogonal"
- different extensions of ALC w/ same concrete domain
  - e.g. ALC($\mathbb{Q}$, <) cannot express restriction with constraint systems e.g. $\exists r\ f, r\ f, r\ f.(x < y \land y < z)$

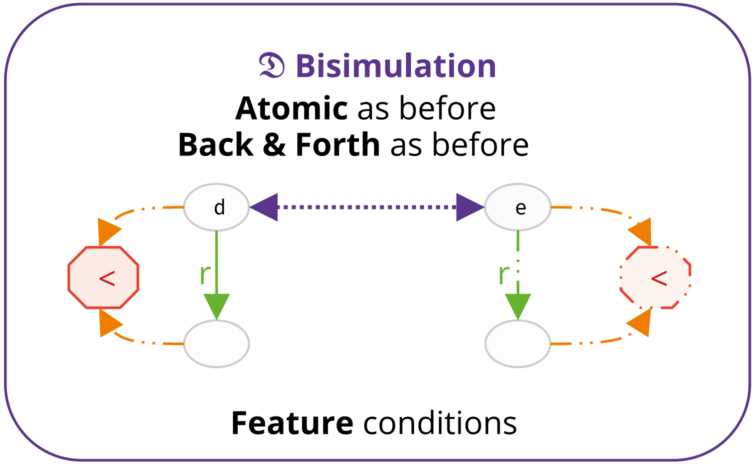
Assume finite sets of names and let be WCUN and have finitely many relations. For all FOL($\mathfrak{D}$) formulae $\varphi(x)$, the following are equivalent:

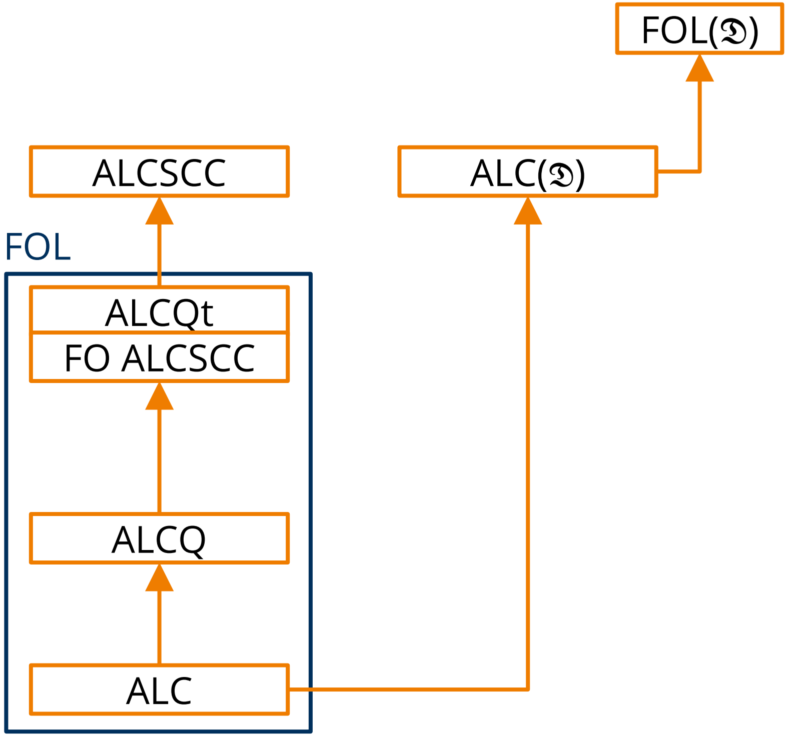
1) $\varphi(x)$ is invariant under bisimulation.
2) $\varphi(x)$ is equivalent to some ALC($\mathfrak{D}$) concept.

- **FroCoS '25:** showed w.r.t all interpretations as well as finitely branching/finite ones

Summary: expressive power







From here on...

- Expressive power of ontologies with numerical constraints*
  - e.g. $ALCSCC^{\infty}$ TBoxes using global Presburger bisimulation (Baader, D. ‘20)
- Expresive power when combining cardinality constraints and concrete domains (CADE ‘25)
- Different notion of expressive power, e.g. conservative extensions

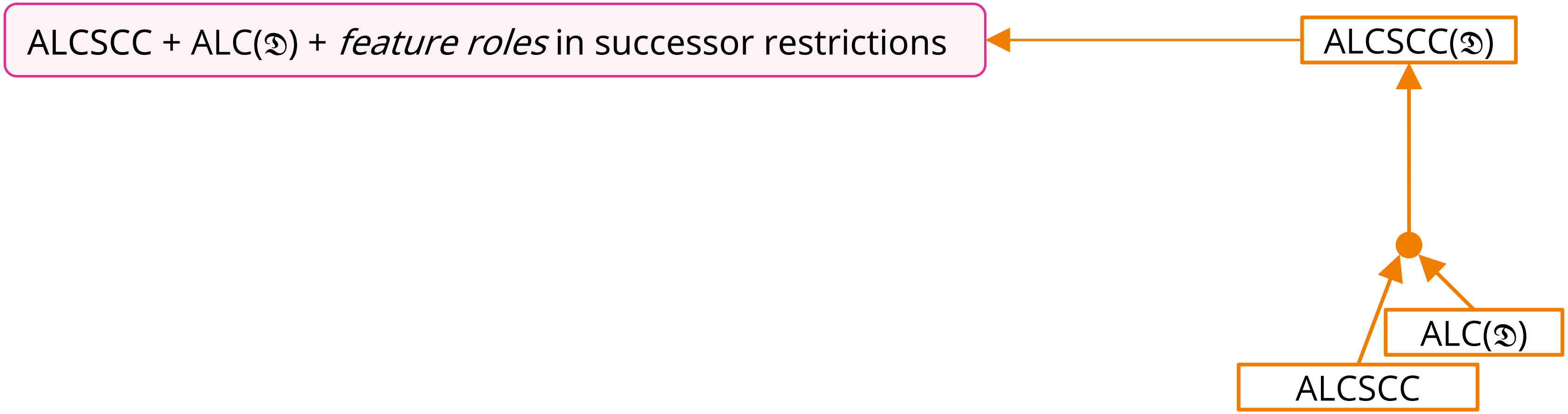
From here on...

- Expressive power of ontologies with numerical constraints*
  - e.g. $\mathcal{ALCSCC}^{\infty}$ TBoxes using global Presburger bisimulation (Baader, D. ‘20)
- Expresive power when combining cardinality constraints and concrete domains (CADE ‘25)
- Different notion of expressive power, e.g. conservative extensions

Thanks! :-)

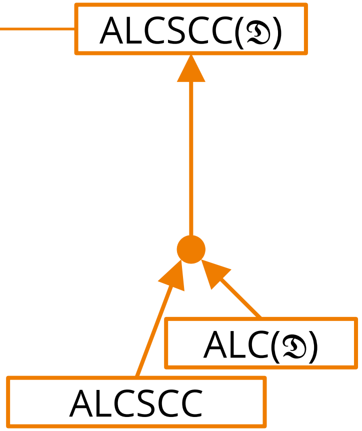
ALCSCC($\mathfrak{D}$) and feature roles (CADE '25)

ALCSCC($\mathfrak{D}$) and feature roles (CADE '25)



ALCSCC($\mathfrak{D}$) and feature roles (CADE '25)

ALCSCC + ALC($\mathfrak{D}$) + *feature roles* in successor restrictions



$\mathrm{succ}(|\mathrm{child} \cap (\mathrm{salary} < \mathtt{next}\ \mathrm{salary})| > |\mathrm{child} \cap (\mathrm{salary} < \mathtt{next}\ \mathrm{salary})^c|)$

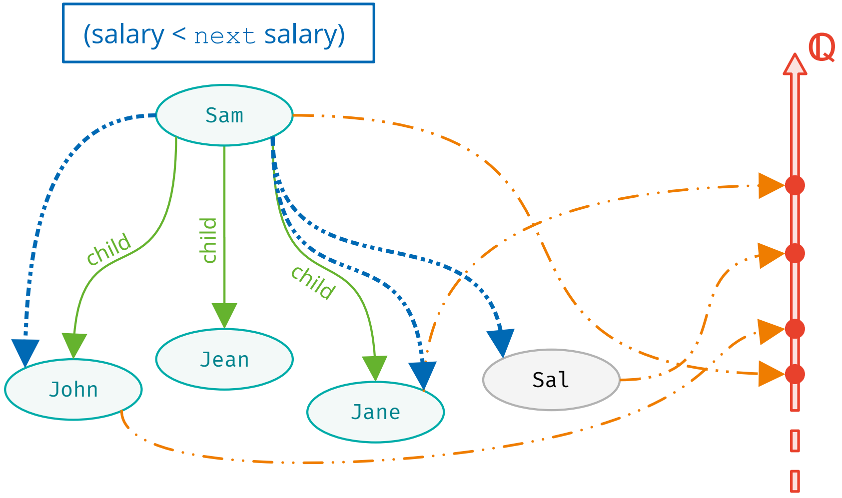
"the salary of this individual is smaller than that of the majority of its children"

ALCSCC($\mathfrak{D}$) and feature roles (CADE '25)

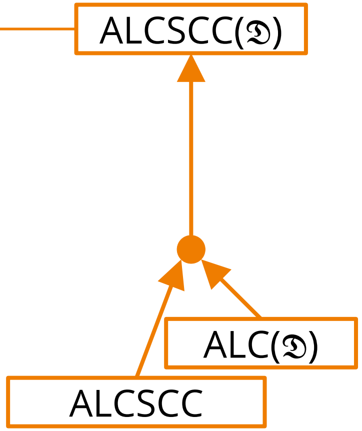
ALCSCC + ALC($\mathfrak{D}$) + *feature roles* in successor restrictions

Feature roles

$$\mathrm{succ}(|\mathrm{child} \cap \underbrace{(\mathrm{salary} < \mathtt{next}\ \mathrm{salary})}| > |\mathrm{child} \cap \underbrace{(\mathrm{salary} < \mathtt{next}\ \mathrm{salary})^c}|)$$



"the salary of this individual is smaller than that of the majority of its children"

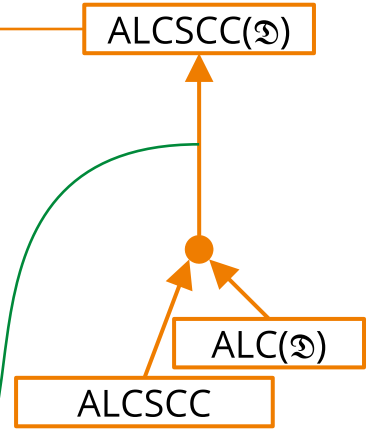


ALCSCC($\mathfrak{D}$) and feature roles (CADE '25)

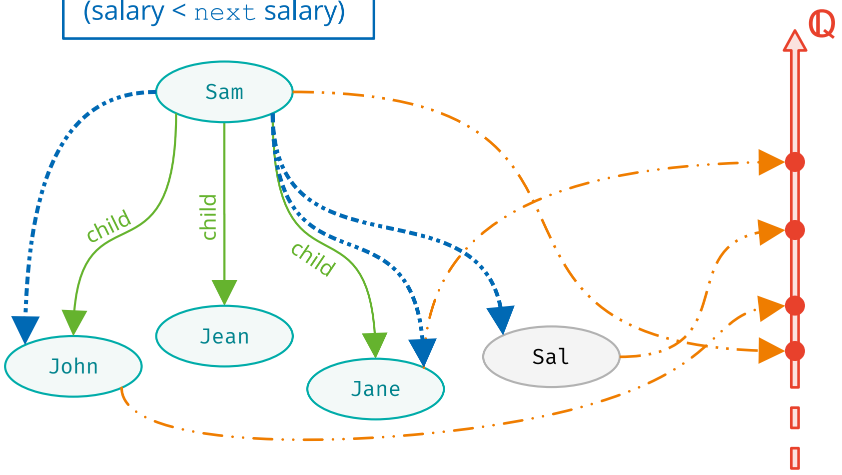
ALCSCC + ALC($\mathfrak{D}$) + *feature roles* in successor restrictions

Feature roles

$\mathrm{succ}(|\mathrm{child} \cap \underbrace{(\mathrm{salary} < \mathtt{next}\ \mathrm{salary})}_{\text{feature role}}| > |\mathrm{child} \cap \underbrace{(\mathrm{salary} < \mathtt{next}\ \mathrm{salary})}_{\text{feature role}}{}^{c}|)$



(salary < $\mathtt{next}$ salary)

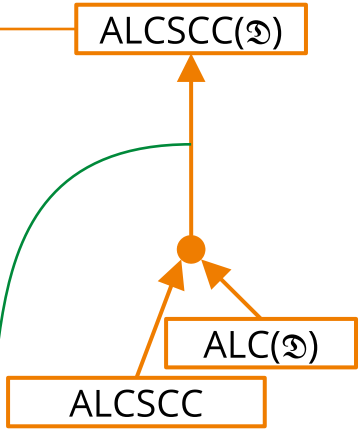


"the salary of this individual is smaller than that of the majority of its children"

It cannot be expressed without feature roles!

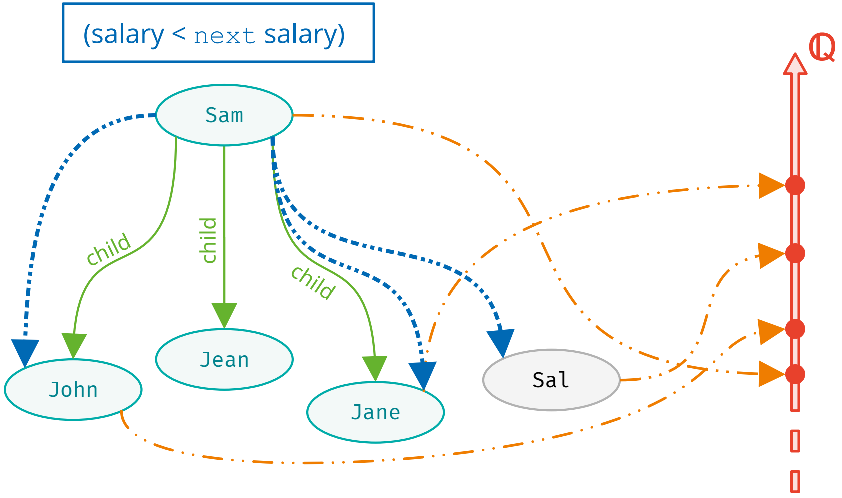
ALCSCC($\mathfrak{D}$) and feature roles (CADE '25)

ALCSCC + ALC($\mathfrak{D}$) + *feature roles* in successor restrictions



Feature roles

$$\mathrm{succ}(|\mathrm{child} \cap (\mathrm{salary} < \mathtt{next}\ \mathrm{salary})| > |\mathrm{child} \cap (\mathrm{salary} < \mathtt{next}\ \mathrm{salary})^c|)$$



"the salary of this individual is smaller than that of the majority of its children"

It cannot be expressed without feature roles!

$$(\exists \mathrm{salary}, \mathrm{child}\ \mathrm{salary}.<) \sqsubseteq (\mathrm{succ}(|\mathrm{child}| \leq 0))$$

Inconsistent, due to the interaction of cardinality constraints and concrete domain!