Tutorial: Definite Descriptions in KR Languages
Part II: Description Logics with Definite Descriptions

Alessandro Artale, Andrzej Indrzejczak, Oliver Kutz, Andrea Mazzullo, Przemysław (Przemek) Wałęga

www.walega.pl | p.walega@qmul.ac.uk

18 of July, 2026

Outline

1. Basic Description Logics – recap
2. Description Logics with DDs – main observations
3. Computational complexity analysis
4. Expressive power analysis
   - Definition of appropriate bisimulation
   - Checking bisimilarity algorithmically
5. Reasoning in practice – implementation

Basic Description Logic $\mathcal{ALC}$

$\mathcal{ALC}$ concepts are given by:

$$C := A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C$$

corresponding to *multi-modal logic formulae*:

$$\varphi := p \mid \neg\varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \Diamond_r \varphi \mid \Box_r \varphi$$

Example:
building $\sqcap$ $\forall$shortThan.$\neg$building — 'a building not shorter than any other building'

Basic Description Logic $\mathcal{ALC}$

$\mathcal{ALC}$ concepts are given by:

$$C := A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C$$

corresponding to *multi-modal logic formulae*:

$$\varphi := p \mid \neg\varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \Diamond_r \varphi \mid \Box_r \varphi$$

Example:
building ⊓ ∀shortThan.¬building — 'a building not shorter than any other building'

Additionally $\mathcal{ALC}$ allows for:

- *TBoxes*, i.e., sets of *axioms*: $C \sqsubseteq D$ 'every C is D'
- *ABoxes*, i.e., sets of *assertions*: $C(a)$ 'a satisfies C'

Extensions of $\mathcal{ALC}$

$\mathcal{ALCO}$ extends $\mathcal{ALC}$ with *nominals*:

$$C := A \mid \{a\} \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C$$

$\{a\}$ holds only at the individual a

$\mathcal{ALC}_u$ extends $\mathcal{ALC}$ with the *universal role*:

$$C := A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C \mid \exists u.\, C \mid \forall u.\, C$$

$\exists u.C$ — 'some individual satisfies C'

Two Types of DDs

Local DDs are of the form $\{\iota C\}$

- $\{\iota(\mathsf{building} \sqcap \forall \mathsf{shortThan}.\neg \mathsf{building})\}$
- expressing *"the tallest building"*

Global DDs are of the form $\iota C.D$

- $\iota(\mathsf{building} \sqcap \forall \mathsf{shortThan}.\neg \mathsf{building}).(\exists \mathsf{locIn}.\mathsf{Dubai})$
- expressing *"the tallest building is located in Dubai"*

Two Types of DDs

Local DDs are of the form $\{\iota C\}$

- $\{\iota(\text{building} \sqcap \forall\text{shortThan}.\neg\text{building})\}$
- expressing *"the tallest building"*

$$(\{\iota C\})^{\mathcal{I}} = \begin{cases} \{d\}, & \text{if } C^{\mathcal{I}} = \{d\}, \\ \emptyset, & \text{otherwise,} \end{cases}$$

Global DDs are of the form $\iota C.D$

- $\iota(\text{building} \sqcap \forall\text{shortThan}.\neg\text{building}).(\exists\text{locIn}.\text{Dubai})$
- expressing *"the tallest building is located in Dubai"*

$$(\iota C.D)^{\mathcal{I}} = \begin{cases} \Delta^{\mathcal{I}}, & \text{if } C^{\mathcal{I}} = \{d\} \subseteq D^{\mathcal{I}}, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Description Logics with DDs

$\mathcal{ALC}\iota_L$ — $\mathcal{ALC}$ with local DDs:

$$C := A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C \mid \{\iota C\}$$

$\mathcal{ALC}\iota_G$ — $\mathcal{ALC}$ with global DDs:

$$C := A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C \mid \iota C.D$$

$\mathcal{ALC}\iota$ — $\mathcal{ALC}$ with local and global DDs:

$$C := A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C \mid \{\iota C\} \mid \iota C.D$$

$\mathcal{ALCO}^{\iota}_{u}$ — $\mathcal{ALC}$ with nominals, universal role, and local DDs:

$$C := A \mid \{a\} \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.\, C \mid \forall r.\, C \mid \exists u.\, C \mid \forall u.\, C \mid \{\iota C\}$$

Observations

$\iota C.D$ allows to express $\{\iota C\}$: $\quad \{\iota C\} \equiv C \sqcap \iota C.\top$

$\exists u$ and $\{\iota C\}$ allow to express $\iota C.D$: $\quad \iota C.D \equiv \exists u.(\{\iota C\} \sqcap D)$

So in terms of the *expressive power* (equivalence preserving translations):

$$\mathcal{ALC} \preceq \mathcal{ALC}\iota_L \preceq \mathcal{ALC}\iota_G \equiv \mathcal{ALC}\iota \preceq \mathcal{ALCO}_u^{\iota}$$

Observations

$\iota C.D$ allows to express $\{\iota C\}$: $\{\iota C\} \equiv C \sqcap \iota C.\top$

$\exists u$ and $\{\iota C\}$ allow to express $\iota C.D$: $\iota C.D \equiv \exists u.(\{\iota C\} \sqcap D)$

So in terms of the *expressive power* (equivalence preserving translations):

$$\mathcal{ALC} \preceq \mathcal{ALC}\iota_L \preceq \mathcal{ALC}\iota_G \equiv \mathcal{ALC}\iota \preceq \mathcal{ALCO}^{\iota}_{u}$$

Interesting research directions:

- Computational complexity?
- Expressive power characterisation?

Observations

$\iota C.D$ allows to express $\{\iota C\}$: $\{\iota C\} \equiv C \sqcap \iota C.\top$

$\exists u$ and $\{\iota C\}$ allow to express $\iota C.D$: $\iota C.D \equiv \exists u.(\{\iota C\} \sqcap D)$

So in terms of the *expressive power* (equivalence preserving translations):

$$\mathcal{ALC} \preceq \mathcal{ALC}\iota_L \preceq \mathcal{ALC}\iota_G \equiv \mathcal{ALC}\iota \preceq \mathcal{ALCO}_u^\iota$$

Interesting research directions:

- Computational complexity? **ExpTime-complete**
- Expressive power characterisation? **New bisimulations**

Computational Complexity

Ontology Satisfiability

Theorem. Ontology satisfiability is ExpTime-*complete* for each of

$$\mathcal{ALC}\iota_L, \ \mathcal{ALC}\iota_G, \ \mathcal{ALC}\iota, \text{ and } \mathcal{ALCO}_u^\iota$$

Proof.

- $\mathcal{ALC}$ and $\mathcal{ALCO}_u$ are both ExpTime-complete
- To show that $\mathcal{ALCO}_u^\iota$ is in ExpTime we reduce to $\mathcal{ALCO}_u$
- The reduction translates
  $\{\iota C\}$ into $A_{\iota C} \sqcap \neg \exists u.(A_{\iota C} \land \neg\{a_{\iota C}\})$
  for fresh atomic concept $A_{\iota C}$ and fresh individual name $a_{\iota C}$

$\square$

Concept Satisfiability

Theorem. Concept satisfiability is ExpTime-*complete* for each of
$\mathcal{ALC}\iota_L$, $\mathcal{ALC}\iota_G$, $\mathcal{ALC}\iota$, and $\mathcal{ALCO}^\iota_u$

Proof.

- To show that $\mathcal{ALC}\iota_L$ is ExpTime-hard we reduce satisfiability of an $\mathcal{ALC}$ concept C wrt a TBox $\mathcal{T}$ (which is ExpTime-complete)

- The reduction translates C and $\mathcal{T}$ into

$$C \sqcap \textstyle\bigsqcap_{(D \sqsubseteq E) \in \mathcal{T}} ((\neg D \sqcup E) \sqcap \{\iota\, (\neg(\neg D \sqcup E) \sqcup A_{D \sqsubseteq E})\}),$$

for fresh atomic concept $A_{D \sqsubseteq E}$

$\square$

Expressive Power

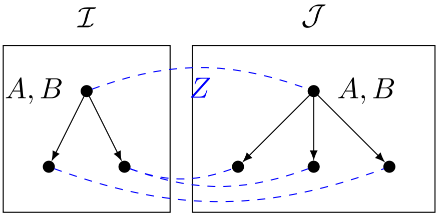
Standard bisimulation...

Definition. An *$\mathcal{ALC}$ bisimulation* is a relation $Z \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ such that for all $(d, e) \in Z$:

Atom: d and e satisfy same atomic concepts

Zig: if $(d, d') \in r^{\mathcal{I}}$, there is e' such that $(e, e') \in r^{\mathcal{J}}$ and $(d', e') \in Z$

Zag: reverse of Zig



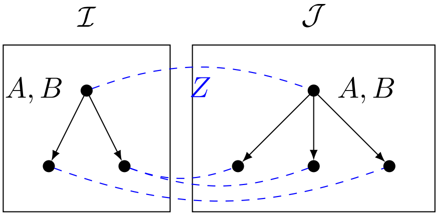
Z is an $\mathcal{ALC}$ bisimulation

Standard bisimulation...

Theorem.

- If d and e are $\mathcal{ALC}$ bisimilar, they satisfy the same $\mathcal{ALC}$ concepts
- If d and e satisfy the same $\mathcal{ALC}$ concepts and $\mathcal{I}, \mathcal{J}$ are ω-saturated, then d and e are $\mathcal{ALC}$ bisimilar

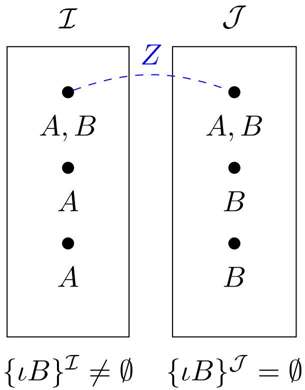
Theorem. An FO formula is invariant under $\mathcal{ALC}$ bisimulations iff it is equivalent to some $\mathcal{ALC}$ concept



Z is an $\mathcal{ALC}$ bisimulation

Standard bisimulation... is inadequate for $\mathcal{ALC}$ with DDs

Z is an $\mathcal{ALC}$ bisimulation, but
does not preserve $\{\iota B\}$ satisfiability.



$\mathcal{ALCO}_u^\iota$ bisimulation

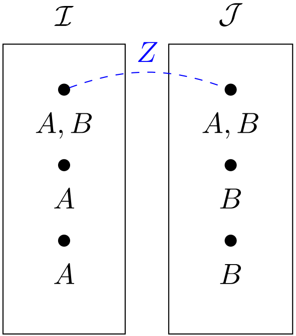
Definition.
Z is an $\mathcal{ALCO}_u^\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALCO}$ bisimualtion
2. Z is total
3. for all $(d, e) \in Z$:
   there is $d' \neq d$ $\mathcal{ALC}$ bisimilar with d iff there is $e' \neq e$ $\mathcal{ALC}$ bisimilar with e such that $(d', e') \in Z$

$\mathcal{ALCO}_u^\iota$ bisimulation

Definition.
Z is an $\mathcal{ALCO}_u^\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALCO}$ bisimualtion
2. Z is total
3. for all $(d, e) \in Z$:
   there is $d' \neq d$ $\mathcal{ALC}$ bisimilar with d iff there is $e' \neq e$ $\mathcal{ALC}$ bisimilar with e such that $(d', e') \in Z$

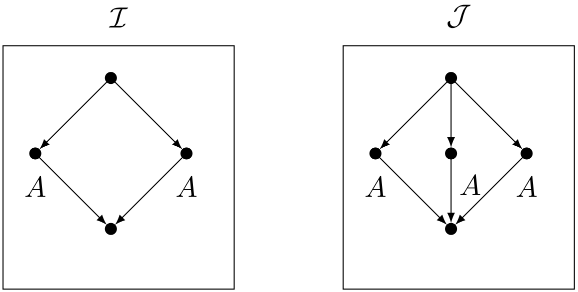


Z is **not an $\mathcal{ALCO}_u^\iota$ bisimulation**
as it does not satisfy conditions 2. and 3.

$\mathcal{ALCO}_u^\iota$ bisimulation

Definition.
Z is an $\mathcal{ALCO}_u^\iota$ *bisimulation* if:

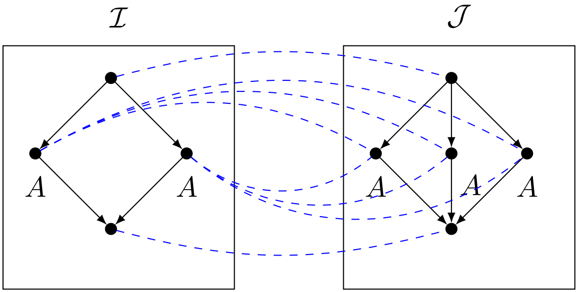
1. Z is an $\mathcal{ALCO}$ bisimualtion
2. Z is total
3. for all $(d, e) \in Z$:
   there is $d' \neq d$ $\mathcal{ALC}$ bisimilar with d iff there is $e' \neq e$ $\mathcal{ALC}$ bisimilar with e such that $(d', e') \in Z$



$\mathcal{ALCO}_u^\iota$ bisimulation

Definition.
Z is an $\mathcal{ALCO}_u^\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALCO}$ bisimualtion
2. Z is total
3. for all $(d, e) \in Z$:
   there is $d' \neq d$ $\mathcal{ALC}$ bisimilar with d iff there is $e' \neq e$ $\mathcal{ALC}$ bisimilar with e such that $(d', e') \in Z$

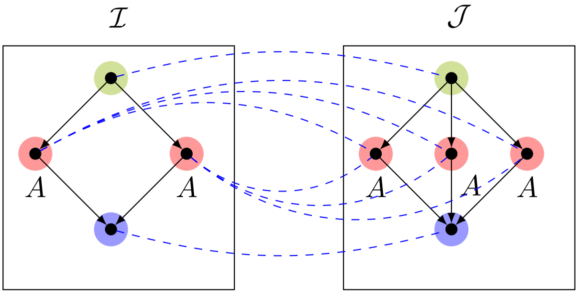


Z is the maximal $\mathcal{ALCO}_u^\iota$ **bisimulation**

$\mathcal{ALCO}_u^\iota$ bisimulation

Definition.
Z is an $\mathcal{ALCO}_u^\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALCO}$ bisimualtion
2. Z is total
3. for all $(d, e) \in Z$:
   there is $d' \neq d$ $\mathcal{ALC}$ bisimilar with d iff there is $e' \neq e$ $\mathcal{ALC}$ bisimilar with e such that $(d', e') \in Z$



Z is the maximal $\mathcal{ALCO}_u^\iota$ **bisimulation**

$\mathcal{ALCO}_u^\iota$ bisimulation

Definition.
Z is an $\mathcal{ALCO}_u^\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALCO}$ bisimualtion
2. Z is total
3. for all $(d, e) \in Z$:
   there is $d' \neq d$ $\mathcal{ALC}$ bisimilar with d iff there is $e' \neq e$ $\mathcal{ALC}$ bisimilar with e such that $(d', e') \in Z$

Theorem.

- If d and e are $\mathcal{ALCO}_u^\iota$ bisimilar, they satisfy the same $\mathcal{ALCO}_u^\iota$ concepts
- If d and e satisfy the same $\mathcal{ALCO}_u^\iota$ concepts and $\mathcal{I}, \mathcal{J}$ are ω-saturated, then d and e are bisimilar

Theorem. An FO formula is invariant under $\mathcal{ALCO}_u^\iota$ bisimulations iff it is equivalent to some $\mathcal{ALCO}_u^\iota$ concept

$\mathcal{ALC}\iota$ bisimulation

Definition.
Z is an $\mathcal{ALC}\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALC}$ bisimualtion
2. *Names*$(\mathcal{I})$ = *Names*$(\mathcal{J})$

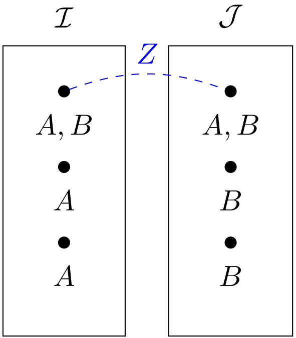
Names$(\mathcal{I}) = \{\mathcal{ALC}$ concept $C \mid$
C holds in a unique individual in $\mathcal{I}\}$

$\mathcal{ALC}\iota$ bisimulation

Definition.
Z is an $\mathcal{ALC}\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALC}$ bisimualtion
2. *Names*$(\mathcal{I})$ $=$ *Names*$(\mathcal{J})$

Names$(\mathcal{I}) = \{\mathcal{ALC}$ concept $C \mid$
C holds in a unique individual in $\mathcal{I}\}$



Z is **not an $\mathcal{ALC}\iota$ bisimulation** because *Names*$(\mathcal{I}) \neq$ *Names*$(\mathcal{J})$:

- $B \in$ *Names*$(\mathcal{I})$
- $B \notin$ *Names*$(\mathcal{J})$

$\mathcal{ALC}\iota$ bisimulation

Definition.
Z is an $\mathcal{ALC}\iota$ *bisimulation* if:

1. Z is an $\mathcal{ALC}$ bisimualtion, and
2. *Names*$(\mathcal{I})$ = *Names*$(\mathcal{J})$.

Names$(\mathcal{I}) = \{\mathcal{ALC}$ concept $C \mid$
C holds in a unique individual in $\mathcal{I}\}$

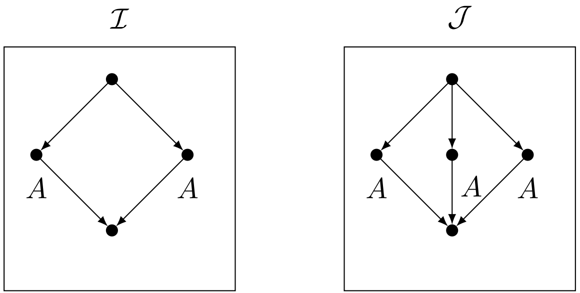
Theorem.

- If d and e are $\mathcal{ALC}\iota$ bisimilar, then they satisfy the same $\mathcal{ALC}\iota$ concepts
- If d and e satisfy the same $\mathcal{ALC}\iota$ concepts and both $\mathcal{I}$ and $\mathcal{J}$ are ω-saturated, then d and e are bisimilar

Theorem. An FO formula is invariant under $\mathcal{ALC}\iota$ bisimulations iff it is equivalent to some $\mathcal{ALC}\iota$ concept

Checking if $Names(\mathcal{I}) = Names(\mathcal{J})$

Ok, but how to check if $Names(\mathcal{J}) = Names(\mathcal{J})$?



$$Names(\mathcal{I}) = \{\exists r.A, \exists r.\exists r.\top, \neg\exists r.\top, \dots\}$$
$$Names(\mathcal{J}) = \{\exists r.A, \exists r.\exists r.\top, \neg\exists r.\top, \dots\}$$

Checking if $Names(\mathcal{I}) = Names(\mathcal{J})$

Ok, but how to check if $Names(\mathcal{J}) = Names(\mathcal{J})$?

Definition.
$\mathit{NamedInd}(\mathcal{I}) = \{d \mid \ d \in C^{\mathcal{I}}$, for some $C \in \mathit{Names}(\mathcal{I})\}$.

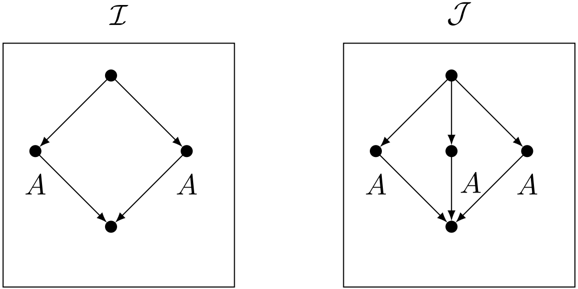
Theorem. The following are equivalent (if $\mathit{Names}(\mathcal{I}) \neq \emptyset$ and $\mathit{Names}(\mathcal{J}) \neq \emptyset$):

1. $\mathit{Names}(\mathcal{I}) = \mathit{Names}(\mathcal{J})$,
2. the maximal $\mathcal{ALC}$ bisimulation Z between $\mathcal{I}$ and $\mathcal{J}$ is a total relation and the restriction of Z to $\mathit{NamedInd}(\mathcal{I}) \times \mathit{NamedInd}(\mathcal{J})$ is also total.

Checking if $Names(\mathcal{I}) = Names(\mathcal{J})$

Theorem. The following are equivalent (if $\mathit{Names}(\mathcal{I}) \neq \emptyset$ and $\mathit{Names}(\mathcal{J}) \neq \emptyset$):

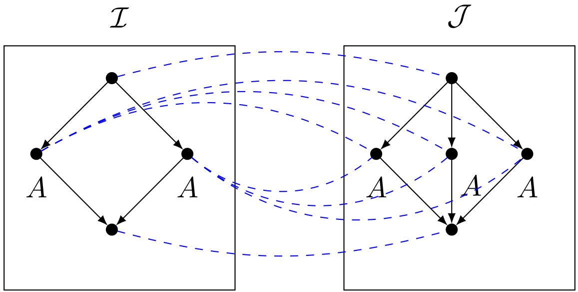
1. $\mathit{Names}(\mathcal{I}) = \mathit{Names}(\mathcal{J})$,
2. the maximal $\mathcal{ALC}$ bisimulation Z between $\mathcal{I}$ and $\mathcal{J}$ is a total relation and the restriction of Z to named individuals is also total.



Checking if $Names(\mathcal{I}) = Names(\mathcal{J})$

Theorem. The following are equivalent (if *Names*$(\mathcal{I}) \neq \emptyset$ and *Names*$(\mathcal{J}) \neq \emptyset$):

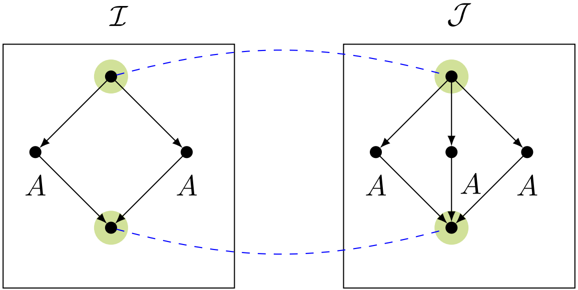
1. *Names*$(\mathcal{I}) =$ *Names*$(\mathcal{J})$,
2. the maximal $\mathcal{ALC}$ bisimulation Z between $\mathcal{I}$ and $\mathcal{J}$ is a total relation and the restriction of Z to named individuals is also total.



Checking if $Names(\mathcal{I}) = Names(\mathcal{J})$

Theorem. The following are equivalent (if $\mathit{Names}(\mathcal{I}) \neq \emptyset$ and $\mathit{Names}(\mathcal{J}) \neq \emptyset$):

1. $\mathit{Names}(\mathcal{I}) = \mathit{Names}(\mathcal{J})$,
2. the maximal $\mathcal{ALC}$ bisimulation Z between $\mathcal{I}$ and $\mathcal{J}$ is a total relation and the restriction of Z to named individuals is also total.



Reasoning in Practice

Tableau Systems

ABox rules:

$$(ABox_{\mathsf{I}})\ \frac{a : C \in ABox}{a : C} \qquad (ABox_r)\ \frac{r(a, a') \in ABox}{r(a, a')}$$

TBox rule:

$$(TBox)\ \frac{C \sqsubseteq D \in TBox, a : E}{a : \neg(C \sqcap \neg D)}$$

Clash rule:

$$(\bot)\ \frac{a : C, a : \neg C}{\bot}$$

Propositional rules:

$$(\neg\neg)\ \frac{a : \neg\neg C}{a : C} \qquad (\sqcap)\ \frac{a : C \sqcap D}{a : C, a : D} \qquad (\neg\sqcap)\ \frac{a : \neg(C \sqcap D)}{a : \neg C \mid a : \neg D}$$

Role rules:

$$(\exists r)\ \frac{a : \exists r.C}{b : C, r : (a, b)} \qquad (\neg\exists r)\ \frac{a : \neg\exists r.C, r : (a, a')}{a' : \neg C}$$

Global definite description rules:

$$(\iota_1^g)\ \frac{a : \iota C.D}{b : C, b : D} \qquad (\iota_2^g)\ \frac{a : \iota C.D, a' : C, a'' : C, a' : E}{a'' : E}$$

$$(\neg\iota^g)\ \frac{a : \neg\iota C.D, a' : E}{a' : \neg C \mid a' : \neg D \mid \begin{array}{c} b : C, b : A_C^g, \\ b' : C, b' : \neg A_C^g \end{array}} \qquad (cut_\iota^g)\ \frac{a : \iota C.D, a' : E}{a' : C \mid a' : \neg C}$$

Local definite description rules:

$$(\iota_1^\ell)\ \frac{a : \{\iota C\}}{a : C} \qquad (\iota_2^\ell)\ \frac{a : \{\iota C\}, a' : C, a'' : C, a' : D}{a'' : D}$$

$$(\neg\iota^\ell)\ \frac{a : \neg\{\iota C\}}{a : \neg C \mid \begin{array}{c} a : \neg A_C, \\ b : C, b : A_C \end{array}} \qquad (cut_\iota^\ell)\ \frac{a : \{\iota C\}, a' : D}{a' : C \mid a' : \neg C}$$

There are *tableau systems for* $\mathcal{ALC}\iota_L$*,* $\mathcal{ALC}\iota_G$*, and* $\mathcal{ALC}\iota$.

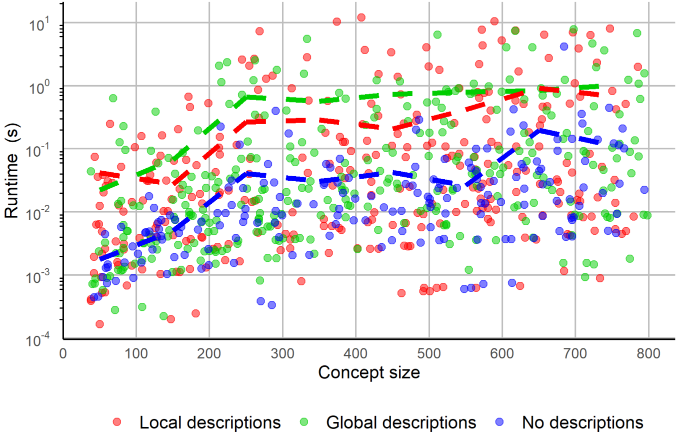
Theorem.
All three systems are
sound, complete, and terminating.

Implementation: OWLeDD Package

Python package: `https://pypi.org/project/OWLeDD/`

```
$ pip install OWLeDD
```

Evaluation:



Takeaways

Definite Descriptions (DDs) introduce new modeling capabilities in description logics

- All $\mathcal{ALC}$ extensions with DDs we consider are ExpTime-*complete* for both concept and ontology satisfiability
- Characterising logics with DDs requires a *new type of bisimulations*
- Local DDs $\{\iota C\}$ are *strictly less expressive* than global DDs $\iota C.D$
- *In practice* global DDs have *higher computational cost*

Future work:

- Lower complexity logics with DDs, e.g. $\mathcal{EL}$ family
- Complexity-optimal tableau systems
- Applications, e.g. computing referring expressions

Thank you for your attention!

p.walega@qmul.ac.uk

Funded by the European Union (ERC, ExtenDD, project number: 101054714). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

European Research Council
Established by the European Commission