Expressive powver of logical languages

Luciano Serafini

FBK-IRST, Trento, Italy

July 18, 2012

Expressive power

distinguishability

The expressive power of any language can be measured through its power of distinction–or equivalently, by the situations it considers indistinguishable.

So, to capture the expressive power of a language, we need to find some appropriate structural invariance between models.

Expressive power – main ingredients

Language

The notion of expressive power is referred to a (formal) language. But it is not an absolute notion, it is relative to the descriptive purpose of the language, i.e., the situations/domain/structures that are described by the language

Set of "items"

The expressive capability of a language always refers to a basic set/class of items. In the informal setting they can be a set of objects, people, situations, etc. In the logical setting they are a class of well defined mathematical structure in which the language is interpreted.

A compatibility/satisfiability relation

I.e., a binary relation that connects symbols with items. This relation intuitively represents the fact that the symbols represents the item, or that the item is represented correctly by the symbol. In logic this relation is the satisfiability relation.

Informal example

The language of smilies

The language of smilies can be used to distinguish people's emotions. In this example we have a language composed of four symbols which can be used to cluster the moods of Hilary Clinton in four subsets.

Language=Smaily4

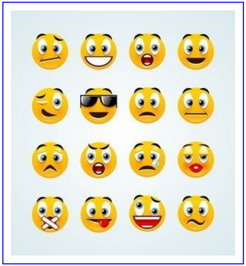
Situations

Informal example

The language of smilies

If we want to have a more accurate description of Hilary's moods, we need more symbols, i.e., a more expressive language such as the following:

Language=smaily15



Situations

Formal example:Propositional logic on finite sets of propositions

Language

Propositional language on a finite set of propositions $P = \{p_1, \dots, p_n\}$

Structures

Truth assignments of P

Satisfiability relation

The standard definition of $\mu \models \phi$

Expressivity

Maximal expressivity w.r.t. the set of truth assignments on finite P. Indeed, for every assignment μ there is a formula which is satisfied only by this assignment

$$\phi_\nu = \bigwedge_{\mu(p)=true} p \wedge \bigwedge_{\mu(p)=false} \neg p$$

This implies that every pair of assignments (models) μ and μ' can be distinguished by the formula ϕ_μ, which is verified by μ and falsified by μ'.

Formal example:Propositional logic on infinite sets of propositions

Language

Propositional language on an infinite set of propositions $P = \{p_1, p_2, \dots\}$

Structures

Truth assignments of P

Satisfiability relation

The standard definition of $\mu \models \phi$

Expressivity

The propositional finite language is not the most expressive since in general it's not possible to describe a single assignment with a finite formula. This will generate an infinite conjunction. However, distinguishability is guaranteed, as two assignments are always distinguished by a formula.

Example: First Order Logics expressivity limitations

Definition (Transitive closure)

Let U be some set and $R \subseteq U \times U$ be a binary relation on the universe U. Then the transitive closure R^+ of relation R is the smallest relation $R^+ \subseteq U \times U$ with

1. $R \subseteq R^+$
2. R^+ is transitive

Expressive limitations of FOL

Transitive closure can not be expressed in a First-order logic.

Example: First Order Logics expressivity limitations

Definition (Definability of a class of structures in FOL)

Let Σ be some signature and K a class of Σ-structures. The class K is definable (over signature Σ) if there is a closed Σ-formula ϕ_K such that for every Σ-structure $\mathcal{I}$

$$\mathcal{I} \models \phi_K \quad \textit{iff} \quad \mathcal{I} \in K$$

Definability of transitive closure

Let Σ be a signature containing the two binary relational symbols R and TCR. The problem of definability of transitive closure in FOL is the problem of finding a formula ϕ_{trans} such that for all Σ-structure $\mathcal{I}$:

$$\mathcal{I} \models \phi_{trans} \quad \textit{iff} \quad (R^{\mathcal{I}})^{+} = (TCR)^{\mathcal{I}}$$

Example: First Order Logics expressivity limitations

Theorem (Transitive closure is not definable in FOL)

Let Σ be a signature containing the two binary relational symbols R and TCR. There is no FOL formula ϕ_{trans} such that

$$\mathcal{I} \models \phi_{trans} \quad \text{iff} \quad (R^{\mathcal{I}})^+ = (TCR)^{\mathcal{I}}$$

Proof (outline)

Suppose by contradiction that there is a ϕ_{trans} that represents transitive closure. For every $n > 0$, we define the following formula ϕ_n

$$\exists x_1 x_n . \left(TCR(x_1, x_n) \wedge \neg \exists x_2, \dots, x_{n-1} . \left(\bigwedge_{i=1}^{n-1} R(x_i, x_{i+1}) \right) \right)$$

The set $\{\phi_{trans}, \phi_{i_1}, \phi_{i_2}, \dots, \phi_{i_k}\}$, satisfiable for every k, while $\{\phi_{trans}, \phi_1, \phi_2, \dots\}$ is not satisfiable. This contradicts compactness theorem in FOL, and therefore ϕ_{trans} cannot exist.

Example: First Order Logics expressivity limitations

Theorem (Transitive closure is not definable in FOL)

Let Σ be a signature containing the two binary relational symbols R and TCR. There is no FOL formula ϕ_{trans} such that

$$\mathcal{I} \models \phi_{trans} \quad \textit{iff} \quad (R^{\mathcal{I}})^{+} = (TCR)^{\mathcal{I}}$$

Proof (outline)

Suppose by contradiction that there is a ϕ_{trans} that represents transitive closure. For every $n > 0$, we define the following formula ϕ_n

$$\exists x_1 x_n . \left(TCR(x_1, x_n) \wedge \neg \exists x_2, \ldots, x_{n-1} . \left(\bigwedge_{i=1}^{n-1} R(x_i, x_{i+1}) \right) \right)$$

The set $\{\phi_{trans}, \phi_{i_1}, \phi_{i_2}, \ldots, \phi_{i_k}\}$, satisfiable for every k, while $\{\phi_{trans}, \phi_1, \phi_2, \ldots\}$ is not satisfiable. This contradicts compactness theorem in FOL, and therefore ϕ_{trans} cannot exist.

Logics that allows to define transitive closure

Second order logics

Second order logics extends FOL with the possibility of quantifying over sets and relations. I.e., possible to write a statement that refers to all the possible binary relation by the quantifier $\forall X$. (capitalized variables usually denote second order quantification)

$$
\begin{aligned}
\phi_{trans} = \ & \forall x, y. R(x, y) \rightarrow TCR(x, y) \wedge \\
& \forall x, y, z. TCR(x, y) \wedge TCR(y, z) \rightarrow TCR(x, z) \wedge \\
& \forall X.((\forall x, y : R(x, y) \rightarrow X(x, y) \wedge \\
& \quad \forall x, y, z. X(x, y) \wedge X(y, z) \rightarrow X(x, z)) \rightarrow \\
& \quad\quad \forall x, y. TCR(x, y) \rightarrow X(x, y))
\end{aligned}
$$

Logics that allows to define transitive closure

Infinitary logic $L_{\omega_1,\omega}$

Infinitary logics extends FOL with the possibility of having infinite disjunction and conjunction.

$$\begin{aligned}\phi_{trans} \quad = \quad & \forall x, y.R(x, y) \rightarrow TCR(x, y) \wedge \\ & \forall x, y, z.TCR(x, y) \wedge TCR(y, z) \rightarrow TCR(x, z) \wedge \\ & \forall x, y.TCR(x, y) \rightarrow \bigvee_{n \geq 1} \left(\exists x_1, \ldots, x_n x = x_1, y = x_n \bigwedge_{i=1}^{n-1} R(x_i, x_{i+1}) \right)\end{aligned}$$

Logics that allows to define transitive closure

Datalog

A datalog program is a first order theory on a function free signature Σ, where all the formulas are in the forms: $p_1(\mathbf{x}_1, \mathbf{y}_1) \wedge \cdots \wedge p_n(\mathbf{x}_n, \mathbf{y}_n) \rightarrow p(\mathbf{x}_1, \dots, \mathbf{x}_n)$ also written as

$$p(\mathbf{x}_1, \dots, \mathbf{x}_n) \leftarrow p_1(\mathbf{x}_1, by_1), \dots, p_n(\mathbf{x}_n, by_y) \qquad (1)$$

Not all the Σ-structure that satisfy all the formulas (1) of a datalog program are models for such a program.

A model for a datalog program is a Σ-structure, that minimizes the interpretation of predicates.

In the following logic program ϕ_{trans},

```
trans-closure-r(x,y) <- r(x,y)
trans-closure-r(x,z) <- trans-closure-r(x,y), trans-closure-r(y,z)
```

the minimal interpretation of `trans-closure-r` is indeed the transitive closure of the interpretation of `r`.

Distinguishability of Interpretations

Distinguishing between models

If M and M' are two models of a logic $\mathcal{L}$, then we say that $\mathcal{L}$ is capable to distinguish M from M' if there is a formula ϕ of the language of $\mathcal{L}$ such that

$$M \models_{\mathcal{L}} \phi \quad \text{end} \quad M \not\models_{\mathcal{L}} \phi$$

Proving non equivalence

To show that two logics $\mathcal{L}_1$ and $\mathcal{L}_2$ with the same class of models, are not equivalent it's enough to show that there are two models m and m' which are distinguishable in $\mathcal{L}_1$ nd non distinguishable in $\mathcal{L}_2$.

Bisimulation

The notion of bisimulation in description logics is intended to capture object equivalences and property equivalences.

Definition (Bisimulation)

A bisimulation ρ between two $\mathcal{ALC}$ interpretations $\mathcal{I}$ and $\mathcal{J}$ is a relation on $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ such that
if $d\rho e$ then the following hold:

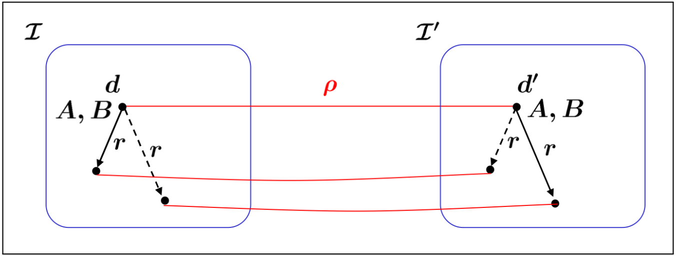
object equivalence $d \in A^{\mathcal{I}}$ if and only if $e \in A^{\mathcal{J}}$;

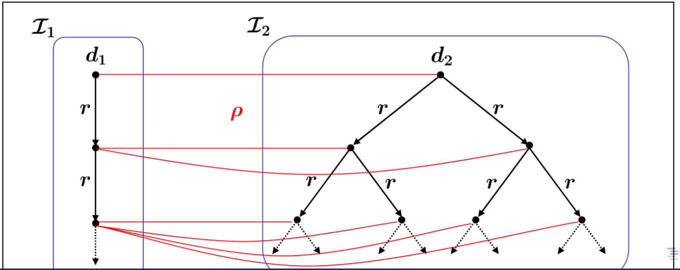
relation equivalence

- for all d' with $\langle d, d' \rangle \in R^{\mathcal{I}}$ there is and e' with $d' \rho e'$ such that $\langle e, e' \rangle \in R^{\mathcal{J}}$
- Same property in the opposite direction

$(\mathcal{I}, d) \sim (\mathcal{J}, e)$ means that there is a bisimulation ρ between $\mathcal{I}$ and $\mathcal{J}$ such that $e\rho e$.

Bisimulation





Bisimulation and $\mathcal{ALC}$

Lemma

$\mathcal{ALC}$ cannot distinguish the interpretations $\mathcal{I}$ and $\mathcal{J}$ when $(\mathcal{I}, d) \sim (\mathcal{J}, e)$.

Exercise

Show by induction on the complexity of concepts, that if $(\mathcal{I}, d) \sim (\mathcal{J}, e)$, then

$$d \in C^{\mathcal{I}} \quad \text{if and only if} \quad e \in C^{\mathcal{J}}$$

Bisimulation and $\mathcal{ALC}$

Definition (Disjoint union)

For every two interpretations $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ and $\mathcal{J} = \langle \Delta^{\mathcal{J}}, \cdot^{\mathcal{J}} \rangle$, the disjoint union of $\mathcal{I}$ and j is:

$$\mathcal{I} \uplus \mathcal{J} = \langle \Delta^{\mathcal{I} \uplus \mathcal{J}}, \cdot^{\mathcal{I} \uplus \mathcal{J}} \rangle$$

where

- $\Delta^{\mathcal{I} \uplus \mathcal{J}} = \Delta^{\mathcal{I}} \uplus \Delta^{\mathcal{J}}$
- $A^{\mathcal{I} \uplus \mathcal{J}} = A^{\mathcal{I}} \uplus A^{\mathcal{J}}$
- $R^{\mathcal{I} \uplus \mathcal{J}} = R^{\mathcal{I}} \uplus R^{\mathcal{J}}$

Exercise

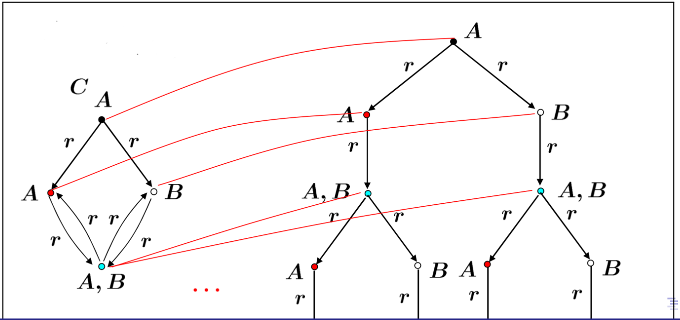
Prove via bisimulation lemma that: if: $\mathcal{I} \models C \sqsubseteq D$ and $\mathcal{J} \models C \sqsubseteq D$ then $\mathcal{I} \uplus J \models C \sqsubseteq D$.

Tree model property

Theorem

An $\mathcal{ALC}$ concept C is satisfiable w.r.t, a T-box $\mathcal{T}$ if and only if there is a tree-shaped interpretation $\mathcal{I}$ that satisfies $\mathcal{T}$, and an object d such that $d \in C^{\mathcal{I}}$.

Proof.



Extensions of $\mathcal{ALC}$

Inverse roles $\mathcal{ALCI}$ R^-. make it possible to use the inverse of a role. For example, we can specify has_Parent as the inverse of has_Child,

$$\text{has\_Parent} \equiv \text{has\_Child}^-$$

meaning that $\text{hasParent}^{\mathcal{I}} = \{(y, x) \mid (x, y) \in \textit{has\_ChildI}^{\mathcal{I}}\}$

Transitive roles tr(R) used to state that a given relation is transitive

$$Tr(hasAncestor)$$

meaning that $(x, y), (y, z) \in \text{hasAncestor}^{\mathcal{I}} \rightarrow (x, z) \in \text{hasAncestor}^{\mathcal{I}}$

Subsumptions between roles $R \sqsubseteq S$ used to state that a relation is contained in another relation.

$$hasMother \sqsubseteq hasParent$$

Inverse role

Exercise

Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCI}$.

Solution

Suggestion: do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $\mathcal{ALCI}$.

Inverse role

Exercise

Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCI}$.

Solution

Suggestion: do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $\mathcal{ALCI}$.

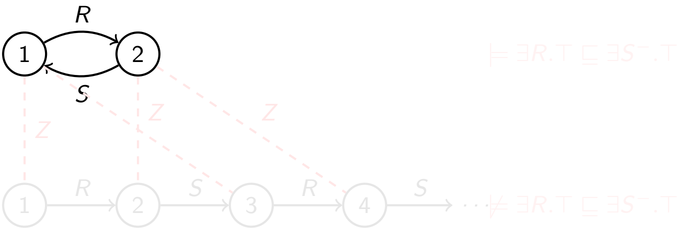
Inverse role

Exercise

Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCI}$.

Solution

Suggestion: do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $\mathcal{ALCI}$.



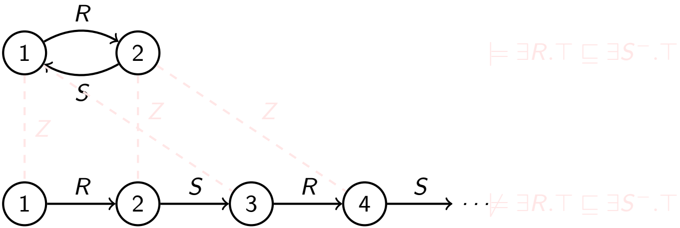
Inverse role

Exercise

Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCI}$.

Solution

Suggestion: do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $\mathcal{ALCI}$.



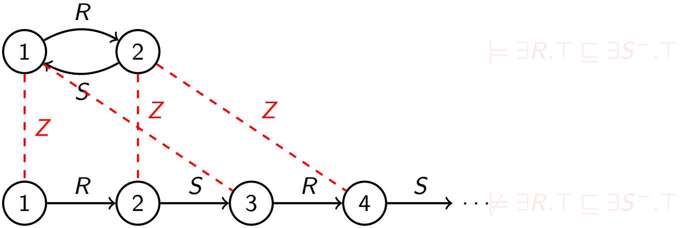
Inverse role

Exercise

Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCI}$.

Solution

Suggestion: do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $\mathcal{ALCI}$.



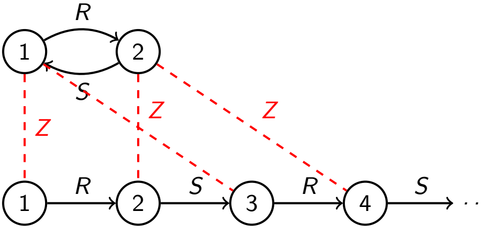
Inverse role

Exercise

Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCI}$.

Solution

Suggestion: do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $\mathcal{ALCI}$.



$\models \exists R.\top \sqsubseteq \exists S^{-}.\top$

$\not\models \exists R.\top \sqsubseteq \exists S^{-}.\top$

Extensions of $\mathcal{ALC}$

Number restrictions $\mathcal{ALCN}$ $(\leq n)R$ $[(\geq n)R]$

$$Persons \sqsubseteq (\leq 1)\text{is\_merried\_with}$$

Number restriction allows to impose that a relation is a function

Qualified Number restrictions $\mathcal{ALCQ}$ $(\leq n)R.C$ $[(\geq n)R.C]$

$$\begin{aligned} \text{football\_team} \sqsubseteq\ & (\geq 1)\text{has\_player.Golly} \sqcap \\ & (\leq 2)\text{has\_player.Golly} \sqcap \\ & (\geq 2)\text{has\_player.Defensor} \sqcap \\ & (\geq 4)\text{has\_player.Defensor} \sqcap \\ & \ldots \end{aligned}$$

Number restriction

Exercise

Prove that number restriction is an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCN}$.

Solution

$\models (\leq 1)R$

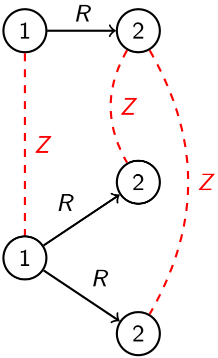
$\not\models (\leq 1)R$

Number restriction

Exercise

Prove that number restriction is an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show that that $\mathcal{ALC}$ is strictly less expressive than $\mathcal{ALCN}$.

Solution



$\models (\leq 1)R$

$\not\models (\leq 1)R$

Qualified number restriction

Exercise

Prove that qualified number restriction is an effective extension of the expressivity of $\mathcal{ALCN}$, i.e., show that that $\mathcal{ALCN}$ is strictly less expressive than $\mathcal{ALCQ}$.

Solution (outline)

1. *Extend the notion of bisimulation relation to $\mathcal{ALCN}$.*
2. *Prove that $\mathcal{ALCN}$ is bisimulation invariant for the bisimulation relation defined in 1*
3. *Prove that $\mathcal{ALCQ}$ is more expressive than $\mathcal{ALCN}$.*

Bisimulation for $\mathcal{ALCN}$

Definition ($\mathcal{ALCN}$-Bisimulation)

A $\mathcal{ALCN}$-bisimulation ρ between two $\mathcal{ALCN}$ interpretations $\mathcal{I}$ and $\mathcal{J}$ is a bisimulation ρ, that satisfies the following additional condition when $d\rho e$:

relation (cardinality) equivalence

- if $d_1, \dots, d_n$ are all the distinct elemnts of $\Delta^{\mathcal{I}}$ such that $\langle d, d_i\rangle \in R^{\mathcal{I}}$ for $1 \leq i \leq n$, then there are exactly n, $e_1, \dots, e_n$ elements of $\Delta^{\mathcal{J}}$ such that $(e, e_i) \in R^{\mathcal{J}}$ for all $1 \leq i \leq n$
- Same property in the opposite direction

$(\mathcal{I}, d) \sim (\mathcal{J}, e)$ means that there is a bisimulation ρ between $\mathcal{I}$ and $\mathcal{J}$ such that $e\rho e$.

Invariance w.r.t. $\mathcal{ALCN}$

Theorem

If $(\mathcal{I}, d) \sim (\mathcal{J}, e)$ then for every $\mathcal{ALCN}$ concept C $(\mathcal{I}, d) \models C$ if and only if $(\mathcal{J}, e) \models C$

Proof.

By induction on the complexity of C, similar as for $\mathcal{ALC}$ bisimulation with the following additional base step:

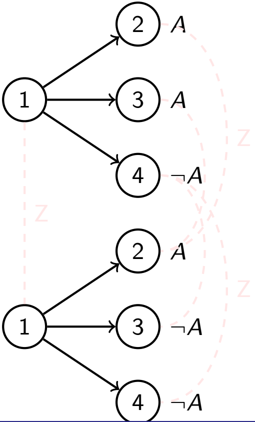
If C is $(\leq n)R$ If $(\mathcal{I}, d) \models (\leq n)R$, then there are $m \leq n$ elements $d_1, \ldots, d_m$ with $R(d, d_i)$. The additional condition on $\mathcal{ALCI}$-bisimulation implies that, there are exactly m elements $e_1, \ldots, e_m$, of $\Delta^{\mathcal{J}}$ such that $(e, e_i) \in R^{\mathcal{J}}$. which implies that $(\mathcal{J}, e) \models (\leq n)R$.

□

$\mathcal{ALCQ}$ is more expressive than $\mathcal{ALCN}$

Proof outline

We show that in $\mathcal{ALCQ}$ we can distinguish two models which are not distinguishable in $\mathcal{ALCN}$



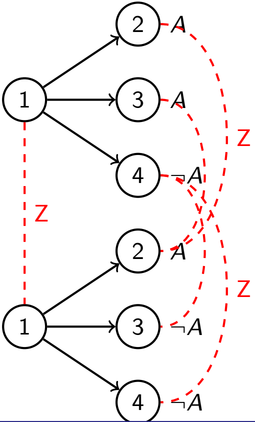
$\models (\leq 1)R.\neg A$

$\not\models (\leq 1)R.\neg A$

$\mathcal{ALCQ}$ is more expressive than $\mathcal{ALCN}$

Proof outline

We show that in $\mathcal{ALCQ}$ we can distinguish two models which are not distinguishable in $\mathcal{ALCN}$



$\models (\leq 1)R.\neg A$

$\not\models (\leq 1)R.\neg A$

Conclusion

Three main messages

- The expressive power of a formal language represent it's capability of distinguishing models/structures.
- Comparing the expressive power of two languages can be done only if they are interpreted on the same class of models/structures
- A language L_1 is more expressive than L_2 if L_1 can distinguish two models that are indistinguishable by L_2