Expressive power 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.
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 $P = \{p_1, \ldots, p_n\}$

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 $\mu$ there is a formula which is satisfied only by this assignment

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

This implies that every pair of assignments (models) $\mu$ and $\mu'$ can be distinguished by the formula $\phi_{\mu}$, which is verified by $\mu$ and falsified by $\mu'$. 
Formal example: Propositional logic on infinite sets of propositions

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

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

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

Definability of transitive closure

Let $\Sigma$ 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 $\phi_{trans}$ such that for all $\Sigma$-structure $\mathcal{I}$:

$$\mathcal{I} \models \phi_{trans} \quad \text{iff} \quad (R^\mathcal{I})^+ = (TCR)^\mathcal{I}$$
Example: First Order Logics expressivity limitations

Theorem (Transitive closure is not definable in FOL)

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

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

Proof (outline)

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

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

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 $\phi_{trans}$ cannot exist.
Example: First Order Logics expressivity limitations

Theorem (Transitive closure is not definable in FOL)

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

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

Proof (outline)

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

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

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 $\phi_{trans}$ 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)

$$
\phi_{trans} = \forall x, y. R(x, y) \rightarrow TCR(x, y) \land \\
\forall x, y, z. TCR(x, y) \land TCR(y, z) \rightarrow TCR(x, z) \land \\
\forall X.((\forall x, y : R(x, y) \rightarrow X(x, y) \land \\
\forall x, y, z. X(x, y) \land X(y, z) \rightarrow X(x, z)) \rightarrow \\
\forall x, y. TCR(x, y) \rightarrow X(x, y))
$$
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.

\[
\phi_{\text{trans}} = \forall x, y. R(x, y) \rightarrow TCR(x, y) \land \\
\forall x, y, z. TCR(x, y) \land TCR(y, z) \rightarrow TCR(x, z) \land \\
\forall x, y. TCR(x, y) \rightarrow \bigvee_{n \geq 1} \left( \exists x_1, \ldots, x_n x = x_1, y = x_n \land \bigwedge_{i=1}^{n-1} R(x_i, x_{i+1}) \right)
\]
Logics that allows to define transitive closure

Datalog

A datalog program is a first order theory on a function free signature $\Sigma$, where all the formulas are in the forms: $p_1(x_1, y_1) \land \cdots \land p_n(x_n, y_n) \rightarrow p(x_1, \ldots, x_n)$ also written as

$$p(x_1, \ldots, x_n) \leftarrow p_1(x_1, by_1), \ldots, p_n(x_n, by_y) \quad (1)$$

Not all the $\Sigma$-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 $\Sigma$-structure, that minimizes the interpretation of predicates.

In the following logic program $\phi_{\text{trans}}$,

- $\text{trans-closure-r}(x,y) \leftarrow \text{r}(x,y)$
- $\text{trans-closure-r}(x,z) \leftarrow \text{trans-closure-r}(x,y), \text{trans-closure-r}(y,z)$

the minimal interpretation of $\text{trans-closure-r}$ is indeed the transitive closure of the interpretation of $\text{r}$.
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 $\phi$ 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 $\rho$ 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 $\rho$ between $\mathcal{I}$ and $\mathcal{J}$ such that $e \rho e$. 
Bisimulation

I

\[ d \]
\[ A, B \]
\[ r \]
\[ r \]

\[ \rho \]

I'

\[ d' \]
\[ A, B \]
\[ r \]
\[ r \]

I₁

\[ d₁ \]
\[ r \]
\[ r \]
\[ r \]

\[ \rho \]

I₂

\[ d₂ \]
\[ r \]
\[ r \]
\[ r \]
\[ r \]
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 $ALC$

**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 $\mathcal{J}$ is:

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

where

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

**Exercise**

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

Luciano Serafini

FBK-IRST, Trento, Italy

Expressive power of logical languages
Tree model property

**Theorem**

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

**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 $\text{hasParent}^\mathcal{I} = \{(y, x) \mid (x, y) \in \text{has Child}^\mathcal{I}\}$

**Transitive roles** $\text{tr}(R)$ used to state that a given relation is transitive

$$\text{Tr}((\text{hasAncestor}))$$

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.

$$\text{hasMother} \sqsubseteq \text{hasParent}$$
**Inverse role**

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

Solution
*Suggestion:* do it via bisimulation. I.e., show that there are two models that bisimulate which are distinguishable in $ALCI$.
Inverse role

**Exercise**
Prove that the inverse role primitive constitutes an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show 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 $\mathcal{ALC}I$.

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

\[
\begin{array}{c}
\exists R.T \sqsubseteq E.S^-T \\
\not\exists R.T \sqsubseteq E.S^-T
\end{array}
\]
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 $\mathcal{ALCI}$.
Extensions of $\mathcal{ALC}$

**Number restrictions** $\mathcal{ALCN}$ \((\leq n)R \sqcap (\geq n)R\)

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

Number restriction allows to impose that a relation is a function

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

$football\_team \sqsubseteq (\geq 1)\text{has\_player}.Golly \sqcap$

\((\leq 2)\text{has\_player}.Golly \sqcap$

\((\geq 2)\text{has\_player}.Defensor \sqcap$

\((\geq 4)\text{has\_player}.Defensor \sqcap$

\ldots$

Luciano Serafini FBK-IRST, Trento, Italy

Expressive power of logical languages
Number restriction

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

Solution

[Diagram showing the relationship between nodes 1 and 2 with restrictions $(\leq 1)R$.]
Number restriction

Exercise
Prove that number restriction is an effective extension of the expressivity of $\mathcal{ALC}$, i.e., show 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 $\mathcal{ALCN}$ 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 $\rho$ between two $\mathcal{ALCN}$ interpretations $\mathcal{I}$ and $\mathcal{J}$ is a bisimulation $\rho$, that satisfies the following additional condition when $d \rho e$:

relation (cardinality) equivalence

- if $d_1, \ldots, d_n$ are all the distinct elements 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 $n$, $e_1, \ldots, e_n$ elements of $\Delta^\mathcal{J}$ such that $\langle e, e_i \rangle \in R^\mathcal{J}$ 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 $\rho$ between $\mathcal{I}$ and $\mathcal{J}$ such that $e \rho e$. 
Invariance w.r.t. $\mathcal{ALCN}$

**Theorem**

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

**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 $(I, d) \models (\leq n)R$, 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 $\Delta^J$ such that $(e, e_i) \in R^J$. which implies that $(J, e) \models (\leq n)R$.
**ALCQ** is more expressive than **ALCN**

**Proof outline**
We show that in **ALCQ** we can distinguish two models which are not distinguishable in **ALCN**

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

$$\not\models (\leq 1)R. \neg A$$
**ALCQ** is more expressive than **ALCN**

**Proof outline**
We show that in **ALCQ** we can distinguish two models which are not distinguishable in **ALCN**

\[
\begin{align*}
1 & \xrightarrow{A} 3 \\
2 & \xrightarrow{A} 4 \\
3 & \xrightarrow{A} 4 \\
1 & \xrightarrow{A} 3 \\
2 & \xrightarrow{A} 4 \\
\end{align*}
\]

\[\models (\leq 1)R. \neg A\]

\[\not\models (\leq 1)R. \neg A\]
Conclusion

Three main messages

- The expressive power of a formal language represents its 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 $L_2$. 