Expressiveness Revisited

Carlos Areces
Dept. of Computer Science
University of Warwick
Coventry CV4 7AL, England
careces@dcs.warwick.ac.uk

Maarten de Rijke
ILLC, University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam, The Netherlands
mdr@wins.uva.nl

Abstract

We consider two recently proposed definitions of the expressive power of description logics, one due to Baader, the other due to Kurtonina and de Rijke. The proposals are non-equivalent, and we explain their differences, similarities, as well as ways in which they are related. We also map out a research agenda resulting from efforts to combine the two approaches.

1 Introduction

The area of description logics is a young and rapidly growing field. New languages keep appearing and new results are added on a regular basis. But as with all young fields, the area still lacks a widely accepted, unifying background theory, and as a consequence it often has a disorganized and incomplete flavor. From a logician's point of view one of the most important notions that had been underdeveloped until recently is that of *expressive power*, especially since a measure of the expressive power of a language is fundamental for comparing different proposals. Description logics aim to represent knowledge and to manipulate it in efficient ways. Hence, it is important to have an exact definition of what a given description logic can express together with a method to compare or measure the difference in expressive power of two formalisms. Naturally, what one is after is to get the highest expressive power at the lowest computational costs.

The issue of expressive power for description logics was first addressed by Baader [1] and Borgida [3], and later by Kurtonina and de Rijke [13, 14]. The approaches proposed by the first two authors are similar, but they differ in many ways from the approach of the latter two authors which has a definite model theoretic character. The different approaches induce different classifications of description logics with respect to their expressive power. For example, according to Baader the description language $\mathcal{TF}$ has the same expressive power as the language $\mathcal{NTF}$. In contrast, according to Kurtonina and de Rijke, $\mathcal{NTF}$ is strictly more expressive than $\mathcal{TF}$ (the definition of the specific description logics mentioned will be given in Section 2.) How can this be? Can both results be correct? The aim of this note is to understand the differences and similarities between the two approaches to the expressive power of description logics and to determine their interrelations, and thereby add new results to both.

This paper is organized as follows. In Section 2 we recall basic definitions and notions pertaining to description logics. In Section 3 we discuss the issue of expressive power, present the approaches to the expressive power of description logics that are due to Baader and Kurtonina and de Rijke, and we identify some of their differences and similarities. In Section 4 we map out a new research agenda resulting from efforts to merge the two approaches; this is very much work in progress. In the final section we draw our conclusions.

2 Background

Description logics are specialized languages related to the KL-ONE system of Brachman and Schmolze [5]. They are designed for representing knowledge, and the general aim is to provide a small set of operations to describe pieces of information together with efficient methods to obtain inferences.

In the early days, it was not clear if a given description logic was really a logic in the strict sense of the notion (i.e., with a formal syntax and semantics, model theory, etc.) or just a tool or methodology. Actually, some of the early description logics were no more than a computer system implementation to store information in an efficient way and derive inferences from it.

Nowadays, description logics are generally considered to be "variations" of first-order logic (FOL) — either restrictions or restrictions plus some added operators. These variations are motivated on one hand by the undecidability of the inference problem for FOL and on the other by an intention to preserve the structure of the knowledge represented.

As a basic way to preserve structure, description logics descending from KL-ONE split information in two kinds: A-boxes which contain assertional information (facts), and T-boxes with terminological knowledge (definitions of derived notions). A-boxes are simple, usually just a list of atomic formulas. The expressive power of a description logic is in the constructions allowed in T-boxes: which derived notions can be defined?

Let's see an example. The following is a valid pair $\langle$ T-Box, A-box $\rangle$ in the description logic $\mathcal{FLEU}_\mathcal{R}$ defined in Table 1 below (with disjointness axioms).

$$\texttt{T-BOX:}\quad T \;=\; \left\{\begin{array}{l} \text{Man} \doteq \text{Human} \sqcap \\ \qquad\quad \text{Male} \sqcap \\ \qquad\quad \text{Rest-Man} \\ \text{Woman} \doteq \text{Human} \sqcap \\ \qquad\qquad \text{Female} \sqcap \\ \qquad\qquad \text{Rest-Woman} \\ \text{Father} \doteq \text{Man} \sqcap \\ \qquad\qquad \exists_{\geq 1}.\text{Child} \\ \text{Loves} \doteq \text{Child} \sqcup \\ \qquad\qquad \text{Rest-Love} \\ \text{dis}(\text{Man}, \text{Woman}) \end{array}\right\}$$

$$\texttt{A-BOX:}\quad A \;=\; \left\{\begin{array}{l} \text{Human}(m) \\ \text{Human}(e) \\ \text{Child}(m, e) \\ \text{Male}(m) \\ \text{Rest-Man}(m) \\ \text{Rest-Love}(m, e) \end{array}\right\}$$

The first three formulas in T define concepts (subsets of the domain to which the represented information refers). For example, the third defines the derived concept 'Father' as all those men which have at least one child. The fourth formula defines a role[1] (a binary relation) which can be read as saying that all parents love their children. Finally, the last formula is an axiom (a rule governing the definition of concepts and roles) establishing that the concepts 'Man' and 'Woman' should be disjoint. This T-box is *acyclic* (no concept or role occurs on both sides of the same definition), but it is not *unfolded* (as a concept occurs on both sides of different definitions.)

It is obvious that the example above can be rewritten as a set of first-order formulas. In general T-boxes can be rewritten in L_3, the three variable fragment of FOL, possibly extended with counting expressions; see [3].

Summing up, concepts and roles are the building blocks of description logics, and each description logic is characterized by the way these can be combined in T-boxes. In [1] description logics are *defined* just as sets of T-boxes. We will take the same definition, but we are convinced that the T-boxes allowed are specified by the concept and role forming constructions the language of a given description logic admits and the kind of axioms it allows.

The *signature* of a description logic is given by specifying three disjoint sets $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{A}$ of concept names and role names, and atomic constants respectively.

Description logics are interpreted on *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty domain, and $\cdot^\mathcal{I}$ is an interpretation function assigning subsets of $\Delta^\mathcal{I}$ to concept names, binary relations over $\Delta^\mathcal{I}$ to role names and single elements of $\Delta^\mathcal{I}$ to atomic constants. Table 1 list operations which appear in different description logic, together with their notation and semantics.

The logic $\mathcal{FL}^-$ [4] is defined as the description logic allowing universal quantification, conjunction and unqualified existential quantifications of the form $\exists R.\top$. $\mathcal{FL}_0$ is the restriction of $\mathcal{FL}$ by not allowing existential quantification. $\mathcal{AL}$ [18] extends $\mathcal{FL}^-$ with negation of atomic concept names. $\mathcal{TF}$ [16] allows concept conjunction, role conjunction, universal quantification and number restric-

[1] The sets of concept and role symbols are assumed to have empty intersection and hence no confusion arises about when a given equation defines a concept or a role.

Constructor	Syntax	Semantics
concept name	C	$C^{\mathcal{I}} \subset \Delta^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\perp$	$\emptyset$
conjunction	$C \sqcap C'$	$C^{\mathcal{I}} \cap C'^{\mathcal{I}}$
disjunction ($\mathcal{U}$)	$C \sqcup C'$	$C^{\mathcal{I}} \cup C'^{\mathcal{I}}$
negation ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
univ. quant.	$\forall R.C$	$\{d_1 \mid \forall d_2(d_1 R^{\mathcal{I}} d_2 \to C^{\mathcal{I}} d_2)\}$
exist. quant. ($\mathcal{E}$)	$\exists R.C$	$\{d_1 \mid \exists d_2(d_1 R^{\mathcal{I}} d_2 \wedge C^{\mathcal{I}} d_2)\}$
numb. restr. ($\mathcal{N}$)	$(\geq nR)$	$\{d_1 \mid \lvert\{(d_1, d_2) \in R^{\mathcal{I}}\}\rvert \geq n\}$
	$(\leq nR)$	$\{d_1 \mid \lvert\{(d_1, d_2) \in R^{\mathcal{I}}\}\rvert \leq n\}$
role name	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
role conj. ($\mathcal{R}$)	$R \sqcap R'$	$R^{\mathcal{I}} \cap R'^{\mathcal{I}}$
role disj. ($\mathcal{U}_{\mathcal{R}}$)	$R \sqcup R'$	$R^{\mathcal{I}} \cup R'^{\mathcal{I}}$

Table 1: Common operators of description logics

tion, while $\mathcal{NTF}$ is the extension of $\mathcal{TF}$ with negation of atomic concept names. The names in parenthesis in Table 1 are the usual ones for defining extensions. For example, $\mathcal{ALC}$ is $\mathcal{AL}$ extended with full negation.

We will only consider first-order description logics and acyclic definitions. As these can always be unfolded and completed (even though this has an impact on their size), we restrict ourselves to full definitions $Q \doteq \varphi$ where Q does not occur anywhere else in the T-box. When only this kind of T-boxes are considered, the left-hand sides of equations become trivial and we are only interested in definitions occurring on the right-hand side.

If T is a T-box, we use $\mathrm{Int}(T)$ to denote the class of interpretations for T, i.e., the interpretations that make all equalities in T true and satisfy its axioms.

3 Two Notions of Expressive Power

In this section we describe and compare the two definitions of expressive power due to Baader, and Kurtonina and de Rijke, respectively. Before doing so, we briefly the discuss the issue of expressive power in a wider setting.

§3.1. Expressiveness versus Complexity. A popular slogan in the area of knowledge representation is 'complexity versus expressiveness': the more expressive a (description) logic is, the higher the complexity of the reasoning tasks that can be performed in it. The complexity of satisfiability and subsumption problems for description logics has been studied extensively, but the problem of expressiveness has not kept pace. Understanding the expressive power of one's representation formalism is important; it may be used by the designer of knowledge based systems to help choose the description logic that best fits his or her descriptive requirements.

More generally, expressive power has been a key interest of logicians and computer scientists in a large number of areas. Without attempting to supply an exhaustive list, we should at least mention model-theoretic logics [2], database query languages [10], finite variable logics [11], and modal and temporal logic [8].

§3.2. Baader [1]. In first-order logic two sentences are equivalent if they have the same class of models. For the comparison of the expressive power of two description logics we may need to compare different languages having different vocabularies and interpretations, but despite these differences we want to relate the meaning of expressions in one language to expressions in the other. To cope with this, Baader introduces *embeddings*: instead of asking for equality of classes of models, we require the classes to be embeddable in each other through translation functions. The formal definition is as follows.

Definition 1 Let $\mathcal{L}_1$, $\mathcal{L}_2$ be description logics, and $T_1 \in \mathcal{L}_1$, $T_2 \in \mathcal{L}_2$ be T-boxes. Also, let $\mathcal{I} \in \mathrm{Int}(T_1)$, $\mathcal{J} \in \mathrm{Int}(T_2)$ be interpretations and f a function from the concept and role names in T_1 to those in T_2 (which assignes concepts to concepts and roles to roles). Then

1. $\mathcal{I}$ *is embedded in* $\mathcal{J}$ *by* f ($\mathcal{I} \subseteq_f \mathcal{J}$) if for all concept and role names Q occurring in T_1, $Q^{\mathcal{I}} = f(Q)^{\mathcal{J}}$.
2. We write $\mathrm{Int}(T_1) =_f \mathrm{Int}(T_2)$ if for all $\mathcal{I} \in \mathrm{Int}(T_1)$ there is $\mathcal{J} \in \mathrm{Int}(T_2)$ such that $\mathcal{I} \subseteq_f \mathcal{J}$ and for all $\mathcal{J} \in \mathrm{Int}(T_2)$ there is $\mathcal{I} \in \mathrm{Int}(T_1)$ such that $\mathcal{I} \subseteq_f \mathcal{J}$.
3. $\mathcal{L}_1$ can be *expressed by* $\mathcal{L}_2$ *through embeddings* ($\mathcal{L}_1 \leq_e \mathcal{L}_2$) if for all $T_1 \in \mathcal{L}_1$ there exists $T_2 \in \mathcal{L}_2$ with $\mathrm{Int}(T_1) =_f \mathrm{Int}(T_2)$. We say that $\mathcal{L}_1$ and $\mathcal{L}_2$ *have the same expressive power through embeddings* ($\mathcal{L}_1 =_e \mathcal{L}_2$) if $\mathcal{L}_1 \leq_e \mathcal{L}_2$ and $\mathcal{L}_2 \leq_e \mathcal{L}_2$.

Some comments are in order. First, it is easy to check that of "having the same expressive power through embeddings" is indeed an equivalence relation. Also, it seems to capture the notion of "everything that can be said in $\mathcal{L}_1$ can be said in $\mathcal{L}_2$" and vice versa. It is related to the standard definition of equivalence of logic systems as presented in [7] on the one hand, and to the notion of equivalence of equational theories in equational logic [19] on the other. But there are important differences between Baader's definition and the other two. Unlike Baader's definition, the definition stemming from equivalence of logic systems applies to systems over the *same* similarity type only, and hence no translation (or embedding) is involved. And unlike Baader's definition, the definition of equivalent equational theories requires explicit translations of the operations which differentiate between theories. Formally, if Γ_1 and Γ_2 are equational theories then Γ_1 *is equivalent to* Γ_2 if there exist sets of definitions Δ_1 and Δ_2 in the corresponding languages such that $\Gamma_1, \Delta_1 \vdash \Gamma_2$ and $\Gamma_2, \Delta_2 \vdash \Gamma_1$ (plus further requirements).

One of the important characteristics of Baader's definition of expressive power is that it permits *translation*. For each of the concept and role names occurring in a T-box T_1 we can pick a corresponding name in T_2 which will be mimicking it. If we can do this for all names in all T-boxes, then the second language can express everything the first language does. In this sense the definition is very general, but in other aspects it is perhaps too specific. Why should we relate names? It seems more natural to translate definitions in T_1 into new definitions in T_2 (compare the definition for equational theories), and it would be even better if we could provide a translation for the full language $\mathcal{L}_1$, independently of the particular T-box we are considering. A further advantage of this *global* approach to translating the full language instead of a given T-box, is that we would be able to measure and compare how much is *needed* in the way of computational resources in the translation. In other words, a global translation would help us to estimate the complexity of interpreting $\mathcal{L}_1$ into $\mathcal{L}_2$. A further point is this: can T_2 be *any* T-box in $\mathcal{L}_2$? If we have axioms in T_2, non trivial encoding can be done at this level which would be hidden by the existential quantification over T_2 in the definition. Finally, observe that Baader's definition does not give an explicit description of what a given description logic can express; it only explains the *relative* expressive power of logics.

§3.3. Kurtonina and de Rijke [13, 14]. The core of this competing definition of expressive power is the notion of *simulation*. The best way to understand simulations is through an example. Let us take $\mathcal{ALC}$ and consider only concepts for a moment.

Definition 2 Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be two interpretations. A non-empty relation $Z \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ is called an *$\mathcal{ALC}$-simulation* if it satisfies the following clauses.

1. If $d_1 Z d_2$, then, for every (atomic) concept name C, $d_1 \in C^{\mathcal{I}}$ iff $d_2 \in C^{\mathcal{J}}$.
2. For every (atomic) role name R, if $d_1 Z d_2$ and $R^{\mathcal{I}} d_1 e_1$, then there exists $e_2 \in \Delta^{\mathcal{J}}$ such that $R^{\mathcal{J}} d_2 e_2$ and $e_1 Z e_2$.
3. For every (atomic) role name R, if $d_1 Z d_2$ and $R^{\mathcal{J}} d_2 e_2$, then there exists $e_1 \in \Delta^{\mathcal{I}}$ such that $R^{\mathcal{J}} d_2 e_2$ and $e_1 Z e_2$.

It is clear that the notion of $\mathcal{ALC}$-simulation is tailored to a specific language: item 1. above takes care of atomic concepts, items 2. and 3. of existential quantifiers, while conjunction and negation are covered by the one-to-one relation (points are related with points) and the "symmetry" of the definition.

Anybody with a basic knowledge of modal logic will recognize the notion of bisimulation in the definition above. This is witness of the close connection between description logics and modal logics as studied in, for example, [17, 9, 6]. Another source for bisimulations is the theory of concurrency; see, for instance, [15].

A first-order formula with a free variable $\varphi(x)$ (i.e., a concept) is said to be *preserved under* a given notion of $\mathcal{L}$-simulation if for all interpretations $\mathcal{I}$, $\mathcal{J}$, $d_1 \in \Delta^{\mathcal{I}}$, and $d_2 \in \Delta^{\mathcal{I}}$, we have that $d_1 \in \varphi^{\mathcal{I}}$ implies $d_2 \in \varphi^{\mathcal{J}}$ whenever there is an $\mathcal{L}$-simulation between $\mathcal{I}$ and $\mathcal{J}$ which connects d_1 and d_2.

The important result concerning simulations is the following.

Theorem 3 *Fix some similarity type for description logics. Let $\varphi(x)$ be a first-order formula over this similarity type. Then $\varphi(x)$ is equivalent to an $\mathcal{ALC}$-concept iff it is preserved under $\mathcal{ALC}$-simulations.*

In this way $\mathcal{ALC}$-concepts are completely characterized by $\mathcal{ALC}$-simulations.

In [13, 14] the authors provide, given a set of concept-forming operations from Table 1 that are admissible in a description logic $\mathcal{L}$, the corresponding relation of $\mathcal{L}$-simulation between $\mathcal{L}$-interpretations, as well as suitable analogs of Theorem 3. Hence, an explicit and exact characterization of the concepts expressible in a given description logic $\mathcal{L}$ is given.

Notice that Theorem 3 gives an explicit characterization of "what can be expressed in a description logic $\mathcal{L}_1$." So far there is no mention of a second language $\mathcal{L}_2$, while Baader's definition is a recipe for determining *relative* expressive power. Nevertheless, once a characterization for $\mathcal{L}_1$ in the style of Theorem 3 is obtained, we can prove that $\mathcal{L}_2$ can express all that $\mathcal{L}_1$ says by using this characterization.

Definition 4 Let $\mathcal{L}_1$, $\mathcal{L}_2$ be description logics. Then $\mathcal{L}_1$ can be *expressed by $\mathcal{L}_2$ through simulations* ($\mathcal{L}_1 \leq_s \mathcal{L}_2$) if all concepts in $\mathcal{L}_1$ are preserved under $\mathcal{L}_2$-simulations. We say that $\mathcal{L}_1$ and $\mathcal{L}_2$ *have the same expressive power under simulations* ($\mathcal{L}_1 =_s \mathcal{L}_2$) if $\mathcal{L}_1 \leq_s \mathcal{L}_2$ and $\mathcal{L}_2 \leq_s \mathcal{L}_1$.

Even though this definition has been shown to produce intuitively correct separation and classification results for a large class of description logics [13, 14], it has some serious limitations: only concepts are considered — but they are characterized completely — and there is no room for translations, and hence for the comparison of concepts defined over different vocabularies.

§3.4. A Comparison. A number of differences between the two approaches is immediately obvious. First, they compare the expressive power of description logics with respect to different things: full T-boxes (Baader) and concepts (Kurtonina and de Rijke). Second, Baader compares description logics based on different sets of (atomic) roles and concepts, and role and concept constructions can be mimicked in a language from which they are absent by using additional roles or concepts in a translation: Baader's definition stands *between* two logics. Kurtonina and de Rijke, on the other hand, aim at explicit characterizations explaining the expressive power of a given description logic in terms of one and only set of (atomic) roles and concepts: simulations are *inside* a given logic. And third, as Kurtonina and de Rijke do not consider translations, it is easier to establish separation results in their framework.

Even though the two approaches talk about different things, it is possible to say something — formally — about their connections. In particular, one can show that Kurtonina and de Rijke's approach gives a finer notion of expressive equivalence of description logics than Baader's.

Proposition 5 *Let $\mathcal{L}_1$, $\mathcal{L}_2$ be two description logics without axioms in T-boxes, then $\mathcal{L}_1 \leq_s \mathcal{L}_2$ implies $\mathcal{L}_1 \leq_e \mathcal{L}_2$.*

Proof. Suppose $\mathcal{L}_1 \leq_s \mathcal{L}_2$. Let $T_1 \in \mathcal{L}_1$ and $Q \doteq \varphi \in T_1$. As $\mathcal{L}_1 \leq_s \mathcal{L}_2$ there is $\varphi' \in \mathcal{L}_2$ such that $\models \varphi \leftrightarrow \varphi'$ (every formula of $\mathcal{L}_1$ has an equivalent in the language of $\mathcal{L}_2$).

Define $T_2 = \{Q \doteq \varphi' \mid Q \doteq \varphi \in T_1\}$, renaming Q if necessary so that it does not appear anywhere else. That is, we replace each formula of $\mathcal{L}_1$ by its equivalent in $\mathcal{L}_2$. Then $\mathrm{Int}(T_1) = \mathrm{Int}(T_2)$ and we can take f to be the identity function. Trivially, $\mathrm{Int}(T_1) =_f \mathrm{Int}(T_2)$. $\dashv$

The converse to Proposition 5 does not hold, even when we restrict ourselves to concepts and T-boxes without axioms: as we said in the introduction $\mathcal{TF} =_e \mathcal{NTF}$ while $\mathcal{TF} <_s \mathcal{NTF}$. It is an interesting exercise to see where an attempted proof of $\mathcal{NTF} \leq_e \mathcal{TF} \Rightarrow \mathcal{NTF} \leq_s \mathcal{TF}$ fails. The fact that translation functions are used in the definition of $\leq_e$ simply destroys any hope of proving that if $\mathcal{I}$ is an interpretation of a negated concept $\varphi(x)$ and $\mathcal{J}$ is $\mathcal{TF}$-similar to $\mathcal{I}$ then $\mathcal{J}$ also satisfies $\varphi(x)$.

Nevertheless, simulations can play an important role in Baader's approach. Using simulations succinct proofs via embeddings can be given. Consider the following example [1, Theorem 4.6].

Proposition 6 $\mathcal{FL}_0 <_e \mathcal{FL}^-$.

Proof. As $\mathcal{FL}^-$ is an extension of $\mathcal{FL}_0$ we have $\mathcal{FL}_0 \leq_e \mathcal{FL}^-$. To prove that $\mathcal{FL}_0 \neq_e \mathcal{FL}^-$, reason as follows. Let $T_1 = \{P \doteq \exists R.\top\} \in \mathcal{FL}^-$. Suppose $T_2 \in \mathcal{FL}_0$ expresses T_1 through embeddings. Let $\mathcal{I} \in \text{Int}(T_1)$ be defined as $\mathcal{I} = (\{a, b\}, \cdot^{\mathcal{I}})$, with $P^{\mathcal{I}} = \{a\}$, $R^{\mathcal{I}} = \{(a, b)\}$. Let $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be the corresponding model in $\text{Int}(T_2)$ such that $\mathcal{I} \subseteq_f \mathcal{J}$. Next, define $\mathcal{J}'$ as $(\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}'})$, with $f(R)^{\mathcal{J}'} = \emptyset$, $Q^{\mathcal{J}'} = Q^{\mathcal{J}}$ for $Q \neq f(R)$ and does not occur on the left-hand side of a definition in T_2, and $Q^{\mathcal{J}'} = \varphi^{\mathcal{J}'}$ if $Q \doteq \varphi \in T_2$. Notice that the definition is not circular because Q does not occur in φ and $\varphi^{\mathcal{J}'}$ can be defined without knowing $Q^{\mathcal{J}'}$.[2]

By definition, $\mathcal{J}' \in \text{Int}(T_2)$. Also, it is easy to prove that $Z = \{(\{m\}, m) \mid m \in \Delta^{\mathcal{J}}\}$ is an $\mathcal{FL}_0$-simulation from $\mathcal{J}$ to $\mathcal{J}'$. Notice that $a \in P^{\mathcal{I}}$ implies $a \in f(P)^{\mathcal{J}}$ and by preservation under simulations, $a \in f(P)^{\mathcal{J}'}$. Now let $\mathcal{I}'$ be such that $\mathcal{I}' \subseteq_f \mathcal{J}'$. By definition $R^{\mathcal{I}'} = f(R)^{\mathcal{J}'} = \emptyset$, but $P^{\mathcal{I}'} = f(P)^{\mathcal{J}'}$ and hence $a \in P^{\mathcal{I}'}$. From which we get $\mathcal{I}' \notin \text{Int}(T_1)$ — a contradiction. $\dashv$

To conclude this section, let us identify what we take to be the main advantages of both approaches. The main characteristic of Baader's approach is its generality in two aspects. First, full T-boxes are considered against only concepts in Kurtonina and de Rijke's definition. Second, translations are allowed and hence the natural idea of coding is captured: if we cannot express a notion *directly*, perhaps there is another way of *explaining* it in terms of others. On the other hand, Kurtonina and de Rijke's approach completely and explicitly characterizes which concepts can be defined in a given description logic. Also, it has a high explanatory power since the characterizations tell us exactly which changes in an interpretation are "noticed" by a given description logic and which are not. Finally, the notion of simulation is not ad-hoc. It has been broadly used in computer science and more recently also in (modal and classical) logic. Some strong and interesting results are known, and some of these may be useful for description logics.

4 The Way Forward

This section is full of questions, and has only very few answers. The main question is: Can we combine the two notions of expressive power to obtain "the best of the two worlds?" To be more precise, in our synthesis we would like to have more general notions of simulation, ones that will allow us to characterize not just the concepts expressible in a given description logic, but also its roles and indeed its T-boxes; in addition, we want to incorporate the notion of a translation into the picture. Of course, this is an ambitious project, and it breaks down in a few natural approximations, which we list below.

1. Characterizing Roles. The first aim is to develop simulation techniques to characterize the roles expressible in a given description logic. We have been successful here, and have extended techniques presented in [13, 14] to account for roles. Basically, simulations should not only relate (sets of) points in one model to (sets of) points in the other, but also pairs of these. The idea is that as connecting (sets of) points preserves concepts, connecting pairs of (sets of) points preserves roles.

We have characterized (first-order definable) operations on roles (as conjunction, disjunction) using these extended simulation techniques. Characterization results for all interesting subsets of the concept- and role-forming operations presented in Table 1 are now available. To substantiate this claim, we present a sample: we describe a two-sorted simulation for $\mathcal{ALCR}$ that allows us to characterize both the concepts and roles that are definable in $\mathcal{ALCR}$.

Definition 7 Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be two interpretations. A *two-sorted* $\mathcal{ALCR}$-*simulation* is a triple $Z = (Z_0, Z_1, Z_2)$ that satisfies the following clauses:

1. (a) $Z_0 \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$.
   (b) $Z_1 \subseteq \mathcal{P}(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \times (\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}})$.
   (c) $Z_2 \subseteq \mathcal{P}(\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}) \times (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$.
2. (a) If $d_1 Z_0 d_2$, then, for every (atomic) concept name C, $d_1 \in C^{\mathcal{I}}$ iff $d_2 \in C^{\mathcal{J}}$.
   (b) If $X_1 Z_1 (d_2, e_2)$, then, for every (atomic) role name R, $X_1 \subseteq R^{\mathcal{I}}$ implies $(d_2, e_2) \in R^{\mathcal{J}}$.
   (c) If $X_2 Z_2 (d_1, e_1)$, then, for every (atomic) role name R, $X_2 \subseteq R^{\mathcal{J}}$ implies $(d_1, e_1) \in R^{\mathcal{I}}$.

[2] This definition fixes an error in Baader's proof where it is not verified that $(\Delta^{\mathcal{J}'}, \cdot^{\mathcal{J}'})$ is still a model of T_2. The same error appears in [1, Theorem 4.9].

3. (a) If $d_1 Z_0 d_2$ and for some e_1 and role R, $(d_1, e_1) \in R^{\mathcal{I}}$, then there exists $e_2 \in \Delta^{\mathcal{J}}$ with $\{(d_1, e_1)\} Z_1 (d_2, e_2)$.
   (b) If $d_1 Z_0 d_2$ and for some e_2 and role R, $(d_2, e_2) \in R^{\mathcal{J}}$, then there exists $e_1 \in \Delta^{\mathcal{I}}$ with $\{(d_2, e_2)\} Z_2 (d_1, e_1)$.
4. (a) If $X_1 Z_1 (d_2, e_2)$, then for all $(d_1, e_1) \in X_1$ we have both $d_1 Z_0 d_2$ and $e_1 Z_0 e_2$.
   (b) If $X_2 Z_2 (d_1, e_1)$, then for all $(d_2, e_2) \in X_2$ we have both $d_1 Z_0 d_2$ and $e_1 Z_0 e_2$.

We say that a unary first-order formula $\varphi(x)$ is *preserved* under two-sorted $\mathcal{ALCR}$-simulations if for all interpretations $\mathcal{I}$ and $\mathcal{J}$, all $d_1 \in \Delta^{\mathcal{I}}$ and $d_2 \in \Delta^{\mathcal{J}}$, we have that $d_1 \in \varphi^{\mathcal{I}}$ iff $d_2 \in \varphi^{\mathcal{J}}$ whenever there is a two-sorted $\mathcal{ALCR}$-simulation $Z = (Z_0, Z_1, Z_2)$ between $\mathcal{I}$ and $\mathcal{J}$ such that $d_1 Z_0 d_2$.

Also, a binary first-order formula $\varphi(x, y)$ is *preserved* under two-sorted $\mathcal{ALCR}$-simulations if for all interpretations $\mathcal{I}$ and $\mathcal{J}$, all $X_1 \subseteq (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$ and $(d_2, e_2) \in (\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}})$, we have that $X_1 \subseteq \varphi^{\mathcal{I}}$ implies $(d_2, e_2) \in \varphi^{\mathcal{J}}$ whenever there is a two-sorted $\mathcal{ALCR}$-simulation $Z = (Z_0, Z_1, Z_2)$ such that $X_1 Z_1 (d_2, e_2)$.

Item 1 of Definition 7 reflects the fact that $\mathcal{ALCR}$-simulations are two-sorted: they have to preserve concepts (item 1(a)) and roles (items 1(b) and 1(c)). Linking objects to objects will provide the right setting for preserving concepts (as in Definition 2), while we the only operation on roles that we want to preserve is role conjunction — for that reason we link sets of pairs of objects to objects (in items 1(b) and 1(c)); see [14] for further intuitions. Item 2 makes sure that atomic concepts and roles preserved. Item 3 encodes the familiar back and forth character of $\mathcal{ALC}$-simulations, and item 4 coordinates the simulations between the two sorts.

Theorem 8 *Fix some similarity type for description logics.*

1. *Then, a unary first-order formula over this similarity type is preserved under two-sorted $\mathcal{ALCR}$-simulations iff it is equivalent to an $\mathcal{ALCR}$-concept.*
2. *Also, a binary first-order formula over this similarity type is preserved under two-sorted $\mathcal{ALCR}$-simulations iff it is equivalent to an $\mathcal{ALCR}$-role.*

Proof. We only prove the 'easy' right-to-left directions of the theorem. For a proof of the 'hard' directions, a combination of the proofs for Theorems 4.2 and 4.18 in [14] works.

To prove the easy directions, we have to show that $\mathcal{ALCR}$-concepts and -roles are preserved under two-sorted $\mathcal{ALCR}$-simulations. First, it is easily seen that all $\mathcal{ALCR}$-roles are so preserved. Let $\mathcal{I}$, $\mathcal{J}$ be two interpretations with $X_1 \subseteq (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$ and $(d_2, e_2) \in (\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}})$. If we have $X_1 \subseteq (R_1^{\mathcal{I}} \cap \cdots \cap R_n^{\mathcal{I}})$ and, for some $\mathcal{ALCR}$-simulation Z, $X_1 Z_1 (d_2, e_2)$, then, by item 2(b) of Definition 7 and a simple induction on n, we have that $(d_2, e_2) \in (R_1^{\mathcal{J}} \cap \cdots \cap R_n^{\mathcal{J}})$, as required.

Next, let us show that $\mathcal{ALCR}$-concepts are indeed preserved. The proof is by induction on concepts, and we will only consider the case of existential quantification $\exists R.C$ (where R may be a complex role). Assume that $d_1 Z_0 d_2$ and $d_1 \in (\exists R.C)^{\mathcal{I}}$; we need to show that $d_2 \in (\exists R.C)^{\mathcal{J}}$. Clearly, there exists $e_1 \in \Delta_{\mathcal{I}}$ with $(d_1, e_1) \in R^{\mathcal{I}}$ and $e_1 \in C^{\mathcal{I}}$. By item 3(a) there exists $e_2 \in \Delta^{\mathcal{J}}$ with $\{(d_1, e_1)\} Z_1 (d_2, e_2)$. By our earlier observations $(d_2, e_2) \in R^{\mathcal{J}}$, and using item 4(a) we find that $e_1 Z_0 e_2$, and hence $e_2 \in C^{\mathcal{J}}$ by the induction hypothesis. $\dashv$

The general strategy used above is worth identifying. What we have done is to take $\mathcal{ALC}$-simulations as defined Definition 2, and to add a second layer characterizing role conjunction on top of it. The same strategy may be used for other description logics $\mathcal{L}$ for which characterizations of the $\mathcal{L}$-definable concepts has been given in terms of $\mathcal{L}$-simulations in [14].

2. Incorporating Translations. With the ideas explored in Step 1 above, we now know how to characterize the concepts and roles that are expressible in a given description logic $\mathcal{L}$, and, hence, we have the tools to compare $\mathcal{L}$ to other logics over the same similarity type with respect the concepts and roles that can be defined. Such comparisons become much harder if *different* similarity types are involved. A notion of simulation is strongly tied to the language: in all the cases covered up to now, we talked about $\mathcal{L}$-simulations for a given $\mathcal{L}$. When trying to incorporate translations so as to account for differences in similarity types, we have to consider two languages. Also, in dealing with such combinations of translations and characterizations via simulations, we want to have control over the complexity of the translation (in terms of the size of the translated formula, the size of the target T-box and new predicates needed).

What is the appropriate modification of the earlier framework? To produce in some way a new notion of simulation for the two logics together, perhaps combining the notions of simulations for $\mathcal{L}_1$ and $\mathcal{L}_2$? To build into the notion of $\mathcal{L}_1$-simulation the translation function used? We do not think that this is the proper solution.

Instead, our best conjecture is that simulations have to be used in a different way than how they are now applied to define the notion of expressive power. In other words, we are after an alternative to Definition 4. The fact that $=_s$ gives rise to a finer notion of equivalence of description logics than $=_b$, gives us the hint that a more liberal use of simulation in Definition 4 is needed. Incorporating translations at this level will keep things in their proper place: simulations are tied to a language and hence are defined for a specific description logic, completely characterizing its definitional power. And when a comparison between description logics based on different vocabularies is needed, the corresponding notions of simulations plus appropriate translations will be used.

After completing this step, we would indeed have a tool which combines the characteristics of the approaches of Baader and Kurtonina and de Rijke for the case of acyclic and unfolded T-boxes, where the key notions are the right-hand side of definitions.

3. Full T-Boxes. The final step, considering full T-boxes, becomes interesting when folded T-boxes are allowed. We should still restrict ourselves to T-boxes without cyclic definition if we want to stay inside first-order logic[3]. In folded definitions, left-hand sides of equations become important as they can appear also on the right-hand side. As we said in Section 2, unfolding definitions might cause an exponential grow in the size of the T-box, something which should be taken into account given that efficiency is a relevant issue for description logics.

To characterize full definitions of the form $Q \doteq \varphi$, we probably need a two-level notion of simulation: on the bottom level, notions of simulation as described in this article will take care of the preservation of φ while the top level will preserve the equivalence of φ with Q.

Finally, full T-boxes also permit axioms that can be used to enforce properties of interpretations. This is probably the hardest point to cope with if we insist to do it through simulations. Preserving general properties of interpretations via simulations is known to be a difficult problem. For example, an axiom may force a given role to be functional, but then in any similar model the role should also be functional, and this global property is difficult to enforce. A solution here might be to consider *relativized* simulations where we try to obtain simulation based characterizations relative to external restrictions on the classes of interpretations that should be considered.

5 Conclusions

A proper notion of the expressive power of description logics is essential for the comparison and understanding of the many formalisms that are continually being proposed in the field. Awkwardly enough, until recently this notion has largely been neglected in the literature.

Two — quite different but related — notions of expressive power have been given for description logics: Baader's definition based on a notion of embeddings and Kurtonina and de Rijke's based on simulations. The different approaches induce different classifications of description logics with respect to their expressive power. In this paper we have compared the two notions, pointing to differences and similarities, and we have also discussed ways in which the two notions are connected.

We have also mapped out a research agenda in which we aim to combine the best features of both approaches. We have indicated three steps in this direction: first, generalize simulations to roles, then introduce translations, and, finally, generalize simulations so as to characterize full T-boxes. So far, we have only given the first of these three steps; the remaining two form part of our ongoing research.

[3] To provide a semantics for cyclic definitions a notion of fix point is needed. In this paper we will not discuss the notion of simulation that is appropriate for such cases, even though relevant results in the area of modal logic do exist; see [12]

Acknowledgments. Carlos Areces is under grant Nr. ARG0100049 of the British Council. Maarten de Rijke is supported by the Spinoza project 'Logic in Action' at ILLC at the University of Amsterdam. Thanks are also due to an anonymous referee whose comments helped improve the paper.

References

[1] F. Baader. A formal definition for the expressive power of terminological knowledge representation languages. *Journal of Logic and Computation*, 6:33–54, 1996.

[2] J. Barwise and S. Feferman, editors. *Model-Theoretic Logics*. Springer-Verlag, 1985.

[3] A. Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82:353–367, 1996.

[4] R. Brachman and H. Levesque. The tractability of subsumption in frame-based description languages. In *AAAI-84*, pages 34–37, 1984.

[5] R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[6] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proc. 12th National Conference on Artificial Intelligence (AAAI'94)*, pages 205–212, 1994.

[7] H. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Undergraduate Texts in Mathematics. Springer-Verlag, 1984.

[8] D. Gabbay. Expressive functional completeness in tense logic. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 91–117. Reidel, Dordrecht, 1981.

[9] W. van der Hoek and M. de Rijke. Counting objects. *Journal of Logic and Computation*, 5(3):325–345, 1995.

[10] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1-3):86–104, 1986.

[11] N. Immerman and D. Kozen. Definability with bounded number of bound variables. *Information and Computation*, 83(2):121–139, 1989.

[12] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus with respect to monadic second order logic. In *Proc. Concur '96*, volume 1119 of *Lecture Notes in Computer Science*, pages 263–277. Springer, Berlin, 1996.

[13] N. Kurtonina and M. de Rijke. Classifying description logics. In *Proceedings DL'97*, 1997.

[14] N. Kurtonina and M. de Rijke. Expressiveness of first-order description logics. Technical report, Dept. of Computer Science, University of Warwick, 1997.

[15] R. Milner. *Communication and Concurrency*. Prentice-Hall International Series in Computer Science. Prentice-Hall, 1989.

[16] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[17] K. Schild. A correspondence theory for terminological logics. In *Proceedings of the 12th IJCAI*, pages 466–471, 1991.

[18] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[19] W. Taylor. Equational logic. *Houston Journal of Mathematics*, 5:1–51, 1979.