Visualizing $\mathcal{ALC}$ Using Concept Diagrams

Gem Stapleton[1], Aidan Delaney[1,2], Michael Compton[3] and Peter Chapman[4]

[1] Centre for Secure, Intelligent and Usable Systems,
University of Brighton, UK
g.e.stapleton@brighton.ac.uk
[2] University of the South Pacific, Fiji
aidan@ontologyengineering.org
[3] unaffiliated
[4] Edinburgh Napier University, UK
p.chapman@napier.ac.uk

Abstract. This paper addresses the problem of how to visualize axioms from $\mathcal{ALC}$ using concept diagrams. We establish that 66.4% of OWL axioms defined for ontologies in the Manchester corpus are formulated over $\mathcal{ALC}$, demonstrating the significance of considering how to visualize this relatively simple description logic. Our solution to the problem involves providing a general translation from $\mathcal{ALC}$ axioms into concept diagrams, which is sufficient to establish that all of $\mathcal{ALC}$ can be expressed. However, the translation itself is not designed to give optimally readable diagrams, which is particularly challenging to achieve in the general case. As such, we also improve the translations for a selected category of $\mathcal{ALC}$ axioms, to illustrate that more effective diagrams can be produced.

1 Introduction

Ontology engineering requires a significant skill set as it involves domain modelling and defining axioms using a formal notation, alongside refining and debugging ontologies until the model is seen as accurate and fit-for-purpose. This engineering task can involve many stakeholders, including domain experts who need not be fluent in or, even, familiar with formal notations such as DL or OWL which are typically used by ontology engineers. Communication problems arise as a result. Thus, the use of symbolic notations is a particular obstacle, with this mode of communication potentially leading to inaccurate ontologies being developed or increased time and effort. This is a shortfall because accurate communication of knowledge is necessary for the production of ontologies.

Visualization techniques have been recognized as possible approaches to addressing accessibility problems associated with symbolic notations. Of the various ontology visualization techniques, the majority exploit node-link diagrams (graphs), with OWLViz [13], OntoGraf [2] and CMap [12] being notable examples, but often they are not formalized. These graph-based visualizations exploit the same syntactic element (arrows) to represent both class subsumption and property restrictions. Consequently, the saliency of these two different types

Concept Subsumption Axioms

$SensingDevice \sqsubseteq Device$ $\quad Device \sqsubseteq System$ $\quad System \sqsubseteq PhysicalObject$
$SensingDevice \sqsubseteq Sensor$ $\quad Sensor \sqsubseteq PhysicalObject$

$Sensing \sqsubseteq Process$ $\quad Process \sqsubseteq Method$ $\quad Method \sqsubseteq Description$ $\quad Description \sqsubseteq SocialObject$
$SensorOutput \sqsubseteq InformationObject$ $\quad InformationObject \sqsubseteq SocialObject$
$Observation \sqsubseteq Situation$ $\quad Situation \sqsubseteq SocialObject$

Concept Disjointness Axioms

$Description \sqcap Situation \sqsubseteq \perp$ $\quad Description \sqcap InformationObject \sqsubseteq \perp$ $\quad InformationObject \sqcap Situation \sqsubseteq \perp$
$PhysicalObject \sqcap SocialObject \sqsubseteq \perp$ $\quad PhysicalObject \sqcap Property \sqsubseteq \perp$ $\quad SocialObject \sqcap Property \sqsubseteq \perp$

Relationships Between Concepts (All Values From Restrictions)

$Sensor \sqsubseteq \forall hasMeasurementCapability.Property$ $\quad Sensor \sqsubseteq \forall observes.Property$
$Observation \sqsubseteq \forall observedproperty.Property$ $\quad Observation \sqsubseteq \forall observationResult.SensorOutput$
$Observation \sqsubseteq \forall sensingMethodUsed.Sensing$ $\quad Observation \sqsubseteq \forall observedBy.Sensor$

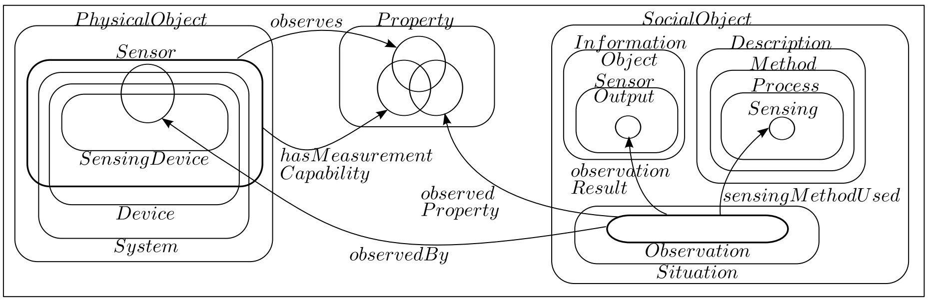


Fig. 1. Description logic axioms converted to a concept diagram.

of information is significantly reduced. Similarity theory tells us saliency is an important factor and, in particular, that different syntactic devices should represent different types of information [8]. This is because when visually searching for particular types of information, increasing degrees of similarity between the target syntax (which represents the required information) and distracter syntax (which represents other information) leads to a corresponding increase in the time taken to perform tasks. Another visualization technique is an adaptation of existential graphs, which represent individuals, conjunction and negation using line segments, juxtaposition and closed curves respectively [7]. The resulting notation is essentially a stylized form of first-order logic that uses only $\exists$, $\wedge$ and $\neg$ to make statements and we are of the opinion that usability suffers as a result.

Concept diagrams were introduced for ontology engineering [14] and aid information saliency by avoiding the use of identical (or, even, similar) syntactic types for different informational types: concepts (sometimes called classes) are represented by closed curves and roles (sometimes called properties) by arrows. Figure 1 shows a set of DL axioms, all from $\mathcal{ALC}$, visualized using a single concept diagram; these axioms correspond to a fragment of the SNN ontology [6]. The concept subsumption, concept disjointness and AllValuesFrom-style axioms are represented by curve inclusion, curve disjointness and arrows respectively.

$\mathcal{ALC}$ is an important DL: $\mathcal{ALC}$ axioms form 64.4% of the Manchester corpus [1], which contains over 4500 ontologies comprising nearly 3 million OWL axioms. Whilst the example just given shows how to visualize 25 DL axioms using one concept diagram, this paper demonstrates how to translate single DL axioms into diagrams. Our first contribution is to establish concept diagrams equivalent to $\mathcal{ALC}$ concepts. We then go on to establish how to visualize ABox and TBox axioms. Thus, concept diagrams can be used to visualize a significant proportion of the axioms from a large number of ontologies. We also show how to simplify the resulting diagrams into arguably more readable forms.

2 The Description Logic $\mathcal{ALC}$

Readers familiar with the formalization of $\mathcal{ALC}$ may choose to omit this section. In $\mathcal{ALC}$, as with all description logics, axioms are written over a vocabulary comprising a set of individuals, a set of atomic concepts and a set of roles, drawn from the pairwise disjoint sets $\mathcal{O}$, $\mathcal{C}$, and $\mathcal{R}$, respectively. There are two special atomic concepts in $\mathcal{C}$: $\top$ and $\bot$. Individuals, concepts and roles represent elements, sets and binary relations respectively; $\top$ represent Thing (the set containing everything) and $\bot$ represents Nothing (the empty set). The vocabulary is used to form axioms in $\mathcal{ALC}$. Firstly, we define concepts, which are built using atomic concepts and roles along with logical operators and quantifiers.

Definition 1. *The following are **concepts** in $\mathcal{ALC}$:*

1. *Any atomic concept is a concept.*
2. *If C and D are a concepts and R is a role then the following are also (complex) concepts: $(C \sqcap D)$, $(C \sqcup D)$, $\neg C$, $\exists R.C$, and $\forall R.C$.*

In more expressive description logics, other types of concepts can be formed, such as $= nR.C$, which is taken to be the set of things that are related to exactly n things in the 'set' C under the 'relation' R. Moreover, roles can be made more complex, too, such as by forming their composition, $R_1 \circ R_2$, and by taking inverses, R^-. As we are focusing on visualizing axioms drawn from $\mathcal{ALC}$, so these more complex constructions are not permitted.

Definition 2. *Given individuals a and b, concepts C and D, and role R the following are **axioms** in $\mathcal{ALC}$: $C(a)$, $R(a,b)$, and $C \sqsubseteq D$. Axioms that involve individuals are **ABox axioms** whereas those which do not are **TBox axioms**.*

We note here that $C \equiv D$ is also sometimes considered an axiom. For the purposes of this paper, we consider $C \equiv D$ to be a pair of subsumption axioms: $C \sqsubseteq D$ and $D \sqsubseteq C$.

Our attention now turns to semantics. Individuals are interpreted as elements, concepts as sets and roles as binary relations.

Definition 3. *An **interpretation** is a pair, $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where*

1. *$\triangle^{\mathcal{I}}$ is a non-empty set, and*
2. *the function $\cdot^{\mathcal{I}}$ maps*
   (a) each individual, a, in $\mathcal{O}$ to an element of $\triangle^{\mathcal{I}}$, that is $a^{\mathcal{I}} \in \triangle^{\mathcal{I}}$,

(b) *each concept, C, in $\mathcal{C}$ to a subset of $\triangle^{\mathcal{I}}$, that is $C^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$, such that $\top^{\mathcal{I}} = \triangle^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$, and*
(c) *each role, R, in $\mathcal{R}$ to a binary relation on $\triangle^{\mathcal{I}}$, that is $R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$.*

The function $\cdot^{\mathcal{I}}$ can then be extended to interpret all concepts as follows:

1. $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
2. $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$,
3. $\neg C^{\mathcal{I}} = \triangle^{\mathcal{I}} \backslash C^{\mathcal{I}}$,
4. $\exists R.C^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} : \exists y\,(y \in C^{\mathcal{I}} \wedge (x,y) \in R^{\mathcal{I}})\}$, *and*
5. $\forall R.C^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} : \forall y\,((y \in \triangle^{\mathcal{I}} \wedge (x,y) \in R^{\mathcal{I}}) \Rightarrow y \in C^{\mathcal{I}})\}$.

Definition 4. *For each axiom, $\mathcal{A}$, an interpretation, $\mathcal{I}$,* ***models*** *$\mathcal{A}$ under the following conditions:*

1. *If $\mathcal{A} = C(a)$ for some concept C and individual a, $\mathcal{I}$ models $C(a)$ whenever $a^{\mathcal{I}} \in C^{\mathcal{I}}$.*
2. *If $\mathcal{A} = C \sqsubseteq D$ for some concepts C and D then $\mathcal{I}$ models $C \sqsubseteq D$ whenever $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.*
3. *If $\mathcal{A} = R(a,b)$ for some role R and individuals a and b then $\mathcal{I}$ models $R(a,b)$ whenever $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.*

3 Concept Diagrams

Here we present the formalization of a first-order fragment of the concept diagram logic that is able to express all of $\mathcal{ALC}$. We adapt the formalization given in [23], removing unnecessary second-order, and some first-order, syntax. Firstly, we note that concept diagrams allow the use of inverse roles. So, for every role, R, in $\mathcal{R}$, R^- is a role and we define $R^{-\mathcal{I}} = \{(y,x) : (x,y) \in R^{\mathcal{I}}\}$. Whilst inverse roles are not permitted in $\mathcal{ALC}$, we make use of them in our translation.

An example of a concept diagram is given in figure 2. It comprises two *unitary* diagrams, β_1 and β_2; unitary diagrams are extended with additional syntax and are called *class and object property diagrams* in [18]. Each of β_1 and β_2 is enclosed by a *boundary rectangle* which represents the universal set, $\triangle^{\mathcal{I}}$. Each of β_1 and β_2 contain a single *spider*; in β_1 the spider is the graph with two nodes joined by an edge whereas in β_2 the spider comprises just a single node. The first

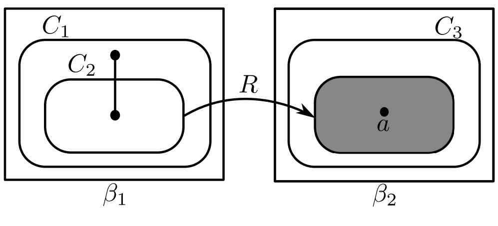


Fig. 2. A concept diagram.

spider represents the existence of an anonymous individual whereas the second spider represent the individual a. The labelled (resp. unlabelled) curves represent atomic (resp. anonymous) concepts and shading is used to place upper bounds on set cardinality: in shaded regions, all elements must be represented by spiders. So, in β_2, the only element in the anonymous set is a. Within each unitary diagram, the spatial relationships between the curves and the spiders convey meaning. In β_1, for instance, we can see that $C_2 \sqsubseteq C_1$, through curve inclusion, and the anonymous individual represented by the spider is in C_1. The shading and spider labelled a in β_2 tell us that the only element in the anonymous set is the individual a.

The arrow joining the two unitary diagrams, thus forming a *concept diagram*, asserts that the elements in C_2 (the arrow's source) are, between them, related to *all and only* the elements in the anonymous set represented by the arrow's target which, in turn, is subsumed by C_3. More informally, the arrow tells us that elements in C_2 can only be related to elements in C_3; in $\mathcal{ALC}$, the arrow expresses $C_2 \sqsubseteq \forall R.C_3$. In general, arrows can be sourced and targeted on boundary rectangles, curves and spiders. In addition, arrows can also be dashed to express partial information. In figure 2, if the arrow was dashed then the diagram would instead assert that the elements in C_2 are, between them, related to at least all of the elements in the anonymous set represented by the arrow's target.

Our formalization of concept diagrams is at an abstract syntax level. Spiders and closed curves are chosen from countably infinite sets $\mathcal{S}$ and $\mathcal{K}$ respectively; note that these are not closed curves in the mathematical sense. Lastly, arrows represent roles – or, rather, role restriction – are of the form $(s, R, t, \circ)$. Here, s is the arrow's source, R is the arrow's label which is a role or inverse role, t is the target and $\circ$ is either $\rightarrow$ or $\dashrightarrow$. As the boundary rectangle in unitary diagrams can be the source or target of an arrow, but is not in $\mathcal{S}$ or $\mathcal{C}$, it will be denoted by $\square$, formally written as $(\square, \beta)$ to identify the diagram, β, in question. Thus, an arrow of the form $((\square, \beta), R, t, \rightarrow)$ indicates that a solid arrow is sourced on the diagram β's boundary rectangle, labelled R with target t.

Definition 5. *A* ***unitary diagram****,* $\beta = (\Sigma, K, \lambda, Z, Z^*, \eta, A)$ *has components that are defined as follows.*

1. $\Sigma = \Sigma(\beta) \subset \mathcal{S}$ *is a finite set of spiders.*
2. $K = K(\beta) \subset \mathcal{K}$ *is a finite set of curves.*
3. $\lambda = \lambda_\beta = \lambda_\Sigma \cup \lambda_K$ *is a partial function such that*
   (a) $\lambda_\Sigma \colon \Sigma \rightarrow \mathcal{O}$ *is a partial function that labels spiders with elements* $\mathcal{O}$ *and*
   (b) $\lambda_K \colon K \rightarrow \mathcal{C}$ *is a partial function that labels curves with elements of* $\mathcal{C}$*.*
4. $Z = Z(\beta)$ *is a set of* ***zones*** *such that* $Z \subseteq \{(in, K \backslash in) : in \subseteq K\}$*.*
5. $Z^* = Z^*(\beta) \subseteq Z$ *is a set of* ***shaded zones****.*
6. $\eta = \eta_\beta \colon \Sigma \rightarrow \mathbb{P}Z \backslash \{\emptyset\}$ *is a function that returns the* ***location*** *of each spider.*
7. $A = A(\beta)$ *is a finite set of arrows such that for all* $(s, R, t, \circ)$ *in* A*,* s *and* t *are in* $\Sigma \cup K \cup \{(\square, \beta)\}$*.*

A spider or curve that does not map to a label under λ *is called* ***unlabelled****. A set of zones is called a* ***region****.*

Briefly, β_1 in figure 2 has $\Sigma = \{\sigma\}$, $K = \{\kappa_1, \kappa_2\}$, $\lambda(\kappa_1) = C_1$, and $\lambda(\kappa_2) = C_2$. There are three zones (the regions in the plane to which the drawn curves give rise), so $Z = \{(\emptyset, \{\kappa_1, \kappa_2\}), (\{\kappa_1\}, \{\kappa_2\}), (\{\kappa_1, \kappa_2\}, \emptyset)\}$ and none of them are shaded. The function η maps σ to the region $\eta(\sigma) = \{(\{\kappa_1\}, \{\kappa_2\}), (\{\kappa_1, \kappa_2\}, \emptyset)\}$. As β_1 does not contain any arrows (but does contains an arrow source), $A = \emptyset$.

Definition 6. *A **concept diagram** is a tuple, $\mathcal{B} = (\mathcal{D}, A)$, where*

1. *$\mathcal{D}$ is a finite set of unitary diagrams such that for any pair of distinct unitary diagrams, β_1 and β_2, in $\mathcal{D}$ we have $\Sigma(\beta_1) \cap \Sigma(\beta_2) = \emptyset$, and $K(\beta_1) \cap K(\beta_2) = \emptyset$.*
2. *$A = A(\mathcal{B})$ is a finite set of arrows such that for all $(s, R, t, \circ)$ in A, $s, t \in \Sigma(\mathcal{B}) \cup K(\mathcal{B}) \cup (\{\square\} \times \mathcal{D})$ where*

$$\Sigma(\mathcal{B}) = \bigcup_{\beta \in \mathcal{D}} \Sigma(\beta), \qquad \text{and} \qquad K(\mathcal{B}) = \bigcup_{\beta \in \mathcal{D}} K(\beta)$$

   and for all unitary diagrams, β, in $\mathcal{D}$ it is not the case that $s \in \Sigma(\beta) \cup K(\beta) \cup \{(\square, \beta)\}$ and $t \in \Sigma(\beta) \cup K(\beta) \cup \{(\square, \beta)\}$.

The last condition above ensures that arrows in the set $A(\mathcal{B})$ go between different unitary diagrams. This condition can be removed without causing any theoretical problems. It might, however, be counterintuitive if arrows in $A(\mathcal{B})$ simply placed an arrow into one of the unitary parts of the concept diagram. Concept diagrams make use of standard logical connectives to build more complex expressions [23] but these are not needed when focusing on $\mathcal{ALC}$.

Turning our attention to the semantics, the meaning of a unitary diagram is determined by how its individual pieces of syntax are related to each other. We start by translating a unitary diagram into a set of *semantic conditions*. These conditions capture the constraints, provided by the diagram, on the relationships between the represented individuals, concepts, and roles. We start by identifying the elements and sets represented by the labelled spiders and labelled curves. This identification allows us to treat labelled and unlabelled spiders and, respectively, curves, in the same way in our semantic conditions.

Definition 7. *Let β be a unitary diagram and let $\mathcal{I}$ be an interpretation. Let s be a labelled spider and c be a labelled curve in β. We define $s^{\mathcal{I}} = \lambda(s)^{\mathcal{I}}$ and $c^{\mathcal{I}} = \lambda(c)^{\mathcal{I}}$.*

Definition 8. *Let $\mathcal{B} = (\mathcal{D}, A)$ be a concept diagram and let $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation, extended so that $(\square, \beta)^{\mathcal{I}} = \triangle^{\mathcal{I}}$, for any β. Then $\mathcal{I}$ is a **model** for $\mathcal{B}$, and $\mathcal{I}$ **satisfies** $\mathcal{B}$, provided there exists an extension of $\mathcal{I}$ to the unlabelled spiders and unlabelled curves in the unitary parts of $\mathcal{B}$, mapping spiders to elements and curves to sets, ensuring the conjunction of the following conditions, called the **semantic conditions**, hold:*

1. *For each unitary diagram, β, in $\mathcal{B}$ the following are true.*

(a) ***The Curves Condition.*** *The union of the sets represented by the zones is equal to* $\triangle^{\mathcal{I}}$*:*

$$\bigcup_{(in,out)\in Z(\beta)} (in, out)^{\mathcal{I}} = \triangle^{\mathcal{I}}$$

where

$$(in, out)^{\mathcal{I}} = \bigcap_{\kappa\in in} \kappa^{\mathcal{I}} \cap \bigcap_{\kappa\in out} (\triangle^{\mathcal{I}}\backslash\kappa^{\mathcal{I}}).$$

(b) ***The Shading Condition.*** *Every shaded zone contains only elements represented by spiders:*

$$\bigwedge_{(in,out)\in Z^*(\beta)} (in, out)^{\mathcal{I}} \subseteq \{\sigma^{\mathcal{I}} : \sigma \in \Sigma\}.$$

(c) ***The Spiders' Location Condition.*** *Each spider,* σ*, represents an element that lies in one of the sets represented by the zones in its location:*

$$\bigwedge_{\sigma\in\Sigma(\beta)} \sigma^{\mathcal{I}} \in \bigcup_{(in,out)\in\eta_\beta(\sigma)} (in, out)^{\mathcal{I}}.$$

(d) ***The Spiders' Distinctness Condition.*** *Any two distinct spiders,* σ_1 *and* σ_2*, represent distinct elements:*

$$\bigwedge_{\sigma_1,\sigma_2\in\Sigma(\beta)} (\sigma_1 \neq \sigma_2 \Rightarrow \sigma_1^{\mathcal{I}} \neq \sigma_2^{\mathcal{I}}).$$

(e) ***The Arrows Condition.*** *For each arrow with source* s*, label* R *and target* t*:*

$$\bigwedge_{(s,R,t,\rightarrow)\in A(\beta)} \{y \in \triangle^{\mathcal{I}} : \exists x\,(x \in s^{\mathcal{I}} \wedge (x, y) \in R^{\mathcal{I}})\} = t^{\mathcal{I}} \quad \text{and}$$

$$\bigwedge_{(s,R,t,\dashrightarrow)\in A(\beta)} \{y \in \triangle^{\mathcal{I}} : \exists x\,(x \in s^{\mathcal{I}} \wedge (x, y) \in R^{\mathcal{I}})\} \supseteq t^{\mathcal{I}}.$$

where we are treating $s^{\mathcal{I}}$ *and* $t^{\mathcal{I}}$ *as singleton sets, rather than elements, in the cases when* s *and* t*, respectively, are spiders.*

2. *For each arrow, with source* s*, label* R *and target* t*, in* $A(B)$*, the arrows condition as just given above holds.*

An extension of $\mathcal{I}$ *that makes the above conditions true is called* ***appropriate****. Moreover, given a region,* r*, we define* $r^{\mathcal{I}} = \bigcup_{z\in r} z^{\mathcal{I}}$*.*

4 Building Diagrams for Concepts

Here we provide an inductive construction of concept diagrams for $\mathcal{ALC}$ concepts. The general construction relies on *merging* unitary diagrams. For this operation, as well as other parts of the construction, we rely on diagrams having disjoint

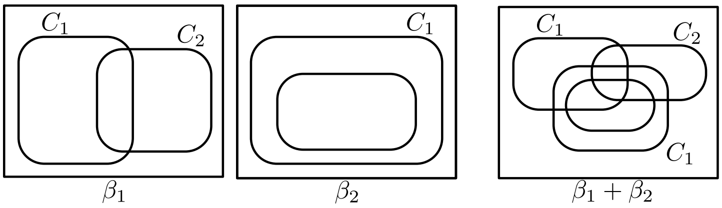


Fig. 3. Merging two diagrams.

curve sets. This reliance is not significant since we can always perform curve substitution, akin to variable substitution in symbolic logics, ensuring that the diagram's components, such as arrow sources and targets and the zones, are updated in the appropriate way; for zones, when substituting κ_1 with κ_2, the zone (in, out) becomes $((in \backslash \{\kappa_1\}) \cup \{\kappa_2\}, out)$ when κ_1 is in in, with the substitution operating similarly when κ_1 is in out. We point out that the construction we give is intended to establish that $\mathcal{ALC}$ axioms *can* all be visualized using concept diagrams. It does not necessarily yield the most effective diagrams, a point to which we return in section 6.

4.1 Merging Diagrams

In order to build diagrams to represent concepts, we need to be able to merge two unitary diagrams that do not contain spiders. An example can be seen in figure 3, where β_1 and β_2 are merged into the single diagram $\beta_1 + \beta_2$.

In order to identify the zones of the merged diagram we use the notion of an *expansion* of a region. To illustrate the idea, in figure 3, suppose that the curves in β_1 are κ_1 and κ_2 and in β_2 the curves are κ_1' and κ. The region $\{(\{\kappa_1\}, \{\kappa_2\})\}$ in β_1 can be expanded, without changing the set represented, to a four-zone region:

$$\{(\{\kappa_1\}, \{\kappa_2, \kappa_1', \kappa\}), (\{\kappa_1, \kappa_1'\}, \{\kappa_2, \kappa\}), (\{\kappa_1, \kappa\}, \{\kappa_2, \kappa_1'\}), (\{\kappa_1, \kappa_1', \kappa\}, \{\kappa_2\})\}.$$

Definition 9. *Let r be a region and let K be a set of fresh curves (that is no zone in r includes any curve in K). The **expansion** of r given K is the region*

$$\mathcal{EXP}(r, K) = \{(in \cup K', out \cup (K \backslash K')) : (in, out) \in r \wedge K' \subseteq K\}.$$

Lemma 1. *Let r be a region and let K be a set of fresh curves. In any interpretation, $\mathcal{I}$, $r^{\mathcal{I}} = \mathcal{EXP}(r, K)^{\mathcal{I}}$, under any extension of $\mathcal{I}$ mapping curves to sets.*

When merging two diagrams, we can start the process by expanding their zone sets using the curves in the other diagram. The zones in the merged diagram will be the intersection of these two expansions, thus not including zones that represent empty sets. For instance, considering the four-zone expansion of $(\{\kappa_1\}, \{\kappa_2\})$ given above, the zone $(\{\kappa_1, \kappa\}, \{\kappa_2, \kappa_1'\})$ represents the empty set and is not included in $\beta_1 + \beta_2$. We are now in a position to define how to merge two unitary diagrams that do not contain any spiders.

Definition 10. *Given unitary diagrams* $\beta_1 = (\Sigma_1, K_1, \lambda_1, Z_1, Z_1^*, \eta_1, A_1)$ *and* $\beta_2 = (\Sigma_2, K_2, \lambda_2, Z_2, Z_2^*, \eta_2, A_2)$*, containing no spiders and with disjoint curve sets, their* ***merger*** *is a unitary diagram,* $\beta = \beta_1 + \beta_2$*, whose (possibly) non-empty components are:* $K(\beta) = K_1 \cup K_2$*,* $\lambda_\beta = \lambda_1 \cup \lambda_2$*,*

$$Z(\beta) = \mathcal{EXP}(Z_1, K_2) \cap \mathcal{EXP}(Z_2, K_1), \quad Z^*(\beta) = Z(\beta) \cap (\mathcal{EXP}(Z_1^*, K_2) \cup \mathcal{EXP}(Z_2^*, K_1)),$$

and $A(\beta) = A_1 \cup A_2$*.*

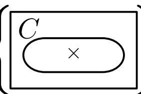
Lemma 2. *Let* β_1 *and* β_2 *be unitary diagrams with no spiders and disjoint curve sets. Interpretation* $\mathcal{I}$ *models* β_1 *and* β_2 *iff* $\mathcal{I}$ *models* $\beta_1 + \beta_2$*.*

Proof (Sketch). Follows readily from lemma 1.

4.2 Translating Concepts into Diagrams

The diagrams we build for concepts express no information, just as the lefthand side and righthand side of an $\mathcal{ALC}$ axiom contain no information when considered in isolation; complex concepts merely describe sets, but do not place any constraints on them (which is done through the use of $\sqsubseteq$ in an axiom, for example). The important feature of diagrams for concepts is that they contain a region that represents the same set as the concept. In what follows, this region is identified diagrammatically by the inclusion of $\times$ as an annotation. The construction is inductive and we begin by defining diagrams for atomic concepts, together with regions that represents the same set as the concept.

Definition 11. *Let* C *be an atomic concept. The* ***concept diagram for*** $\boldsymbol{C}$*, denoted* $\mathcal{DIAG}(C)$*, and the* ***region for*** $\boldsymbol{C}$*, denoted* $\mathcal{REG}(C)$*, are as follows:*

$\mathcal{DIAG}(C) = \Big(\Big\{$ 

$\Big\}, \emptyset\Big)$ and $\mathcal{REG}(C) = \{(\{\kappa\}, \emptyset)\}$

where κ *is the curve labelled* C*. Moreover, the unitary part of* $\mathcal{DIAG}(C)$ *is called the* ***merging diagram for*** $\boldsymbol{C}$*, denoted* $\mathcal{MER}(C)$*.*

Strictly speaking, the translation of an atomic concept to a diagram returns the *abstract syntax of the concept diagram* but our definition presents a drawing of $\mathcal{DIAG}(C)$ for readability.

Lemma 3. *Let* C *be an atomic concept. In any interpretation,* $\mathcal{I}$*,* $C^{\mathcal{I}} = \mathcal{REG}(C)^{\mathcal{I}}$*.*

Using this simple base case, we can now build diagrams for complex concepts. In these diagrams, we need to build anonymous concepts using arrows for concepts that involve quantifiers. To facilitate this, we need to add curves inside regions, since arrows cannot be sourced or targeted on the *regions* which

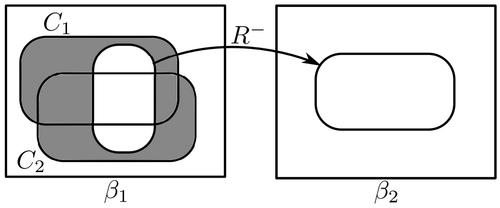


Fig. 4. Translating $\exists R.(C_1 \sqcup C_2)$.

represent concepts. To illustrate, figure 4 shows a diagram for $\exists R.(C_1 \sqcup C_2)$. Here, the unlabelled curve in β_1 represents the same set as $C_1 \sqcup C_2$. The arrow labelled R^- constructs the set of elements that are related to by some element in $C_1 \sqcup C_2$ and, thus, the unlabelled curve in β_2 represents $\exists R.(C_1 \sqcup C_2)$.

Definition 12. *Let β be a unitary diagram containing no spiders and let r be a region in β. Let κ be a fresh curve. The diagram obtained by* ***adding*** *κ inside r, denoted $\beta + (r, \kappa)$ has the same components as β except that the curves are $K(\beta) \cup \{\kappa\}$, the zones are*

$$Z(\beta + (r, \kappa)) = \{(in, out \cup \{\kappa\}) : (in, out) \in Z(\beta) \backslash r\} \cup \mathcal{EXP}(r, \{\kappa\})$$

and the shaded zones are

$$Z^*(\beta + (r, \kappa)) = \{(in, out) \in Z(\beta + (r, \kappa)) : (in \backslash \{\kappa\}, out \backslash \{\kappa\}) \in Z^*(\beta)\} \cup \{(in, out \cup \{\kappa\}) : (in, out) \in r\}.$$

Lemma 4. *Let β be a unitary diagram containing no spiders and let r be a region in β. Let κ be a fresh curve. Let $\mathcal{I}$ be an interpretation. Then*

1. *$r^{\mathcal{I}} = \kappa^{\mathcal{I}}$ under any appropriate extension of $\mathcal{I}$ for $\beta + (r, \kappa)$, and*
2. *$\mathcal{I}$ models β iff $\mathcal{I}$ models $\beta + (r, \kappa)$.*

Before we present a definition of the concept diagram for an arbitrary non-atomic concept, we illustrate the key features of the translation by considering $\exists R_1.\neg C_1 \sqcap \forall R_2.(C_2 \sqcup C_3)$. The construction, being inductive, starts by translating the atomic concepts C_1, C_2 and C_3 as in definition 11. The next stage is to form diagrams for $\neg C_1$ and $C_2 \sqcup C_3$. In fact, the diagram for $\neg C_1$ is the *same* as that for C_1, but the region for $\neg C_1$ differs: it is the complement of the region for C_1. The diagram for $C_2 \sqcup C_3$ is the merger of the diagrams for C_2 and C_3 with, roughly speaking, the associated region being the 'union' of the regions for C_2 and C_3. The diagrams for $\neg C_1$ and $C_2 \sqcup C_3$ are β_1 and β_2 respectively, figure 5, with their associated regions indicated by $\times$.

We can now build diagrams for $\exists R_1.\neg C_1$ and $\forall R_2.(C_2 \sqcup C_3)$. Considering $\exists R_1.\neg C_1$, we obtain the concept diagram $(\{\beta_3, \beta_4\}, \{(\kappa_3, R_1^-, \kappa_4, \rightarrow)\})$, where κ_3 and κ_4 are the unlabelled curves in β_3 and β_4 respectively. Here, the diagram which contains the region representing the concept $\exists R_1.\neg C_1$ is β_4; this

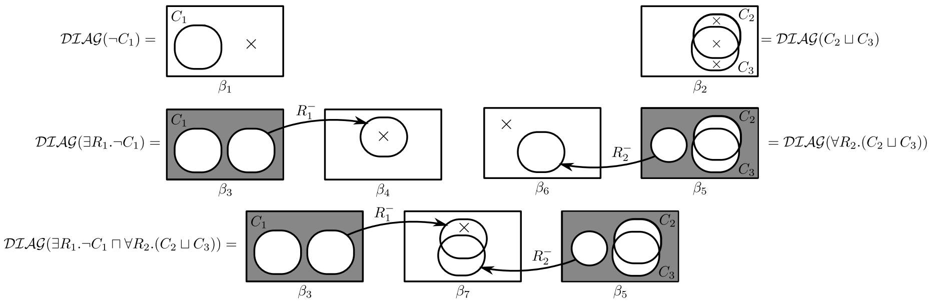


Fig. 5. Translating $\neg C_1$, $C_2 \sqcup C_3$, $\exists R_1.\neg C_1$, $\forall R_2.(C_2 \sqcup C_3)$ and $\exists R_1.\neg C_1 \sqcap \forall R_2.(C_2 \sqcup C_3)$.

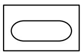
is the *merging diagram.* For $\forall R_2.(C_2 \sqcup C_3)$, we obtain the concept diagram $(\{\beta_5, \beta_6\}, \{(\kappa_5, R_2^-, \kappa_6, \rightarrow)\})$, where β_6 is the merging diagram. Here, the arrow, together with its source, is used to construct the set of things related to by something not in $C_1 \sqcup C_2$. Thus, the complement of this set – represented by region outside the curve in β_6 – contains exactly the elements that are in $\forall R_2.(C_1 \sqcup C_3)$. The last step is to form a diagram for the entire concept of interest: $\exists R_1.\neg C_1 \sqcap \forall R_2.(C_2 \sqcup C_3)$. We merge β_4 and β_6, leaving β_3 and β_5 unchanged, with the result being the concept diagram $(\{\beta_3, \beta_5, \beta_7\}, \{(\kappa_3, R_1^-, \kappa_4, \rightarrow), (\kappa_5, R_2^-, \kappa_6, \rightarrow)\})$, again with the region representing the entire concept indicated with the inclusion of $\times$.

Definition 13. *Let C be a non-atomic concept. The* ***concept diagram for C****, denoted $\mathcal{DIAG}(C)$, the* ***region for C****, denoted $\mathcal{REG}(C)$, and the* ***merging diagram for C****, denoted $\mathcal{MER}(C)$, are defined as follows:*

1. *If $C = C_1 \sqcap C_2$ then*
   (a) $\mathcal{MER}(C_1 \sqcap C_2) = \mathcal{MER}(C_1) + \mathcal{MER}(C_2)$,
   (b) $\mathcal{DIAG}(C_1 \sqcap C_2) = (\mathcal{D}, A_1 \cup A_2)$ *where*
   $$\mathcal{D} = (\mathcal{D}_1 \backslash \{\mathcal{MER}(C_1)\}) \cup (\mathcal{D}_2 \backslash \{\mathcal{MER}(C_2)\}) \cup \{\mathcal{MER}(C_1 \sqcap C_2)\},$$
   and
   (c) $\mathcal{REG}(C_1 \sqcap C_2) = \mathcal{EXP}(\mathcal{REG}(C_1), K_2) \cap \mathcal{EXP}(\mathcal{REG}(C_2), K_1)$.
2. *If $C = C_1 \sqcup C_2$ then*
   (a) $\mathcal{MER}(C_1 \sqcup C_2) = \mathcal{MER}(C_1) + \mathcal{MER}(C_2)$,
   (b) $\mathcal{DIAG}(C_1 \sqcup C_2) = (\mathcal{D}, A_1 \cup A_2)$ *where*
   $$\mathcal{D} = (\mathcal{D}_1 \backslash \{\mathcal{MER}(C_1)\}) \cup (\mathcal{D}_2 \backslash \{\mathcal{MER}(C_2)\}) \cup \{\mathcal{MER}(C_1 \sqcup C_2)\},$$
   and
   (c) $\mathcal{REG}(C_1 \sqcup C_2) = Z(\mathcal{MER}(C_1 \sqcup C_2)) \cap (\mathcal{EXP}(\mathcal{REG}(C_1), K_2) \cup \mathcal{EXP}(\mathcal{REG}(C_2), K_1))$.
3. *If $C = \neg C_1$ then*
   (a) $\mathcal{MER}(\neg C_1) = \mathcal{MER}(C_1)$,
   (b) $\mathcal{DIAG}(\neg C_1) = \mathcal{DIAG}(C_1)$ *and*

(c) $\mathcal{REG}(\neg C_1) = Z(\mathcal{MER}(C_1)) \backslash \mathcal{REG}(C_1)$.

4. If $C = \exists R.C_1$ then
   (a) $\mathcal{MER}(\exists R.C_1)$ is a unitary diagram containing a fresh curve, κ_t:

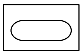


   (b) $\mathcal{DIAG}(\exists R.C_1) = (\mathcal{D}, A_1 \cup \{(\kappa_s, R^-, \kappa_t, \rightarrow)\})$ where

$$\mathcal{D} = (\mathcal{D}_1 \backslash \{\mathcal{MER}(C_1)\}) \cup \{\mathcal{MER}(C_1) + (\mathcal{REG}(C_1), \kappa_s), \mathcal{MER}(\exists R.C_1)\}$$

   and κ_s is a fresh curve, and
   (c) $\mathcal{REG}(\exists R.C_1) = \{(\{\kappa_t\}, \emptyset)\}$.

5. If $C = \forall R.C_1$ then
   (a) $\mathcal{MER}(\forall R.C_1)$ is a unitary diagram containing a fresh curve, κ_t:



   (b) $\mathcal{DIAG}(\forall R.C_1) = (\mathcal{D}, A_1 \cup \{(\kappa_s, R^-, \kappa_t, \rightarrow)\})$ where

$$\mathcal{D} = (\mathcal{D}_1 \backslash \{\mathcal{MER}(C_1)\}) \cup \\ \{\mathcal{MER}(C_1) + (Z(\mathcal{MER}(C_1)) \backslash \mathcal{REG}(C_1), \kappa_s), \mathcal{MER}(\forall R.C_1)\}$$

   and κ_s is a fresh curve, and
   (c) $\mathcal{REG}(\forall R.C_1) = \{(\emptyset, \{\kappa_t\})\}$.

where $\mathcal{DIAG}(C_1) = (\mathcal{D}_1, A_1)$, $\mathcal{DIAG}(C_2) = (\mathcal{D}_2, A_2)$, and K_1 and K_2 are the sets of curves in $\mathcal{MER}(C_1)$ and $\mathcal{MER}(C_2)$ respectively.

An important property of diagrams for concepts is that they are satisfied in every interpretation. This allows us to readily use them when constructing diagrams for $\mathcal{ALC}$ axioms.

Lemma 5. *Let C be a concept. Then $\mathcal{DIAG}(C)$ is satisfied by all interpretations, that is $\mathcal{DIAG}(C)$ is valid.*

Corollary 1. *Let C be a concept. Then all unitary parts of $\mathcal{DIAG}(C)$ are valid.*

We now establish the crucial result that $\mathcal{REG}(C)$ represents the same set as C.

Theorem 1. *Let C be a concept. For all interpretations, $\mathcal{I}$, $C^{\mathcal{I}} = \mathcal{REG}(C)^{\mathcal{I}}$ under any appropriate extension of $\mathcal{I}$ for $\mathcal{DIAG}(C)$.*

Proof (Sketch). The proof proceeds by induction with the base case provided by lemma 3. We include the remainder of the proof for the $\exists R.C_1$ and $\forall R.C_1$ cases. In the first of these two cases the curve, κ_t, in $\mathcal{DIAG}(\exists R.C_1)$ that is the target of the arrow represents the image of R^- when its domain is restricted to C_1. Formally, we have

$$\begin{aligned} \kappa_t^{\mathcal{I}} &= \{x \in \triangle^{\mathcal{I}} : \exists y\,(y \in C_1^{\mathcal{I}} \wedge (y, x) \in R^{-\mathcal{I}})\}, \text{ by definition 8} \\ &= \{x \in \triangle^{\mathcal{I}} : \exists y\,(y \in C_1^{\mathcal{I}} \wedge (x, y) \in R^{\mathcal{I}})\} = \exists R.C_1^{\mathcal{I}}. \end{aligned}$$

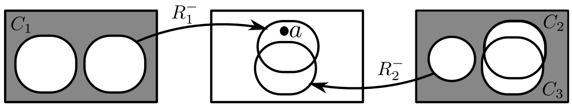


Fig. 6. $(\exists R_1.\neg C_1 \sqcap \forall R_2.(C_2 \sqcup C_3))(a)$.

It is straightforward to verify that $\mathcal{REG}(\exists R.C_1)^{\mathcal{I}} = \{(\{\kappa_t\}, \emptyset)\}^{\mathcal{I}} = \kappa_t^{\mathcal{I}}$ and we are done.

For the $\forall R.C_1$ case, we must show that $\mathcal{REG}(\forall R.C_1) = \{(\emptyset, \{\kappa_t\})\}$ represents the same set as $\forall R.C_1$. Consider $\mathcal{MER}(C_1) + (Z(\mathcal{MER}(C_1), \kappa_s) \backslash \mathcal{REG}(C_1))$. We can show that κ_s, which is the source of the arrow labelled R^-, represents the set $\triangle^{\mathcal{I}} \backslash C_1^{\mathcal{I}}$, using the inductive assumption. Therefore in $\mathcal{MER}(\forall R.C_1)$ the curve, κ_t, which is the target of the arrow labelled R^-, represents the set

$$\kappa_t^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} : \exists y\, (y \in \triangle^{\mathcal{I}} \backslash C_1^{\mathcal{I}} \wedge (x, y) \in R^{\mathcal{I}})\} = \exists R.\neg C_1.$$

Thus, $\triangle^{\mathcal{I}} \backslash \kappa_t^{\mathcal{I}}$ contains precisely the elements in $\triangle^{\mathcal{I}}$ that are related only to things in the set $C_1^{\mathcal{I}}$ under $R^{\mathcal{I}}$. More formally,

$$\triangle^{\mathcal{I}} \backslash \kappa_t^{\mathcal{I}} = \Big\{x \in \triangle^{\mathcal{I}} : \forall y \Big((y \in \triangle^{\mathcal{I}} \wedge (x, y) \in R^{\mathcal{I}}) \Rightarrow y \in C_1^{\mathcal{I}}\Big)\Big\} = \forall R.C_1^{\mathcal{I}}.$$

Since $\mathcal{REG}(\forall R.C_1) = \{(\emptyset, \{\kappa_t\})\}$, we readily see that $\mathcal{REG}(\forall R.C_1)^{\mathcal{I}} = \triangle^{\mathcal{I}} \backslash \kappa_t^{\mathcal{I}}$, by definition, and we are done. Hence, in all cases, $C^{\mathcal{I}} = \mathcal{REG}(C)^{\mathcal{I}}$, as required.

5 Visualizing Axioms

In this section we show how to visualize $\mathcal{ALC}$ axioms using concept diagrams.

5.1 ABox Axioms

The Manchester OWL corpus [1] contains over 1.5 million ABox axioms of which 64.3% are in $\mathcal{ALC}$[5]. Using the diagrams constructed for concepts, we are now readily able to establish that all A-box axioms in $\mathcal{ALC}$ can be visualized using concept diagrams. The basic principle for ABox axioms of the form $C(a)$ is to place a spider labelled a in $\mathcal{REG}(C)$. To illustrate, the ABox axiom for $(\exists R_1.\neg C_1 \sqcap \forall R_2.(C_2 \sqcup C_3))(a)$ is visualized in figure 6.

Definition 14. *The* ***ABox diagram*** *for $\mathcal{ALC}$ axiom $C(a)$, denoted $\mathcal{DIAG}(C(a))$, is obtained from $\mathcal{DIAG}(C)$ by adding a spider labelled a to $\mathcal{REG}(C)$ in $\mathcal{MER}(\mathcal{C})$.*

[5] To count axioms, we used OWL API's DL expressiveness checker. Each axiom is extracted and provided to the OWL API which determines whether the axiom is syntactically in $\mathcal{ALC}$. This approach is somewhat crude, in that some OWL non-$\mathcal{ALC}$ axioms can be reduced to a set of axioms including some in $\mathcal{ALC}$; we count such OWL axioms as not being in $\mathcal{ALC}$. Of the ontologies in the corpus, we could parse 4019. Our counting software is an extension of an existing ontology statistics processing package [11] and can be found at https://github.com/hammar/OntoStats.

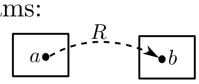
Theorem 2. *Let $C(a)$ be an ABox axiom in $\mathcal{ALC}$ and let $\mathcal{I}$ be an interpretation. $\mathcal{I}$ satisfies $C(a)$ iff $\mathcal{I}$ satisfies $\mathcal{DIAG}(C(a))$.*

Proof. Suppose $\mathcal{I}$ satisfies $C(a)$. Lemma 5 tells us that $\mathcal{I}$ satisfies $\mathcal{DIAG}(C)$. Moreover, corollary 1 tells us that $\mathcal{I}$ satisfies $\mathcal{MER}(C)$, the unitary part of $\mathcal{DIAG}(C)$ into which the spider, σ say, labelled a has been placed. The only difference between the semantic conditions for $\mathcal{DIAG}(C)$ and $\mathcal{DIAG}(C(a))$ arise from the inclusion of this spider, whereby $\mathcal{DIAG}(C(a))$ asserts:

$$\sigma^{\mathcal{I}} \in \bigcup_{(in,out)\in\mathcal{REG}(C)} (in, out)^{\mathcal{I}} = \mathcal{REG}(C)^{\mathcal{I}} \qquad (*).$$

By theorem 1, $\mathcal{REG}(C)^{\mathcal{I}} = C^{\mathcal{I}}$. Since $\mathcal{I}$ satisfies C, we know that $a^{\mathcal{I}} \in C^{\mathcal{I}}$. By definition, $\sigma^{\mathcal{I}} = a^{\mathcal{I}}$, so (*) is true and we conclude that $\mathcal{I}$ satisfies $\mathcal{DIAG}(C(a))$. The proof for the converse, if $\mathcal{I}$ satisfies $\mathcal{DIAG}(C(a))$ then $\mathcal{I}$ satisfies $C(a)$, is similar. Hence $\mathcal{I}$ satisfies $C(a)$ if and only if $\mathcal{I}$ satisfies $\mathcal{DIAG}(C(a))$.

The remaining ABox case is for axioms of the form $R(a, b)$. These are trivially expressed using concept diagrams:



Hence, concept diagrams can express all of $\mathcal{ALC}$'s ABox axioms.

Theorem 3. *All ABox axioms in $\mathcal{ALC}$ can be visualized by a semantically equivalent concept diagram.*

5.2 TBox Axioms

The Manchester OWL corpus [1] contains over 1.3 million TBox axioms of which 66.3% are in $\mathcal{ALC}$. Using the diagrams constructed for concepts, we can establish that all TBox axioms in $\mathcal{ALC}$ can be visualized, although the process is not as straightforward as for ABox axioms. To illustrate, the TBox axiom for $\exists R_1.\neg C_1 \sqsubseteq \forall R_2.(C_2 \sqcup C_3)$ can be seen in figure 7 (the diagrams for $\exists R_1.\neg C_1$ and $\forall R_2.(C_2 \sqcup C_3)$ are in figure 5). The first step in the construction process is to merge the two merging diagrams for the two sides of the subsumption relationship. This is followed by shading the appropriate zones in order to obtain the correct subsumption relationship. In this example, there is one zone inside the region for $\exists R_1.\neg C_1$ but not in the region for $\forall R_2.(C_2 \sqcup C_3)$; this zone is shaded to assert that no elements can be in the corresponding set. Formally, the zones which require shading are captured by considering expansions of the regions for $\mathcal{REG}(\exists R_1.\neg C_1)$ and $\mathcal{REG}(\forall R_2.(C_2 \sqcup C_3))$.

Definition 15. *Let $C_1 \sqsubseteq C_2$ be a TBox axiom in $\mathcal{ALC}$ where $\mathcal{DIAG}(C_1) = (\mathcal{D}_1, A_1)$ and $\mathcal{DIAG}(C_2) = (\mathcal{D}_2, A_2)$. The **TBox diagram** for $C_1 \sqsubseteq C_2$, denoted $\mathcal{DIAG}(C_1 \sqsubseteq C_2)$, is obtained from the concept diagram*

$$\big((\mathcal{D}_1 \backslash \{\mathcal{MER}(C_1)\}) \cup (\mathcal{D}_2 \backslash \{\mathcal{MER}(C_1)\}) \cup \{\mathcal{MER}(C_1) + \mathcal{MER}(C_2)\}, A_1 \cup A_2\big)$$

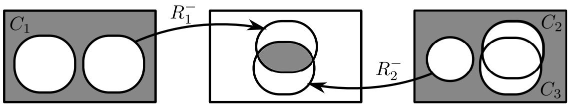


Fig. 7. $\exists R_1.\neg C_1 \sqsubseteq \forall R_2.(C_2 \sqcup C_3)$.

by shading the zones in

$$\mathcal{EXP}(\mathcal{REG}(C_1), K(\mathcal{MER}(C_2))) \backslash \mathcal{EXP}(\mathcal{REG}(C_2), K(\mathcal{MER}(C_1)))$$

in $\mathcal{MER}(C_1) + \mathcal{MER}(C_2)$.

Lemma 6. *Let C_1 and C_2 be $\mathcal{ALC}$ concepts. Let $\mathcal{I}$ be an interpretation. The following statements are equivalent.*

(1) $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
(2) $\mathcal{REG}(C_1)^{\mathcal{I}} \subseteq \mathcal{REG}(C_2)^{\mathcal{I}}$.
(3) $\mathcal{EXP}(\mathcal{REG}(C_1), K(\mathcal{MER}(C_2)))^{\mathcal{I}} \subseteq \mathcal{EXP}(\mathcal{REG}(C_2), K(\mathcal{MER}(C_1)))^{\mathcal{I}}$.

Theorem 4. *Let $C_1 \sqsubseteq C_2$ be a TBox axiom in $\mathcal{ALC}$ and let $\mathcal{I}$ be an interpretation. $\mathcal{I}$ satisfies $C_1 \sqsubseteq C_2$ iff $\mathcal{I}$ satisfies $\mathcal{DIAG}(C_1 \sqsubseteq C_2)$.*

Proof. Suppose that $\mathcal{I}$ satisfies $C_1 \sqsubseteq C_2$. Since $\mathcal{DIAG}(C_1)$ and $\mathcal{DIAG}(C_1)$ are valid, by lemma 5, we only need to show that $\mathcal{I}$ satisfies the merged unitary diagram $\mathcal{MER}(C_1)+\mathcal{MER}(C_2)$ with the shading added to it as in definition 15; call this diagram β. First, by corollary 1, $\mathcal{MER}(C_1)$ and $\mathcal{MER}(C_2)$ are both valid. By lemma 2, $\mathcal{MER}(C_1)+\mathcal{MER}(C_2)$ is also valid. Therefore, we only need to consider the shading condition for β. This condition reduces to

$$Z^*(\beta)^{\mathcal{I}} = \Big(\mathcal{EXP}(\mathcal{REG}(C_1), K(\mathcal{MER}(C_2))) \backslash \mathcal{EXP}(\mathcal{REG}(C_2), K(\mathcal{MER}(C_1)))\Big)^{\mathcal{I}} = \emptyset \ (*)$$

since there are no spiders. Now, since $C_1 \sqsubseteq C_2$ is satisfied by $\mathcal{I}$, we know that $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. Lemma 6 tells us, therefore, that

$$\mathcal{EXP}(\mathcal{REG}(C_1), K(\mathcal{MER}(C_2)))^{\mathcal{I}} \subseteq \mathcal{EXP}(\mathcal{REG}(C_2), K(\mathcal{MER}(C_1)))^{\mathcal{I}}$$

from which (*) follows, as required. Thus, $\mathcal{I}$ satisfies $\mathcal{DIAG}(C_1 \sqsubseteq C_2)$. The converse, omitted for space reasons, is similar. Hence $\mathcal{I}$ satisfies $C_1 \sqsubseteq C_2$ iff $\mathcal{I}$ satisfies $\mathcal{DIAG}(C_1 \sqsubseteq C_2)$.

Theorem 5. *All TBox axioms in $\mathcal{ALC}$ can be visualized by a semantically equivalent concept diagram.*

Theorems 3 and 5 establish that $\mathcal{ALC}$ can be visualized using concept diagrams.

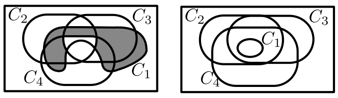


Fig. 8. $C_1 \sqsubseteq C_2 \sqcap C_3 \sqcap C_4$.

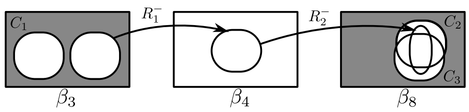


Fig. 9. $\exists R_1 \neg C_1 \sqsubseteq \forall R_2.(C_2 \sqcup C_3)$.

6 Improving the General Translations

The translations just defined sometimes return diagrams involving shaded zones. It is possible to simplify these diagrams by removing the shaded zones. An example is given in figure 8, where removing shaded zones reduces *clutter*. John et al. [15] defined a clutter score for Euler diagrams (which are concept diagrams that do not include any spiders or arrows): the clutter score for Euler diagram, β, denoted $CS(\beta)$ is

$$CS(\beta) = \sum_{(in,out) \in Z(\beta)} |in|.$$

In figure 8, the clutter score reduces from 31 to 16 when removing the shaded zones. *All* diagrams arising from TBox axioms involve shading and can be simplified in this way. Moreover, axioms involving quantifiers also give rise to diagrams that include shading.

Lemma 7. *Let $\mathcal{A}$ be an $\mathcal{ALC}$ axiom. Removing shaded zones from $\mathcal{DIAG}(\mathcal{A})$ reduces the clutter score.*

It is known that diagrams with a higher clutter score are harder for people to interpret [4] and it has further been shown that Euler diagrams without shading are easier to interpret [5]. Indeed, removing shaded zones makes the resulting diagram exploit spatial relations to assert information, making them well-matched to their semantics [10].

We can also simplify the translation of axioms of the form $C_1 \sqsubseteq \forall R.C_2$, where C_1 and C_2 are arbitrary concepts. For instance, in figure 7 the diagram unnaturally uses R_2^- in order to produce a region in β_6 that represents $\forall R_2.(C_2 \sqcup C_3)$. In fact, whilst helpful for a *general translation mechanism*, this construction step can be eliminated, instead making direct use of β_2. An alternative diagram can be seen in figure 9. Here, we have added a curve to β_2, figure 5, representing a subset of $C_2 \sqcup C_3$. This curve represents the set of all elements that things in $\exists R_1.\neg C_1$ are related to under R_2, though the use of the arrow targeting it. Thus, the diagram expresses $\exists R_1.\neg C_1 \sqsubseteq \forall R_2.(C_2 \sqcup C_3)$. This process readily generalizes to axioms of the form $C_1 \sqsubseteq C_2$ where C_2 involves top-level universal quantifiers. We note here that the use of inverse roles for existentially quantified concepts can also be avoided, see [14, 21] for examples.

7 Conclusion

This paper shows how to use concept diagrams to visualize $\mathcal{ALC}$ axioms. Our approach was to build diagrams for concepts and then use these diagrams to express ABox and TBox axioms. A substantial proportion of axioms from OWL ontologies are drawn from $\mathcal{ALC}$, establishing that concept diagrams can visualize a significant proportion of ontology axioms that have been developed. We view the contribution in this paper to be an important foundational step towards producing *usable* visualizations of description logic. Whilst our general translation from $\mathcal{ALC}$ may not produce ideal diagrams from a usability perspective, we have demonstrated some improvements can be readily achieved. Further improving the resulting diagrams is a key future ambition. For this, it is likely that extensive empirical studies will be required, to establish how to choose between semantically equivalent, yet syntactically different, concept diagrams. This was started in [3], where diagrams for common styles of axioms where empirically compared to ascertain their relative usability.

There are a number of other exciting avenues for future work. We plan to extend the translations to richer description logics, establishing that most, if not all, ontologies can be visualized using concept diagrams. It will be a particular challenge to produce improved versions of these visualizations, to ensure that the results of translations are most usable. Indeed, we envisage a much more general translation from DL axioms to concept diagrams, which identifies sets of DL axioms that can be translated to single diagrams, as in figure 1. We plan to automate the translation process, allowing the results to be readily used in practice. This brings with it substantial diagram drawing and layout problems, building on the body of work on Euler diagram generation [9, 16, 20, 22]. Work towards a theorem prover for concept diagrams has already begun [17], where it has been designed using empirical insights into what constitutes understandable inference rules [19]. Our ultimate vision is to devise a framework that allows concept diagrams to be used for ontology engineering, not merely as a visualization aid, either as a stand-alone notation or fully integrated with existing symbolic approaches.

Acknowledgement

Gem Stapleton was partially funded by a Leverhulme Trust Research Project Grant (RPG-2016-082) for the project entitled Accessible Reasoning with Diagrams.

References

1. Manchester owl corpus. http://owl.cs.manchester.ac.uk/publications/supporting-material/owlcorpus/ (accessed February 2014)
2. OntoGraf. http://protegewiki.stanford.edu/wiki/OntoGraf (accessed July 2013)
3. Alharbi, E., Howse, J., Stapleton, G., Hamie, A.: Reasoning with concept diagrams about antipatterns in ontologies. In: 9th International Conference on the Theory and Application of Diagrams. pp. 51–66. Springer (2016)
4. Alqadah, M., Stapleton, G., Howse, J., Chapman., P.: Evaluating the impact of clutter in Euler diagrams. In: 8th International Conference on the Theory and Application of Diagrams. pp. 108–122. Springer (2014)

5. Chapman, P., Stapleton, G., Rodgers, P., Micallef, L., Blake, A.: Visualizing sets: An empirical comparison of diagram types. In: 8th International Conference on the Theory and Application of Diagrams. pp. 146–160. Springer (2014)
6. Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the W3C semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web 17, 25 – 32 (2012)
7. Dau, F., Ekland, P.: A diagrammatic reasoning system for the description logic $\mathcal{ALC}$. Journal of Visual Languages and Computing 19(5), 539–573 (2008)
8. Duncan, J., Humphreys, G.: Visual search and stimulus similarity. Psychological Review 96, 433–458 (1989)
9. Flower, J., Howse, J.: Generating Euler diagrams. In: Proceedings of 2nd International Conference on the Theory and Application of Diagrams. pp. 61–75. Springer, Georgia, USA (2002)
10. Gurr, C.: Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. Journal of Visual Languages and Computing 10(4), 317–342 (1999)
11. Hammar, K.: Reasoning performance indicators for ontology design patterns. In: 4th Workshop on Ontology and Semantic Web Patterns (2013)
12. Hayes, P., Eskridge, T., Mehrotra, M., Bobrovnikoff, D., Reichherzer, T., Saavedra, R.: Coe: Tools for collaborative ontology development and reuse. In: Knowledge Capture Conference (2005)
13. Horridge, M.: Owlviz. www.co-ode.org/downloads/owlviz/ (accessed June 2009)
14. Howse, J., Stapleton, G., Taylor, K., Chapman, P.: Visualizing ontologies: A case study. In: International Semantic Web Conference. pp. 257–272. Springer (2011)
15. John, C., Fish, A., Howse, J., Taylor, J.: Exploring the notion of clutter in Euler diagrams. In: 4th International Conference on the Theory and Application of Diagrams. pp. 267–282. Springer, Stanford, USA (2006)
16. Riche, N., Dwyer, T.: Untangling Euler diagrams. IEEE Transactions on Visualization and Computer Graphics 16(6), 1090–1099 (2010)
17. Shams, Z., Jamnik, M., Stapleton, G., Sato, Y.: Reasoning with concept diagrams about antipatterns. In: 21st International Conference on Logic for Programming, Artificial Intelligence, and Reasoning. pp. 27–42. Kapla Publications in Computing (2017)
18. Shams, Z., Jamnik, M., Stapleton, G., Sato, Y.: Reasoning with concept diagrams about antipatterns in ontologies. In: Intelligent Computer Mathematics. pp. 255–271. Springer (2017)
19. Shams, Z., Sato, Y., Jamnik, M., Stapleton, G.: Accessible reasoning with diagrams: from cognition to automation. In: 10th International Conference on the Theory and Application of Diagrams. Springer (2018)
20. Simonetto, P., Auber, D., Archambault, D.: Fully automatic visualisation of overlapping sets. Computer Graphics Forum 28(3) (2009)
21. Stapleton, G., Compton, M., Howse, J.: Visualizing OWL 2 using diagrams. In: IEEE Symposium on Visual Languages and Human-Centric Computing. pp. 245–253. IEEE (2017)
22. Stapleton, G., Flower, J., Rodgers, P., Howse, J.: Automatically drawing Euler diagrams with circles. J. of Visual Languages and Computing 23, 163–193 (2012)
23. Stapleton, G., Howse, J., Chapman, P., Delaney, A., Burton, J., Oliver, I.: Formalizing concept diagrams. In: 19th International Conference on Distributed Multimedia Systems. pp. 182–187. KSI (2013)