Appendix 1

Description Logic Terminology

Franz Baader

Abstract

The purpose of this appendix is to introduce (in a compact manner) the syntax and semantics of the most prominent DLs occurring in this handbook. More information and explanations as well as some less familiar DLs can be found in the respective chapters. For DL constructors whose semantics cannot be described in a compact manner, we will only introduce the syntax and refer the reader to the respective chapter for the semantics. Following Chapter 2 on Basic Description Logics, we will first introduce the basic DL $\mathcal{AL}$, and then describe several of its extensions. Thereby, we will also fix the notation employed in this handbook. Finally, we will comment on the naming schemes for DLs that are employed in the literature and in this handbook.

A1.1 Notational conventions

Before starting with the definitions, let us introduce some notational conventions. The letters A, B will often be used for atomic concepts, and C, D for concept descriptions. For roles, we often use the letters R, S, and for functional roles (features, attributes) the letters f, g. Nonnegative integers (in number restrictions) are often denoted by n, m, and individuals by a, b. In all cases, we may also use subscripts. This convention is followed when defining syntax and semantics and in abstract examples. In concrete examples, the following conventions are used: concept names start with an uppercase letter followed by lowercase letters (e.g., Human, Male), role names (also functional ones) start with a lowercase letter (e.g., hasChild, marriedTo), and individual names are all uppercase (e.g., CHARLES, MARY).

A1.2 Syntax and semantics of common Description Logics

In this section, we introduce the standard concept and role constructors as well as knowledge bases. For more information see Chapter 2.

A1.2.1 Concept and role descriptions

Elementary descriptions are *atomic concepts* and *atomic roles* (also called *concept names* and *role names*). Complex descriptions can be built from them inductively with *concept constructors* and *role constructors.* Concept descriptions in $\mathcal{AL}$ are formed according to the following syntax rule:

$$\begin{array}{lcll} C, D & \longrightarrow & A \mid & \text{(atomic concept)} \\ & & \top \mid & \text{(universal concept)} \\ & & \bot \mid & \text{(bottom concept)} \\ & & \neg A \mid & \text{(atomic negation)} \\ & & C \sqcap D \mid & \text{(intersection)} \\ & & \forall R.C \mid & \text{(value restriction)} \\ & & \exists R.\top & \text{(limited existential quantification).} \end{array}$$

Following our convention, A denotes an atomic concept and C, D denote concept descriptions. The role R is atomic since $\mathcal{AL}$ does not provide for role constructors.

An *interpretation* $\mathcal{I}$ consist of a non-empty set $\Delta^\mathcal{I}$ (the domain of the interpretation) and an interpretation function, which assigns to every atomic concept A a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and to every atomic role R a binary relation $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The interpretation function is extended to concept descriptions by the following inductive definitions:

$$\begin{array}{rcl} \top^\mathcal{I} & = & \Delta^\mathcal{I} \\ \bot^\mathcal{I} & = & \emptyset \\ \neg A^\mathcal{I} & = & \Delta^\mathcal{I} \setminus A^\mathcal{I} \\ (C \sqcap D)^\mathcal{I} & = & C^\mathcal{I} \cap D^\mathcal{I} \\ (\forall R.C)^\mathcal{I} & = & \{a \in \Delta^\mathcal{I} \mid \forall b.\ (a, b) \in R^\mathcal{I} \to b \in C^\mathcal{I}\} \\ (\exists R.\top)^\mathcal{I} & = & \{a \in \Delta^\mathcal{I} \mid \exists b.\ (a, b) \in R^\mathcal{I}\}. \end{array}$$

There are several possibilities for extending $\mathcal{AL}$ in order to obtain a more expressive DL. The three most prominent are adding additional concept constructors, adding role constructors, and formulating restrictions on role interpretations. Below, we start with the third possibility, since we need to refer to restrictions on roles when defining certain concept constructors. For these extensions, we also introduce a naming scheme. Basically, each extension is assigned a letter or symbol. For concept constructors, the letters/symbols are written after the starting $\mathcal{AL}$, for role

Table A1.1. *Some Description Logic concept constructors.*

Name	Syntax	Semantics	Symbol
Top	$\top$	$\Delta^{\mathcal{I}}$	$\mathcal{AL}$
Bottom	$\bot$	$\emptyset$	$\mathcal{AL}$
Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$\mathcal{AL}$
Union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$\mathcal{U}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	$\mathcal{C}$
Value restriction	$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \to b \in C^{\mathcal{I}}\}$	$\mathcal{AL}$
Existential quant.	$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$	$\mathcal{E}$
Unqualified number restriction	$\geqslant n\,R$ $\leqslant n\,R$ $= n\,R$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}\rvert = n\}$	$\mathcal{N}$
Qualified number restriction	$\geqslant n\,R.C$ $\leqslant n\,R.C$ $= n\,R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\rvert \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\rvert \leq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\rvert = n\}$	$\mathcal{Q}$
Role-value-map	$R \subseteq S$ $R = S$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \to (a,b) \in S^{\mathcal{I}}\}$ $\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \leftrightarrow (a,b) \in S^{\mathcal{I}}\}$	
Agreement and disagreement	$u_1 \doteq u_2$ $u_1 \neq u_2$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}}.\ u_1^{\mathcal{I}}(a) = b = u_2^{\mathcal{I}}(a)\}$ $\{a \in \Delta^{\mathcal{I}} \mid \exists b_1, b_2 \in \Delta^{\mathcal{I}}.\ u_1^{\mathcal{I}}(a) = b_1 \neq b_2 = u_2^{\mathcal{I}}(a)\}$	$\mathcal{F}$
Nominal	I	$I^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ with $\lvert I^{\mathcal{I}}\rvert = 1$	$\mathcal{O}$

constructors, we write the letters/symbols as superscripts, and for restrictions on the interpretation of roles as subscripts. As an example, the DL $\mathcal{ALCQ}^{-1}_{R^+}$ extends $\mathcal{AL}$ with the concept constructors negation ($\mathcal{C}$) and qualified number restrictions ($\mathcal{Q}$), the role constructor inverse ($^{-1}$), and the restriction that some roles are transitive ($_{R^+}$).

Restrictions on role interpretations

These restrictions enforce the interpretations of roles to satisfy certain properties, such as functionality and transitivity. We consider these two prominent examples in more detail. Others would be symmetry or connections between different roles.[1]

(i) *Functional roles.* Here one considers a subset N_F of the set of role names N_R, whose elements are called *features*. An interpretation must map features

[1] One could also count role hierarchies as imposing such restrictions. Here we will, however, treat role hierarchies in the context of knowledge bases.

Table A1.2. *Concrete syntax of concept constructors.*

Name	Concrete syntax	Abstract syntax
Top	`TOP`	$\top$
Bottom	`BOTTOM`	$\bot$
Intersection	(and $C_1 \cdots C_n$)	$C_1 \sqcap \cdots \sqcap C_n$
Union	(or $C_1 \cdots C_n$)	$C_1 \sqcup \cdots \sqcup C_n$
Negation	`(not C)`	$\neg C$
Value restriction	`(all R C)`	$\forall R.C$
Limited existential quantification	`(some R)`	$\exists R.\top$
Existential quantification	`(some R C)`	$\exists R.C$
At-least number restriction	`(at-least n R)`	$\geqslant n\, R$
At-most number restriction	`(at-most n R)`	$\leqslant n\, R$
Exact number restriction	`(exactly n R)`	$= n\, R$
Qualified at-least restriction	`(at-least n R C)`	$\geqslant n\, R.C$
Qualified at-most restriction	`(at-most n R C)`	$\leqslant n\, R.C$
Qualified exact restriction	`(exactly n R C)`	$= n\, R.C$
Same-as, agreement	(same-as u_1 u_2)	$u_1 \doteq u_2$
Role-value-map	(subset R_1 R_2)	$R_1 \subseteq R_2$
Role fillers	(fillers R $I_1 \cdots I_n$)	$\exists R.I_1 \sqcap \cdots \sqcap \exists R.I_n$
One-of	(one-of $I_1 \cdots I_n$)	$I_1 \sqcup \cdots \sqcup I_n$

f to functional binary relations $f^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, i.e., relations satisfying $\forall a, b, c. f^{\mathcal{I}}(a, b) \wedge f^{\mathcal{I}}(a, c) \rightarrow b = c$. Sometimes functional relations are viewed as partial function, and thus one writes $f^{\mathcal{I}}(a) = b$ rather than $f^{\mathcal{I}}(a, b)$. $\mathcal{AL}$ extended with features is denoted by $\mathcal{AL}_f$.

(ii) *Transitive roles.* Here one considers a subset N_{R^+} of N_R. Role names $R \in N_{R^+}$ are called *transitive roles*. An interpretation must map transitive roles $R \in N_{R^+}$ to transitive binary relations $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. $\mathcal{AL}$ extended with transitive roles is denoted by $\mathcal{AL}_{R^+}$.

Concept constructors

Concept constructors take concept and/or role descriptions and transform them into more complex concept descriptions. Table A1.1 shows the syntax and semantics of common concept constructors. In order to have them all in one place, we also repeat

Table A1.3. *Some Description Logic role constructors.*

Name	Syntax	Semantics	Symbol
Universal role	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	U
Intersection	$R \sqcap S$	$R^{\mathcal{I}} \cap S^{\mathcal{I}}$	$\sqcap$
Union	$C \sqcup D$	$R^{\mathcal{I}} \cup S^{\mathcal{I}}$	$\sqcup$
Complement	$\neg R$	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$	$\neg$
Inverse	R^{-}	$\{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}$	-1
Composition	$R \circ S$	$R^{\mathcal{I}} \circ S^{\mathcal{I}}$	$\circ$
Transitive closure	R^{+}	$\bigcup_{n \geq 1} (R^{\mathcal{I}})^n$	$+$
Reflexive-transitive closure	R^{*}	$\bigcup_{n \geq 0} (R^{\mathcal{I}})^n$	$*$
Role restriction	$R\|_C$	$R^{\mathcal{I}} \cap (\Delta^{\mathcal{I}} \times C^{\mathcal{I}})$	r
Identity	$id(C)$	$\{(d,d) \mid d \in C^{\mathcal{I}}\}$	id

the ones from $\mathcal{AL}$, minus atomic negation and limited existential quantification since they are special cases of negation and existential quantification.

Some explanatory remarks are in order. The symbols u_1, u_2 in the agreement constructor stand for chains of functional roles, i.e., $u_1 = f_1 \cdots f_m$ and $u_2 = g_1 \cdots g_n$ where $n, m \geq 0$ and the f_i, g_j are features. The semantics of such a chain is given by the composition of the partial functions interpreting its components, i.e., $u_1^{\mathcal{I}}(a) = f_n^{\mathcal{I}}(\cdots f_1^{\mathcal{I}}(a) \cdots)$. Nominals (or individuals) in concept expression are interpreted as singleton sets, consisting of one element of the domain. We assume that names for individuals come from a name space disjoint from the set of concept and role names. Since role-value-maps cause undecidability and thus are no longer used in DL systems, there is no special symbol for them in the last column of Table A1.1.

Many DL systems employ a Lisp-like concrete syntax. Table A1.2 introduces this syntax and gives a translation into the abstract syntax introduced in Table A1.1.

Role constructors

Role constructors take role and/or concept descriptions and transform them into more complex role descriptions. Table A1.3 shows the syntax and semantics of common role constructors.

The symbol $\circ$ denotes the usual composition of binary relations, i.e.,

$$R^{\mathcal{I}} \circ S^{\mathcal{I}} = \{(a,c) \mid \exists b.\ (a,b) \in R^{\mathcal{I}} \wedge (b,c) \in S^{\mathcal{I}}\}.$$

Table A1.4. *Concrete syntax of role constructors.*

Name	Concrete syntax	Abstract syntax
Universal role	`top`	U
Intersection	(and $\mathtt{R}_1$ $\cdots$ $\mathtt{R}_n$)	$R_1 \sqcap \cdots \sqcap R_n$
Union	(or $\mathtt{R}_1$ $\cdots$ $\mathtt{R}_n$)	$R_1 \sqcup \cdots \sqcup R_n$
Complement	`(not R)`	$\neg R$
Inverse	`(inverse R)`	R^-
Composition	(compose $\mathtt{R}_1$ $\cdots$ $\mathtt{R}_n$)	$R_1 \circ \cdots \circ R_n$
Transitive closure	`(transitive-closure R)`	R^+
Reflexive-transitive closure	`(transitive-reflexive-closure R)`	R^*
Role restriction	`(restrict R C)`	$R\|_C$
Identity	`(identity C)`	$id(C)$

Iterated composition is denoted in the form $(R^\mathcal{I})^n$. To be more precise,

$$(R^\mathcal{I})^0 = \{(d,d) \mid d \in \Delta^\mathcal{I}\} \quad \text{and} \quad (R^\mathcal{I})^{n+1} = (R^\mathcal{I})^n \circ R^\mathcal{I}.$$

Transitive and reflexive-transitive closure are the only constructors among the ones introduced until now that cannot be expressed in first-order predicate logic.

The Lisp-like concrete syntax for role constructors can be found in Table A1.4.

A1.2.2 Knowledge bases

A DL knowledge base usually consists of a set of terminological axioms (often called TBox) and a set of assertional axioms or assertions (often called ABox). Syntax and semantics of these axioms can be found in Table A1.5. An interpretation $\mathcal{I}$ is called a *model* of an axiom if it satisfies the statement in the last column of the table.

An equality whose left-hand side is an atomic concept (role) is called concept (role) *definition.* A finite set of definitions is called a *terminology* or *TBox* if the definitions are unambiguous, i.e., no atomic concept occurs more than once as left-hand side. Axioms of the form $C \sqsubseteq D$ for a complex description C are often called *general inclusion axioms.* A set of axioms of the form $R \sqsubseteq S$ where both R and S are atomic is called *role hierarchy.* Such a hierarchy obviously imposes restrictions on the interpretation of roles. Thus, the fact that the knowledge base may contain a role hierarchy is sometimes indicated by appending a subscript $\mathcal{H}$ to the name of the DL (see "Restrictions on role interpretations" above).

Table A1.5. *Terminological and assertional axioms.*

Name	Syntax	Semantics
Concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Concept equality	$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
Role equality	$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$
Concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Role assertion	$R(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Table A1.6. *Concrete syntax of axioms.*

Name	Concrete syntax	Abstract syntax
Concept definition	`(define-concept A C)`	$A \equiv C$
Primitive concept introduction	`(define-primitive-concept A C)`	$A \sqsubseteq C$
General inclusion axiom	`(implies C D)`	$C \sqsubseteq D$
Role definition	`(define-role R S)`	$R \equiv S$
Primitive role introduction	`(define-primitive-role R S)`	$R \sqsubseteq S$
Concept assertion	`(instance a C)`	C(a)
Role assertion	`(related a b R)`	R(a,b)

The concrete LISP-like syntax distinguishes between terminological axioms with atomic concepts as left-hand sides and the more general ones. Following the convention mentioned at the beginning of this appendix, A denotes an atomic concept. In the table, R is also meant to denote an atomic role.

A1.3 Additional constructors

Here we mention some of the additional constructors that occur somewhere in the handbook. For most of them, the semantics cannot be described in a compact manner, and thus we refer to the respective chapter for details.

A1.3.1 Concept and role constructors

Many additional constructors are introduced in Chapter 6. In DLs with *concrete domains* one can us concrete predicates to constrain fillers of feature chains, similarly to the use of the equality predicate in feature agreements. For example, if hasAge is a feature and $\geq_{18}$ the unary concrete predicate consisting of all nonnegative integers greater than or equal to 18, then $\exists\mathsf{hasAge}.\geq_{18}$ describes the individuals whose age is greater than or equal to 18. In general, an *existential predicate restriction* is of the form

$$\exists(u_1, \cdots, u_n).P,$$

where P is an n-ary predicate of the underlying concrete domain and $u_1, \ldots, u_n$ are feature chains. One can also use concrete domain predicates to define new roles. For example, $\exists(\mathsf{hasAge})(\mathsf{hasAge}).>$ consists of all pairs of individuals having an age such that the first individual is older than the second one. The general form of such a *complex role* is

$$\exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P,$$

where P is an $(n + m)$-ary predicate of the underlying concrete domain and $u_1, \ldots, u_n$, $v_1, \ldots, v_m$ are feature chains.

In *modal extensions* of description logics, one can apply modal operators to concepts and/or roles, i.e., if $\Box$ is such a modal operator, C is a concept, and R is a role, then

$$\Box C \quad \text{and} \quad \Box R$$

is a concept and a role, respectively. Similarly, one can also use diamond operators $\Diamond$ to obtain new concepts and roles. A special such modal operator is the *epistemic operator* $\mathbf{K}$, which can be used to talk about things that are known to the knowledge base.

Chapter 5 introduces several additional constructors. Least and greatest fixpoint semantics for cyclic terminologies (see Chapter 2) can be generalized by introducing *fixpoint constructors* directly into the description language. Let X be a concept name and C a concept description containing the name X. Then

$$\mu X.C \quad \text{and} \quad \nu X.C$$

is a new concept description respectively obtained by applying the least and the greatest fixpoint constructor to C. To ensure that the least and the greatest fixpoint exist, one must restrict C to be syntactically monotonic, i.e., every occurrence of X in C must be in the scope of an even number of complement operators. For example, given an interpretation $\mathsf{Man}^{\mathcal{I}}$ of Man and $\mathsf{hasChild}^{\mathcal{I}}$ of $\mathsf{hasChild}$, the concept $\nu\mathsf{Momo}.(\mathsf{Man} \sqcap \forall\mathsf{hasChild}.\mathsf{Momo})$ looks for the greatest interpretation $\mathsf{Momo}^{\mathcal{I}}$ of Momo such that $\mathsf{Momo}^{\mathcal{I}} = (\mathsf{Man} \sqcap \forall\mathsf{hasChild}.\mathsf{Momo})^{\mathcal{I}}$. It is easy to see that this is the set of all men having only male offspring (see Chapter 2 for the corresponding example with a cyclic TBox).

Chapter 5 also considers the DL $\mathcal{DLR}$, in which the restriction to at most binary predicates is no longer enforced. If $\mathbf{R}$ is an n-ary predicate, $i \in \{1, \ldots, n\}$, and k is a nonnegative integer, then

$$\exists[\$i]\mathbf{R}$$

denotes the concept collecting those individuals that occur as ith component in some tuple of $\mathbf{R}$, and

$$\leq k\,[\$i]\mathbf{R}$$

denotes the concept collecting those individuals d for which the predicate $\mathbf{R}$ contains at most k tuples whose ith component is d. Conversely, if C is a concept, n a nonnegative integer, and $i \in \{1, \ldots, n\}$, then

$$(\$i/n : C)$$

denotes the n-ary predicate consisting of the tuples whose ith component belongs to C. The DL $\mathcal{DLR}$ also allows for Boolean operators on both concepts and predicates.[1]

A1.3.2 Axioms

In addition to the semantics for terminological axioms introduced above, Chapter 2 also considers fixpoint semantics for cyclic TBoxes.

Chapter 6 introduces several ways of extending the terminological and the assertional component of a DL system. In DLs with *concrete domains* one can use concrete predicates also in the ABox in assertions of the form

$$P(x_1, \ldots, x_n),$$

where P is an n-ary predicate of the underlying concrete domain and $x_1, \ldots, x_n$ are names for concrete individuals.

In some *modal extensions* of description logics, one can apply modal and Boolean operators also to terminological and assertional axioms: if φ, ψ are axioms, then so are

$$\varphi \wedge \psi, \quad \neg\varphi, \quad \Box\varphi.$$

In *probabilistic extensions* of description logics, one can use probabilistic terminological axioms of the form

$$\mathrm{P}(C|D) = p,$$

which state that the conditional probability for an object known to be in D to belong to C is p.

[1] Note, however, that negation on predicates has a non-standard semantics (see Chapter 5 for details).

The integration of Reiter's default logic into DLs yields *terminological defaults* of the form

$$\frac{C(x) : D(x)}{E(x)},$$

where C, D, E are concept descriptions (viewed as first-order formulae with one free variable x). Intuitively, such a default rule can be applied to an ABox individual a, i.e., $E(a)$ is added to the current set of beliefs, if its prerequisite $C(a)$ is already believed for this individual and its justification $D(a)$ is consistent with the set of beliefs.

Rules of the form

$$C \Rightarrow E$$

(as introduced in Chapter 2) can be seen as a special case of terminological defaults where the justification is empty. Their intuitive meaning is: "if an individual is known to be an instance of C, then add the information that it is also an instance of E."

A1.4 A note on the naming scheme for Description Logics

In Section A1.2 we have introduced a naming scheme for DLs, which extends the naming scheme for the $\mathcal{AL}$-family introduced in Chapter 2 by writing letters/symbols for role constructors as superscripts, and for restrictions on the interpretation of roles as subscripts. The reason was that this yields a consistent naming scheme, which distinguishes typographically between the three different possibilities for extending the expressive power of $\mathcal{AL}$.

In the literature, and also in this handbook, other naming schemes are employed as well. One reason for this, in addition to the fact that such schemes have evolved over time, is that it is very hard to pronounce a name like $\mathcal{ALCQ}^{-1}_{R^+}$. We will here point out the most prominent such naming schemes.

The historically first scheme is the one for the $\mathcal{AL}$-family introduced in Chapter 2, and extended in this appendix. However, in the literature the typographical distinction between role constructors, concept constructors, and restrictions on the interpretation of roles is usually not made. For example, many papers use $\mathcal{I}$ to denote inverse of roles, $\mathcal{R}$ to denote intersection of roles, and $\mathcal{H}$ to denote role hierarchies. Thus, $\mathcal{ALCRI}$ denotes the extension of $\mathcal{ALC}$ by intersection and inverse of roles, and $\mathcal{ALCH}$ denotes the extension of $\mathcal{ALC}$ by role hierarchies. In some cases, the letter $\mathcal{F}$, which we employed to express the presence of feature agreements and disagreements, is used with a different meaning. Its presence states that number restrictions of the form $\leqslant 1\, R$ can be used to express functionality of roles.[1] The

[1] Unlike the restriction of R to be functional, which we express with a subscript f, this allows for *local*

subscript "trans" (or "reg") is often employed to express the presence of union, composition, and transitive closure of roles (sometimes also including the identity role). The Greek letter μ in front of a language name, like in $\mu\mathcal{ALC}$, usually indicates the extension of this DL by fixpoint operators.

All members of the $\mathcal{AL}$-family include $\mathcal{AL}$ as a sublanguage. In some cases on does not want all the constructors of $\mathcal{AL}$ to be present in the language. The DL $\mathcal{FL}^-$ is obtained from $\mathcal{AL}$ by disallowing atomic negation, and $\mathcal{FL}_0$ is obtained from $\mathcal{FL}^-$ by, additionally, disallowing limited existential quantification. If these languages are extended by other constructors, one can indicate this in a way analogous to extensions of $\mathcal{AL}$. For example, $\mathcal{FL}^-\mathcal{U}$ denotes the extension of $\mathcal{FL}^-$ by union of concepts.

All the DLs mentioned until now contain the concept constructors intersection and value restriction as a common core. DLs that allow for intersection of concepts and existential quantification (but not value restriction) are collected in the $\mathcal{EL}$-family. The only constructors available in $\mathcal{EL}$ are intersection of concepts and existential quantification. Extensions of $\mathcal{EL}$ are again obtained by adding appropriate letters/symbols.

In order to avoid very long names for expressive DLs, the abbreviation $\mathcal{S}$ was introduced for $\mathcal{ALC}_{R^+}$, i.e., the DL that extends $\mathcal{ALC}$ by transitive roles. Prominent members of the $\mathcal{S}$-family are $\mathcal{SIN}$ (which extends $\mathcal{ALC}_{R^+}$ with number restrictions and inverse roles), $\mathcal{SHIF}$ (which extends $\mathcal{ALC}_{R^+}$ with role hierarchies, inverse roles, and number restrictions of the form $\leqslant 1\, R$), and $\mathcal{SHIQ}$ (which extends $\mathcal{ALC}_{R^+}$ with role hierarchies, inverse roles, and qualified number restrictions). Actually, the DLs $\mathcal{SIN}$, $\mathcal{SHIF}$, and $\mathcal{SHIQ}$ are somewhat less expressive than indicated by their name since the use of roles in number restrictions is restricted: roles that have a transitive subrole must not occur in number restrictions.

The DL $\mathcal{DLR}$ mentioned in the previous section also gives rise to a family of DLs, with members like $\mathcal{DLR}_{reg}$, which extends $\mathcal{DLR}$ with union, composition, and transitive closure of binary relations obtained as projections of n-ary predicates onto two of their components.

functionality statements, i.e., R is functional at a certain place, but may be non-functional at other places.