2 Preliminaries

This chapter introduces topics that are relevant in the course of this thesis; the presentations are hereby restricted to the aspects required in the remainder of this work and not intended to be complete. Where needed to simplify presentation later on, we also introduce new terminology and prove related lemmas to simplify proofs later on. Section 2.1 gives a short overview on ontologies and their relation to description logics as well as classical reasoning tasks for DLs. Section 2.2 complements this account with an introduction to abductive reasoning both from a general perspective and in the context of description logics. Alternative definitions of abduction from literature are introduced, and their relation to the definition given here is formally analysed. As solving certain classes of relaxed abduction problems will be reduced to finding optimal hyperpaths later on, a number of relevant order-theoretic notions is introduced in Section 2.3; a basic account on hypergraphs, an extension of standard graphs, can be found in Section 2.4.

2.1 Ontologies and Description Logics

Knowledge about the entities of a domain and their numerous interrelations can conveniently be captured in an explicit and intelligible way by means of an *ontology*. Representing domains by means of ontologies provides a deeper understanding of the domain as compared to subsymbolic approaches or other representations acquired by learning that often do not support the representation of relational information. This makes it possible for domain experts to gradually extend and adapt the formalisation to integrate their knowledge. Nowadays, the method of choice for representing ontologies is the *Web Ontology Language (OWL)* which is semantically founded on the description logic (DL) formalism described in Nardi et al. (2003).

Description logics are a family of formal knowledge representation formalisms and can be understood as fragments of first-order logic restricted to at most binary predicates. The main representational elements are *concepts* (unary predicates), *roles* (binary predicates), and *individuals* (terms). Description logics have model-theoretic semantics defined by means of an

T. Hubauer, *Relaxed Abduction*,
DOI 10.1007/978-3-658-14407-4_2

interpretation function $\cdot^{\mathcal{I}}$ that maps individuals to elements of the domain Δ, concepts to sets of elements, and roles to sets of element pairs. The entailment relation between a set X of DL axioms and a single DL axiom x, denoted $X \models x$, is defined as usual. For two sets of axioms X, Y, we let $X \models Y$ stand for $X \models \bigwedge Y$ or equivalently: $X \models y$ for all $y \in Y$. $X \mathrel{=\joinrel|} Y$ stands for $Y \models X$, and $X \equiv Y$ is an abbreviation for $X \mathrel{=\joinrel|} Y$ and $X \models Y$. Later in this thesis, we need the intuition of an axiom set not containing any redundant axioms. To capture this concept, we define the notion of a *logically independent set of axioms* as follows:

Definition 2.1 (Logically independent axiom set)
Let X be a set of axioms with subsets S and S'. Then X is logically independent *if and only if $S' \models S$ implies $S \subseteq S'$.*

In other words, an axiom set X is logically independent if *no* subset $S \subseteq X$ is redundant in the sense specified by Grimm & Wissmann (2011).

A concrete description logic is characterized by the constructors available for forming complex concepts from simple ones, and types of *axioms* that can be stated. This choice typically constitutes a tradeoff between expressiveness of the resulting language and computational complexity of deciding satisfiability of a set of axioms called a *knowledge base* (KB) or *ontology*. This task also known as *consistency checking* is one of the central reasoning tasks in description logics, its decidability is a property shared by all DLs. There is a number of other *standard inference tasks* which can be solved by reduction to consistency checking for most description logics[1]: *subsumption checking* (determine whether every instance of concept C must necessarily be an instance of concept D as well, denoted $\mathsf{C} \sqsubseteq \mathsf{D}$), *classification* (determine all subsumption relations between concepts of the knowledge base)[2], *instance checking* (determine whether individual i is an instance of a concept C, denoted $\mathsf{C}(i)$ or $i : \mathsf{C}$), and *realization* (for every individual of the knowledge base determine the most specific concept names it is an instance of).

[1] Basically, reducibility for concept-oriented reasoning tasks rests on the capability of expressing disjointness of concepts, whereas reducibility of individual-oriented tasks also requires the existence of nominals, or enumerated concepts.

[2] It should be pointed out that this definition coming from the community around lightweight description logics differs from the standard definition used in most other parts of the DL community, where classification determines only the most specific subsumption relations between any pair of concepts. The difference can be understood as an additional filter step on the full set of all subsumptions, retaining only the most specific ones. Therefore, classification according to the standard definition is typically somewhat more expensive from a computational perspective.

In its second version OWL 2 (W3C OWL Working Group, 2009a), the Web Ontology Language is largely based on the highly expressive description logic $\mathcal{SROIQ}$ (Horrocks et al., 2006). $\mathcal{SROIQ}$ provides many powerful features including enumerated classes $\{i_1, \ldots, i_k\}$, full negation $\neg\mathsf{C}$, various qualified restrictions, role hierarchies, and inverse roles. This however leads to high computational cost of the reasoning tasks introduced before. Deciding concept satisfiability was shown to be N2ExpTime-complete in Kazakov (2008). Since the doubly exponential complexity of $\mathcal{SROIQ}$ may be prohibitive for certain applications, OWL 2 comes with a number of so-called profiles which offer language variants with reduced expressiveness and better computational properties. One of these profiles is OWL 2 EL, which has been shown to be especially well suited for the representation of large taxonomic structures such as the well-known SNOMED-CT[3] and GALEN[4] medical ontologies. The EL profile is based on the description logic $\mathcal{EL}^{++}$ (Baader et al., 2005a), for which consistency is decidable in PTime. This significant reduction of complexity is made possible by the absence of a number of constructors such as universal quantification, full negation and disjointness axioms for concepts and roles, along with a number of other constructs that could be used to express one of the previous. It has been shown by Baader et al. that $\mathcal{EL}^{++}$ is maximal in the sense that none of these dropped features can be included without sacrificing tractability. Despite these syntactic restrictions, $\mathcal{EL}^{++}$ supports expressing information about individuals by means of nominals (concepts comprised of exactly one named individual), which allows to reduce reasoning over individuals to standard concept-level reasoning. The same complexity bounds hold for the sublanguage $\mathcal{EL}^{+}$ which has been proved sufficient for the representation of diagnostic ontologies in various contexts (see e. g. Baader et al. (2007)). An $\mathcal{EL}^{+}$ ontology $\mathcal{T}$ is a set of concept inclusion axioms of the form $\mathsf{Turbine} \sqcap \exists\, \mathsf{propelledBy} \,.\, \mathsf{Gas} \sqsubseteq \mathsf{GasTurbine}$ (which states that a gas turbine is a turbine propelled by gas) and role inclusion axioms of the form $\mathsf{directlyControls} \circ \mathsf{hasSubComponent} \sqsubseteq \mathsf{controls}$ (stating that control is propagated over the partonomy of a system). Table 2.1 summarizes the syntax and semantics of the concept constructors and the axioms that can be represented in $\mathcal{EL}^{+}$ (where A is an atomic concept, C, D are arbitrary concepts, and $\mathsf{r}_i, \mathsf{r}, \mathsf{s}$ are roles). For details on the description logic constructors we refer the reader to Nardi et al. (2003); W3C OWL Working Group (2009b).

For making explicit that an axiom *ax* is expressed in the description logic

[3] `http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html`
[4] `http://www.openclinical.org/prj_galen.html`

Table 2.1: $\mathcal{EL}^+$ syntax and semantics

	Syntax	Semantics
top concept	$\top$	$\Delta^\mathcal{I}$
atomic concept	A	$\mathsf{A}^\mathcal{I}$
concept conjunction	$\mathsf{C} \sqcap \mathsf{D}$	$\mathsf{C}^\mathcal{I} \cap \mathsf{D}^\mathcal{I}$
existential restriction	$\exists\, \mathsf{r}\,.\, \mathsf{C}$	$\{x \in \Delta^\mathcal{I} \mid \exists y \in \Delta^\mathcal{I} :$ $(x, y) \in \mathsf{r}^\mathcal{I} \wedge y \in \mathsf{C}^\mathcal{I}\}$
concept inclusion axiom	$\mathsf{C} \sqsubseteq \mathsf{D}$	$\mathsf{C}^\mathcal{I} \subseteq \mathsf{D}^\mathcal{I}$
role inclusion axiom	$\mathsf{r}_1 \circ \cdots \circ \mathsf{r}_n \sqsubseteq \mathsf{s}$	$\mathsf{r}_1^\mathcal{I} \circ \cdots \circ \mathsf{r}_n^\mathcal{I} \subseteq \mathsf{s}^\mathcal{I}$

L we sometimes call ax an *L-axiom*. Analogously, a *L-knowledge base* is a set of L-axioms. Given a description logic knowledge base $\mathcal{T}$, we denote the set of all role names occurring in $\mathcal{T}$ by N_R, the set of all concept names by N_C, and define $N_\mathrm{C}^\top = N_\mathrm{C} \cup \{\top\}$.

2.2 Abductive Reasoning

This section provides an overview on abductive reasoning. We first give a general introduction on the framework of (axiom-based) abduction and its instantiations. Next, we relate this definition to a relatively new formulation called concept abduction and show how the former subsumes the latter. The section concludes with an analysis of the shortcomings of logic-based abduction with respect to the processing of deficient domain representations, one of the main challenges addressed in this thesis.

2.2.1 Standard Notions of Abduction

Abduction was introduced in philosophy of logic in the late 19th century by Charles Sanders Pierce, who characterized it as the sole reasoning method generating new information (Hartshorne & Weiss, 1931). It was rediscovered as a method for determining diagnoses in the 1970s by Pople (Pople, 1973), and has since then been employed as a method for interpreting incomplete information in various applications such as text interpretation (Hobbs et al., 1993), plan generation and analysis (Appelt & Pollack, 1992), and analysis of sensor (Shanahan, 2005) or multimedia data (Möller & Neumann, 2008; Peraldi et al., 2007). In the spirit of Pierce, abduction can be seen as an inference scheme capable of providing possible explanations for some observation. It is conveniently represented by the rule

$$\frac{\phi \supset \omega \qquad \omega'}{\phi\Theta}$$

and interpreted as an inversion of the well-known generalised modus ponens rule. This new derivation rule allows to derive $\phi\Theta$ as a hypothetical explanation for the occurrence of ω', given that the presence of ϕ in some sense justifies ω (where Θ is a unifier[5] for ω and ω'). Such explanations need not be unique, and while each single explanation is typically required to fulfil some consistency requirement, different competing explanations may well be contradictory to each other. Hypotheses can be falsified by adding information (e. g. representing additional observations), making abduction an inherently non-monotonic reasoning scheme.

Note that this general notion of abduction does not presuppose any causality between ϕ and ω, as indicated by the notation $\phi \supset \omega$ of the *major premise*. On the contrary, various notions of how ϕ sanctions the presence of ω give rise to different definitions of abduction as explicated in detail in Paul (1993):

- *Set-cover abduction* implements the major hypothesis $\phi \supset \omega$ by means of an explicit mapping $e : \Phi \mapsto \Omega$ called *explanatory power* that relates sets of hypotheses to the sets of observations they explain. Determining abductive solutions can then be reduced to finding minimum covers of the observation set, based on the explanatory power. For realistic examples however, the mapping e is prone to get huge and it is unlikely that it can be defined correctly by hand, let alone updated on changes in the domain.
- *Logic-based abduction* can be understood to replace the explicit mapping used in the set-cover approach by a knowledge base $\mathcal{T}$, making the full power of the formal language available for representing expressive yet concise mappings. In addition to the domain formalisation, a set $\mathcal{A}$ of abducible axioms is provided. A logic-based abduction problem is solved by determining subsets of $\mathcal{A}$ which are consistent with the axiom set $\mathcal{T}$ and explain the observation(s) given $\mathcal{T}$. Hypothesis construction for logic-based abduction is typically more involved than in the set-cover variant since the mapping e is only known implicitly.

[5] Given terms $\omega_1, \ldots, \omega_n$, a *substitution* is a mapping from variables to terms which can be understood as a generalised type of renaming, replacing (some of the) variables with potentially complex terms comprising constants, variables, and functions. A unifier Θ for $\omega_1, \ldots, \omega_n$ is a special substitution such that $\omega_1\Theta = \ldots = \omega_n\Theta$.

- *Knowledge-level abduction* generalizes logic-based abduction with a conceptual model of belief. In a nutshell, ϕ explains ω w. r. t. an epistemic state if (in this state) it is believed that $\phi \rightarrow \omega$, and the negation of ϕ is not believed. This approach is claimed to be relevant e. g. for situated agents where the epistemic state might change regularly due to external factors.

Obviously, set-cover abduction will typically be too weak to meet Requirement **R1**. As representation of and reasoning over belief is not relevant for the application scenarios at hand, we focus on logic-based reasoning over L-knowledge bases (for decidable logics L), i. e. we understand abduction as a *non-standard inference task* over formal, logic-based domain representations. In this context, an abduction problem can be defined as follows (c.f. Eiter & Gottlob, 1995):

Definition 2.2 (Abduction problem)
An abduction problem *is a 3-tuple* $\mathbf{AP} = (\mathcal{T}, \mathcal{A}, \mathcal{O})$*, where*

- $\mathcal{T}$ *is a set of axioms that formalise the domain,*
- $\mathcal{A}$ *is a set of abducible axioms representing assumptions, and*
- $\mathcal{O}$ *is a set of axioms representing observations.*

A solution *to* $\mathbf{AP}$ *is a set* $A \subseteq \mathcal{A}$ *such that* $\mathcal{T} \cup A$ *is consistent and* $\mathcal{T} \cup A \models \mathcal{O}$*. Naturally extending the standard notation, we write* $A \models \mathbf{AP}$ *to state that* A solves $\mathbf{AP}$*. The* solution set *for* $\mathbf{AP}$ *is defined by* $Sol_{\mathbf{AP}} := \{A \mid A \models \mathbf{AP}\}$*.*

It should be pointed out that abduction does not at all require that every conclusion of $\mathcal{T} \cup A$ has been observed (i. e. is an element of $\mathcal{O}$). This separates abduction from mere backward-chaining rule-based deduction, and enables it to flexibly address situations where the observation set is incomplete.[6]

Due to the fact that an abduction problem does in general not have one unique solution A but a collection of alternative answers $A_1, \ldots, A_k$, one typically selects optimal solutions by means of a (not necessarily total) preference order $\preceq$, with $A_i \preceq A_j$ expressing that A_i is at least as good as A_j (see Section 2.3 for an introduction to orders). Abductive inference can then conveniently be formulated as an optimization problem as shown next:

[6] Backward-chaining rule-based reasoning can obviously emulate this feature by breaking down rules with multiple conjuncts in the precedent. However, this typically leads to a significant explosion of the number of rules to consider.

Definition 2.3 (Preferential abduction problem)
A preferential abduction problem *is a 4-tuple* **PrefAP** $= (\mathcal{T}, \mathcal{A}, \mathcal{O}, \preceq_{\mathcal{A}})$*, where*

- $\mathcal{T}$ *is a set of axioms that formalise the domain,*
- $\mathcal{A}$ *is a set of abducible axioms representing assumptions,*
- $\mathcal{O}$ *is a set of axioms representing observations, and*
- $\preceq_{\mathcal{A}} \subseteq \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{A})$ *is a partial order over sets of assumptions.*

A pre-solution *to* **PrefAP** *is a set* $A \subseteq \mathcal{A}$ *such that* $\mathcal{T} \cup A$ *is consistent and* $\mathcal{T} \cup A \models \mathcal{O}$*. A* solution *is an* $\preceq_{\mathcal{A}}$*-minimal element of the set of pre-solutions. We write* $A \models$ **PrefAP** *to state that* A solves **PrefAP**. *The* solution set *for* **PrefAP** *is defined by* $Sol_{\mathbf{PrefAP}} := \{A \mid A \models \mathbf{PrefAP}\}$.

Typical orders over sets include subset minimality ($A_i \preceq^{\mathrm{s}} A_j$ if and only if $A_i \subseteq A_j$), minimum cardinality ($A_i \preceq^{\mathrm{c}} A_j$ if and only if $|A_i| \leq |A_j|$), and weight-based approaches defined by a function w that assigns numerical weights to subsets of $\mathcal{A}$ (i. e. $A_i \preceq^{\mathrm{w}} A_j$ if and only if $w(A_i) \leq w(A_j)$).

2.2.2 Concept-based Notions of Abduction

A different formulation of abduction due to Bienvenu (2008) employs atomic concepts for representing the abducibles and the observation. Note that we use the term *subsumption-based abduction* for this approach instead of the original name *concept-based abduction*, in order to clearly distinguish it from *concept abduction* introduced later. For similar reasons, we will from now on use the term axiom-based abduction for the standard definition (Definition 2.2). After introducing subsumption-based abduction in Definition 2.4, we show in Lemma 2.1 that this notion is less general than axiom abduction.

Definition 2.4 (Subsumption-based abduction problem)
A subsumption-based abduction problem *is a 3-tuple* **SAP** $= (\mathcal{T}, \mathcal{H}, \mathsf{O})$*, where*

- $\mathcal{T}$ *is a set of axioms that formalise the domain,*
- $\mathcal{H}$ *is a set of atomic concepts (each representing a possible assumption), and*
- O *is an atomic concept representing the observation.*

A solution *to* **SAP** *is a set* $H \subseteq \mathcal{H}$ *such that* $\bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i$ *is consistent w. r. t.* $\mathcal{T}$ *and* $\mathcal{T} \models \bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i \sqsubseteq \mathsf{O}$.

Lemma 2.1 (Axiom-based can simulate subsumption-based abd.) *A subsumption-based abduction problem* $\mathbf{SAP} = (\mathcal{T}, \mathcal{H}, \mathsf{O})$ *can be solved by determining solutions to the corresponding axiom-based abduction problem* $\mathbf{AP} = (\mathcal{T}, \mathcal{A}, \mathcal{O})$ *defined by:*

- $\mathcal{A} = \{\mathsf{C}^* \sqsubseteq \mathsf{C}_i \mid \mathsf{C}_i \in \mathcal{H}\}$,
- $\mathcal{O} = \{\mathsf{C}^* \sqsubseteq \mathsf{O}\}$, *and*
- C^* *is a new concept name,* $\mathsf{C}^* \notin N_\mathrm{C}$

More concretely, $A = \{\mathsf{C}^* \sqsubseteq \mathsf{C}_i\}$ *solves* $\mathbf{AP}$ *if* $H = \{\mathsf{C}_i \mid \mathsf{C}^* \sqsubseteq \mathsf{C}_i \in A\}$ *solves* $\mathbf{SAP}$.

Proof. Let H solve $\mathbf{SAP}$, i. e. $\mathcal{T} \models \bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i \sqsubseteq \mathsf{O}$. Then $\mathcal{T} \equiv \mathcal{T} \cup \{\bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i \sqsubseteq \mathsf{O}\}$, and obviously $\mathcal{T} \cup \{\bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i \sqsubseteq \mathsf{O}\} \cup \{\mathsf{C}^* \sqsubseteq \mathsf{C}_i \mid \mathsf{C}_i \in H\} \models \mathsf{C}^* \sqsubseteq \mathsf{O}$. Thus $\mathcal{T} \cup A \models \mathcal{O}$, A therefore solves $\mathbf{AP}$. □

In Colucci et al. (2003) the similar term *concept abduction* is used for a slightly different task: Given a L-knowledge base $\mathcal{T}$ and two L-concepts C and D that are satisfiable in $\mathcal{T}$, determine a L-concept H such that $\mathcal{T} \not\models \mathsf{C} \sqcap \mathsf{H} \sqsubseteq \bot$ and $\mathcal{T} \models \mathsf{C} \sqcap \mathsf{H} \sqsubseteq \mathsf{D}$. This problem can be reduced to subsumption-based abduction as shown in the following lemma:

Lemma 2.2 (Subsumption-based can simulate concept abd.) *Let* $(\mathcal{T}, \mathsf{C}, \mathsf{D})$ *be a concept abduction problem over language L as defined by Colucci et al. (2003). Then, all solutions* H *to* $(\mathcal{T}, \mathsf{C}, \mathsf{D})$ *can be determined using subsumption-based abduction as defined in Definition 2.4.*

Proof. Define a subsumption-based abduction problem $\mathbf{SAP}$ for $(\mathcal{T}, \mathsf{C}, \mathsf{D})$ by $\mathbf{SAP}_{(\mathcal{T},\mathsf{C},\mathsf{D})} = (\mathcal{T}, \{\mathsf{C}\} \cup L^*(d), \mathsf{D})$, where $L^*(d)$ denotes the set of all semantically distinct L-concepts of role depth $\leq d$, with d defined as the maximum role depth occurring in C, D, or any axiom of $\mathcal{T}$. Further, let $Sol_{\mathbf{SAP}_{(\mathcal{T},\mathsf{C},\mathsf{D})}}$ be the solution set for the induced subsumption-based abduction problem, and $Sol_{\mathbf{SAP}_{(\mathcal{T},\mathsf{C},\mathsf{D})}|\mathbf{C}}$ its restrictions to subsets of $\mathcal{H}$ that do contain C.
Assume that H solves $\mathbf{SAP}_{(\mathcal{T},\mathsf{C},\mathsf{D})}$, and $H \in Sol_{\mathbf{SAP}_{(\mathcal{T},\mathsf{C},\mathsf{D})}|C}$. Then $H \subseteq \mathcal{H}$, $\bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i$ is consistent w. r. t. $\mathcal{T}$ (i. e. $\mathcal{T} \not\models \bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i \sqsubseteq \bot$), and $\mathcal{T} \models \bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i \sqsubseteq \mathsf{D}$. Now define $\mathsf{C}_\mathsf{H} = \bigsqcap_{\mathsf{C}_i \in H} \mathsf{C}_i$. As $\mathsf{C} \in H$ by construction, it is also true that $\mathcal{T} \not\models \mathsf{C} \sqcap \mathsf{C}_\mathsf{H} \sqsubseteq \bot$ and $\mathcal{T} \models \mathsf{C} \sqcap \mathsf{C}_\mathsf{H} \sqsubseteq \mathsf{D}$, i. e. H solves the original concept abduction problem $(\mathcal{T}, \mathsf{C}, \mathsf{D})$. □

The notion of abduction introduced in Definition 2.2 is therefore non-trivial and practically relevant, making it a good basis for our extension presented in Chapter 3.

2.2.3 A Critique of Logic-Based Abduction

A joint property of all formalizations of abduction introduced so far (independently of the concrete logic L used) is their requirement that a solution must completely explain the observation(s): There is no way of having a solution "explain the largest part of $\mathcal{O}$" or "make C almost subsume D". Typical application fields, such as media analysis (Castano et al., 2009) and industrial diagnostics (Hubauer et al., 2011b), are however characterized by an abundance of low-level observations due to a large number of sensors, whereas the knowledge base formalising the domain is often rough or incomplete since it must be created manually. As the following example shows, existing definitions of abduction are not sufficient in this context:

Example 2.1 (Diagnostics over incomplete domain formalisations) *The simple production system introduced before consists of a main control unit (MCU), a mechanical gripper and a conveyor which are both part of the transportation subsystem, and a PROFINET communication link to other systems of the factory. The system and its structure can be expressed using $\mathcal{EL}^+$ axioms as follows:*

$$\begin{aligned}
\mathsf{MCU} &\sqsubseteq \exists\,\mathsf{partOf}\,.\,\mathsf{System}\\
\mathsf{Communications} &\sqsubseteq \exists\,\mathsf{subsystemOf}\,.\,\mathsf{System}\\
\mathsf{Transportation} &\sqsubseteq \exists\,\mathsf{subsystemOf}\,.\,\mathsf{System}\\
\mathsf{PROFINET} &\sqsubseteq \exists\,\mathsf{belongsTo}\,.\,\mathsf{Communications}\\
\mathsf{Gripper} &\sqsubseteq \exists\,\mathsf{belongsTo}\,.\,\mathsf{Transportation}\\
\mathsf{Conveyor} &\sqsubseteq \exists\,\mathsf{belongsTo}\,.\,\mathsf{Transportation}\\
\mathsf{belongsTo} \circ \mathsf{subsystemOf} &\sqsubseteq \mathsf{partOf}
\end{aligned}$$

The diagnostic expert knowledge of the system laid out before informally in Example 1.1 can then be formalized with the following set $\mathcal{T}$ of axioms:

$$\begin{aligned}
\mathsf{MCU} &\sqcap \exists\,\mathsf{partOf}\,.\,(\exists\,\mathsf{operatesIn}\,.\,\mathsf{PowerSupplyFluctuations})\\
&\qquad\qquad \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{IntermittentOutages}\\
\mathsf{PROFINET} &\sqcap \exists\,\mathsf{partOf}\,.\,(\exists\,\mathsf{operatesIn}\,.\,\mathsf{PowerSupplyFluctuations})\\
&\qquad\qquad \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{SendingReceivingOK}\\
\mathsf{Gripper} &\sqcap \exists\,\mathsf{partOf}\,.\,(\exists\,\mathsf{operatesIn}\,.\,\mathsf{PowerSupplyFluctuations})\\
&\qquad\qquad \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{FullyFunctional}\\
\mathsf{MCU} &\sqcap \exists\,\mathsf{partOf}\,.\,(\exists\,\mathsf{operatesIn}\,.\,\mathsf{ControlSWMalfunction})\\
&\qquad\qquad \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{IntermittentOutages}
\end{aligned}$$

$$\begin{aligned}\mathsf{Conveyor} &\sqcap \exists\,\mathsf{partOf}\,.\,(\exists\,\mathsf{operatesIn}\,.\,\mathsf{ControlSWMalfunction})\\ &\qquad\qquad \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{IrregularMovements}\\ \mathsf{Gripper} &\sqcap \exists\,\mathsf{partOf}\,.\,(\exists\,\mathsf{operatesIn}\,.\,\mathsf{ControlSWMalfunction})\\ &\qquad\qquad \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{IrregularMovements}\end{aligned}$$

The first axiom, for instance, expresses that a MCU (motor control unit) component being part of some (unspecified) larger system having a fluctuating power supply will be affected by showing intermittent outages. Similarly, the last axiom states that the mechanical gripper of a system affected by a control software malfunction will show irregularities in its movement patterns. We will return on the topic of how to model diagnostic knowledge later on in Section 3.1.

Let us now assume that based on the available sensor measurements, intermittent outages of the main control unit have been confirmed, and motion sensors signal flawless action of the gripper. Due to a general network problem throughout the factory, however, the working state of the PROFINET component cannot be asserted. These observations can be represented by the axiom set $\mathcal{O} = \{\mathsf{MCU} \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{IntermittentOutages}, \mathsf{Gripper} \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{FullyFunctional}\}$*. In this situation, assuming fluctuations in the power supply as expressed by the assumption set* $A_1 = \{\mathsf{System} \sqsubseteq \exists\,\mathsf{operatesIn}\,.\,\mathsf{PowerSupplyFluctuations}\}$ *yields a valid solution (that is* $\mathcal{T} \cup \{A_1\} \models \mathcal{O}$*, the observations are logically entailed by the assumed diagnosis, given the knowledge base* $\mathcal{T}$*). Assuming a software malfunction, on the contrary, (i. e.* $A_2 = \{\mathsf{System} \sqsubseteq \exists\,\mathsf{operatesIn}\,.\,\mathsf{ControlSWMalfunction}\}$*) would not be considered a solution as it cannot account for the observation regarding the PROFINET system. Similarly, the explanation candidate assuming that both defects occur at the same time (*$A_3 = A_1 \cup A_2$*) would typically be rejected since it does not provide any additional expressive power over* A_1 *alone.*

Extending on this situation, assume that a new vibration sensor is added to the system. The new sensor emits a signal whenever it detects vibrations that exceed a predefined threshold. It is not atypical in an application context that the diagnostic knowledge base is not updated instantly, possible reasons ranging from lack of information on how different faults affect vibrations to mundane lack of time. Continuing with the example, we now assume that the diagnostic unit is signalled the presence of significant low-frequency vibrations in addition to the previously discussed symptoms. This results in the extended observation set $\mathcal{O}' = \mathcal{O} \cup \{\mathsf{MCU} \sqsubseteq \exists\,\mathsf{shows}\,.\,\mathsf{LowFrequencyVibrations}\}$*). As motivated before, naïve approaches to diagnostic or abductive reasoning would now try to explain the full observation set* $\mathcal{O}$*'. Consequently, this would lead to* *the invalidation of the previously determined solution A_1, as it cannot explain the observed vibrations (which are not even part of the diagnostic model). Moreover, unless the knowledge bases are updated accordingly, the system may fail to find any solution at all. This illustrates the counterintuitive effect that an additional (and potentially completely unrelated) piece of information may cause the diagnostic problem to fail completely.*[7]

This behaviour severely hinders the practical applicability of logic-based abduction to real-world industrial applications, where an ever-growing amount of sensor data almost inevitably generates pieces of information that the current representation of the domain knowledge cannot account for. We therefore suggest that the classic definition of logic-based abduction is too strict for information-intensive applications, leading to overly complex diagnoses or even the failure to produce any solution at all in the presence of such spurious observations. This is justified by revisiting the requirements defined in Section 1.2, where logic-based abduction alone fails to meet Requirements **R3** and **R5**. For very simple knowledge bases, a remedy could be to identify and remove problematic observations in a preprocessing step, resulting in an approach that meets Requirements **R1** to **R3**, and the restriction of Requirement **R4** to classical solutions (but nevertheless fails w. r. t. Requirement **R5**). This is however not feasible for reasonably complex formalisations since the (ir-)relevance of a piece of information depends on the analysis result and is thus not known beforehand.

A more general solution can be found in the definition of Scott-entailment relations (Scott, 1974) and derived approaches such as the notion of causal abduction presented by Bochman (2007), where an axiom set is considered as an explanation for a set of observations if it entails at least one of the observations[8]. However, the very general framework laid out by Bochman does not define an order over the set of observations, and thus permits no detailed ranking over the typically large set of possible explanations. More recently, Kaya (2011); Peraldi (2011) proposed a different approach

[7] As can easily be seen, even this simple scenario makes use of relational structures both for describing the partonomy of the system and for defining how faults in the system affect its observable state. While for a specific case it would obviously be possible as well to use a simple propositional logic knowledge base to represent the same information, the availability of relations and existential quantification makes the representation much more concise and intelligible. This is a vital factor to ensure that experts can work efficiently with the domain model and update it as needed without getting lost in hundreds of propositional rules.

[8] More formally, Scott-entailment between two axiom sets X and Y holds if X entails some subset of Y, i. e. $Cn(X) \cup Y \neq \emptyset$ with $Cn(X)$ denoting the deductive closure of the axiom set X.

to address the problem of inexplicable observations in an automated way: any observation that cannot be explained from the formal representation is simply assumed to be true. Although this solution has been shown to provide reasonable results e. g. in the context of media interpretation, the equalisation of assumptions and observations leads to a loss of information and subsequently to the loss of other potentially relevant solutions). Extending on this research, Nafissi (2013) employs a beam-search approach to control the generation of solution candidates in an abductive Markov Logic setup. However, while the authors can significantly reduce the size of solution space that is explored using a heuristic approach, they do not explicitly address the trade-off between expressiveness and simplicity of a single solution (instead, they use one-dimensional quality score). The variant of logic-based abduction proposed in this thesis can be understood as a natural continuation of this line of research: It provides robustness w. r. t. deficient domain representations, but does neither require a tradeoff between expressiveness and simplicity, nor assume their separability. We present our solution in Chapter 3, but beforehand we introduce basics from order and hypergraph theory required for a clear definition of both the problem and our solution to it.

We conclude this section with a short explanation why we represent observations by means of terminological axioms in Example 2.1. Firstly, note that this representation is not required by the framework itself, but only for a specific instantiation for the description logic $\mathcal{EL}^+$ which we will introduce in Section 3.3.2 (a restriction which can however be lifted by the introduction of nominals or enumerated classes, as explicated in Section 3.4.1). Yet, to keep the running example consistent throughout this thesis, we use a terminology-based notation for observations right from the start.

2.3 Orders

By a *preorder* $\preceq$ on some set X we understand as usual a binary relation $\preceq \subseteq X \times X$ that is reflexive, i. e., for all $x \in X : x \preceq x$, and transitive, i. e., $\forall x, y, z \in X$: if $(x \preceq y)$ and $(y \preceq z)$, then $(x \preceq z)$. If we want to make explicit the set X over which a relation $\preceq$ is defined, we write $\preceq_X$. Two additional important orders can be derived from $\preceq$: $x \simeq y$ (equivalence) is a shorthand for $x \preceq$ and $y \preceq x$ being both valid, whereas $x \prec y$ (strict preorder) is used as an abbreviation for $x \preceq y$ and it is not the case that $y \preceq x$. Two elements $x, y \in X$ are called incomparable w. r. t. to $\preceq_X$ if any only if neither $x \preceq_X y$ nor $y \preceq_X x$. The preorder $\preceq_X$ is called total if no two elements of its domain X are incomparable, otherwise $\preceq_X$ is a partial preorder. An element $x \in X$ is minimal w. r. t. $\preceq_X$, or an $\preceq_X$*-minimal* element, if and only if there is no $y \in X$ such that $y \prec_X x$. More generally, $x \in X'$ is a $\preceq_X$ minimal element of $X' \subseteq X$ if there is no $y \in X'$ with $y \prec_X x$.

A (partial or total) *order* is a (partial or total) preorder that additionally fulfils the antisymmetry property, i. e., $\forall x, y \in X$: if $x \preceq y$ and $y \preceq x$, then $x = y$. Note that many "natural" preference relations (such as $\leq$ over set or real numbers) are antisymmetric and therefore orders. However, there exist interesting preference relations over sets of axioms (as needed for logic-based diagnosis) which are not antisymmetric (for instance, two sets X, Y containing the same number of axioms need not be identical although). As this thesis aims at defining a general framework first, and only study specific instantiations and their properties after that, the more general setting of preorders is of special interest.

Given any binary relations R_X, R_Y (not necessarily preorders) over the sets X, Y respectively, one can define the Cartesian product $R_{X\times Y} \subseteq R_X \times R_Y = (X \times Y) \times (X \times Y)$ as usual, letting

$$(x_1, y_1)\, R_{X\times Y}(x_2, y_2) \quad \text{iff} \quad x_1 R_X x_2 \text{ and } y_1\, R_Y\, y_2$$

It is a well known fact from model theory that properties that can be represented as universal Horn formulas are preserved under the Cartesian product constructions. Hence, as reflexivity, transitivity, and antisymmetry are all universal Horn formulas, we may restate the following fact that the Cartesian product of preorders (partial orders) is again a preorder (partial) order:

Lemma 2.3 (Cross product of (pre-)orders)
If $\preceq_X, \preceq_Y$ are preorders (partial orders), then $\preceq_X \times \preceq_Y$ is a preorder (partial order), too.

As the Cartesian product of preorders is again a preorder, it gives rise to the notion of minimal elements $(x, y) \in X \times Y$ over the cross-product of both sets. Considering the preorders as preference relations on equally relevant dimensions X and Y, it is a natural assumption to think of the minimal pairs over the pairs on the Cartesian product as the most preferable ones. This idea can be formally grounded within the notion of Pareto-optimality from the theory of economics. Intuitively, the so-called weak definition of Pareto-optimality states that trying to make one component of a Pareto-optimal element strictly better must lead to one component being not better

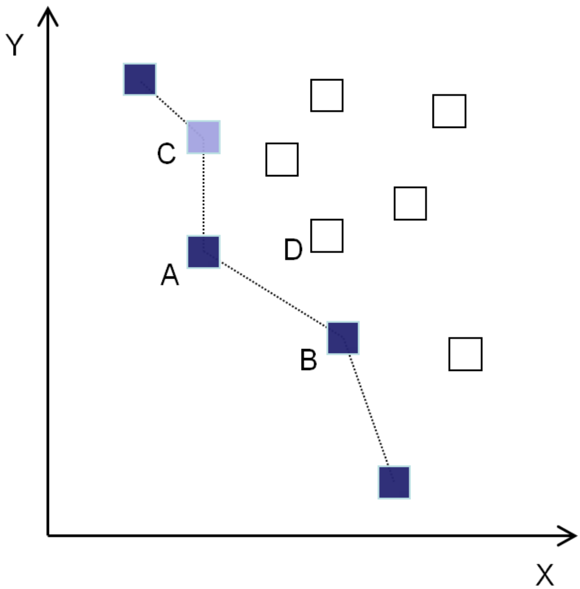

Figure 2.1: Pareto-optimality and Dominance

(i. e. either incomparable or worse), the strong notion demands a component being strictly better. Figure 2.1 visualizes the different notions for total orders. In this example, the points A and B are both weakly and strongly Pareto-optimal, whereas C is weakly Pareto-optimal, but not strongly. Point D is said to be *dominated* by A (A is smaller than D in all dimensions). In the next definition, we formalize these intuitions mathematically.

Definition 2.5 (Pareto-optimality)
Let $\overrightarrow{X} = X_1 \times \ldots \times X_n$ *be the Cartesian of the sets* X_i *and* $\preceq_i$ *be preorders on* X_i*. Then* $\vec{x} = (x_1, \ldots, x_n) \in X$ *is said to be* weakly Pareto-optimal for X *iff:*
For all $\vec{x}' \in \overrightarrow{X}$*: If there is some* i *with* $1 \leq i \leq n$ *s.t.* $x'_i \prec_i x_i$*, then there is a* j *with* $1 \leq j \leq n$ *such that not* $x'_j \preceq x_j$*.*

Moreover, $\vec{x} = (x_1, \ldots, x_n) \in \overrightarrow{X}$ *is said to be* strongly Pareto-optimal for X *iff:*
For all $\vec{x}' \in \overrightarrow{X}$*: If there is some* i *with* $1 \leq i \leq n$ *s.t.* $x'_i \prec_i x_i$*, then there is a* j *with* $1 \leq j \leq n$ *such that* $x_j \prec x'_j$*.*

As we show next, minimal and weakly Pareto-optimal elements coincide for the Cartesian product construction introduced before:

Lemma 2.4 (Weakly Pareto-optimal elements)
The weakly Pareto-optimal elements over the Cartesian product of preorders relations are exactly the minimal elements.

Proof. Let $\preceq_X$ denote the Cartesian product of component preorders $\preceq_{X_i}$ $(1 \leq i \leq n)$ from Definition 2.5. Let be $\vec{x}$ be a (weakly) Pareto-optimal element. Assume for contradiction that there is a $\vec{x}' \prec_X \vec{x}$. That means that that for all i one has $x_i' \preceq x_i$ and at least for one j $x_j' \prec_j x_j$, which contradicts the Pareto-optimality of $\vec{x}$. To prove the other direction assume that $\vec{x}$ is a minimal element. Let $\vec{x}'$ be such that there is some j with $x_j' \prec_j x_j$. If for all other i we had $x_i' \preceq_i x_i$, then we would have $\vec{x}' \prec \vec{x}$ contradicting the minimality of $\vec{x}$. □

If the component preorders also fulfil the property of connectedness or totality (i. e., for all x, x' either $x \preceq x'$ or $x' \preceq x$ or $x = x'$ holds), then the notions of weak Pareto-optimality and strong Pareto-optimality collapse into one, as $\neg(x \preceq x')$ entails $x' \prec x$. So we get:

Lemma 2.5 (Strongly Pareto-optimal elements)
The minimal elements of Cartesian products of total preorders are just the (strongly, weakly) Pareto-optimal elements.

Total preorders are an important structure in belief revision and non-monotonic reasoning (for details see Booth & Meyer, 2011).

Set inclusion $\subseteq$ over the power set $\mathcal{P}(X) = \{Y \mid Y \subseteq X\}$ of some set X is a special partial order which will be relevant for abduction. Then, by Lemma 2.3, $\preceq \subseteq \mathcal{P}(X) \times \mathcal{P}(X)$ is a preorder on the Cartesian product of $\mathcal{P}(X)$ with itself. This preorder $\preceq$ is said to be *monotonic (anti-monotonic) for set inclusion* if and only if for all $x, x' \in \mathcal{P}(X)$ with $x \subseteq x'$ one also has $x \preceq x'$ $(x' \preceq x)$.

2.4 Hypergraphs

A *graph* is a mathematical structure that connects pairs of *vertices* by means of (directed or undirected) *edges*. *Hypergraphs* generalise graphs by extending the definition of an edge from a binary to an n-ary relation. This way, any (non-empty) subset of vertices can form an edge that may again be undirected or directed. In the following, we introduce key notions about hypergraphs which are needed for our work. Additional information can be found, among others, in the work by Nielsen (2001), whose notation is used here in adapted form.

Definition 2.6 (Directed hypergraph)
A directed hypergraph *is a tuple* $\mathcal{H} = (V, E)$ *where* V *is a finite, nonempty set of* vertices*, and* $E \subseteq (\mathcal{P}(V) \setminus \emptyset) \times V$ *is a finite set of* edges*. For each edge* $e = (T, h) \in E$*,* $T(e)$ *denotes the* tail *of* e*, and* $h(e)$ *denotes its* head*.*

$\mathcal{H} = (V, E)$ *is a* weighted directed hypergraph *if* V *is defined like before, and* $E \subseteq \mathcal{P}(V) \times V \times W$ *for an arbitrary space* W *of weights. The weight of an edge* $e = (T, h, w)$ *is denoted by* $w(e)$*.*

Weights in graphs or hypergraphs are typically used to represent some evaluation of the edge in terms of quality, cost, or the like. It is typically desirable to extend this evaluation from single edges to sets of edges. To this end, we need an operator for combining weights; the mathematical structure of a monoid captures basic requirements for such a combination:

Definition 2.7 (Monoid)
Let S *be a set and* $\otimes$ *a binary operation.* $(S, \otimes)$ *is a* monoid *if and only if*

a) $\forall s_1, s_2 \in S : s_1 \otimes s_2 \in S$ *(*S *is closed under* $\otimes$*)*

b) $\forall s_1, s_2, s_3 \in S : (s_1 \otimes s_2) \otimes s_3 = s_1 \otimes (s_2 \otimes s_3)$ *(*$\otimes$ *is associative)*

c) $\exists e \in S \forall s \in S : e \otimes s = s = s \otimes e$ *(existence of a neutral element)*

$(S, \otimes)$ *is a* commutative monoid *if and only if it is a monoid and*

d) $\forall s_1, s_2 \in S : s_1 \otimes s_2 = s_2 \otimes s_1$ *(*$\otimes$ *is commutative)*

In standard graphs, a path (of length n) is a sequence $e_1, \ldots, e_i, e_{i+1}, \ldots, e_\mathrm{n}$ of edges such that two consecutive edges share a node, for example $e_i = (v_i, v_{i+1}), e_{i+1} = (v_{i+1}, v_{i+2})$, and so on. If the graph is weighted (that is, the edges each have an attributed weight), the edge weights along the path can be aggregated to yield the path weight. Reflecting the generalization of edges to arbitrary sets of vertices, a hyperpath represents a consecutive chain of connections between two sets of vertices. Intuitively, there is a hyperpath from S (start vertices) to g (goal vertex) if there is a hyperedge connecting some intermediate set of vertices Y to g, and each $y_i \in Y$ is in turn reachable from S via a hyperpath. Definition 2.8 formalizes this intuition.

Definition 2.8 (Directed hyperpath)
Let $\mathcal{H} = (V, E)$ be a directed hypergraph. Then $p_{S,g} = (V_{S,g}, E_{S,g})$ is a simple directed hyperpath *in $\mathcal{H}$ from S to g if and only if*

a) $g \in S$ *and* $p_{S,g} = (\{g\}, \emptyset)$*, or*

b) $\exists e \in E : h(e) = g \wedge T(e) = \{y_1, \ldots, y_k\} \wedge \forall 1 \leq i \leq k \; \exists p_{S,y_i} :$
$(V \supseteq V_{S,g} = \{g\} \cup \bigcup_{y_i \in T(e)} V_{S,y_i} \wedge E \supseteq E_{S,g} = \{e\} \cup \bigcup_{y_i \in T(e)} E_{S,y_i})$

Moreover, $p_{S,G} = (V_{S,G}, E_{S,G})$ is a directed hyperpath *in $\mathcal{H}$ from S to G if and only if*

c) $\forall g \in g : p_{S,g} = (V_{S,g}, E_{S,g})$ *is a directed hyperpath in $\mathcal{H}$ from S to g,*
$V_{S,G} = \bigcup_{g \in G} V_{S,g}$ *and* $E_{S,G} = \bigcup_{g \in G} E_{S,g}$

If $\mathcal{H}$ is a weighted hypergraph, let W denote its weight space and let $\otimes$ be a binary operation such that $(W, \otimes)$ is a commutative monoid called the weight system *of $\mathcal{H}$. The notion of a* weight *then extends from hyperedges e to simple hyperpaths $p_{S,g} = (V_{S,g}, E_{S,g})$ by $w(p_{S,g}) = \bigotimes_{e \in E_{S,g}} w(e)$, and to hyperpaths $p_{S,G} = (V_{S,G}, E_{S,G})$ by $w(p_{S,G}) = \bigotimes_{e \in E_{S,G}} w(e)$.*

It is easy to see that a simple hyperpath is thus a sub-hypergraph of $\mathcal{H}$ connecting (a subset of) the vertices in S to the vertex g. Similarly, a hyperpath from S to G is a sub-hypergraph of $\mathcal{H}$ that connects (a part of) the vertices in S to all vertices in G. The cost of a hyperpath is determined by "adding up" the costs of its hyperedges. The requirement of $(W, \otimes)$ being a commutative monoid is derived from the frequently-used weight system $(\mathbb{N}, +)$. While other systems such as $(\mathbb{R}, +)$ correspond to algebraic structures that extend commutative monoids with additional properties (such as the existence of inverses), these additional properties are not relevant for our work.

In this chapter, we have provided the tools needed to give a clear, formal definition of relaxed abduction in Section 3.2. In a nutshell, we will define solutions to a relaxed abduction problem to be the Pareto-optimal elements of a solution space. Then, we will show how derivations of an axiom in a so-called proof tree correspond to directed weighted hyperpaths, and employ this finding to devise an algorithm for computing the set of solutions of a relaxed abduction problem as shortest hyperpaths in the proof tree.

http://www.springer.com/978-3-658-14406-7

Relaxed Abduction
Robust Information Interpretation for Industrial Applications
Hubauer, Th.
2016, XVII, 134 p. 11 illus., 1 illus. in color., Softcover
ISBN: 978-3-658-14406-7