Towards Formalizations in Case-Based Reasoning for Synthesis *

Katy Börner
HTWK Leipzig, Department of Informatics
P.O.Box 66, 04251 Leipzig, FRG
katy@informatik.th-leipzig.de

Abstract

This paper presents the formalization of a novel approach to structural similarity assessment and adaptation in case-based reasoning (CBR) for synthesis. The approach has been informally presented, exemplified, and implemented for the domain of industrial building design (Börner 1993). By relating the approach to existing theories we provide the foundation of its systematic evaluation and appropriate usage. Cases, the primary repository of knowledge, are represented structurally using an algebraic approach. Similarity relations provide structure preserving case modifications modulo the underlying algebra and an equational theory over the algebra (so available). This representation of a modeled universe of discourse enables theory-based inference of adapted solutions. The approach enables us to incorporate formally generalization, abstraction, geometrical transformation, and their combinations into CBR.

Introduction

In CBR the universe of discourse is represented by a finite set of already solved cases stored in the case-base and a similarity or distance relation over them (Richter 1992). An actual problem is solved by searching for a most similar problem inside the case-base and solving the actual one accordingly. Usually, cases consist of two disjoint parts describing the problem and the solution by attribute value pairs (Riesbeck & Schank 1989 and Kolodner 1993). Similarities are measured via some metrics by taking into account the number of shared attribute values (weighted or not).

Aiming at the solution of synthesis tasks functional dependencies between objects rather than their simple attributes are important. Discovery of homomorphic mappings between object sets and their inherent relations becomes necessary. Homomorphism detection, however, may become computationally intractable.

In our approach we represent the modeled universe of discourse by structural case descriptions and structural similarity relations over them. Structural similarity is defined as a mapping into a highly structured, partially ordered space. It represents background knowledge about structure preserving case modifications and is able to guide adaptation.

Evaluations of CBR approaches and systems are mostly performed by "real world" exemplifications. The problems with this are the unavailability of unique databases for synthesis tasks and doubts whether this is the best way to compare approaches. Formalizations of approaches and their explicit relation to underlying theories may be much more helpful. That's exactly what we want to do.

The paper is organized as follows. Section 2 introduces the basic ideas behind structural similarity assessment and adaptation in synthesis. Section 3 gives the formalization of the approach. The corresponding algorithm is presented in section 4. We finish with some concluding remarks in section 5. For the sake of clarity but without loss of generality Example starts easily accessible exemplifications given in *slanted* letters.

Methodology and Framework

Aiming at the solution of synthesis tasks, cases are no longer given by sets of attribute values and corresponding concepts. Problems and solutions are often complex structures of objects and inherent relations. Similarity assessment proceeds over complex structures.

Example: Domain of geometric figures

Driven by the original application domain, industrial building design (Hovestadt 1993), and for purposes of simplicity and generality we will use a two-dimensional domain of simple geometric figures as exemplified in Fig. 1. In this micro-domain different states correspond to arrangements of objects (figures for short) like circles, ellipses, and squares etc. All objects are arranged on a natural grid. Following the arrows in Fig. 1

*This research was supported by the German Ministry for Research and Technology (BMFT) within the joint project FABEL under contract no. 413-4001-01IW104. Project partners in FABEL are German National Research Center of Computer Science (GMD), Sankt Augustin, BSR Consulting GmbH, München, Technical University of Dresden, HTWK Leipzig, University of Freiburg, and University of Karlsruhe.

each state (figure) provides an improvement (rearranging objects, introducing new objects, etc.) of the ongoing one, e.g., figure-a=Circle; figure-b=two copies of "figure-a" arranged in X direction; figure-c=one copy of "figure-b" arranged in Y direction; figure-d=cover "figure-c"; figure-e=two copies of "figure-d" arranged in X direction.

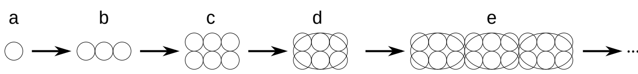

Figure 1: Design of geometric figures

Algebraic approaches to knowledge representation allow us to represent solutions as consisting of the problem and applied operators. The underlying algebra and an equational theory (so available) over the algebra provide background knowledge to determine *structural similarity* of problem states. Given a *complete* and *consistent* model of the universe of discourse the transfer of operational solutions results in *correctly* adapted solutions.

Note, that similarity assessed by the system may differ considerably from that assessed by human users, even assuming that both have the same set of cases at hand. Due to the similarity relations derived from prior cases their strategies to use and reason from these cases may be quite different.

Structural Similarity Assessment, Adaptation, and Learning

Now we want to present the approach to structural similarity assessment, adaptation, and learning formally. First, we give the underlying knowledge representation as a basis for reasoning. Inspired by the work (Indurkhya 1991 and O'Hara 1992), we will use an algebraic approach. Next we define what we understand by structural similarity and show how to learn *proper* modification rules for similarity assessment from a set of prior cases. Based on this we are able to define solution transfer and adaptation.

Knowledge Representation

As a basis for knowledge representation we assume any finite, heterogeneous, and finitary *signature* Σ. The signature provides a set S of *sorts*, a set O of *operator symbols*, and an *arity function* α over O. Operator symbols of arity 0 are called *constants*. Additionally, we need a sorted, countably infinite set X of *variable symbols* with $\Sigma \cap X = \emptyset$. We will use *indices* as appropriate. The *ground term algebra* over Σ is denoted by $T(\Sigma, \emptyset)$. A *ground term* $t \in T(\Sigma, \emptyset)$ is a constant or the application of an operator symbol to the appropriate number of ground terms. The *free term algebra* over Σ and X is denoted by $T(\Sigma, X)$.

Example: Term algebra of geometric figures

Throughout the paper the signature and variables used to describe figures are as follows:

sorts	$I\!N, Direction, Object$;
constants	$X, Y : Direction$;
	$0, 1, 2, ..., 10 : I\!N$;
	$Circle, Ellipse, Square : Object$;
operators	copy: $Direction \times I\!N \times Object \rightarrow Object$;
	divide: $Object \rightarrow Object$;
	cover: $Object \rightarrow Object$.

First order variables z and w will be used to substitute constants of sort $I\!N$ and $Object$. The second order variable f will be used to substitute operators like copy, divide, and cover. The interpretation of the signature will become obvious by its use. We insist that identical figures are represented by unique representations.

States of the world correspond to different figures and are represented by ground terms $t \in T(\Sigma, \emptyset)$.

Example: Representation of states

Let t_s be a ground term representing a state s. Using the introduced term algebra the five states (figures) in Fig. 1 may be represented by:

$t_a = Circle$; $t_b = copy(X, 2, Circle)$;

$t_c = copy(Y, 1, copy(X, 2, Circle))$;

$t_d = cover(copy(Y, 1, copy(X, 2, Circle)))$;

$t_e = copy(X, 2, cover(copy(Y, 1, copy(X, 2, Circle))))$

as illustrated in Fig. 2.

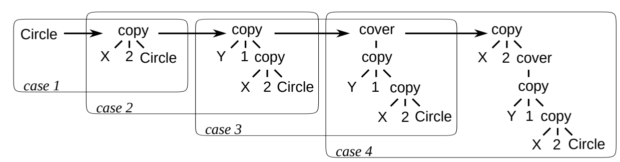

Figure 2: Term-based knowledge representation of the states (figures) given in Fig. 1

As a basis for reasoning subsequent states are pairwise related to each other by means of cases. The first state will be named t^p, the *problem*, the following describes t^s, its *solution*, with $t^p, t^s \in T(\Sigma, \emptyset)$. The fact becomes obvious, that problem and solution may be seen as some kind of role states have.

To represent the case solution in terms of the case problem we need to refer to subterms. Terms are identified with finite labeled trees as usual (see Fig. 2). Intermediate nodes correspond to operators and leaves to constant symbols. Every sub-term within a given term t can be uniquely referred to by its place. Places q within terms are denoted by words of natural numbers excluding zero $I\!N^+$, and are defined recursively as follows. The term at place $\langle i \rangle$ within $f(t_1, ..., t_m)$ is t_i. The term at place $\langle i_1, ..., i_n \rangle$ within $f(t_1, ..., t_m)$ is the term at place $\langle i_2, ..., i_n \rangle$ in t_{i_1}. The top-most (root) place in a term is the empty word called ϵ. The symbol at place q is denoted $t(q)$, the subterm of t at place q is denoted $t|_q$. The result of replacing $t|_q$ with a term u at place q is denoted $t[q \leftarrow u]$.

A *structural case description* c is defined by $c = \langle t^s, q \rangle$ where the problem state t^p corresponds to a subterm of the solution state t^s at place q, i.e., $t^s|_q = t^p$.

As usual in CBR the set of all cases is named *CB*.

Example: Representation of cases

With $t_b\,|_3 = t_a$, $t_c\,|_3 = t_b$, $t_d\,|_1 = t_c$ *and* $t_e\,|_3 = t_d$ *the states introduced in Fig. 1 may be related by four cases:* $c_1 = \langle t_b, 3\rangle$, $c_2 = \langle t_c, 3\rangle$, $c_3 = \langle t_d, 1\rangle$, *and* $c_4 = \langle t_e, 3\rangle$.

Structural Similarity Assessment

Structural similarity of problem states of cases is defined by structure preserving, homomorphic mappings. As proposed by (Jantke 1993) the mappings are formalized close to antiunification. We distinguish two kinds of structural similarity, namely *syntactic* and *semantic similarity.*

Syntactic Similarity To define *syntactic similarity* we are interested in mappings of *ground terms* in $T(\Sigma, \emptyset)$ into some term in the *free term algebra* $T(\Sigma, X)$ containing variables. Therefore we need the definition of substitution and their inverse.

A *substitution* θ is a mapping from a set of variables X into $T(\Sigma, X)$. Such a mapping is finitely representable and denoted as a set of correspondences between constants (terms) and distinct variables. The application of θ to a term t is denoted $t\theta$. If θ_1 and θ_2 are substitutions then $t\theta_1\theta_2 = (t\theta_1)\theta_2$. Given a term t and a substitution θ we assume there exists an unique inverse substitution θ^{-1} such that $t\theta\theta^{-1} = t$. Whereas the substitution maps variables into terms, the inverse substitution θ^{-1} maps terms into variables. Thus if: $\theta = \{x_1 \mid t_1, ..., x_n \mid t_n\}$ we denote the corresponding *inverse substitution* by: $\theta^{-1} = \{t_1(q_{1,1}, ..., q_{1,m_1}) \mid x_1, ..., t_n(q_{n,1}, ..., q_{n,m_n}) \mid x_n\}$ in which q_{i,m_j} are the places at which the variables x_i are found within t. Inverse substitutions are applied by replacing all t_i at places $\{q_{i,1}, ..., q_{i,m_i}\}$ within t by x_i.

Syntactic similarity σ is defined as the union of proper modifications which relate a set of problems $t_i^p \in T^p$, $i = 1, ..., n$ to each other. That is $\sigma := \bigcup_{i=1}^{n} \theta_i^{-1}$ satisfying $t_i^p\theta_i^{-1} = \text{MSCS}^p$, $i = 1, ..., n$. The unique term MSCS^p is called the *most specific common structure*[1]. For any other specific common structure t of T^p there exists a substitution θ^{-1} s.t. $\text{MSCS}^p\theta^{-1} = t$. Note that substitutions of function symbols are either *first order* or *incomplete.*

Example: Syntactic similarity

We illustrate syntactic similarity by generalization and abstraction.

Generalization *is a mapping from constant symbols into variables formalized by inverse substitutions. It is best known for descriptive generalization and one of the techniques most often used in inductive inference (Mitchell et al. 1996).*

Abstraction *is the process of mapping a given representation of an universe of discourse onto another logically equivalent, abstract representation (Giunchiglia & Walsh 1992). We restrict abstractions here to mappings from function symbols into variables again formalized by inverse substitutions. We will consider only function symbol abstractions which allow collapse of the function name, or change of the arity.*

Fig. 3 and Fig. 4 provide examples of generalization and abstraction using the signature and variables given. Both examples provide the pictorial and term-based representations of two figures, the applied inverse substitutions, and the MSCS *corresponding to the generalization respectively abstraction of the figures.*

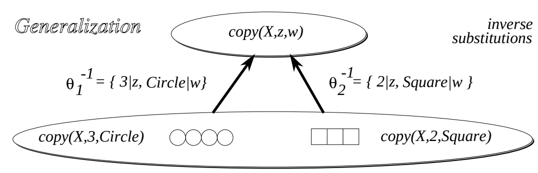

Figure 3: **generalization**

In Fig. 3 $copy(X, 3, Circle)$ *corresponds to "take one Circle, copy it three times and arrange all in X direction".* $copy(X, 2, Square)$ *represents "take one Square, copy it two times and arrange all in X direction". Applying the inverse substitutions* θ_1^{-1} *and* θ_2^{-1} *correspondingly results in* MSCS= $copy(X, z, w)$.

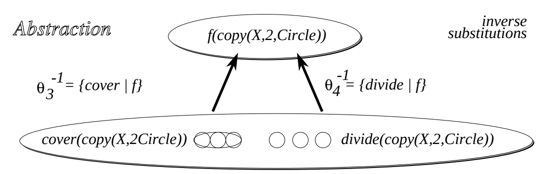

Figure 4: **abstraction**

Analogous is the example for abstraction given in Fig. 4. Here $cover(copy(X, 2, Circle))$ *corresponds to "take one Circle, copy it two times, arrange all in X-direction, cover all objects",* $divide(copy(X, 2, Circle))$ *corresponds to "take one Circle, copy it two times, arrange all in X-direction, and enlarge the distances between all objects". Applying the inverse substitutions* θ_3^{-1} *and* θ_4^{-1} *results in* MSCS=$f(copy(X, 2, Circle))$.

Semantic Similarity To define *semantic similarity* we are interested in mappings from $T(\Sigma, \emptyset)$ into some sub-algebra $T(\Sigma, \emptyset)\,|_{\equiv^E}$ or $T(\Sigma, X)\,|_{\equiv^E}$ induced by an equational theory over $T(\Sigma, \emptyset)$ or $T(\Sigma, X)$. That is, we map *ground terms* into some equation $e \in E$.

An *equation* e or *equational axiom* is an unordered pair of terms, written $t_1 = t_2$ where terms are of the same sort $s \in S$. We let $E(\Sigma, X)_s$ denote the set of

[1] This definition of syntactic similarity is close to the concept of *syntactic antiunification.* Given two terms t_1, t_2 one is searching for some term t, called anti-unifier, and corresponding substitutions θ_1, θ_2 satisfying both $t\theta_1 = t_1$ and $t\theta_2 = t_2$. The term t is called *least general anti-unifier* (Muggleton 1992) or *most specific generalization* (Plotkin 1970) iff there is no other anti-unifier $t\prime$ with $t\prime = t\theta$.

all equations over Σ and X of sort $s \in S$, and we let $E(\Sigma, X) = \cup_{s \in S} E(\Sigma, X)_s$. An equation is said to be a *ground equation* iff $t_1, t_2 \in T(\Sigma, \emptyset)_s$. We term $\equiv^E$ the relation of provable equivalence on $T(\Sigma, X)$ induced by E and denote the resulting subalgebra $T(\Sigma, X) \mid_{\equiv^E}$. Thus $t_1 \equiv^E t_2 \Leftrightarrow E \vdash t_1 = t_2$.

Semantic similarity of a finite set of problem states $t_i^p \in T^p$, $i = 1, ..., n$ modulo an equational theory E is defined as the union of inverse substitutions $\sigma := \cup_{i=1}^n \theta_i^{-1}$ satisfying $t_i^p \theta_i^{-1} \equiv^E \text{MSCS}^p$, $i = 1, ..., n$. For any other specific common structure t of T^p there exists an inverse substitution θ^{-1} s.t. $\text{MSCS}^p \theta^{-1} \equiv^E t$. Note that there is no unique MSCS^p when one is using equational theory. In fact, this suggests that the formalization is addressing the real problem.

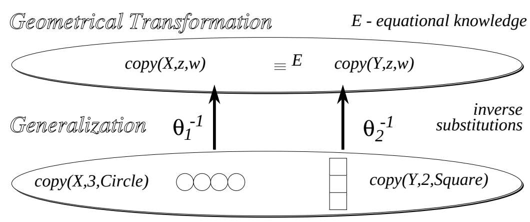

Figure 5: **generalization–geometrical transformation**

Example: Semantic similarity
In our domain semantic similarity may be illustrated by incorporating geometrical transformations represented by an equational theory over a set of cases.

The specific equational scheme for a ground term t_1^p *and its 90-degree-rotated version* t_2^p *is obviously simple:* $E_{scheme} = \{t_1^p = t_2^p, t_1^p = t_1^p, t_2^p = t_2^p\}$.
As an example for pure geometrical transformation we may instantiate this scheme by ground terms:
$t_1^p = copy(X, 2, Circle), t_2^p = copy(Y, 2, Circle)$.

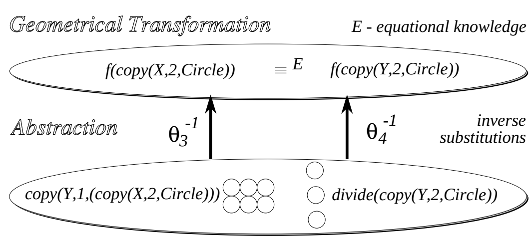

Figure 6: **abstraction–geometrical transformation**

Usually we are interested in the combination of inverse substitutions (for generalization and abstraction) and equational theories. The equational theory has to be enlarged to cover corresponding variables. Using the inverse substitutions θ_1^{-1} *to* θ_4^{-1} *(see Fig. 3 and Fig. 4) and instantiations*
$t_1^p = copy(X, z, w), t_2^p = copy(Y, z, w)$
we are able to cover generalization (see Fig. 5) or $t_1^p = f(copy(X, 2, Circle)), t_2^p = f(copy(Y, 2, Circle))$ *abstraction (see Fig. 6).*

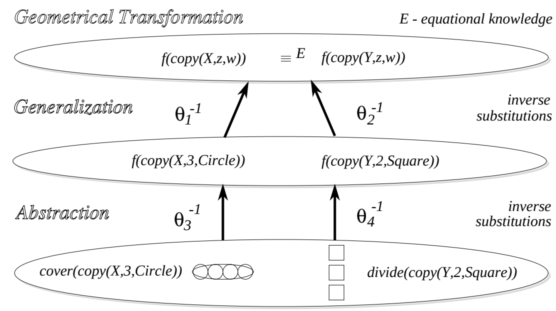

Figure 7: **abstraction–generalization–geometrical transformation**

With $t_1^p = f(copy(X, z, w)), t_2^p = f(copy(Y, z, w))$ *we are able to incorporate generalization, abstraction, and geometrical transformations (see Fig. 7).*

Learning

Learning proceeds incrementally by collecting cases in *CB* and inducing similarity relations about these cases using the underlying algebra. The latter proceeds by determining *proper* inverse substitutions of problem states belonging to one case-set. There are different possibilities to define case-sets in *CB*. Motivated by case-based classification one may divide the entire *CB* into case-sets $CB_j := \{c_{j_1}, c_{j_2}, ..., c_{j_n}\}$, with equal solution parts $t_j^{s-part} := t_j^s[q_j \leftarrow w]$, $j = 1, ..., m$. For each set CB_j we determine $\sigma_j := \cup_{i=1}^n \theta_{j_i}^{-1}$ with $t_{j_i}^p \theta_{j_i}^{-1} \equiv^E \text{MSCS}_j^p$ for all $t_{j_i}^p \in CB_j$. $E = \emptyset$ is possible. Thus case-based knowledge is represented by $CB = \{CB_1, ..., CB_m\}$ and corresponding similarity relations $\{\sigma_1, ..., \sigma_m\}$.

Example: Learning about structural similarity
We assume the figures in Fig. 3 and Fig. 4 share equal solution parts. In the example for generalization $\sigma_G = \theta_1^{-1} \cup \theta_2^{-1} = \{3 \mid z, Circle \mid w, 2 \mid z, Square \mid w\}$ *holds. Analogously for the example of abstraction, we derive* $\sigma_A = \{cover \mid f, divide \mid f\}$. *The similarity of all four terms is* $\sigma = \sigma_G \cup \sigma_A$.

Solution Transfer and Adaptation

Given structural similarity solution transfer and adaptation proceeds by transferring the prior solution part to the actual problem. That is $t_{actual}^s := t_j^s\ [q_j \leftarrow t_{actual}^p]$ iff $\sigma_j : (t_{actual}^p) \equiv^E \text{MSCS}_j^p$. With $q_{actual} := q_j$ we receive $c_{actual} = \langle t_{actual}^s, q_{actual} \rangle$. Adaptation is done implicitly by representing the solution operationally and transferring it to the actual problem.

Example: Solution transfer and adaptation
We come back to our example of generalization. Assume there are exactly two cases in CB with $c_1 = \langle cover(copy(X, 3, Circle)), 1\rangle$ *and* $c_2 = \langle cover(copy(X, 2, Square)), 1\rangle$. *We learned about* MSCS_G^p *and*

$\sigma_G = \theta_1^{-1} \cup \theta_2^{-1}$. *Given an actual problem* $t^p_{actual} = copy(X, 2, Circle)$ *we are able to derive* MSCS^p_G *by applying inverse substitutions provided by* σ_G, *i.e.,* $copy(X, 2, Circle)\{3 \mid z, Circle \mid w, 2 \mid z, Square \mid w\} = copy(X, z, w)$. *With* $t^s_{actual} := t^s_G\ [q_G \leftarrow t^p_{actual}]$ *we receive* $c_{actual} = \langle cover(copy(X, 2, Circle)), 1\rangle$.

Algorithm

Next we want to introduce the algorithm for structural similarity assessment, adaptation, and learning. **Given** are a set of cases in *CB*, a set of variables X, and an equational theory E over Σ and X. **Input** is the actual problem state t^p_{actual}. **Output** is the complete actual case or the remark *not solvable.*

Reasoning proceeds as follows: as in standard CBR we need first to know about similarity relations σ over case-sets. That's why we start by dividing the *CB* into case-sets with equal solution-parts. Next we determine structural similarity relations over these case-sets.

begin
$CB := \{CB_j \mid \forall t^p_{j_i} \in CB_j\ (t_j^{s-part} = t^s_{j_i}), j = 1, ..., m\}$
solved:=false; j:=0;
repeat
  j:=j+1;
  $\sigma_j := \{\theta^{-1}_{j_i} \mid \forall t^p_{j_i} \in CB_j (t^p_{j_i}\theta^{-1}_{j_i} \equiv^E \text{MSCS}^p_j), i = 1, ..., n\}$
  if $(\sigma_j : (t^p_{actual}) \equiv^E \text{MSCS}^p_j)$ **then**
    begin
    $t^s_{actual} := t^s_j[q_j \leftarrow t^p_{actual}]$;
    $q_{actual} := q_j$;
    output $(c_{actual} = \langle t^s_{actual}, q_{actual}\rangle)$;
    solved:=true;
    $CB_j := CB_j \cup c_{actual}$
    end
until (solved) **or** $(j = m)$
if not solved **then output** (not solvable)
end.

To check each case stored in the case-base during structural similarity assessment is both computationally unfeasible and psychologically implausible in any realistic situation. That's why we apply proper inverse substitutions provided by the structural similarity σ_j of case-sets CB_j to the actual problem and compare the resulting term with the MSCS^p_j of CB_j. Given equivalence, the prior solution is transferred by replacing the substructure of prior problems by the actual one. The actual problem will be solved corresponding to the first similar case-set and stored in *CB*.

Conclusions

The contribution of this paper is a formalization of a novel approach to structural similarity assessment and adaptation which was developed and implemented for the domain of industrial building design. The approach may be used practically to represent and perform reasoning in other application domains where structures to a large degree determine or facilitate classification, part identification, similarity and adaptability, and where examples and underlying theories are available.

We hope to encourage formalizations in CBR. They provide not only the basis to evaluate and compare approaches in a more formal setting but show how to improve them by work not always associated with CBR.

Acknowledgements

I would like to acknowledge discussions with Dietmar Janetzko, Oliver Jäschke, and Eberhard Pippig which helped to clarify the presented formalization. I am indebted to my advisor Klaus P. Jantke for his continuing support. Phil Watson deserves thanks for his aid in revising an earlier version of this paper.

References

Katy Börner. Structural similarity as guidance in case-based design. In *First European Workshop on Case-Based Reasoning (EWCBR-93)*, LNAI, page to appear. Springer Verlag, 1994.

Fausto Giunchiglia and Toby Walsh. A theory of abstraction. *Artificial Intelligence*, 57:323–389, 1992.

Ludger Hovestadt. A4 – digital building – extensive computer support for the design, construction, and management of buildings. In *CAAD Futures '93, Proceedings of the Fifth International Conference on Computer-Aided Architectural Design Futures*, pages 405–422. North-Holland, 1993.

Bipin Indurkhya. On the role of interpretive analogy in learning. *New Generation Computing*, 8(4):385–402, 1991.

Klaus P. Jantke. Nonstandard concepts of similarity in case-based reasoning. In M. M. Richter H. H. Bock, W. Lenski, editor, *Information Systems and Data Analysis: Prospects –Foundations–Applications, Proceedings of the 17th Annual Conference of the GfKl, Univ. of Kaiserslautern, 1993*, pages 29–44. Springer Verlag, 1994.

Janet L. Kolodner. *Case-based reasoning.* Morgan Kaufmann, 1993.

Tom M. Mitchell, Richard M. Keller, and Smadar T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1:47–80, 1986.

Steven Muggleton. *Inductive logic programming.* Academic Press, 1992.

Scott O'Hara. A model of the 'redescription' process in the context of geometric proportional analogy problems. In Klaus P. Jantke, editor, *Proceedings of the International Workshop on Analogical and Inductive Inference*, pages 268–293. Springer Verlag, 1992.

Gordon D. Plotkin. A note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pages 153–163. American Elsevier, 1970.

Michael M. Richter. Classification and learning of similarity measures. In *Proceedings of the 16th Annual Meeting of the German Society for Classification in Kaiserslautern.* Springer Verlag, 1992.

Christopher K. Riesbeck and Roger C. Schank. *Inside Case-Based Reasoning.* Lawrence Erlbaum Associates, Hillsdale, 1989.