

On Knowledge Transfer in Case-Based Inference

Santiago Ontañón¹ and Enric Plaza²

¹ Computer Science Department
Drexel University
Philadelphia, PA, USA 19104
santi@cs.drexel.edu

² IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain)
enric@iia.csic.es

Abstract. While similarity and retrieval in case-based reasoning (CBR) have received a lot of attention in the literature, other aspects of CBR, such as case reuse are less understood. Specifically, we focus on one of such, less understood, problems: *knowledge transfer*. The issue we intend to elucidate can be expressed as follows: *what knowledge present in a source case is transferred to a target problem in case-based inference?* This paper presents a preliminary formal model of knowledge transfer and relates it to the classical notion of analogy.

1 Introduction

In case-based reasoning (CBR), a problem is solved by first *retrieving* one or several relevant cases from a case-base, and then *reusing* the knowledge in the retrieved case (or cases) to solve the new problem. The retrieval stage in CBR has received a lot of attention in the literature, however, other aspects of CBR have received less attention and are less well understood; specifically, what knowledge can be reused from a previous case (source) to solve a new (target) case?

There is no generally agreed upon model of this process, which we will call the *knowledge transfer* process. This paper presents a model of knowledge transfer in case-based inference (CBI). Case-based inference, as described in [7] corresponds only to a part of the complete CBR cycle [1]. CBI basically accounts for the general inference process performed when predicting or characterizing a solution to a problem from a given set of cases, it does not include the process of adaptation or revision of the proposed solutions.

Consequently, in this paper, we intend to model the process of pure knowledge transfer, without intending to model the complete case reuse process, nor trying to encompass the whole variety of approaches to reuse in case-based reasoning, like rule-based adaptation. The issue we intend to elucidate can be expressed as follows: *what knowledge present in the source case is transferred to a target problem during case-based inference?*

In our model, we take a different direction from the CBI model of Hüllermeier [7], where they focus on prediction, i.e. classification and regressions tasks, since

we focus on design tasks. While on prediction, the solution is selecting a solution from a set of possible solutions, on design tasks the solution is achieved by building a complex structure from “solution elements” (usually nodes and their relationships). The goal of this paper is then to give an account of what is transferred from a previous case to a solution case when it is a complex structure. Our model of knowledge transfer is based on the notions of refinement, subsumption, partial unification and amalgam, defined over a generalization space. This model is applicable to any representation formalism for which a relevant generalization space can be defined. Consequently, albeit we do take into account the notion of similarity, numerical measures of similarity are downplayed in this model, and we focus on a more symbolic notion of similarity. In our approach, it is more important to reason about *what is shared* among cases than the *degree* to which two cases share some of their content.

The work presented in this paper is an extension of the work in [15], where we introduced a preliminary version of this model. In this paper, we take one step forward, generalize the model to also cover multi-case adaptation and make more emphasis on its relation with analogical reasoning, as one of the underlying principles of case-based reasoning.

The remainder of this paper is organized as follows. First we introduce the idea of knowledge transfer in CBR in Section 2. Then, Section 3 briefly presents some necessary theoretical background for our formal model of knowledge transfer presented in Section 4. Finally, Section 5 discusses knowledge transfer in computational analogy and its relations with CBR.

2 Knowledge Transfer in CBR

In standard models of CBR, cases are typically understood as problem/solution pairs (p, s) or situation/outcome pairs. Therefore, solving a problem p' means finding or constructing a solution s' by adapting the solution of one or more retrieved cases. In this paper, we will consider a more general model, where cases are a single description, and where the problem and the solution are just two parts of this single description. In this view, an unsolved problem is just a partial description that needs completion.

The task of solving a problem in our view consists of two steps (in accordance to recent formal models of CBR [7]):

1. (case-based inference) finding a complete description by transferring information from retrieved cases to the problem at hand. Thus, the process of case-based inference can be further divided into two steps: case retrieval and knowledge transfer.
2. (adaptation) later performing any additional domain specific adaptations required to turn the complete description found by case-based inference into a valid solution for the domain at hand.

In the traditional CBR cycle [1], the reuse process encompasses both knowledge transfer and adaptation. The model presented in this section focuses exclusively

on the process of knowledge transfer, rather than on the whole reuse process. Therefore, the outcome of the knowledge transfer process is not a valid solution, but the result of transferring knowledge from the one or more source cases to the target, which might still need to be adapted by using some domain specific rules, or any other reuse procedure. For that reason, we will refer to the result of knowledge transfer as a *conjecture*. Thus, we say that a conjecture is formed by transferring knowledge from source cases to a target problem —or, in other words, conjectures are the outcome of case-based inference. Some conjectures might constitute solutions, while some others might require adaptation.

There are multiple scenarios that define different knowledge transfer tasks:

- Transfer may be from a single or multiple retrieved cases.
- The unsolved problem description can be understood as a hard requirement (i.e. when the solved problem can only add elements to the unsolved problem description, but not change or remove anything to the problem description), or not (when the unsolved problem description just expresses some preferences of over the final solution).

For the sake of clarity, in this paper we will only provide a formalization of the hard requirements scenario. However, we will provide insights into how the soft requirement scenario can be easily modeled in our framework.

Our formalization is based on the notions of *generalization space* and that of *amalgam* and *partial unification*. For a more in-depth description of these ideas, the reader is referred to [13], here, we will just provide their intuitive ideas, sufficient to present our model of knowledge transfer.

3 Background

In this paper we will make the assumption that cases are terms in some *generalization space*. We define a generalization space as a partially ordered set $\langle \mathcal{L}, \sqsubseteq \rangle$, where \mathcal{L} is a language, and \sqsubseteq is a subsumption between the terms of the language \mathcal{L} . We say that a term ψ_1 subsumes another term ψ_2 ($\psi_1 \sqsubseteq \psi_2$) when ψ_1 is more general (or equal) than ψ_2 ¹. Additionally, we assume that \mathcal{L} contains the infimum element \perp (or “any”), and the supremum element \top (or “none”) with respect to the subsumption order.

Next, for any two terms ψ_1 and ψ_2 we can define their *unification*, $(\psi_1 \sqcup \psi_2)$, which is the *most general specialization* of two given terms, and their *anti-unification*, defined as the *least general generalization* of two terms, representing the most specific term that subsumes both. Intuitively, a unifier (if it exists) is a term that has all the information in both the original terms, and an anti-unifier is a term that contains only all that is common between two terms. Also, notice that, depending on \mathcal{L} , anti-unifier and unifier might be unique or not.

¹ In machine learning terms, $A \sqsubseteq B$ means that A is more general than B , while in description logics it has the opposite meaning, since it is seen as “set inclusion” of their interpretations.

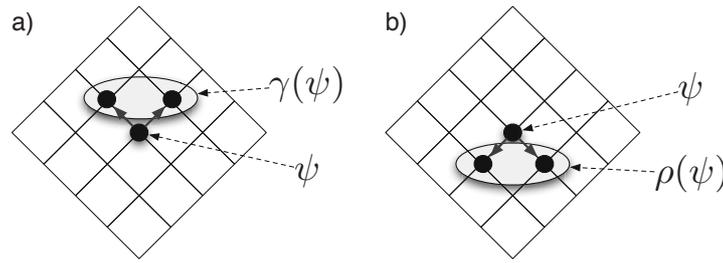


Fig. 1. A generalization refinement operator γ , and a specialization operator ρ

Let us now summarize the basic notions of *refinement operator* over partially ordered sets and introduce the concepts relevant for this paper —see [9] for a more in-depth analysis. Refinement operators are defined as follows:

Definition 1. A downward refinement operator ρ over a partially-ordered set $\langle \mathcal{L}, \sqsubseteq \rangle$ is a function such that $\rho(\psi) \subseteq \{\psi' \in \mathcal{L} \mid \psi \sqsubseteq \psi'\}$ for all $\psi \in \mathcal{L}$.

Definition 2. An upward refinement operator γ over a partially-ordered set $\langle \mathcal{L}, \sqsubseteq \rangle$ is a function such that $\gamma(\psi) \subseteq \{\psi' \in \mathcal{L} \mid \psi' \sqsubseteq \psi\}$ for all $\psi \in \mathcal{L}$.

In other words, upward refinement operators generate elements of \mathcal{L} which are more general, whereas downward refinement operators generate elements of \mathcal{L} which are more specific, as illustrated by Figure 1. Typically, the symbol γ is used for upward refinement operators, and ρ for downward refinement operators.

Refinement operators can be used to navigate the generalization space using different search strategies, and are widely used in Inductive Logic Programming. For instance, if we have a term representing “a German minivan”, a generalization refinement operator would return generalizations like “a European minivan”, or “a German vehicle”. Moreover, in practice, it is preferable to have refinement operators that do not perform large generalization or specialization leaps, i.e. that make the smallest possible change in a term when generalizing or specializing, to better explore the space of generalizations as a search space [14].

3.1 Amalgams

The notion of *amalgam* can be conceived of as a generalization of the notion of unification over terms. The unification of two terms (or descriptions) ψ_a and ψ_b is a new term $\phi = \psi_a \sqcup \psi_b$, called unifier. All that is true for ψ_a or ψ_b is also true for ϕ ; e.g. if ψ_a describes “a red vehicle” and ψ_b describes “a German minivan” then their unification yields the description “a red German minivan.” Two terms are not unifiable when they possess contradictory information; for instance “a red French vehicle” is not unifiable with “a red German minivan”. The strict definition of unification means that any two descriptions with only one item with contradictory information cannot be unified.

An *amalgam* of two terms (or descriptions) is a new term that contains *parts from these two terms*. For instance, an amalgam of “a red French vehicle” and “a

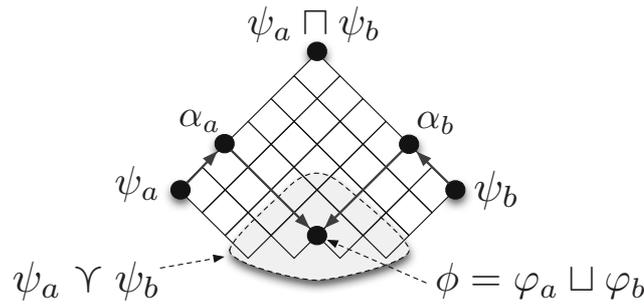


Fig. 2. Illustration of the idea of amalgam between two terms ψ_a and ψ_b

German minivan” is “a red German minivan”; clearly there are always multiple possibilities for amalgams, since “a red French minivan” is another example of amalgam. The notion of amalgam, as a form of partial unification, was formally defined in [13]. For the purposes of this paper, we will introduce a few necessary concepts.

Definition 3. (Amalgam) *The set of amalgams of two terms ψ_a and ψ_b is the set of terms such that:*

$$\psi_a \nabla \psi_b = \{\phi \in \mathcal{L}^+ \mid \exists \alpha_a, \alpha_b \in \mathcal{L} : \alpha_a \sqsubseteq \psi_a \wedge \alpha_b \sqsubseteq \psi_b \wedge \phi = \alpha_a \sqcup \alpha_b\}$$

where $\mathcal{L}^+ = \mathcal{L} - \{\top\}$

Thus, an amalgam of two terms ψ_a and ψ_b is a term that has been formed by unifying two generalizations α_a and α_b such that $\alpha_a \sqsubseteq \psi_a$ and $\alpha_b \sqsubseteq \psi_b$ —i.e. an amalgam is a term resulting from combining some of the information in ψ_a with some of the information from ψ_b , as illustrated in Figure 2. Formally, $\psi_a \nabla \psi_b$ denotes the set of all possible amalgams; however, whenever it does not lead to confusion, we will use $\psi_a \nabla \psi_b$ to denote one specific amalgam of ψ_a and ψ_b .

The terms α_a and α_b are called the *transfers* of an amalgam $\psi_a \nabla \psi_b$. α_a represents all the information from ψ_a which is *transferred* to the amalgam, and α_b is all the information from ψ_b which is transferred into the amalgam. As we will see later, this idea of transfer is akin to the idea of *transferring* knowledge from the source to target in CBR, and also in computational analogy [4].

Intuitively, an amalgam is *complete* when all which can be transferred from both terms into the amalgam has been transferred, i.e. if we wanted to transfer more information, α_a and α_b would not have a unifier.

For the purposes of case reuse, we introduce the notion of asymmetric amalgam, where one term is fixed while only the other term is generalized in order to compute an amalgam.

Definition 4. (Asymmetric Amalgam) *The asymmetric amalgams $\psi_s \vec{\nabla} \psi_t$ of two terms ψ_s (source) and ψ_t (target) is the set of terms such that:*

$$\psi_s \vec{\nabla} \psi_t = \{\phi \in \mathcal{L}^+ \mid \exists \alpha_s \in \mathcal{L} : \alpha_s \sqsubseteq \psi_s \wedge \phi = \alpha_s \sqcup \psi_t\}$$

In an asymmetric amalgam, the target term is transferred completely into the amalgam, while the source term is generalized. The result is a form of partial unification that conserves all the information in ψ_t while relaxing ψ_s by generalization and then unifying one of those more general terms with ψ_t itself. Finally, an asymmetric amalgam is *maximal* when all knowledge in ψ_s that is consistent with ψ_t is transferred to the solution ψ'_t —i.e. $\psi'_t \in \psi_s \xrightarrow{\vec{\gamma}} \psi_t$ is maximal iff $\nexists \psi''_t \in \psi_s \xrightarrow{\vec{\gamma}} \psi_t$ such that $\psi'_t \sqsubset \psi''_t$.

4 A Model of Knowledge Transfer

This section provides a formalization of the idea of knowledge transfer in CBR for the scenarios of single and multi-case retrieval, but only considering problems as a *hard requirement* (see Section 2).

4.1 Knowledge Transfer with Hard Requirements

Let us define the task of knowledge transfer for single case reuse with hard requirements as follows.

Given. A case base $\Delta = \{\psi_1, \dots, \psi_m\}$ and a target description ψ_t

Find. A ‘maximal’ case ψ'_t such that $\psi_t \sqsubset \psi'_t$ (a *conjecture*)

Clearly, if there is some $\psi_i \in \Delta$ such that $\psi_t \sqsubset \psi_i$ then ψ_i is a solution, and the conjecture can be built simply by unifying query and solution: $\psi_t \sqcup \psi_i = \psi_i$. This specific situation is called in CBR literature “solution copy with variable substitution” [8]. Also, notice that the CBI model worries about maximal amalgams, while determining whether such case is *complete* or not corresponds to the whole CBR task and is beyond the scope of the knowledge transfer model.

In general, when there is no case such that $\psi_t \sqsubset \psi_i$, unification is not enough, and knowledge transfer requires the use of amalgams, and in particular of the asymmetric amalgam. Knowledge transfer from a source ψ_s with hard requirements produces *hard conjectures*, defined as follows:

Definition 5. (Hard Transfer) A hard transfer α for target ψ_t from a source ψ_s is a term $\alpha \sqsubseteq \psi_s$ such that $\alpha \sqcup \psi_t \neq \top$, i.e. a generalization of ψ_s that unifies with ψ_t . Thus, the set of hard transfers for target ψ_t from a source ψ_s is: $G(\psi_s, \psi_t) = \{\alpha \in \mathcal{L} \mid \alpha \sqsubseteq \psi_s \wedge \alpha \sqcup \psi_t \neq \top\}$.

Definition 6. (Hard Conjecture) Given a hard transfer $\alpha \in G(\psi_s, \psi_t)$, a conjecture for target ψ_t is a term in $\psi_s \xrightarrow{\vec{\gamma}} \psi_t$ where α is the transfer. The set of hard conjectures K_H for target ψ_t from a source ψ_s is $K_H(\psi_s, \psi_t) = \psi_s \xrightarrow{\vec{\gamma}} \psi_t$.

We will be interested in the most specific conjectures, which are the ones coming from maximal asymmetric amalgams, and as a subset of $G(\psi_s, \psi_t)$. Whether a maximal conjecture is a complete solution is discussed later in Section 4.2

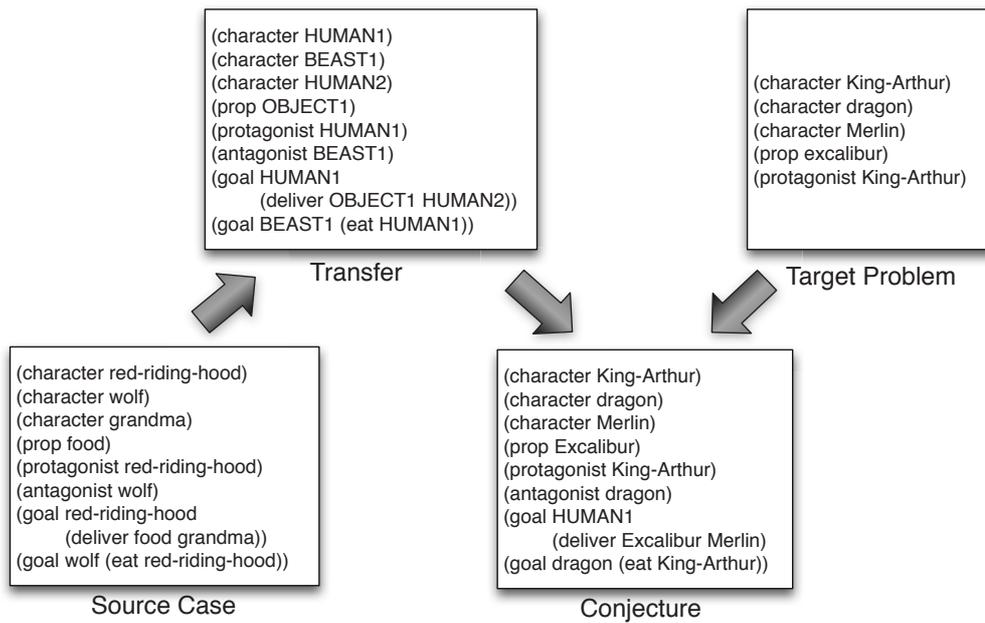


Fig. 3. Exemplification of the concepts of source, target, transfer and conjecture in a story generation domain

In order to illustrate our model with an example let us consider the task of story generation (which has been addressed using CBR by many authors [19]). In this domain, the goal is to generate a story or a story schema (decide which characters exist in the story, which props, which are the goals of the characters, which actions will they perform, etc.). The case base contains a collection of predefined stories, and a problem corresponds to a set of requirements over the story we want the system to generate. We can see, first of all, that there is no clear distinction between problem and solution. A case is just a complete story, whereas a problem is just a partially specified story. Figure 3 illustrates our model showing the following elements: a target problem consisting of an incomplete story specifying three characters (from the King Arthur fantasy world), and asking the system to generate a story that has three characters, King Arthur, Merlin and a dragon, where King Arthur is the main character and where Excalibur is involved. The system happens to retrieve a case with the story of *Little Red Riding Hood* (shown on the bottom left). We don't show the complete case, for space limitations, but in addition to the definitions shown in Figure 3, the retrieved case should contain the list of actions that constitute the plot of the story. We show the transfer, which is a generalization of the retrieved case, and a possible conjecture, which is a unification of the transfer with the target problem. In this example, the resulting story has King Arthur wanting to deliver excalibur to Merlin, while the dragon wants to eat Kind Arthur. We show one possible conjecture, but notice that many different conjectures could be formed here, by transferring different aspects from the retrieved case.

The result of CBI is a *conjecture* in the sense that it is a plausible solution for ψ_t . Notice that, (1) a conjecture may be an incomplete solution, and (2) a

conjecture is not assured to be correct. Moreover, since there may be more than one conjecture, (3) the issue of which conjecture should be selected has also to be specified. Let us review them in turn.

4.2 Conjecture Incompleteness

The purpose of knowledge transfer in case reuse is to transfer to the target as much knowledge as possible (consistent with the target). This “as much as possible” is satisfied if we take as *transfer* a term α that is one of the most specific generalizations of the source that are unifiable with the target, what we called maximal amalgams in Section 3.1. Nevertheless, some information is lost in the generalization path $\psi_s \xrightarrow{\gamma} \alpha$, which corresponded to the *remainder* [14]. Specifically, the remainder $r(\psi, \alpha)$ of a term ψ and a generalization $\alpha \sqsubseteq \psi$ is a term ϕ such that $\alpha \sqcup \phi = \psi$ (and there is no $\phi' \sqsubseteq \phi$ such that $\alpha \sqcup \phi' = \psi$). That which is lost from the source case will be called *source differential* in our model.

Definition 7. (Source Differential) *The source differential ψ_D of a source term ψ_s with respect to a transfer $\alpha \in G(\psi_s, \psi_t)$ is the remainder $r(\psi_s, \alpha)$.*

Notice that, even assuming the source ψ_s to be a consistent and complete case in a case base, now we view the source as having two parts with respect to the target, namely $\psi_s = \alpha \sqcup r(\psi_s, \alpha)$, and only one of this parts (α) is transferred to the target. Therefore, we cannot assume, in general, that the result of case-based inference $\alpha \sqcup \psi_t$ (even when $\alpha \sqcup \psi_t$ is maximal) is a complete solution for the new case (that depends on what is in $r(\psi_s, \alpha)$ and what are the requirements for a solution to be ‘complete’).

Depending on the task a CBR system is performing, this partial solution may be enough. Classical analogy systems take this approach: the goal is to transfer (as much as possible) knowledge from source to target —there is no notion of an externally enforced task that demands some kind of completeness to solutions. Thus, our model of case-based inference encompasses maximal conjectures, but solution completeness is out of its scope, since it depends on the whole CBR process beyond case-based inference.

4.3 Conjecture Correctness

A conjecture $\psi_t \sqcup \alpha$ may be maximal, but even so this might be a correct solution or not with respect to ψ_t . If we see ψ_t as a set of requirements that the complete solved target case must satisfy, then if a conjecture $\psi_t \sqcup \alpha$ is maximal, then the conjecture $\psi_t \sqcup \alpha$ is correct. Although this supplementary assumptions makes sense in theory (if ψ_t expresses the “requirements” to be satisfied), often CBR systems operate in domains where it is not feasible to assure that ψ_t is a complete requirement on the correctness and completeness of solutions; it is more reasonable to assume that ψ_t is a partial requirement and the final acceptability or correctness is left to be assessed by the Revise process.

Therefore, knowledge transfer in case reuse produces a solution that is consistent and maximal, but possibly partial, and not assured to be correct; i.e.

produces a conjecture. Since there are multiple transfers that can produce multiple conjectures, we turn now into the issue of assessing, comparing, and ranking conjectures.

4.4 Conjecture Ordering

Multiplicity of maximal conjectures may have two causes. The first is that $\Gamma(\psi_s, \psi_t)$ might not be unique. The second is when, even if $\Gamma(\psi_s, \psi_t)$ is unique, more than one source is taken into account (as considered in the next section): a set of k precedent cases $\mathbb{P}_k = (\psi_1, \dots, \psi_k)$ produce a set of transfers $\Psi(\mathbb{P}_k) = \Psi_1 \cup \dots \cup \Psi_k$, which in turn generates a set of conjectures.

Conjectures in $K_H(\mathbb{P}_k, \psi_t)$ may be complete, but from a practical point of view it is useful to rank them according to their estimated plausibility, their degree of completeness, or any other heuristic that can be used in a particular application domain. Typically, the Retrieve phase estimates relevance of precedent cases with some similarity measure, so we can use the similarity degrees ($s_1 \geq \dots \geq s_k$) of the k retrieved cases $\mathbb{P}_k = \{\psi_1, \dots, \psi_k\}$ to induce a partial order on the set of transfers: $\langle \Psi(\mathbb{P}_k), \geq \rangle = (\Psi_1 \geq \dots \geq \Psi_k)$. Thus, the conjectures coming from transfers originating in more similar precedent cases (or those transferring more knowledge from more similar cases, in the case of multi-case reuse) are preferred to those from less similar cases.

Since conjectures are in general partial solutions, using some measure that estimates the degree of completeness of conjectures may also be used for ranking conjectures. Domain knowledge can be used to estimate conjecture completeness. In previous work [15] we proposed a measure called *preservation degree* for this purpose. This ranking can be combined with the similarity based ordering to establish a combined partial order on conjectures.

4.5 Knowledge Transfer from Multiple Cases

There are scenarios when the conjecture generated using case-based reasoning is a combination of more than one case in the case base. The intuitive idea in this scenario is that, instead of an asymmetric amalgam where a generalization of a single retrieved case (transfer) is unified with the target problem, we will have an asymmetric amalgam where a generalization of each of the source cases (one transfer per source case) is unified with the target problem. Therefore, instead of a single transfer, we will have multiple transfers (one per source case).

This process can be formally modeled again as an asymmetric amalgam.

Definition 8. (*Hard Conjecture from Multiple Cases*) Given a set of source cases $\{\psi_s^1, \dots, \psi_s^n\}$, a target ψ_t and a set of hard transfers $\alpha_1, \dots, \alpha_n$, where $\alpha_i \in G(\psi_s^i, \psi_t)$, a conjecture for a target ψ_t is a term in $\{\psi_s^1, \dots, \psi_s^n\} \vec{\Upsilon} \psi_t$, where $\alpha_1, \dots, \alpha_n$ are the transfers. The set of hard conjectures K_H for target ψ_t is $K_H(\{\psi_s^1, \dots, \psi_s^n\}, \psi_t) = \{\psi_s^1, \dots, \psi_s^n\} \vec{\Upsilon} \psi_t$.

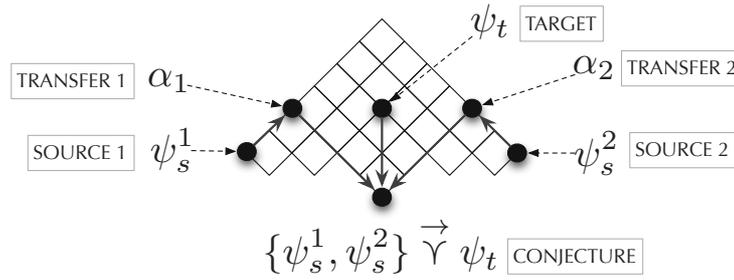


Fig. 4. An schema showing multi-case hard conjecture from two sources ψ_s^1 and ψ_s^2

This idea is illustrated in Figure 4 for the situation of two source cases.

Although $K_H(\{\psi_s^1, \dots, \psi_s^n\}, \psi_t)$ is formally well defined, complexity clearly increases as the number of sources increases, since the number of possible conjectures grows. In practical approaches, a CBR system will typically use a small number of source cases, say 2 or 3, and will use heuristics or domain knowledge that restrict the set of amalgams to consider.

Let us illustrate the idea with the same story generation domain used before. This time, assume that two cases were retrieved: *Little Red Riding Hood* and *Star Wars*. The target problem is the same as the one shown in Figure 3. This time, there will be two different transfers, one from each case, and the conjecture will be the unification of the two transfers with the target problem. For example, if the transfer from *Little Red Riding Hood* is that the dragon wants to eat the main character, and the transfer from *Star Wars* is that the main character wants to learn how to use a sword to defeat the villain and asks another character to train him/her, the resulting story would be the following: King Arthur wants Merlin to train him in the use of Excalibur to defeat the dragon, and the dragon wants to eat King Arthur. Notice, that by transferring from more than one story, there is a wider variety of conjectures that can be formed, and thus, the chances of finding a good solution are also higher.

For the sake of space, in this paper we have only considered the scenario of seeing problems as hard requirements. This means that the conjectures proposed by a CBR system always satisfy the target problem. In the soft requirements scenario, the term representing the problem is considered to just express the preferences over the kind of solutions we want. Therefore, instead of considering asymmetric amalgams, the soft requirements scenario is modeled with the symmetric amalgams, where both the retrieved case and the target problem can be generalized in order to produce the final conjecture. That is, if the system cannot find any solution that completely satisfies the target problem, it can relax the problem, and find a solution that only partially satisfies the target problem.

5 Knowledge Transfer in Analogy

We turn now to discuss how the classic concept of analogical reasoning [4] is related to our model of knowledge transfer, and to CBR in general. It is well accepted that CBR and analogical reasoning are tightly related and share some

common underlying principles [10]. In this section we will see how our model of knowledge transfer underlies both CBR and some forms of analogical reasoning, showing that CBR and analogy indeed share a common underlying formal reasoning mechanism, at least in the limited scope of knowledge transfer.

Computational models of analogy operate by identifying similarities and transferring knowledge between a source domain S and a target domain T . This process can be divided into four stages [6]: 1) *recognition* of a candidate analogous source, S , 2) *elaboration* of an analogical mapping between source domain S and target domain T , 3) *evaluation* of the mapping and inferences, and 4) *consolidation* of the outcome of the analogy for other contexts (i.e. learning). At a superficial level, those 4 processes can be likened to the 4 processes of CBR: retrieve, reuse, revise and retain, although some differences exist. For example, while the reuse process in CBR aims at generating a candidate solution for the problem at hand, the elaboration step in computational analogy limits itself to mapping a source domain to a target domain and proposing candidate inferences (conjectures, in the vocabulary used in this paper). Another piece of evidence that the 4 process of analogy can be likened to those in the CBR cycle is that CBR theoretical frameworks, such as Richter's *knowledge containers* can be applied to analyze computational analogy processes [16].

Moreover, we would like to emphasize that *analogy* is an overloaded concept. The previous 4 step process models the complete cycle of analogical reasoning as understood in cognitive science. However, the term analogical reasoning in mathematics and logic corresponds just to the elaboration step. In the remainder of this paper, we will specifically focus our attention on this elaboration step, which is the most studied in computational models of analogy like SME [4].

5.1 Analogy as a Special Case of Induction

Analogy in the logical sense is typically defined as the process of transferring knowledge or inferences from a particular source domain S to another particular target domain T . John Stuart Mill [12, Ch. XX] argued that analogy is simply a special case of induction. In his view, analogy could be reduced to: "Two things resemble each other in one or more respects; a certain proposition is true of the one; therefore it is true of the other". That is to say, analogy between two situations S and T can be interpreted as having two steps:

Inductive Step: In a first step we perform an inductive leap. Assume that S and T are similar in some aspects, we will use anti-unification $S \sqcap T$ to denote all the information shared between S and T (i.e. that in which they are similar). Now, given a proposition α which is true in S (i.e. $\alpha \sqsubseteq S$) but we don't know if it is true in T , we assume (inductive leap) that $S \sqcap T$ is the cause of α —i.e. $S \sqcap T$ implies α (which we will write as $S \sqcap T \rightarrow \alpha$).

Deductive Step: Then, in a second step, we apply the inductively derived assumption $S \sqcap T \rightarrow \alpha$ to derive that α is also true in T , and conclude $\alpha \sqcup T$ is true (i.e. the target with the added piece of knowledge α is also true).

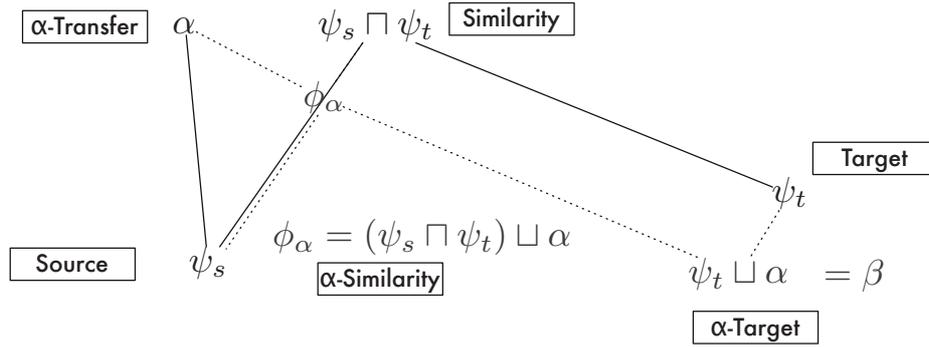


Fig. 5. Subsumption relations among the terms involved in analogy $\psi_s \xrightarrow[\psi_t]{a} \beta$

Let us illustrate this analogical reasoning principle with a typical example. Let us consider our solar system as the source domain T , and Bohr’s model of the atom as the target domain S . Both domains are similar in some aspects, $S \sqcap T =$ “There are smaller elements orbiting a larger element in the center”. We know that the following statement is true for the solar system S : $\alpha =$ “there is an attraction force between the small elements and the larger element in the center”. We can now use the previous model of analogical reasoning in the following way. In the first (inductive) step of analogy we reach the following assumption: $S \sqcap T \rightarrow \alpha$ (which means that the fact that there are elements orbiting is enough to conclude that there is an attraction force). In the second (deductive) step, we apply $S \sqcap T \rightarrow \alpha$ to T and conclude that there in Bohr’s model of the atom there must also be an attraction force between the small elements and the larger element in the center. By adding this new piece of information to T , now we can conclude $\alpha \sqcup T$, that represents the model of the atom with the added piece of knowledge referring to the attraction force.

5.2 Knowledge Transfer in Analogy

This view of analogy can be defined as follows in a generalization space.

Definition 9. Given two terms $\psi_s, \psi_t \in \mathcal{L}$ (called source and target respectively) a formula $\beta \neq \top$ is derived by analogy whenever:

1. $\exists \alpha : \alpha \sqsubseteq \psi_s \wedge \alpha \not\sqsubseteq (\psi_s \sqcap \psi_t)$ (α is true in source only)
2. $\beta = \alpha \sqcup \psi_t$ (knowledge α is transferred to target)

where α is the knowledge transferred from source to target.

Since $\alpha \not\sqsubseteq \psi_s \sqcap \psi_t$ we cannot (deductively) derive that α is true in ψ_t . Therefore, this analogical reasoning requires an inductive step, which can be seen as a defeasible or conjectural inference. This “inductive analogy” model is illustrated in Figure 5. The solid lines depict sound inferences, i.e. the subsumption relationships among terms (ψ_s, ψ_t and α). Analogy makes some conjectural inferences shown as dotted lines. Specifically, if α is not inconsistent with ψ_t (i.e. $\alpha \sqcup \psi_t \neq \top$) then possibly both situations may also have α in common; this is represented as the term $\phi_\alpha = \alpha \sqcup \psi_t$ that is conjectured to be true. Now,

assuming ϕ_α is true, and ψ_t is true, we can then conjecture that $\beta = \psi_t \sqcup \alpha$ is true (i.e. that α can be “transferred to” ψ_t).

This conjectural inference can be seen in two ways: induction or knowledge transfer, that nonetheless are equivalent. In the knowledge transfer approach, we derive β by conjecturing α is also true in the target (i.e. we derive $\alpha \sqcup \psi_t$); i.e. we use the idea of asymmetric amalgam to derive β by transferring α to ψ_t .

In the inductive model of analogy we conjecture that the implicit generalization should also include α as being true (that is we move from $\psi_s \sqcap \psi_t$ to ϕ_α). Later, since the target also shares $\psi_s \sqcap \psi_t$, we can (deductively) infer that α is true in the target. Figure 5 shows how both views reach the same conjecture.

5.3 Analogy and Case-Based Reasoning

The “inductive analogy” model sheds some light on the nature of analogical reasoning, and also provides insights on how to assess when the conclusions reached by analogy are stronger or weaker.

Conclusions reached by analogy are considered strong when the similarity between source and target is high. In Stuart Mill’s words: “[...] it follows that where the resemblance is very great, the ascertained difference very small, and our knowledge of the subject-matter tolerably extensive, the argument from analogy may approach in strength very near to a valid induction”. This follows from his inductive view of analogy, because if source and target are not very similar, then $S \sqcap T$ would contain very little information, and thus the rule $S \sqcap T \rightarrow \alpha$ reached by induction would most likely be an over generalization. Moreover, if $S \sqcap T$ contains a lot of information, then $S \sqcap T \rightarrow \alpha$ would be a rule with a very narrow scope (only applicable to those domains satisfying $S \sqcap T$), and thus more likely to be correct.

As stated above, conclusions reached by analogy can be seen as a knowledge transfer process from the source domain to the target domain. It should be now clear that the principles underlying knowledge transfer are a special case of analogical reasoning. This can be seen when we bear in mind that the assumption behind CBR, namely “similar problems have similar solutions”, is just a special case of the analogical reasoning principle: “if two things resemble each other in one or more respects; a certain proposition is true of the one; therefore it is true of the other”. Moreover, the second principle of analogy, stating that analogical reasoning reaches stronger conclusion when the two domains are more similar, explains the principle behind the most common approaches to case retrieval in CBR, that simply look for the most similar case to the problem at hand.

Notice that what we are stating is that knowledge transfer (and thus CBI) is a special case of analogical reasoning, not that the whole case-based reasoning paradigm is. Moreover, we also state that the goal of the retrieval step in CBR should be to provide a source domain from which the conclusions reached by analogy (knowledge transfer) are stronger. Solution adaptation, typically domain dependent, is not explained by analogical reasoning, and constitutes the main theoretical difference between CBR and computational models of analogy in cognitive science.

6 Discussion

This paper has presented a model of knowledge transfer in case-based inference based on the idea of partial unification. We have focused on cases are represented as terms in a generalization space. In our model, case reuse is seen as having two steps: a first step (case-based inference) where knowledge is transferred from one or several source cases to the target case (called a conjecture), and a second step (adaptation) where the conjecture might need to be adapted. This paper has focused on a model of the first step.

Our model of knowledge transfer offers insights on the relation between case reuse and analogical reasoning. Previous work on relating CBR with analogy has focused on superficial aspects such as CBR being typically intra-domain, whereas analogy is inter-domain [18]. In our model, we can see that analogical reasoning is related to the knowledge transfer step of case reuse rather than with the second (adaptation) step. The work of Prade and Richard [17] is an exception, and proposed a Boolean model of analogical reasoning and suggested it could be used for adaptation in CBR, following an inductive view of analogy as we presented above. An interesting line of future work is the relation of knowledge transfer with conceptual blending [5]. We have seen that analogy can be likened to an asymmetric amalgam, whereas conceptual blending could be seen as a form of symmetric amalgam.

Our work is related to existing general models of case reuse, like [2]. However, such models focus on the adaptation step, and typically obviate the process of knowledge transfer (transfer is seen as a mere “solution copy” from source to target). We believe that this oversimplification of the knowledge transfer problem is at the root of the difficulty of finding general models of multi-case reuse. The work presented in this paper is a step towards that direction, since it can easily cope with transferring knowledge from multiple sources.

Also related is the work on case-based inference [7], but they focus on prediction (classification and regression tasks where the outcome is a form of similarity-based inference). The difference with our work is that we have focused on how complex solutions and conjectures can be formed by transferring knowledge from one or multiple source cases to a partial solution of a target case.

Part of our long term goal is understanding case reuse and its relation to other forms of reasoning. We envision case-based inference as a form of conjectural or defeasible inference, like other forms of non-monotonic reasoning (induction, abduction and hypothetical reasoning). The model presented in this paper is one step towards this goal. As future work, we want to formalize the process of knowledge transfer from multiple source cases, and develop case reuse methods based on the idea of knowledge transfer.

Finally, although the model presented in this paper is theoretical, practical implementations of the underlying principles are possible. For example, our previous work on similarity measures over generalization spaces [14], and on case adaptation in multiagent systems using amalgams [11] are steps in this direction.

Acknowledgements. This research was partially supported by project Next-CBR (TIN2009-13692-C03-01).

References

- [1] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications* 7(1), 39–59 (1994)
- [2] Bergmann, R., Wilke, W.: Towards a new formal model of transformational adaptation in case-based reasoning. In: *European Conference on Artificial Intelligence (ECAI 1998)*, pp. 53–57. John Wiley and Sons (1998)
- [3] Cojan, J., Lieber, J.: Belief Merging-Based Case Combination. In: McGinty, L., Wilson, D.C. (eds.) *ICCBR 2009. LNCS*, vol. 5650, pp. 105–119. Springer, Heidelberg (2009)
- [4] Falkenhainer, B., Forbus, K.D., Gentner, D.: The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41, 1–63 (1989)
- [5] Fauconnier, G.: Conceptual blending and analogy. In: Gentner, D., Holyoak, K.J., Kokinov, B.K. (eds.) *The analogical mind Perspectives from cognitive science*, pp. 255–285. MIT Press (2001)
- [6] Hall, R.P.: Computational approaches to analogical reasoning: a comparative analysis. *Artificial Intelligence* 39(1), 39–120 (1989)
- [7] Hüllermeier, E.: *Case-Based Approximate Reasoning. Theory and Decision Library*, vol. 44. Springer (2007)
- [8] Kolodner, J.: *Case-based reasoning*. Morgan Kaufmann (1993)
- [9] van der Laag, P.R.J., Nienhuys-Cheng, S.H.: Completeness and properness of refinement operators in inductive logic programming. *Journal of Logic Programming* 34(3), 201–225 (1998)
- [10] Mántaras, R.L.D., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. *Knowl. Eng. Rev.* 20(3), 215–240 (2005)
- [11] Manzano, S., Ontañón, S., Plaza, E.: Amalgam-Based Reuse for Multiagent Case-Based Reasoning. In: Ram, A., Wiratunga, N. (eds.) *ICCBR 2011. LNCS*, vol. 6880, pp. 122–136. Springer, Heidelberg (2011)
- [12] Mill, J.S.: *The Collected Works of John Stuart Mill*, vol. 7. Liberty Fund (2006)
- [13] Ontañón, S., Plaza, E.: Amalgams: A Formal Approach for Combining Multiple Case Solutions. In: Bichindaritz, I., Montani, S. (eds.) *ICCBR 2010. LNCS*, vol. 6176, pp. 257–271. Springer, Heidelberg (2010)
- [14] Ontañón, S., Plaza, E.: Similarity measure over refinement graphs. *Machine Learning Journal* 87, 57–92 (2012)
- [15] Ontañón, S., Plaza, E.: Toward a knowledge transfer model of case-based inference. In: *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society (FLAIRS)*. AAAI Press (2012)
- [16] Ontañón, S., Zhu, J.: On the role of domain knowledge in analogy-based story generation. In: *IJCAI*, pp. 1717–1722 (2011)
- [17] Prade, H., Richard, G.: Analogy-Making for Solving IQ Tests: A Logical View. In: Ram, A., Wiratunga, N. (eds.) *ICCBR 2011. LNCS*, vol. 6880, pp. 241–257. Springer, Heidelberg (2011)
- [18] Seifert, C.M.: Analogy and case-based reasoning. In: *Proc. of a Workshop on Case-Based Reasoning, Pensacola Beach, FL*, pp. 125–129 (1989)
- [19] Turner, S.R.: *Minstrel: a computer model of creativity and storytelling*. Ph.D. thesis. University of California at Los Angeles, Los Angeles, CA, USA (1993)