The Use of Reformation to Repair Faulty Analogical Blends

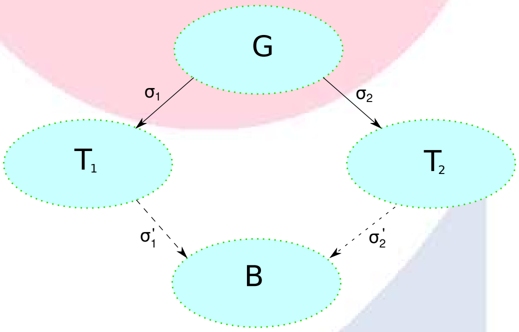
Citation for published version:
Bundy, A & Maclean, E 2016, The Use of Reformation to Repair Faulty Analogical Blends. in UK Ontology Network 2016. Newcastle, UK Ontology Network 2016, Newcastle upon Tyne, United Kingdom, 14/04/16.

Published In:
UK Ontology Network 2016

The Use of Reformation to Repair Faulty Analogical Blends

Alan Bundy
A.Bundy@ed.ac.uk
http://homepages.inf.ed.ac.uk/bundy/

An analogical blend is the formation of a new concept from two old ones, e.g., houseboat or boathouse from house and boat, depending on whether you align the boat with the house or its occupant. This can be automated via the colimit algorithm from Category Theory, applied to two logical theories, plus an initial alignment between the terms in each theory. This is illustrated by the following diagram.



T_1 and T_2 are the parent theories, and B is the blend constructed from them. Alignments between concepts in each of the two parents are given by the morphisms σ_1 and σ_2 between the general theory, G, and the two parent theories, T_1 and T_2. The colimit algorithm then constructs the morphisms σ'_1 and σ'_2, which together define B. Analogical blends are often faulty, e.g. inconsistent or incomplete, so need to be repaired. For instance, the following two parent database theories T_1 and T_2 have been blended into the merged theory B, where Own & $Sold\_To$ are aligned and so are $Part\_Num$ & Ser_Num.

T_1

$Own(Cust_a, Prod_a),\ Part\_Num(Prod_a) = 123$
$Own(Cust_b, Prod_b),\ Part\_Num(Prod_b) = 123$
$Prod_a \neq Prod_b$

T_2

$Sold\_To(Cust_c, Prod_c),\ Ser\_Num(Prod_c) = 234$
$Ser\_Num(x) = Ser\_Num(y) \implies x = y$

B

$Sold\_To(Cust_a, Prod_a),\ Ser\_Num(Prod_a) = 123$
$Sold\_To(Cust_b, Prod_b),\ Ser\_Num(Prod_b) = 123$
$Sold\_To(Cust_c, Prod_c),\ Ser\_Num(Prod_c) = 234$
$Ser\_Num(x) = Ser\_Num(y) \implies x = y$
$Prod_a \neq Prod_b$

Unfortunately, B is inconsistent. The error was to align $Part\_Num$ from T_1 with Ser_Num from T_2. Part numbers name a particular kind of product, whereas serial numbers are unique to each instance of a product. $Prod_a$ and $Prod_b$ are two different instances of the same product. The problematic parts of the blend B are highlighted in red. The following proof of falsity ($\perp$) shows the inconsistency of B.

$$\frac{\frac{\frac{\frac{Ser\_Num(x) = Ser\_Num(y) \implies x = y \quad Prod_a \neq Prod_b}{Ser\_Num(Prod_a) \neq Ser\_Num(Prod_b)} \quad Ser\_Num(Prod_b) = 123}{Ser\_Num(Prod_a) \neq 123} \quad Ser\_Num(Prod_a) = 123}{123 \neq 123} \quad z = z}{\perp}$$

At each proof step, a pair of blue or red expressions are unified. Reformation is an algorithm we have developed for the diagnosis and repair of faulty logical theories, such as B. It is an adaption of the unification algorithm. Unification steps are paired. Paired steps both apply to input in the same syntactic form, but with inverse pre-conditions: one step leading to success and one to failure. Faulty theories can be repaired by analysis of either the derivation of false conjectures or the failed proofs of true conjectures. A key unification step is inverted by changing a failed step into a successful one, or vice versa. This is realised by changing the theory so that its partner step in the pair is triggered instead of it. We are investigating the application of reformation to the diagnosis and repair of faulty analogical blends.

In the inconsistency proof above, reformation can prohibit the unification of the pair of red expressions by replacing one of the two red occurrences of Ser_Num with a different property, e.g., $Part\_Num$. This can be implemented by withdrawing the alignment between $Part\_Num$ and Ser_Num. The colimit operation now generates the new analogical blend $\nu(B)$, pronounced 'new B'. The corrected parts of $\nu(B)$ are highlighted in green. In $\nu(B)$, the inconsistency proof above will now fail at the red step.

$\nu(B)$

$Sold\_To(Cust_a, Prod_a),\ Part\_Num(Prod_a) = 123$
$Sold\_To(Cust_b, Prod_b),\ Part\_Num(Prod_b) = 123$
$Sold\_To(Cust_c, Prod_c),\ Ser\_Num(Prod_c) = 234$
$Ser\_Num(x) = Ser\_Num(y) \implies x = y$
$Prod_a \neq Prod_b$

Both colimit and reformation are generic algorithms that have widespread applications in areas as diverse as ontology merging, program debugging and cognitive science. We will be exploring diverse applications in Human-Like Computing.

References

[Eppe et al., 2015] Eppe, M., Confalonieri, R., Maclean, E., Kaliakatsos, M., Cambouropoulos, E., Codescu, M., Schorlemmer, M., and Kühnberger, K. (2015). Computational invention of cadences and chord progressions by conceptual chord-blending. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.