Math. Struct. in Comp. Science (2016), *vol.* 26, *pp.* 655–657. © Cambridge University Press 2014
doi:10.1017/S0960129514000280 First published online 10 November 2014

General conditions for full abstraction

JOACHIM PARROW

Department of Information Technology
Uppsala University, Uppsala, Sweden
Email: joachim@it.uu.se

Received 24 January 2014; revised 8 April 2014

Full abstraction, i.e. that a function preserves equivalence from a source to a target, has been used extensively as a correctness criterion for mappings between models of computation. I here show that with fixed equivalences, fully abstract functions almost always exist. Also, with the function and one of the equivalences fixed the other equivalence can almost always be found.

1. Introduction

A function $f : \mathbf{S} \to \mathbf{T}$ is fully abstract with respect to equivalences $\simeq_{\mathbf{S}}$ and $\simeq_{\mathbf{T}}$ on $\mathbf{S}$ and $\mathbf{T}$ respectively if, intuitively, f maps $\simeq_{\mathbf{S}}$ to $\simeq_{\mathbf{T}}$. As discussed at length in Gorla and Nestmann (2014), full abstraction has been used extensively as a correctness criterion when comparing models for concurrency. They point out that on occasion the mere existence of a fully abstract function has been considered evidence of relative expressiveness of such models, and argue that this view can be dangerous. In this short paper, I support their conclusion by showing that given any two elements of a triple $(f, \simeq_{\mathbf{S}}, \simeq_{\mathbf{T}})$, it is almost always possible to find a third element to satisfy full abstraction. More precisely:

1. Given $\simeq_{\mathbf{S}}$ and $\simeq_{\mathbf{T}}$, there exists f unless $\simeq_{\mathbf{T}}$ has strictly fewer equivalence classes than $\simeq_{\mathbf{S}}$.
2. Given f and $\simeq_{\mathbf{T}}$ there always exists $\simeq_{\mathbf{S}}$.
3. Given f and $\simeq_{\mathbf{S}}$, there exists $\simeq_{\mathbf{T}}$ unless f maps two $\simeq_{\mathbf{S}}$-inequivalent elements to the same element of $\mathbf{T}$.

This in no way diminishes the value of a full abstraction result for a *particular* triple under consideration. It can still be regarded as a correctness criterion, in the same way as proving that a mapping between formalisms preserves deadlock or divergence properties. My main point here is that in isolation such a result is not very informative if any one component of the triple can be chosen freely.

2. Results

Definition 1. Let $\mathbf{S}$ and $\mathbf{T}$ be two sets. Let $\simeq_{\mathbf{S}}$ be an equivalence relation on $\mathbf{S}$ and $\simeq_{\mathbf{T}}$ be an equivalence relation on $\mathbf{T}$. Let $f : \mathbf{S} \to \mathbf{T}$ be a (total) function. Then $(f, \simeq_{\mathbf{S}}, \simeq_{\mathbf{T}})$ is *fully abstract* if for all $s_1, s_2 \in \mathbf{S}$ it holds that

$$s_1 \simeq_{\mathbf{S}} s_2 \quad \Leftrightarrow \quad f(s_1) \simeq_{\mathbf{T}} f(s_2).$$

We write $\mathbf{S}/\simeq_\mathbf{S}$ to mean the set of equivalence classes of $\simeq_\mathbf{S}$ and $[s]_{\simeq_\mathbf{S}}$ to mean the equivalence class of $\simeq_\mathbf{S}$ to which s belongs, and similarly for $\simeq_\mathbf{T}$.

Theorem 1 (existence of f given $\simeq_\mathbf{S}$ and $\simeq_\mathbf{T}$). Let $\mathbf{S}$ and $\mathbf{T}$ be sets with equivalence relations $\simeq_\mathbf{S}$ and $\simeq_\mathbf{T}$ respectively. Then there exists a function $f : \mathbf{S} \to \mathbf{T}$ such that $(f, \simeq_\mathbf{S}, \simeq_\mathbf{T})$ is fully abstract if and only if the cardinality of $\mathbf{T}/\simeq_\mathbf{T}$ is greater than or equal to the cardinality of $\mathbf{S}/\simeq_\mathbf{S}$.

Proof. (If). Assume the cardinality of $\mathbf{T}/\simeq_\mathbf{T}$ is greater than or equal to the cardinality of $\mathbf{S}/\simeq_\mathbf{S}$. Then there exists an injection $g : \mathbf{S}/\simeq_\mathbf{S} \to \mathbf{T}/\simeq_\mathbf{T}$. Define $f : \mathbf{S} \to \mathbf{T}$ by letting $f(s)$ be an arbitrary member of $g([s]_{\simeq_\mathbf{S}})$. In other words take the equivalence class of s, apply g to it, and choose an arbitrary element. (To be strict this construction assumes the axiom of choice.) Then $(f, \simeq_\mathbf{S}, \simeq_\mathbf{T})$ is fully abstract: if $s_1 \simeq_\mathbf{S} s_2$ then $[s_1]_{\simeq_\mathbf{S}} = [s_2]_{\simeq_\mathbf{S}}$ so by construction $f(s_1) \simeq_\mathbf{T} f(s_2)$. If $s_1 \not\simeq_\mathbf{S} s_2$ then they belong to different equivalence classes and $g([s_1]_{\simeq_\mathbf{S}}) \neq g([s_2]_{\simeq_\mathbf{S}})$ since g is an injection, and since different equivalence classes are disjoint we have $f(s_1) \not\simeq_\mathbf{T} f(s_2)$.

(Only if). Assume $(f, \simeq_\mathbf{S}, \simeq_\mathbf{T})$ is fully abstract. Then for all equivalence classes $S \in \mathbf{S}/\simeq_\mathbf{S}$ it holds that for all $s_1, s_2 \in S$, $f(s_1) \simeq_\mathbf{T} f(s_2)$. So we can uniquely define $g : \mathbf{S}/\simeq_\mathbf{S} \to \mathbf{T}/\simeq_\mathbf{T}$ by $g(S) = T$ if for all $s \in S$ it holds that $f(s) \in T$. Now assume two equivalence classes S_1 and S_2 such that $g(S_1) = g(S_2)$. Then for $s_1 \in S_1$ and $s_2 \in S_2$ it holds that $f(s_1) \simeq_\mathbf{T} f(s_2)$, so by full abstraction $s_1 \simeq_\mathbf{S} s_2$, whence $S_1 = S_2$. Thus g is an injection, and proves that the cardinality of $\mathbf{T}/\simeq_\mathbf{T}$ is greater than or equal to the cardinality of $\mathbf{S}/\simeq_\mathbf{S}$. □

Theorem 2 (existence of $\simeq_\mathbf{S}$ given f and $\simeq_\mathbf{T}$). Let $\mathbf{S}$ and $\mathbf{T}$ be sets, $\simeq_\mathbf{T}$ an equivalence relation on $\mathbf{T}$, and $f : \mathbf{S} \to \mathbf{T}$. Then there exists an equivalence relation $\simeq_\mathbf{S}$ on $\mathbf{S}$ such that $(f, \simeq_\mathbf{S}, \simeq_\mathbf{T})$ is fully abstract.

Proof. Define $\simeq_\mathbf{S}$ by $s_1 \simeq_\mathbf{S} s_2$ if $f(s_1) \simeq_\mathbf{T} f(s_2)$, then $\simeq_\mathbf{S}$ is an equivalence relation since $\simeq_\mathbf{T}$ is one, and f is fully abstract by definition. □

Definition 2. Let f be a function with domain $\mathbf{S}$ and $\simeq_\mathbf{S}$ an equivalence on $\mathbf{S}$. Then f *respects* $\simeq_\mathbf{S}$ if for all $s_1, s_2 \in \mathbf{S}$ it holds $s_1 \not\simeq_\mathbf{S} s_2 \Rightarrow f(s_1) \neq f(s_2)$.

Theorem 3 (existence of $\simeq_\mathbf{T}$ given f and $\simeq_\mathbf{S}$). Let $\mathbf{S}$ and $\mathbf{T}$ be sets, $\simeq_\mathbf{S}$ an equivalence relation on $\mathbf{S}$, and $f : \mathbf{S} \to \mathbf{T}$. Then there exists an equivalence relation $\simeq_\mathbf{T}$ on $\mathbf{T}$ such that $(f, \simeq_\mathbf{S}, \simeq_\mathbf{T})$ is fully abstract if and only if f respects $\simeq_\mathbf{S}$.

Proof. (If) Define $\simeq'_\mathbf{T}$ by $t_1 \simeq'_\mathbf{T} t_2$ if there exist $s_1, s_2 \in \mathbf{S}$ such that $f(s_1) = t_1, f(s_2) = t_2$ and $s_1 \simeq_\mathbf{S} s_2$. We first prove that $\simeq'_\mathbf{T}$ is a partial equivalence relation on $\mathbf{T}$. Obviously $\simeq'_\mathbf{T}$ is reflexive and symmetric on the image of f since $\simeq_\mathbf{S}$ is reflexive and symmetric. For transitivity, assume $t_1 \simeq'_\mathbf{T} t_2$ and $t_2 \simeq'_\mathbf{T} t_3$. By the first equivalence there are $s_1, s_2 \in \mathbf{S}$ such that $f(s_1) = t_1, f(s_2) = t_2$ and $s_1 \simeq_\mathbf{S} s_2$. By the second equivalence there are $s_3, s_4 \in \mathbf{S}$ such that $f(s_3) = t_2, f(s_4) = t_3$ and $s_3 \simeq_\mathbf{S} s_4$. We have $f(s_2) = f(s_3) = t_2$, thus since f respects $\simeq_\mathbf{S}$ we get $s_2 \simeq_\mathbf{S} s_3$, which through transitivity of $\simeq_\mathbf{S}$ gives that $s_1 \simeq_\mathbf{S} s_4$, and thus by definition of $\simeq'_\mathbf{T}$ that $t_1 \simeq'_\mathbf{T} t_3$. In conclusion, $\simeq'_\mathbf{T}$ is an equivalence on $\mathbf{T}$ restricted to the image of f. Extend $\simeq'_\mathbf{T}$ to $\simeq_\mathbf{T}$ on all of $\mathbf{T}$ by adding all members of $\mathbf{T}$ not in the image of f to any equivalence class in $\simeq'_\mathbf{T}$. Thus $\simeq_\mathbf{T}$ is an equivalence on all of $\mathbf{T}$. To establish

https://doi.org/10.1017/S0960129514000280 Published online by Cambridge University Press

full abstraction, direction $\Rightarrow$ follows directly from the definition of $\simeq'_{\mathbf{T}}$. For direction $\Leftarrow$ assume that $f(s_1) \simeq'_{\mathbf{T}} f(s_2)$. Then there are s_3 and s_4 such that $f(s_3) = f(s_1), f(s_4) = f(s_2)$ and $s_3 \simeq_{\mathbf{S}} s_4$. Since f respects $\simeq_{\mathbf{S}}$ we get $s_3 \simeq_{\mathbf{S}} s_1$ and $s_4 \simeq_{\mathbf{S}} s_2$. Since $\simeq_{\mathbf{S}}$ is an equivalence, we conclude $s_1 \simeq_{\mathbf{S}} s_2$.

(Only if) Let there be s_1, s_2 such that $s_1 \not\simeq_{\mathbf{S}} s_2$ and $f(s_1) = f(s_2)$. Let $\simeq_{\mathbf{T}}$ be any equivalence relation on $\mathbf{T}$. Then since $\simeq_{\mathbf{T}}$ is reflexive, we have $f(s_1) \simeq_{\mathbf{T}} f(s_2)$, which with $s_1 \not\simeq_{\mathbf{S}} s_2$ contradicts full abstraction. □

3. Discussion

In retrospect, the theorems here appear so trivial and the proofs so obvious that it is almost surprising they have not been presented previously. They were developed in January 2013 when reading early versions of Gorla and Nestmann (2014), and I am greatly indebted to Gorla and Nestmann for discussions on the role of full abstraction in comparing models of concurrency. In particular they refer to a result by Beauxis *et al.* (2008) that there is a fully abstract encoding between Turing machines and finite automata with their respective language equivalences. Their proof simply goes by lining up the equivalence classes, which in both cases are countably many, and was an inspiration for Theorem 1 above.

My colleague Tjark Weber has formalized the definitions and proofs of Theorems 1–3 in the interactive theorem prover Isabelle/HOL. The proofs follow my sketches above and comprise 187 lines (the pdf output file is four pages) of Isabelle code, and required around six hours of his time. In this he also devised some small optimizations. The time that I spent on this effort myself is hard to estimate since it was done intermittently over a year, but it is surely not less than a week in total. It is an interesting observation that the added effort of formalizing the proofs is comparatively small. The added value is twofold: it establishes beyond doubt that the results are correct despite some sweeping statements in the sketches, and it provides a basis for developments and variants. A conclusion is that a theorem prover in projects such as this is a valuable and regrettably underused tool.

References

Beauxis, R., Palamidessi, C. and Valencia, F. D. (2008) On the asynchronous nature of the asynchronous pi-calculus. In: Degano, P., De Nicola, R. and Meseguer, J. (eds.) Concurrency, Graphs and Models. *Springer Lecture Notes in Computer Science* **5065** 473–492.

Gorla, D. and Nestmann, U. (2014) Full abstraction for expressiveness: History, myths and facts. In this issue of *Mathematical Structures in Computer Scinece.*

https://doi.org/10.1017/S0960129514000280 Published online by Cambridge University Press