The Associativity of Equivalence and the Towers of Hanoi Problem

Roland Backhouse* Maarten Fokkinga†

Dedicated to Edsger W. Dijkstra
from whom we have learnt a great deal.

Abstract

Dijkstra and Scholten have argued that greater use should be made of the associativity of equivalence. This note shows how the property is used in specifying the rotation of the disks in the well-known Towers of Hanoi problem.

Keywords: propositional calculus, program correctness.

1 Introduction

Equality is transitive. This important fact is reflected by the *conjunctional* use of the infix equality symbol in continued expressions — a "continued equality" is an expression of the form $a=b=c$ and it means $a=b$ and $b=c$. The (not insignificant) notational advantage of the continued equality is the avoidance of repeating the subexpression b (and thus the avoidance of the need to check that the two occurrences are indeed identical).

Equivalence of boolean values is also transitive and, for this reason, many logicians, mathematicians and computing scientists would interpret a continued equivalence conjunctionally. That is, with the understanding that the symbol "$\equiv$" denotes equivalence, many would interpret an expression of the form $p\equiv q\equiv r$ as the conjunction $(p\equiv q)\wedge(q\equiv r)$.

In the early nineteen eighties, Dijkstra and Scholten (see eg. [DS90]) remarked that equivalence, unlike equality in general, is also associative. Thus $p\equiv q\equiv r$ can be read as $(p\equiv q)\equiv r$ or (equivalently) as $p\equiv(q\equiv r)$ in just the same way as we are used to reading $i+j+k$ or $l\times m\times n$. That the conjunctional and associative readings are quite different is illustrated by the expression $\mathsf{false}\equiv\mathsf{false}\equiv\mathsf{true}$. Read conjunctionally this is

*School of Computer Science and IT, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, UK. rcb@cs.nott.ac.uk

†Department of Computer Science, University of Twente, PO Box 217, NL 7500 AE Enschede, The Netherlands. fokkinga@cs.utwente.nl

$(\mathsf{false}\equiv\mathsf{false})\wedge(\mathsf{false}\equiv\mathsf{true})$, which simplifies to false, whereas read associatively this is $(\mathsf{false}\equiv\mathsf{false})\equiv\mathsf{true}$ which simplifies to true. (Example due to Gries and Schneider [GS93].)

With the postulate that continued equivalences must be read associatively, Dijkstra and Scholten went on to present a very elegant axiomatisation of the propositional calculus. The beauty of their presentation lies in the fact that almost all of the axioms can be parsed in more than one way so that, in effect, each axiom captures more than one property of the operators involved.

A very, very beautiful example is the following delightfully simple and easily remembered property of the predicate even on numbers: for all numbers m and n,

$$\mathsf{even}(m+n) \;\equiv\; \mathsf{even}(m) \;\equiv\; \mathsf{even}(n) \quad .$$

Note that this must be read associatively, not conjunctionally. Doing so leads to a different property for each different way that we parse the formula. One parsing gives the property that even distributes through addition:

$$\mathsf{even}(m+n) \equiv (\mathsf{even}(m)\equiv\mathsf{even}(n)) \quad .$$

The alternative parsing gives the property that adding a number n to any number m has no effect on the evenness of the number if and only if n is even:

$$(\mathsf{even}(m+n) \;\equiv\; \mathsf{even}(m)) \;\equiv\; \mathsf{even}(n) \quad .$$

The fact that equivalence is associative was, inevitably, “well-known” long before the publication of Dijkstra and Scholten’s book. Indeed, it is mentioned by Alfred Tarski in the paper “On the primitive term of logistic” [Tar56] where it is attributed to J. Lukasiewicz[1]. There is also a growing number of examples of the effective use of the associativity of equivalence in mathematical and computing science problems. These include Wiltink’s [Wil87] solution of a number of challenging logic puzzles, the design of an electronic circuit for a full adder in Gries and Schneider’s text [GS93], and Sybrand Dijkstra’s solution of a synchronisation problem (documented by Edsger W. Dijkstra in EWD904[2]). Even so, judging by textbooks that have been published in recent years, most computing scientists are ignorant of the property. Indeed, equivalence is still most commonly couched as “if and only if”, thus the conjunction of two other operators rather than as an operator in its own right. A notable exception is Gries and Schneider’s introductory text [GS93] which follows Dijkstra and Scholten’s lead.

This paper discusses another example of the exploitation of the associativity of equivalence. The context is the well-known Towers of Hanoi problem, a problem that is often used in computing science texts to illustrate the power of recursion. The next section summarises two solutions to the problem: the well-known recursive solution and a less well-known iterative solution. The problem that concerns us in this paper is showing that the recursive and iterative solutions are equivalent.

---

[1] We are grateful to Peter Hancock for pointing out this reference.

[2] The series of “EWD’s” is available from `http://www.cs.utexas.edu/users/ham/scanning/`.

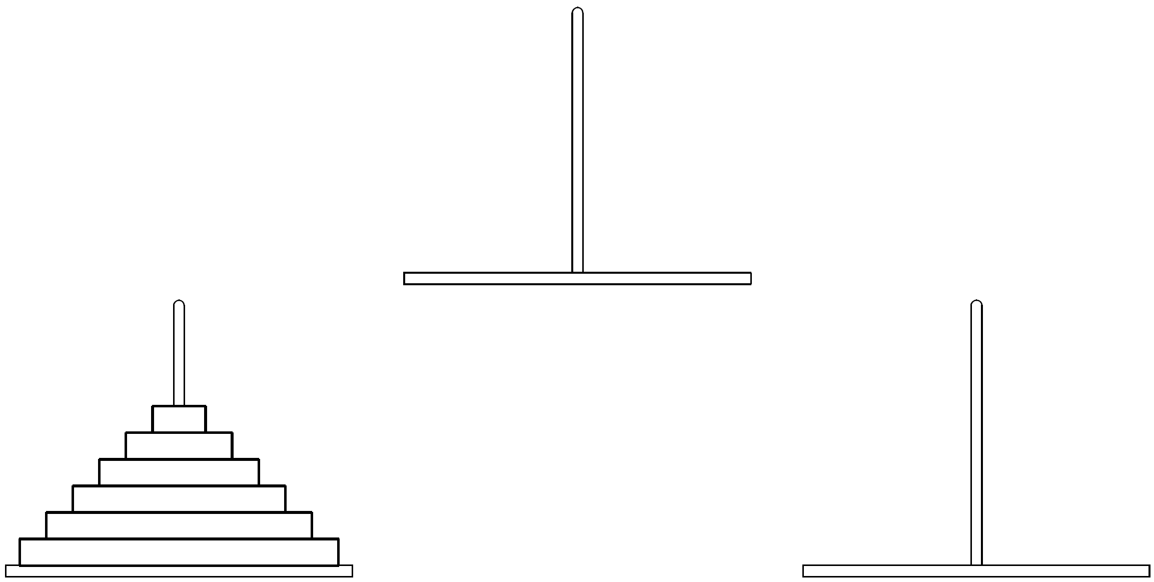

Figure 1: Towers of Hanoi problem.

2 Problem and Solutions

In this section we state the Towers of Hanoi problem and present an inductive and an iterative solution. (All of this is well-known so the discussion is brief.) The (non-evident) correctness of the iterative solution is used to motivate the analysis that follows in the next section, where the associativity of equivalence comes into play.

2.1 Problem Statement

The Towers of Hanoi problem comes from a puzzle marketed in 1883 by the French mathematician Édouard Lucas, under the pseudonym M.Claus [Ste97].

The puzzle is based on a legend according to which there is a temple, apparently in Bramah rather than in Hanoi as one might expect, where there are three giant poles fixed in the ground. On the first of these poles, at the time of the world's creation, God placed sixty four golden disks, each of different size, in decreasing order of size. The Bramin monks were given the task of moving the disks, one per day, from one pole to another subject to the rule that no disk may ever be above a smaller disk. The monks' task would be complete when they had succeeded in moving all the disks from the first of the poles to the second and, on the day that they completed their task the world would come to an end!

2.2 Inductive Solution

There is a well-known inductive solution to the problem given by the pseudo-code below.

In this solution we make use of the fact that the given problem is symmetrical with respect to all three poles. Thus it is undesirable to name the individual poles. Instead we visualise the poles as being arranged in a circle; the problem is to move the tower of disks from one pole to the next pole in a specified direction around the circle.

The code defines $H_n.d$ to be a sequence of pairs $\langle k, d' \rangle$ where n is the number of disks, k is a disk number and d and d' are directions. Disks are numbered from 0 onwards, disk 0 being the smallest[3]. Directions are boolean values, true representing a clockwise movement and false an anti-clockwise movement. The pair $\langle k, d' \rangle$ means move the disk numbered k from its current position in the direction d'. The semicolon operator concatenates sequences together, $[]$ denotes an empty sequence and $[x]$ is a sequence with exactly one element x. Taking the pairs in order from left to right, the complete sequence $H_n.d$ prescribes how to move the n smallest disks one-by-one from one pole to the next pole in the direction d following the rule of never placing a larger disk on top of a smaller disk.

$$\begin{array}{rcl} H_0.d & = & [] \\ H_{n+1}.d & = & H_n.\neg d \; ; \; [\langle n, d \rangle] \; ; H_n.\neg d \end{array}$$

2.3 Iterative Solution

The inductive solution to the Towers of Hanoi problem just given is often used as an illustration of the power of recursion as a problem solving technique. It is a solution, however, that human beings find very difficult to implement. The solution used by the Bramin monks is more likely to be the following remarkably simple iterative solution [BL80].

> On every alternate day, beginning on the first day, the smallest disk is moved. The rule for moving the smallest disk is that it should *cycle* around the poles. The direction of rotation depends on the total number of disks. If the total number of disks is odd the smallest disk should be rotated in the direction specified for the whole, original, tower. Otherwise it should be rotated in the opposite direction.
>
> On every other day a disk other than the smallest disk is moved — subject to the rule that no disk may ever be above a smaller disk. It is easy to see that because of this rule there is exactly one move possible so long as not all the disks are on one and the same pole.

[3]Assigning number 0 to the smallest rather than the largest disk has the advantage that the number of the disk that is moved on any day is independent of the total number of disks to be moved.

3 Iterative Solution — Specification and Proof

3.1 WHY?

The iterative solution to the Towers of Hanoi problem illustrates well the difference between the WHAT of a problem specification, the HOW of an implementation of a solution to the problem and the WHY relating the implementation to the specification. The iterative solution is easy to carry out and it seems to work. But why?

Here is a good opportunity to demonstrate the art of effective reasoning. The particular aspect of the problem that concerns us is that of turning the English description of the movement of the disks into a formal statement that can be proven by syntactic formula manipulation. Our solution involves exploiting the associativity of equivalence.

In fact, it isn't just the smallest disk that cycles around the poles. All the disks do too. This note is about formalising and proving this fact in a way that is economical and convincing.

In addition, for completeness, we also formally prove that the disk that is moved on each day alternates between the smallest disk and the other disks.

3.2 Formalisation and Proof

We begin by showing that each disk cycles around the poles.

Suppose the objective is to move the smallest n disks in the direction d. In natural language, the claim is that, on any day, the direction of movement of the disk moved on that day is d if and only if the total number of disks n is as odd as the number of the disk is even. Formally, avoiding specific references to days, the claim is that, for all n,

$$\langle \forall k, d, d' : \langle k, d' \rangle \in H_n.d : d' \equiv d \equiv \mathrm{odd}.n \equiv \mathit{even}.k \rangle$$

where $x \in xs$ is true whenever x is an element of the sequence xs. We prove the claim by induction on n. For $n=0$ we have:

$$\begin{array}{ll}
 & \langle \forall k, d, d' : \langle k, d' \rangle \in H_0.d : d' \equiv d \equiv \mathrm{odd}.0 \equiv \mathit{even}.k \rangle \\
= & \qquad \{ \quad \text{definition of } H_0.d \quad \} \\
 & \langle \forall k, d, d' : \langle k, d' \rangle \in [] : d' \equiv d \equiv \mathrm{odd}.0 \equiv \mathit{even}.k \rangle \\
= & \qquad \{ \quad \text{empty range} \quad \} \\
 & \mathsf{true} \quad .
\end{array}$$

For the induction step, we have:

$$\begin{array}{ll}
 & \langle \forall k, d, d' : \langle k, d' \rangle \in H_{n+1}.d : d' \equiv d \equiv \mathrm{odd}.(n+1) \equiv \mathit{even}.k \rangle \\
= & \qquad \{ \quad \text{definition of } H_{n+1}.d \quad \} \\
 & \langle \forall k, d, d' : \langle k, d' \rangle \in (H_n.\neg d \ ; \ [\langle n, d \rangle] \ ; H_n.\neg d) : d' \equiv d \equiv \mathrm{odd}.(n+1) \equiv \mathit{even}.k \rangle
\end{array}$$

$$
\begin{array}{ll}
= & \{ \quad \text{membership of a sequence} \quad \} \\
& \langle \forall k, d, d' : \langle k, d' \rangle \in H_n.\neg d : d' \equiv d \equiv \mathsf{odd}.(n{+}1) \equiv \mathit{even}.k \rangle \\
& \wedge \ \langle \forall k, d, d' : \langle k, d' \rangle = \langle n, d \rangle : d' \equiv d \equiv \mathsf{odd}.(n{+}1) \equiv \mathit{even}.k \rangle \\
& \wedge \ \langle \forall k, d, d' : \langle k, d' \rangle \in H_n.\neg d : d' \equiv d \equiv \mathsf{odd}.(n{+}1) \equiv \mathit{even}.k \rangle \\
= & \{ \quad \text{conjunction is idempotent and symmetric,} \\
& \quad\ \ \text{one-point rule} \quad \} \\
& \langle \forall k, d, d' : \langle k, d' \rangle \in H_n.\neg d : d' \equiv d \equiv \mathsf{odd}.(n{+}1) \equiv \mathit{even}.k \rangle \\
& \wedge \ (d \equiv d \equiv \mathsf{odd}.(n{+}1) \equiv \mathit{even}.n) \\
= & \{ \quad \text{1st conjunct:} \quad \mathsf{odd}.(n{+}1) \equiv \neg\mathsf{odd}(n) \ , \\
& \quad\ \ \text{2nd conjunct:} \quad \mathsf{odd}.(n{+}1) \equiv \mathit{even}(n) \ . \quad \} \\
& \langle \forall k, d, d' : \langle k, d' \rangle \in H_n.\neg d : d' \equiv d \equiv \neg\mathsf{odd}(n) \equiv \mathit{even}.k \rangle \\
= & \{ \quad d \equiv \neg\mathsf{odd}(n) \equiv \neg d \equiv \mathsf{odd}(n) \quad \} \\
& \langle \forall k, d, d' : \langle k, d' \rangle \in H_n.\neg d : d' \equiv \neg d \equiv \mathsf{odd}(n) \equiv \mathit{even}.k \rangle \\
= & \{ \quad \text{induction hypothesis, with } d := \neg d \quad \} \\
& \mathsf{true} \ .
\end{array}
$$

It remains to prove that the disk that is moved on each day alternates between the smallest disk and the other disks.

Let us call a sequence of numbers *alternating* if it has two properties. The first property is that consecutive elements alternate between zero and a non-zero value; the second property is that if the sequence is non-empty then it begins and ends with the value zero. We write $\mathsf{alt}.ks$ if the sequence ks has these two properties.

The sequence of disks moved on each day, which we denote by $disk_n.d$, is obtained by mapping the fst function (the function that returns the first of a pair of values) over the sequence $H_n.d$. Using well-known properties of mapping, $disk_n.d$ is easily shown to satisfy the recursive equations

$$
\begin{array}{rcl}
disk_0.d & = & [] \\
disk_{n+1}.d & = & disk_n.\neg d \ ; \ [n] \ ; \ disk_n.\neg d \ .
\end{array}
$$

Our goal is to prove $\mathsf{alt}.(disk_n.d)$. The proof is by induction on n. The base case, $n{=}0$, is clearly true because an empty sequence has no consecutive elements. For the induction step, the property of alternating sequences on which the proof depends is that, for a sequence ks and number k,

$$
\mathsf{alt}.(ks ; [k] ; ks) \ \Leftarrow \ \mathsf{alt}.ks \wedge ((ks{=}[]) \equiv (k{=}0)) \ .
$$

The proof is then:

$$
\begin{array}{ll}
 & \mathsf{alt}.(\mathtt{disk}_{n+1}.d) \\
= & \quad \{ \quad \text{definition} \quad \} \\
 & \mathsf{alt}.(\mathtt{disk}_n.\neg d\ ;\ [n]\ ;\ \mathtt{disk}_n.\neg d) \\
\Leftarrow & \quad \{ \quad \text{above property of alternating sequences} \quad \} \\
 & \mathsf{alt}.(\mathtt{disk}_n.\neg d)\ \wedge\ ((\mathtt{disk}_n.\neg d = [\,]) \equiv (n{=}0)) \\
= & \quad \{ \quad \text{induction hypothesis applied to first conjunct,} \\
 & \quad\quad\ \ \text{straightforward property of } \mathtt{disk}_n \text{ for the second.} \quad \} \\
 & \mathsf{true}\ \ .
\end{array}
$$

4 Conclusion

The use of continued infix notation for binary operators is geared to the use of associativity properties. The use of continued infix notation for binary relations is geared to the use of transitivity properties. Both associativity and transitivity are used extensively in the calculational style of reasoning, and to great effect — largely because their use is almost subconscious. Early recognition and exploitation of such properties is therefore very important.

Equivalence stands out because it is both associative (when viewed as a binary function) and transitive (when viewed as a binary relation). Its transitivity has long been recognized and is reflected in the way many scientists read a continued equivalence. But, as Dijkstra and Scholten rightly point out, its associativity is also very important and should be exploited to the full. The Towers of Hanoi problem provides a pleasing illustration.

Acknowledgement We are grateful to the Eindhoven Tuesday Afternoon Club (in particular Netty van Gasteren) and to Graham Hutton for critical comments and suggestions for improvement.

References

[BL80] P. Buneman and L. Levy. The Towers of Hanoi Problem. *Information Processing Letters*, 10:243–244, 1980.

[DS90] Edsger W. Dijkstra and Carel S. Scholten. *Predicate Calculus and Program Semantics*. Texts and monographs in Computer Science. Springer-Verlag, 1990.

[GS93] David Gries and Fred B. Schneider. *A Logical Approach to Discrete Math*. Springer-Verlag, 1993.

[Ste97] Ian Stewart. *The Magical Maze*. Weidenfield and Nicolson, London, 1997.

[Tar56] Alfred Tarski. *Logic, Semantics, Metamathematics, papers from 1923 to 1938.* Oxford University Press, 1956. Translated by J.H.Woodger.

[Wil87] J.G. Wiltink. A deficiency of natural deduction. *Information Processing Letters*, 25:233–234, 1987.