10-2010

Towards Simpler Description of Properties like Commutativity and Associativity: Using Expression Fragments

Shubhra Datta
The University of Texas at El Paso, sdatta2@miners.utep.edu

Valeria Fierro
The University of Texas at El Paso

Krasen Petrov
The University of Texas at El Paso

Jessica Romo
The University of Texas at El Paso, jromo2@miners.utep.edu

Gesuri Ramirez
The University of Texas at El Paso

See next page for additional authors

Comments:
Technical Report: UTEP-CS-10-45
Published in *Journal of Uncertain Systems*, 2011, Vol. 5, No. 2, pp. 84-89.

Authors
Shubhra Datta, Valeria Fierro, Krasen Petrov, Jessica Romo, Gesuri Ramirez, and Cesar Valenzuela

Journal of Uncertain Systems
Vol.5, No.x, pp.x-x, 2011
Online at: www.jus.org.uk

Towards Simpler Description of Properties like Commutativity and Associativity: Using Expression Fragments

Shubhra Datta*, Valeria Fierro, Krasen Petrov,
Jessica Romo, Gesuri Ramirez, and Cesar Valenzuela,
Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA

Received 15 December 2009; Revised 23 February 2010

Abstract

Properties like commutativity and associativity are very important in many applications. To make detecting these properties easier, it is desirable to reformulate them in the simplest possible way. A known way to simplify these descriptions is to use postfix notations, in which there is no need to use parentheses. We show that some of these properties can be further simplified if we represent them as equalities not between well-defined expressions, but as equalities between expression fragments.

Keywords: commutativity, associativity, expression fragments

1 Formulation of the Problem

Formulation of the problem: it is desirable to have simpler descriptions of basic properties like commutativity or associativity. Many binary operations $\circ$ have properties like commutativity

$$a \circ b = b \circ a \tag{1}$$

or associativity

$$a \circ (b \circ c) = (a \circ b) \circ c. \tag{2}$$

Since these properties are very important in many applications, it is desirable to describe them in the simplest possible way. This may help, e.g., if we want to check these properties – the simpler the property's description, usually the easier (and faster) it is to check this property.

For example, the above standard description of commutativity consists of 7 symbols, and the associativity consists of 15 symbols. Can we describe this property by using fewer symbols?

Postfix notations: a way to get shorter descriptions. For properties whose description contains parentheses – like associativity – we can get a shorter description if we use *postfix* notations. For readers who are not familiar with these notations, let us explain what they are and how they can help.

In the standard notations, the operation symbol is placed between the two operands, so that the result of applying the operation $\circ$ to a and b is denoted by $a \circ b$. In these notations, it is important to have parentheses. For example, $a \circ b \circ c$ can mean either $(a \circ b) \circ c$ or $a \circ (b \circ c)$. These two expressions have the same value for associative arithmetic operations like addition or multiplication, but for non-associative operations like division, the results are, in general, different.

It is known that the need for parentheses disappears if we use *postfix* notations in which the operation symbol is listed *after* the two operands. In these notations, the result of applying the operation $\circ$ to a and b is denoted by $ab\circ$. To find the postfix expression for $(a \circ b) \circ c$, i.e., for the result of applying the operation $\circ$ to $a \circ b$ and c,

- we first list the first operand $a \circ b$ – which in postfix notation has the form $ab\circ$;
- then, we list the second operand c;
- finally, we list the operation symbol $\circ$.

*Corresponding author. Email: sdatta2@miners.utep.edu (S. Datta).

The resulting postfix expression is $ab \circ c\circ$.

Postfix notations are actively used in computer science: to compute the value of an arithmetic expression, a compiler usually first transforms this expression into the postfix form, and then uses this form to compute the value; see, e.g., [1].

Postfix notations do not need parentheses and are, therefore, usually shorter. For example, in these notations, commutativity has the form

$$ab\circ = ba\circ \tag{3}$$

which happens to have the same length (7 symbols) as in the standard notation. However, for associativity, the postfix description

$$ab \circ c\circ = abc \circ \circ \tag{4}$$

is shorter: it requires 11 symbols instead of the original 15.

This is well known. The next natural question is: can we make this description even shorter?

2 Main Idea: Using Expression Fragments

Our main idea: use expression fragments instead of complete expressions. In the equality (3) that describes commutativity, the last symbol on both sides is the same: the operation symbol $\circ$. Thus, the right-hand side can be obtained from the left-hand side if we replace all the other symbols, i.e., if we replace ab with ba.

It therefore seems reasonable to represent commutativity as the formal equality

$$ab = ba. \tag{5}$$

Please note that while $ab\circ$ is a well-defined expression, the combination ab is an *expression fragment* but *not* a well-defined expression because it lacks the operation symbol.

The meaning of the original formula $ab\circ = ba\circ$ is that whenever a and b are well-formed expressions, the corresponding expressions $ab\circ$ and $ba\circ$ have the same value. Since the equality (5) is about expression *fragments* not well-formed expressions, and the values are defined only for well-formed expressions, it is reasonable to interpret this equality in fragment terms:

- whenever we have a well-formed expression with a fragment ab, for which replacing ab with ba also leads to a well-formed expression,
- these two expressions – the original one and the one formed by replacement – must have the same value.

Similarly, in the equality (4) that describes associativity, the last symbol on both sides is the same: the operation symbol $\circ$, and the first two symbols are the same in both sides: a and b. Thus, the right-hand side can be obtained from the left-hand side if we replace all the other symbols, i.e., if we replace $\circ c$ with $c\circ$.

It therefore seems reasonable to represent associativity as the formal equality

$$\circ c = c \circ . \tag{6}$$

Here, in contrast to commutativity, where replacing ab with ba and replacing ba with ab mean the same thing – swapping two fragments, here, replacing $\circ c$ with $c\circ$ and replacing $c\circ$ with $\circ c$ are two different replacement operations, so they must be described separately.

Historical comment. The idea of using fragments came from the bra- and ket-notations in quantum mechanics and quantum computing; see, e.g., [2]. In quantum mechanics, originally, researchers used the "bracket" notation $\langle a|b\rangle$ to describe the amplitude corresponding to the transition from the state a to the state b. However, P. A. M. Dirac, a future Nobel prize winner, noticed that many formulas become much easier if we formulate them in terms of expression fragments $\langle a|$ and $|b\rangle$. Dirac started using the word "bra" for the first fragment $\langle a|$ of the bracket notation, and the word "ket" for the second fragment $|b\rangle$.

For example, in bracket notations, the fact that the state b is a superposition of states b' and b'' with equal coefficients $\dfrac{1}{\sqrt{2}}$, can be described as

$$\langle a|b\rangle = \frac{1}{\sqrt{2}} \cdot \langle a|b'\rangle + \frac{1}{\sqrt{2}} \cdot \langle a|b''\rangle. \tag{7}$$

In bra-ket notations, we have a simpler formula

$$|b\rangle = \frac{1}{\sqrt{2}} \cdot |b'\rangle + \frac{1}{\sqrt{2}} \cdot |b''\rangle. \tag{8}$$

The original formula (7) can be recovered if we "multiply" the bra fragment $\langle a|$ by both sides of the new formula (8).

3 Definitions and Results

Towards formal definitions. Let us now give formal definitions of the fragment equality and analyze whether the fragment equalities (5) and (6) are indeed equivalent to, correspondingly, commutativity and associativity.

Let us first recall the standard definitions.

Definition 1. *Let V, C, and Op be sets, and let $ar : Op \to N$ be a function from the set Op to the set N of natural numbers (= non-negative integers).*

- *Elements v of the set V are called* variables*.*
- *Elements c of the set C are called* constants*.*
- *Elements op of the set Op are called* operation symbols*.*
- *For each operation symbol $op \in Op$, the value $ar(op)$ is called its* arity*.*
- *A* well-formed expression *is defined as follows:*
  - *every variable and every constant is a well-formed expression;*
  - *if $\circ$ is an operation of arity $m = ar(op)$ and $t_1, \ldots, t_N$ are well-formed expressions, then $t_1 \ldots t_m \circ$ is a well-formed expression;*
  - *nothing else is a well-formed expression.*
- *By an* interpretation *f, we mean a mapping that assigns, to every operation symbol $op \in Op$ of arity $m = ar(op)$, a function $f_{op} : C^m \to C$.*
- *For every interpretation f and for every well-formed expression t that does not contain variables, its* value *$v_f(t)$ is defined as follows:*
  - *for every constant c, its value is this same constant: $v_f(c) = c$;*
  - *if an expression has the form $t_1 \ldots t_m \circ$ for some operation $\circ$ of arity $m = ar(op)$, then its value is defined as $v_f(t) = f_\circ(v_f(t_1), \ldots, v_f(t_m))$.*
- *By a* property*, we mean a set of equalities of the type $t = t'$, where t and t' are well-formed expressions.*
- *We say that the property $t = t'$ is* satisfied *for a given interpretation if, every time we substitute constants instead of the variables into the formula $t = t'$, the resulting expressions T and T' have the same value.*

Comment. Instead of requiring that the values are equal when we substitute constants, we can also require that the values are equal when we substitute well-formed expressions; the result will be the same since for the above definition,

- when we first substitute the expressions and then compute the value of the result, and
- when we first compute the values of the expressions, substitute these values, and then compute the result,

we get the exact same result.

Let us now give formal definitions of expression fragments and equality between them.

Definition 2. *Let V, C, and Op be sets, and let $ar : Op \to N$ be a function from the set Op to the set N of natural numbers.*

- *An arbitrary sequence of variables, constants, and operation symbols is called an* expression fragment.
- *We say that an expression t* contains *a fragment e if t has the form $\underline{e}e\overline{e}$ for some fragments $\underline{e}$ and $\overline{e}$.*
- *By a* fragment equality*, we mean an equality of the type $e = e'$, where e and e' are expression fragments.*
- *We say that the fragment equality $e = e'$ with variables $v_1, \ldots, v_p$, is* satisfied *for a given interpretation f if, for every p expression fragments $E_1, \ldots, E_p$ without variables and for the results E and E' of substituting E_j instead of v_j into e and e', the following two statements hold:*
  - *for every well-defined expression t that contains E, if the result t' of replacing E with E' is also a well-formed expression, then t and t' have the same value: $v_f(t) = v_f(t')$;*
  - *for every well-defined expression t' that contains E', if the result t of replacing E' with E is also a well-formed expression, then t and t' have the same value: $v_f(t) = v_f(t')$.*

Comment. In the definition of fragment equality being satisfied, we have a condition that the replacement result must be a well-formed expression. The following simple example shows that this condition is necessary: for the fragment equality

$$E \stackrel{\text{def}}{=} c\circ = \circ c \stackrel{\text{def}}{=} E' \tag{9}$$

that describes associativity,

- the expression $t = ac\circ$ contains E and is well-defined,
- but the result $t' = a \circ c$ of replacing E with E' is no longer well-defined.

Proposition 1. *For a single operation $\circ$, the fragment equality $a\circ = \circ a$ is equivalent to associativity.*

Comment. For readers' convenience, all the proofs are placed at the end of this paper.

What if there are other operations? When we have other operations in addition to $\circ$, the fragment equality $a\circ = \circ a$ corresponding to associativity of $\circ$ is, in general, no longer equivalent to associativity of $\circ$. As an example, let us take the set of integers with two operations: multiplication $\circ = \cdot$ and addition $+$. Multiplication is associative. However, for $a = +$, the fragment equality $a\circ = \circ a$ corresponding to multiplication $\circ = \cdot$ would allow us to transform the expression $c_1c_2c_3 \cdot +$ corresponding to $c_1 + (c_2 \cdot c_3)$ into a different expression $c_1c_2c_3 + \cdot$ corresponding to $c_1 \cdot (c_2 + c_3)$ whose value is, in general, different.

Commutativity: discussions. As we have mentioned earlier, for commutativity, a natural fragment equality is $ab = ba$. At first glance, it sound reasonable to assume that this fragment equality is equivalent to commutativity. However, if we take $b = \circ$, we see that it also implies the fragment equality corresponding to associativity. The next natural hypothesis – that this fragment equality is equivalent to commutativity *and* associativity – turns out to be absolutely correct.

Proposition 2. *For a single operation $\circ$, the fragment equality $ab = ba$ is equivalent to a combination of commutativity and associativity.*

Comment. When there is a single operation, sometimes, the symbol for this operation is omitted, so we right ab instead of $a \circ b$. In such notations, $ab = ba$ would mean $a \circ b = b \circ a$, i.e., commutativity. In this notation, both ab and ba are well-defined expressions.

To avoid confusion, it is important to emphasis that we are *not* using this operation-less notation. In our notation, there is an explicit operation symbol. So, in our notation, ab and ba are *not* well-defined expressions, they are expression fragments.

What if there are other operations? When we have several operations, this equivalence is no longer true, even if all these operations are commutative and associative. To show this, we can use the same example as we used after Proposition 1. Both addition and multiplication are commutative and associative. However, the fragment equality $ab = ba$ with $a = \cdot$ and $b = +$ allows us to transform the expression $c_1c_2c_3 \cdot +$ corresponding to $c_1 + (c_2 \cdot c_3)$ into a different expression $c_1c_2c_3 + \cdot$ corresponding to $c_1 \cdot (c_2 + c_3)$ whose value is, in general, different.

Remaining open problems. We have described simple equivalents for associativity and for a combination of associativity and commutativity. It is desirable to use expression fragments to find similar simpler expressions for other algebraic properties.

4 Proofs

Proof of Proposition 1. To prove the desired equivalence, we need to show that associativity implies the fragment equality $c\circ = \circ c$ and that, vice versa, this fragment equality implies associativity.

It is clear that the fragment equality implies associativity. Indeed, if the fragment equality $c\circ = \circ c$ holds, then, whenever a, b, and c are well-formed expressions, the expressions $ab \circ c\circ$ and $abc \circ \circ$ can be obtained from each other by replacing the fragment $\circ c$ with $c\circ$. Thus, the two expressions $ab \circ c\circ$ and $abc \circ \circ$ have the same value – i.e., we have associativity.

Vice versa, let us show that associativity implies the fragment equality $c\circ = \circ c$, i.e., that associativity implies all the equalities that follow from this fragment equality.

Indeed, every time we apply the transformation $c\circ = \circ c$, we simply move the operation symbol $\circ$ to a different place in the original expression. This transformation does not change the order in which the constants appear in the expression. Thus, when we start with an expression with constants $c_1, \ldots, c_n$ (in this order), we end up with an expression in which these same constants appear in this same order – but the order of operations may change.

For $n = 2$, there is only one expression: $c_1 \circ c_2$ whose postfix expression is $c_1c_2\circ$.

For $n = 3$, there are two possible expressions and their equality is what associativity is about:

- the expression $(c_1 \circ c_2) \circ c_3$ whose postfix description is $c_1c_2 \circ c_3\circ$;
- the expression $c_1 \circ (c_2 \circ c_3)$ whose postfix description is $c_1c_2c_3 \circ \circ$.

For $n = 4$, we have the following expressions corresponding to all possible ways to combine c_1, c_2, c_3, and c_4. In all case, we start by applying the operation $\circ$ to two neighboring values of c_i.

When we start with $c_1 \circ c_2$, we get the following two expressions:

- the expression $(c_1 \circ c_2) \circ (c_3 \circ c_4)$ whose postfix description is $c_1c_2 \circ c_3c_4\circ$;
- the expression $(c_1 \circ c_2) \circ c_3) \circ c_4$ whose postfix description is $c_1c_2 \circ c_3 \circ c_4\circ$.

When we start with $c_2 \circ c_3$, we get the following two expressions:

- the expression $(c_1 \circ (c_2 \circ c_3)) \circ c_4$ whose postfix description is $c_1c_2c_3 \circ \circ c_4\circ$;
- the expression $c_1 \circ ((c_2 \circ c_3) \circ c_4)$ whose postfix description is $c_1c_2c_3 \circ c_4 \circ \circ$.

Finally, if we start with $c_3 \circ c_4$, then, in addition to the expression $(c_1 \circ c_2) \circ (c_3 \circ c_4)$ that we have already counted earlier, we also have the following new expression:

- the expression $c_1 \circ (c_2 \circ (c_3 \circ c_4))$ whose postfix description is $c_1c_2c_3c_4 \circ \circ\circ$.

It is know that, in general, for associative operations, all these expressions coincide. The proposition is thus proven.

Proof of Proposition 2. We have already shown that the fragment equality $ab = ba$ implies both commutativity and associativity.

Vice versa, if we allow transformations related to the fragment equality $ab = ba$, this means that we can change the order of all the symbols in the original sequence. Thus, if we have any sequence of operations applied to the constants $c_1, \ldots, c_n$ in arbitrary order of c_i and arbitrary order of operations, we may end up with a different order of c_i and different order of operations.

For example, we may add up with expressions

$$(c_1 \circ c_2) \circ (c_3 \circ c_4) \text{ and } c_2 \circ ((c_4 \circ c_3) \circ c_1), \tag{10}$$

i.e., in postfix notations, $c_1 c_2 \circ c_3 c_4 \circ \circ$ and $c_2 c_4 c_3 \circ c_1 \circ \circ$.

Here:

- commutativity makes sure that the value is preserved when we change the order of the constants, and
- associativity makes sure that the value is preserved when we change the order of operations.

Thus, the new expression indeed has the same value as the original one.

The proposition is proven.

References

[1] Collins, W. *Data Structures and the Java Collections Framework*, McGraw Hill, 2004.

[2] Nielsen, M. A., and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, Massachusetts, 2000.