University of Amsterdam
Theory of Computer Science

Note on Paraconsistency and the Logic of Fractions

J.A. Bergstra
I. Bethke

Report TCS1413 October 2014

J.A. Bergstra

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XH   Amsterdam
the Netherlands

tel. +31 20 525.7591
e-mail: J.A.Bergstra@uva.nl

I. Bethke

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XH   Amsterdam
the Netherlands

tel. +31 20 525.7583
e-mail: I.Bethke@uva.nl

Theory of Computer Science Electronic Report Series

Note on paraconsistency and the logic of fractions

Jan A. Bergstra & Inge Bethke

Informatics Institute, University of Amsterdam*

Abstract

We apply a paraconsistent logic to reason about fractions.

1 Introduction

Suppose we want to define an arithmetic framework in which it is possible to reason about fractions in a consistent and reliable way and in which the usual laws of arithmetic hold: e.g. for natural numbers n, l and $k \neq 0 \neq m$ the equations

$$\frac{n}{m} + \frac{l}{k} = \frac{nk + lm}{mk} \qquad \text{and} \qquad \frac{n}{m} + \frac{l}{m} = \frac{n + l}{m}$$

should be valid. At the same time, we want to consider fractions as mathematical expressions with typical syntactic operations like the numerator $num(\ )$ satisfying

$$num(\frac{n}{m}) = n.$$

Then equational logic dictates

$$n + l = num(\frac{n + l}{m}) = num(\frac{n}{m} + \frac{l}{m}) = num(\frac{nm + lm}{mm}) = nm + lm.$$

for arbitrary l, n and $m \neq 0$, and our framework is inconsistent. Nevertheless, fractions are of great practical and abstract importance and we should be able to reason about them without lapsing into absurdity.

In mathematics education, one can get around this predicament in various ways: avoiding the concepts of numerator and denominator (see e.g. [6]), viewing fractions as heterogeneous subject (see e.g. [5]), or accepting cognitive conflicts (see e.g. [7]). In this note, we propose to apply paraconsistent logic.

A *paraconsistent* logic is a way to reason about inconsistent information without *exploding* in the sense that if a contradiction is obtained, then everything can be obtained.

*Science Park 904, 1098 XH Amsterdam

Paraconsistent logics come in a broad spectrum, ranging from logics with the thought that if a contradiction were true, then everything would be true, to logics that claim that some contradictions really are true. For a brief discussion see e.g. [4]. In this note, we choose a particular paraconsistent logic to tackle the dilemma sketched above. We do not claim that this is the only possible way how to proceed in our scenario; other paraconistent logics may be suitable as well.

2 The binary C & P structure

The approach taken is known as the *preservationist* school. There the fundamental idea is that, given an inconsistent collection of premises, one should not try to reason about the collection of premises as a whole, but rather focus on internally consistent subsets of premises. In 2004, Brown and Priest [3] introduced the *Chunk and Permeate* (C & P) strategy for dealing with reasoning situations involving incompatible assumptions. In this preservationist logic, a theory is broken up into chunks and only restricted information is allowed to pass from one chunk to another.

Here we only give the rendering of the simple binary structure which is sufficient in our case. We let Σ be the two-sorted signature containing $0, 1$, the numerator and denominator *num* and *denom*, the fraction $-$, symbols for addition and multiplication of natural numbers and fractions, respectively, and an additional error element a produced by division by zero. The binary C & P structure for our problem is $\langle \mathcal{T}_s, \mathcal{T}_t, \rho \rangle$ where

- $\mathcal{T}_s$ is the source chunk,
- $\mathcal{T}_t$ is the target chunk, and
- ρ is the information that is allowed to flow from source to target.

$\mathcal{T}_s$ is the equational theory given in Table 1. Here k, l, m, n range over natural numbers or can take the error value, and α, β, γ denote fractions. Observe that in the source, fractions with identical denominators can be added (†):

$$\begin{aligned} \frac{n}{m} + \frac{k}{m} &= \frac{n}{1} \cdot \frac{1}{m} + \frac{k}{1} \cdot \frac{1}{m} && (9) \\ &= (\frac{n}{1} + \frac{k}{1}) \cdot \frac{1}{m} && (6), (7) \\ &= \frac{n+k}{1} \cdot \frac{1}{m} && (10) \\ &= \frac{n+k}{m} && (9) \end{aligned}$$

That $\mathcal{T}_s$ is consistent can be seen as follows. We let $\mathbb{M}$ be the Σ-algebra with the sorts $\mathbb{N}_a = \mathbb{N} \cup \{a\}$ and $\mathbb{F} = \mathbb{N} \times \mathbb{N}$ where the operations are interpreted as follows. If $\circ \in \{+, \cdot\}$, then $\circ : \mathbb{N}_a \times \mathbb{N}_a \to \mathbb{N}_a$ is defined by

$$x \circ y = \begin{cases} x \circ y & \text{if } x, y \in \mathbb{N} \\ a & \text{otherwise,} \end{cases}$$

$$
\begin{aligned}
n + 0 &= n && (1)\\
(n + m) + l &= n + (m + l) && (2)\\
n + m &= m + n && (3)\\
n \cdot 1 &= n && (4)\\
(n \cdot m) \cdot l &= n \cdot (m \cdot l) && (5)\\
n \cdot m &= m \cdot n && (6)\\
n \cdot (m + l) &= n \cdot m + n \cdot l && (7)\\
\frac{n}{m} \cdot \frac{l}{k} &= \frac{n \cdot l}{m \cdot k} && (8)\\
\frac{n}{m} &= \frac{n}{1} \cdot \frac{1}{m} && (9)\\
\frac{n}{1} + \frac{m}{1} &= \frac{n + m}{1} && (10)\\
(\alpha + \beta) + \gamma &= \alpha + (\beta + \gamma) && (11)\\
\alpha + \beta &= \beta + \alpha && (12)\\
(\alpha \cdot \beta) \cdot \gamma &= \alpha \cdot (\beta \cdot \gamma) && (13)\\
\alpha \cdot \beta &= \beta \cdot \alpha && (14)\\
\alpha \cdot (\beta + \gamma) &= \alpha \cdot \beta + \alpha \cdot \gamma && (15)\\
m \neq 0 \ \wedge \ m \neq a \rightarrow \quad num(\frac{n}{m}) &= n && (16)\\
m \neq 0 \ \wedge \ n \neq a \rightarrow \quad denom(\frac{n}{m}) &= m && (17)
\end{aligned}
$$

Table 1: The equational theory of the source $\mathcal{T}_s$

and $\frac{-}{-} : \mathbb{N}_a \times \mathbb{N}_a \to \mathbb{F}$ is defined by

$$\frac{x}{y} = \begin{cases} (x,y) & \text{if } x \in \mathbb{N}, y \in \mathbb{N} \\ (0,0) & \text{otherwise,} \end{cases}$$

$num, denom : \mathbb{F} \to \mathbb{N}_a$ are defined by

$$num((n,m)) = \begin{cases} n & \text{if } m \neq 0 \\ a & \text{otherwise,} \end{cases}$$

and

$$denom((n,m)) = \begin{cases} m & \text{if } m \neq 0 \\ a & \text{otherwise,} \end{cases}$$

and $+, \cdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ are defined by

$$(n,m) + (l,k) = \begin{cases} (n+l, m) & \text{if } m = k \neq 0 \\ (0,0) & \text{otherwise,} \end{cases}$$

and

$$(n,m) \cdot (l,k) = \begin{cases} (nl, mk) & \text{if } m \neq 0 \neq k \\ (0,0) & \text{otherwise.} \end{cases}$$

It is easy to see that $\mathbb{M}$ is a model for $\mathcal{T}_s$. The target theory $\mathcal{T}_t$ consists of the single conditional equation

$$k \neq 0 \to \frac{nk}{mk} = \frac{n}{m}. \quad (18)$$

Clearly, $\mathcal{T}_t$ is consistent. However, $\mathcal{T}_s \cup \mathcal{T}_t$ is inconsistent since otherwise we have for arbitrary $k \neq 0 \neq m$

$$mk = denom(\frac{nk}{mk}) = denom(\frac{n}{m}) = m.$$

Moreover, we let ρ be the set of equations (1) – (15). $\mathcal{T}_t \cup \rho$ is consistent: $\mathbb{Q}_0$ and $\mathbb{Q}_a$ —the zero-totalized and the a-totalized meadow of the rational numbers (see e.g. [1, 2]), respectively—are both models of $\mathcal{T}_t \cup \rho$. We have thus arrived at a consistent theory were we have full addition of fractions:

$$\begin{aligned} \frac{n}{m} + \frac{k}{l} &= \frac{nl}{ml} + \frac{km}{lm} && (18) \\ &= \frac{nl}{ml} + \frac{km}{ml} && (6) \\ &= \frac{nl + km}{ml} && (\dagger). \end{aligned}$$

References

[1] Bergstra, J.A., Bethke, I., and Ponse, A. (2013). Cancellation meadows: a generic basis theorem and some applications. *The Computer Journal*, 56(1):3–14, `doi:10.1093/comjnl/bsx147`. (Also available at arXiv:0803.3969 [math.RA, cs.LO].)

[2] Bergstra, J.A. and Ponse, A. (2014). Division by zero in common meadows. University of Amsterdam, Section Theory of Computer Science, Report TCS1410, June 2014. (Also available at arXiv:1406.3280v2 [cs.LO].)

[3] Brown, B. and Priest, G. (2004). Chunk and Permeate, a paraconsistent inference strategy. Part I: the infinitesimal calculus. *Journal of Philosophical Logic*, 33:379 –388.

[4] Middelburg, C.A. (2011). A Survey of Paraconsistent Logics. arXiv:1103.4324[cs.LO].

[5] Padberg, F. (2012). *Didaktik der Bruchrechnung* (4_{th} edition). Series: Mathematik, Primar- und Sekundarstufe, Springer-Spektrum.

[6] Rollnik, S. (2009). *Das pragmatische Konzept für den Bruchrechenunterricht.* PhD thesis, University of Flensburg, Germany.

[7] Tall, D. and Vinner, S. (1981). Concept image and concept definition in mathematics, with special reference to limits and continuity. *Educational Studies in Mathematics*, 12 151–169.

Electronic Reports Series of section Theory of Computer Science

Within this series the following reports appeared.

[TCS1412] J.A. Bergstra, I. Bethke, and A. Ponse, *Rekenen-Informatica*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1411] J.A. Bergstra, *Bitcoin: not a Currency-like Informational Commodity*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1409v2] J.A. Bergstra and A. Ponse, *Three Datatype Defining Rewrite Systems for Datatypes of Integers each extending a Datatype of Naturals (version 2)*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1410] J.A. Bergstra and A. Ponse, *Division by Zero in Common Meadows*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1407v3] J.A. Bergstra, *Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers (version 3)*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1409] J.A. Bergstra and A. Ponse, *Three Datatype Defining Rewrite Systems for Datatypes of Integers each extending a Datatype of Naturals*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1406v3] J.A. Bergstra, *Bitcoin and Islamic Finance (version 3)*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1407v2] J.A. Bergstra, *Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers (version 2)*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1408] J.A. Bergstra, *Bitcoin: Informational Money en het Einde van Gewoon Geld*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1407] J.A. Bergstra, *Four Complete Datatype Defining Rewrite Systems for an Abstract Datatype of Natural Numbers*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1406v2] J.A. Bergstra, *Bitcoin and Islamic Finance (version 2)*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1406] J.A. Bergstra, *Bitcoin and Islamic Finance*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1405] J.A. Bergstra, *Rekenen in een Conservatieve Schrapwet Weide*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1404] J.A. Bergstra, *Division by Zero and Abstract Data Types*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1403] J.A. Bergstra, I. Bethke, and A. Ponse, *Equations for Formally Real Meadows*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1402] J.A. Bergstra and W.P. Weijland, *Bitcoin, a Money-like Informational Commodity*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1401] J.A. Bergstra, *Bitcoin, een "money-like informational commodity"*, section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1301] B. Diertens, *The Refined Function-Behaviour-Structure Framework*, section Theory of Computer Science - University of Amsterdam, 2013.

[TCS1202] B. Diertens, *From Functions to Object-Orientation by Abstraction*, section Theory of Computer Science - University of Amsterdam, 2012.

[TCS1201] B. Diertens, *Concurrent Models for Object Execution*, section Theory of Computer Science - University of Amsterdam, 2012.

[TCS1102] B. Diertens, *Communicating Concurrent Functions*, section Theory of Computer Science - University of Amsterdam, 2011.

[TCS1101] B. Diertens, *Concurrent Models for Function Execution*, section Theory of Computer Science - University of Amsterdam, 2011.

[TCS1001] B. Diertens, *On Object-Orientation*, section Theory of Computer Science - University of Amsterdam, 2010.

Within former series (PRG) the following reports appeared.

[PRG0914] J.A. Bergstra and C.A. Middelburg, *Autosolvability of Halting Problem Instances for Instruction Sequences*, Programming Research Group - University of Amsterdam, 2009.

[PRG0913] J.A. Bergstra and C.A. Middelburg, *Functional Units for Natural Numbers*, Programming Research Group - University of Amsterdam, 2009.

[PRG0912] J.A. Bergstra and C.A. Middelburg, *Instruction Sequence Processing Operators*, Programming Research Group - University of Amsterdam, 2009.

[PRG0911] J.A. Bergstra and C.A. Middelburg, *Partial Komori Fields and Imperative Komori Fields*, Programming Research Group - University of Amsterdam, 2009.

[PRG0910] J.A. Bergstra and C.A. Middelburg, *Indirect Jumps Improve Instruction Sequence Performance*, Programming Research Group - University of Amsterdam, 2009.

[PRG0909] J.A. Bergstra and C.A. Middelburg, *Arithmetical Meadows*, Programming Research Group - University of Amsterdam, 2009.

[PRG0908] B. Diertens, *Software Engineering with Process Algebra: Modelling Client / Server Architecures*, Programming Research Group - University of Amsterdam, 2009.

[PRG0907] J.A. Bergstra and C.A. Middelburg, *Inversive Meadows and Divisive Meadows*, Programming Research Group - University of Amsterdam, 2009.

[PRG0906] J.A. Bergstra and C.A. Middelburg, *Instruction Sequence Notations with Probabilistic Instructions*, Programming Research Group - University of Amsterdam, 2009.

[PRG0905] J.A. Bergstra and C.A. Middelburg, *A Protocol for Instruction Stream Processing*, Programming Research Group - University of Amsterdam, 2009.

[PRG0904] J.A. Bergstra and C.A. Middelburg, *A Process Calculus with Finitary Comprehended Terms*, Programming Research Group - University of Amsterdam, 2009.

[PRG0903] J.A. Bergstra and C.A. Middelburg, *Transmission Protocols for Instruction Streams*, Programming Research Group - University of Amsterdam, 2009.

[PRG0902] J.A. Bergstra and C.A. Middelburg, *Meadow Enriched ACP Process Algebras*, Programming Research Group - University of Amsterdam, 2009.

[PRG0901] J.A. Bergstra and C.A. Middelburg, *Timed Tuplix Calculus and the Wesseling and van den Berg Equation*, Programming Research Group - University of Amsterdam, 2009.

[PRG0814] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences for the Production of Processes*, Programming Research Group - University of Amsterdam, 2008.

[PRG0813] J.A. Bergstra and C.A. Middelburg, *On the Expressiveness of Single-Pass Instruction Sequences*, Programming Research Group - University of Amsterdam, 2008.

[PRG0812] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences and Non-uniform Complexity Theory*, Programming Research Group - University of Amsterdam, 2008.

[PRG0811] D. Staudt, *A Case Study in Software Engineering with PSF: A Domotics Application*, Programming Research Group - University of Amsterdam, 2008.

[PRG0810] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Poly-Threading,* Programming Research Group - University of Amsterdam, 2008.

[PRG0809] J.A. Bergstra and C.A. Middelburg, *Data Linkage Dynamics with Shedding,* Programming Research Group - University of Amsterdam, 2008.

[PRG0808] B. Diertens, *A Process Algebra Software Engineering Environment,* Programming Research Group - University of Amsterdam, 2008.

[PRG0807] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *Tuplix Calculus Specifications of Financial Transfer Networks,* Programming Research Group - University of Amsterdam, 2008.

[PRG0806] J.A. Bergstra and C.A. Middelburg, *Data Linkage Algebra, Data Linkage Dynamics, and Priority Rewriting,* Programming Research Group - University of Amsterdam, 2008.

[PRG0805] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *UvA Budget Allocatie Model,* Programming Research Group - University of Amsterdam, 2008.

[PRG0804] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Sequential Poly-Threading,* Programming Research Group - University of Amsterdam, 2008.

[PRG0803] J.A. Bergstra and C.A. Middelburg, *Thread Extraction for Polyadic Instruction Sequences,* Programming Research Group - University of Amsterdam, 2008.

[PRG0802] A. Barros and T. Hou, *A Constructive Version of AIP Revisited,* Programming Research Group - University of Amsterdam, 2008.

[PRG0801] J.A. Bergstra and C.A. Middelburg, *Programming an Interpreter Using Molecular Dynamics,* Programming Research Group - University of Amsterdam, 2008.

The above reports and more are available through the website: ivi.fnwi.uva.nl/tcs/

Electronic Report Series

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XG Amsterdam
the Netherlands

ivi.fnwi.uva.nl/tcs/