Datatype defining rewrite systems for naturals and integers

Bergstra, J.A.; Ponse, A.

DOI
10.23638/LMCS-17(1:17)2021

Publication date
2021

Document Version
Final published version

Published in
Logical Methods in Computer Science

Logical Methods in Computer Science
Volume 17, Issue 1, 2021, pp. 17:1–17:31
https://lmcs.episciences.org/

Submitted Jan. 15, 2020
Published Feb. 18, 2021

DATATYPE DEFINING REWRITE SYSTEMS FOR NATURALS AND INTEGERS

JAN A. BERGSTRA AND ALBAN PONSE

Informatics Institute, section Theory of Computer Science, University of Amsterdam
e-mail address: j.a.bergstra@uva.nl
e-mail address: a.ponse@uva.nl

Abstract. A datatype defining rewrite system (DDRS) is an algebraic (equational) specification intended to specify a datatype. When interpreting the equations from left-to-right, a DDRS defines a term rewriting system that must be ground-complete. First we define two DDRSs for the ring of integers, each comprising twelve rewrite rules, and prove their ground-completeness. Then we introduce natural number and integer arithmetic specified according to unary view, that is, arithmetic based on a postfix unary append constructor (a form of tallying). Next we specify arithmetic based on two other views: binary and decimal notation. The binary and decimal view have as their characteristic that each normal form resembles common number notation, that is, either a digit, or a string of digits without leading zero, or the negated versions of the latter. Integer arithmetic in binary and decimal notation is based on (postfix) digit append functions. For each view we define a DDRS, and in each case the resulting datatype is a canonical term algebra that extends a corresponding canonical term algebra for natural numbers. Then, for each view, we consider an alternative DDRS based on tree constructors that yields comparable normal forms, which for that view admits expressions that are algorithmically more involved. For all DDRSs considered, ground-completeness is proven.

1. Introduction

We specify natural number arithmetic and integer arithmetic by algebraic specifications, according to three different "views": unary, binary, and decimal notation. This paper is based on the specifications for natural numbers from [Ber14] that define addition and multiplication, and we follow the same strategy to develop these different views. Each of the specifications provided is a so-called DDRS (datatype defining rewrite system) and consists of a number of equations that define a term rewriting system (TRS) when interpreting the equations from left-to-right. A DDRS must be ground-complete, that is, terminating and ground-confluent; for some general information on TRSs see e.g. Terese [Ter03].

The *unary view* in [Ber14] is defined by a DDRS for which 0 and successor terms are the normal forms. The unary view is also used to provide a semantic specification of binary and decimal notation, using operator symbols for appending a digit. These two positional notations were modified with respect to conventional notations in such a way that

Key words and phrases: Datatype defining rewrite system, Equational specification, Integer arithmetic, Natural number arithmetic.

[CR1]	$x + (y + z) = (x + y) + z$	[CR5]	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
[CR2]	$x + y = y + x$	[CR6]	$x \cdot y = y \cdot x$
[CR3]	$x + 0 = x$	[CR7]	$1 \cdot x = x$
[CR4]	$x + (-x) = 0$	[CR8]	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

Table 1: Axioms for commutative rings

syntactic confusion between these notations cannot arise. It seems to be the case that for the unary view the specification of the integers is entirely adequate, whereas all subsequent specifications for binary and decimal view may provide no more than a formalisation of a topic which must be somehow understood before taking notice of that same formalisation. It remains to be seen to what extent the DDRSs for the unary case may serve exactly that expository purpose. Furthermore, each of these DDRSs contains equations for rewriting constructor terms in one of the other views to a term in the DDRS's view, e.g., the DDRSs for the unary view contain the equation $1 = S(0)$ for the constant 1 in binary view and in decimal view. The definitions of these DDRSs are geared towards obtaining comprehensible specifications of natural number and integer arithmetic in binary and decimal view. The successor function $S(x)$ and predecessor function $P(x)$ appeared to be instrumental auxiliary functions for this purpose, thereby justifying the incorporation of the unary view as a separate view.

This paper constitutes a further stage in the development of a family of arithmetical datatypes with corresponding specifications. The resulting DDRSs incorporate different views on the same abstract datatype (ADT), where an ADT may be understood as the isomorphism class of its instantiations. The datatypes considered in [Ber14] are so-called canonical term algebras (we further discuss these in Section 2).

The strategy of this work is somewhat complicated: on the one hand we look for specifications that may genuinely be considered introductory, that is, descriptions that can be used to construct the datatype at hand for the first time in the mind of a person. On the other hand awareness of the datatype in focus may be needed to produce an assessment of the degree of success achieved in the direction of the first objective. The described models aim to represent the natural and integer numbers. The question whether this is really the case depends on one's conception of the natural numbers as well as on the requirements one maintains of the notion of proof. As an independent foundational question, this question is neither posed nor answered in this paper, though the arguments in favour of the recognition of representations of natural and integer numbers in the given specifications are in line with conventional work on abstract datatypes. For such arguments ground-completeness is clearly an important aspect, and is the key focus of the current paper. The most closely related work seems to be that of Walters & Zantema [WZ95]; in Section 6 we briefly discuss a comparison with this work.

The paper is structured as follows. In Section 2 we start with DDRSs for commutative rings over the signature $\Sigma_r = \{0, 1, -(\_), +, \cdot\}$, which are defined by the axioms Table 1 (we shall often write $-t$ for $-(t)$). These axioms characterise integer arithmetic, while leaving open how numbers are represented (apart from the constants $0, 1 \in \Sigma_r$). For the ADT $\mathbb{Z}$ determined by these axioms we introduce two simple DDRSs that both have the same normal forms. In Section 3 we define two DDRSs for the unary view that relate to the DDRSs for the ring of integers, with numbers represented in unary view, that is, in a numeral system based on the constant 0 and a *unary append* constructor (a form of tallying). Moreover, a part of each of these DDRSs specifies natural number arithmetic. In Section 4 we define DDRSs for natural number and integer arithmetic that follow the approach in [Ber14]. These DDRSs combine unary view with binary and decimal view and are based on *digit append* constructors. In Section 5 we define similar DDRSs that employ *digit tree* constructors instead, which are algorithmically more involved. In Section 6 we come up with some conclusions. The paper contains two appendices with detailed proofs.

This paper subsumes, and thereby also replaces and improves on, our earlier paper [BP16a]. The main differences are that the DDRSs defined in Tables 2 – 6 are new, and that the DDRSs defined in all remaining tables are simplified and contain fewer rules, except those in Table 13. All termination proofs were found by the tool AProVE [Gie+17], downloads of these proofs are available at `https://arxiv.org/src/1608.06212/anc/`.

This paper has been written for the occasion of the retirement of Jos Baeten, lately as the director of the CWI in Amsterdam. For Alban, Jos was his highly valued second promotor at the University of Amsterdam. For Jan, in addition to being a former UvA colleague, Jos has been a highly respected coauthor of many papers. Jos has been participating in the ACP process algebra project from a quite early stage and the outcome of our joint work has been important for Jan's work ever since.

2. DDRSs for the ring of integers

In this section we introduce two DDRSs that both specify the ring of integers. In Section 2.1 we first give a general definition of a DDRS and discuss in what way it specifies a datatype, and then provide a DDRS for the ring of integers. In Section 2.2 we consider an alternative DDRS for the ring of integers that is deterministic with respect to rewriting a sum of two nonnegative closed normal forms.

2.1. **A DDRS for the ring of integers.** We start with a formal definition of a DDRS.

Definition 2.1. Given a many-sorted signature Σ and a finite set E of equations over Σ, a specification (Σ, E) is a **DDRS** (Dataype Defining Rewrite System) if for each sort S in Σ the following two requirements are satisfied:
(1) there is a closed term of sort S, thus, S is inhabited,
(2) the equations in E over sort S when interpreted from left to right define a ground-complete TRS.

The datatypes specified by the forthcoming DDRSs are *canonical term algebras*, which means that carriers are non-empty sets of closed terms, and for each congruence class of closed terms, a unique representing term is chosen, and this set of representing closed terms, the *normal forms*, is closed under taking subterms.

In this paper we will consider single-sorted DDRSs for integer arithmetic. We require that for each such DDRS, 0 is a constant in its signature and for each closed term t both $t + 0 = t$ and $t + (-t) = 0$ hold (so, e.g. exchanging the roles of 0 and 1 in such a DDRS is not permitted).

In [BP16, BP16a] we defined a DDRS consisting of fifteen equations for the ADT $\mathbb{Z}$ (which is determined by the axioms for commutative rings in Table 1). Normal forms for

[R1]	$x + 0 = x$	[R7]	$-0 = 0$
[R2]	$0 + x = x$	[R8]	$(-1) + 1 = 0$
[R3]	$x + (y + z) = (x + y) + z$	[R9]	$(-(x + 1)) + 1 = -x$
		[R10]	$-(-x) = x$
[R4]	$x \cdot 0 = 0$		
[R5]	$x \cdot 1 = x$	[R11]	$x + (-y) = -((-x) + y)$
[R6]	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$	[R12]	$x \cdot (-y) = -(x \cdot y)$

Table 2: The DDRS D_1 for the datatype $\mathbb{Z}_r$ that specifies the ring of integers

this DDRS are 0 for zero, the positive normal forms 1 and $t + 1$ with t a positive normal form, and the negations of positive normal forms, thus $-t$ for each positive normal form t. Clearly, two different closed normal forms have distinct values in $\mathbb{Z}$. The DDRS D_1 in Table 2 over the signature Σ_r defines the datatype $\mathbb{Z}_r$ that has the same normal forms and is hence the same CTA.

Clearly, all equations of D_1 are semantic consequences of the axioms for commutative rings (Table 1). The difference between D_1 in Table 2 and the DDRS for $\mathbb{Z}_r$ defined in [BP16, BP16a] is that equation [R11] replaces the four equations

[r5] $1 + (-1) = 0,$

[r6] $(x + 1) + (-1) = x,$

[r7] $x + (-(y + 1)) = (x + (-y)) + (-1),$

[r11] $(-x) + (-y) = -(x + y).$

Theorem 2.2. *The DDRS D_1 for $\mathbb{Z}_r$ defined in Table 2 is ground-complete.*

Proof. The AProVE tool [Gie+17] finds that this DDRS is terminating. For ground-confluence, see Corollary 2.4. □

2.2. **An alternative DDRS for the ring of integers.** The DDRS D_1 can be simplified by instantiating some of its equations. In Table 3 we provide the DDRS D_2 that also specifies the datatype $\mathbb{Z}_r$, where the differences with D_1 show up in the tags: equations [R2′], [R3′] and [R6′] replace [R2], [R3] and [R6], respectively.

Theorem 2.3. *The DDRS D_2 for $\mathbb{Z}_r$ defined in Table 3 is ground-complete.*

Proof. The AProVE tool [Gie+17] finds that this DDRS is terminating, so it remains to be proven that D_2 is ground-confluent. Define the set N of closed terms over Σ_r as follows:

$$N = \{0\} \cup N^+ \cup N^-,$$
$$N^+ = \{1\} \cup \{t + 1 \mid t \in N^+\},$$
$$N^- = \{-t \mid t \in N^+\}.$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite step applies). In order to prove ground-confluence of the DDRS D_2 it suffices to show that for each closed term t over Σ_r, either $t \in N$ or t has a rewrite step, so that each normal form is in N. This

[R1]	$x + 0 = x$	[R7]	$-0 = 0$
[R2′]	$0 + 1 = 1$	[R8]	$(-1) + 1 = 0$
[R3′]	$x + (y + 1) = (x + y) + 1$	[R9]	$(-(x + 1)) + 1 = -x$
		[R10]	$-(-x) = x$
[R4]	$x \cdot 0 = 0$		
[R5]	$x \cdot 1 = x$	[R11]	$x + (-y) = -((-x) + y)$
[R6′]	$x \cdot (y + 1) = (x \cdot y) + x$	[R12]	$x \cdot (-y) = -(x \cdot y)$

Table 3: The DDRS D_2 for the datatype $\mathbb{Z}_r$ that specifies the ring of integers

is sufficient because D_2 is terminating, all equations of D_2 are semantic consequences of the axioms for commutative rings, and distinct closed normal forms have distinct values in $\mathbb{Z}$.

We prove this by structural induction on t. The base cases $t \in \{0, 1\}$ are trivial. For the induction step we have to consider three cases:

(1) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [R7],
- if $r \in N^+$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [R10].

(2) Case $t = u + r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [R1],
- if $r = 1$, then apply case distinction on u:
  - if $u = 0$, then t has a rewrite step by equation [R2′],
  - if $u \in N^+$, then $t \in N$,
  - if $u = -1$, then t has a rewrite step by equation [R8],
  - if $u = -(u' + 1)$, then t has a rewrite step by equation [R9],
- if $r = r' + 1$, then t has a rewrite step by equation [R3′],
- if $r = -r'$ with $r' \in N^+$, then t has a rewrite step by equation [R11].

(3) Case $t = u \cdot r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [R4],
- if $r = 1$, then t has a rewrite step by equation [R5],
- if $r = r' + 1$, then t has a rewrite step by equation [R6′],
- if $r = -r'$ with $r' \in N^+$, then t has a rewrite step by equation [R12]. □

Corollary 2.4. *The DDRS D_1 for $\mathbb{Z}_r$ defined in Table 2 is ground-confluent.*

Proof. It suffices to consider the proof of Theorem 2.3 and to observe that each rewrite step by one of the equations [R2′], [R3′], and [R6′] implies a rewrite step of the associated equation in D_1, which implies ground-confuence. □

A particular property of D_2 concerns the addition of two nonnegative normal forms.

Proposition 2.5. *With respect to addition of two nonnegative closed normal forms, the DDRS D_2 in Table 3 is* ***deterministic****, that is, for nonnegative closed normal forms t, t', in each state of rewriting of $t + t'$ to its normal form, only one equation (rewrite rule) applies.*

Proof. By structural induction on t'. For $t' \in \{0, 1\}$ this is immediately clear.

If $t' = r + 1$, then the only possible rewrite step is $t + (r + 1) \xrightarrow{[R3']} (t + r) + 1$. If $r = 1$, this is a normal form, and if $r = r' + 1$, the only redex in $(t + (r' + 1)) + 1$ is in $t + (r' + 1)$, and by induction the latter rewrites deterministically to some normal form u.

Consider this reduction: $t + (r' + 1) \to u' \twoheadrightarrow u$ (where $\twoheadrightarrow$ denotes zero or more rewrite steps). Then $(t + (r' + 1)) + 1 \to u' + 1 \twoheadrightarrow u + 1$, and in each state of this reduction the rightmost addition ... $+ 1$ does not establish a new redex according to equations [R1], [R2′] and [R3′], and results in $u + 1$, which also is a normal form. $\square$

It is clear that for example $1 + t'$ with t' a closed negative normal form also rewrites deterministically in D_2, and that this determinism is not preserved for $t + t'$ if $t = 0$, e.g.,

$$0 + (-(r+1)) \xrightarrow{[R11]} -((-0) + (r+1)) \begin{bmatrix} \xrightarrow{[R3']} -(((-0) + r) + 1) \xrightarrow{[R7]} \\ \xrightarrow{[R7]} \quad -(0 + (r+1)) \quad\quad \xrightarrow{[R3']} \end{bmatrix} -((0 + r) + 1),$$

or if t is a closed negative normal form $(-t')$, e.g.,

$$(-t') + (-(r+1)) \xrightarrow{[R11]} -((-(-t')) + (r+1)) \begin{bmatrix} \xrightarrow{[R3']} -(((-(-t')) + r) + 1) \xrightarrow{[R10]} \\ \xrightarrow{[R10]} \quad -(t' + (r+1)) \quad\quad\quad \xrightarrow{[R3']} \end{bmatrix} -((t' + r) + 1).$$

However, our interest in deterministic reductions concerns nonnegative closed normal forms and we return to this point in the next section. Finally, note that with respect to multiplication of two nonnegative normal forms, the DDRS D_2 is not deterministic:

$$0 \cdot (1+1) \xrightarrow{[R6']} (0 \cdot 1) + 0 \begin{bmatrix} \xrightarrow{[R4]} 0 + 0 \xrightarrow{[R1]} \\ \xrightarrow{[R1]} \;\; 0 \cdot 1 \;\; \xrightarrow{[R4]} \end{bmatrix} 0.$$

3. DDRSs for natural number and integer arithmetic in unary view

Given the signature $\{0, 1, +, \cdot\}$, natural number arithmetic can be characterised by the axioms in Table 4 and we write $\mathbb{N}$ for the ADT captured by these axioms. In the remainder of this paper we will also discuss single-sorted DDRSs for natural number arithmetic. For each DDRS that specifies natural number arithmetic we require that 0 is a constant in its signature and that both $t + 0 = t$ and $0 \cdot t = 0$ hold for each closed term t.

All DDRSs further discussed in this paper are based on a signature Σ that contains the unary minus function $-(\_)$ and will be defined in a pairwise manner:

(1) a DDRS $(\Sigma \setminus \{-(\_)\}, E_n)$ for natural number arithmetic, and
(2) a DDRS (Σ, E_z), where $E_n \subset E_z$.

In this section we consider a simple form of number representation that is related to tallying and establishes a unary numeral system based on the constant 0. The *unary append* is the one-place (postfix) function

$$\_ :_u 1 : \mathbb{Z} \to \mathbb{Z}$$

and is an alternative notation for the successor function $S(x)$. The signature we work in is

$$\Sigma_U = \{0, -(\_),\ \_ :_u 1, +, \cdot\}.$$

[Nat1]	$x + (y + z) = (x + y) + z$	[Nat5]	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
[Nat2]	$x + y = y + x$	[Nat6]	$x \cdot y = y \cdot x$
[Nat3]	$x + 0 = x$	[Nat7]	$1 \cdot x = x$
[Nat4]	$0 \cdot x = 0$	[Nat8]	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

Table 4: Axioms for natural number arithmetic

For natural numbers, the intended normal forms are 0 for zero, and applications of the unary append function that define all successor values: each natural number n is represented by n applications of the unary append to 0 and can be seen as representing a sequence of 1's of length n having 0 as a single prefix, e.g.

$$(0 :_u 1) :_u 1$$

is the normal form that represents 2 and can be abbreviated as 011. We name the resulting datatype $\mathbb{N}_U$. Clearly, two different closed normal forms have distinct values in $\mathbb{N}$.

For integers, each minus instance $-t$ of a nonzero normal form t in $\mathbb{N}_U$ is a normal form over Σ_U, e.g.

$$-((0 :_u 1) :_u 1)$$

is the normal form that represents -2 and can be abbreviated as -011. We name the resulting datatype $\mathbb{Z}_U$, which satisfies the property that two distinct closed normal forms have distinct values in $\mathbb{Z}$.

In Section 3.1 we introduce DDRSs based on Σ_U. In Section 3.2 we investigate in what way the DDRS D_2 for the ring of integers is related.

3.1. DDRSs for $\mathbb{N}_U$ and $\mathbb{Z}_U$.

In the left column of Table 5 we define the DDRS Nat_1 for the datatype $\mathbb{N}_U$ over the signature $\Sigma_U \setminus \{-(\_)\}$. With the interpretation rule

$$[\![x :_u 1]\!] = [\![x]\!] + 1,$$

it follows that the equations in Table 5 are semantic consequences of the axioms for natural number arithmetic in Table 4.

The transition to DDRSs for $\mathbb{Z}_U$ can be taken in different ways. In Table 5 we provide the DDRS Int_1 that defines an extension of Nat_1 to integer numbers (thus, to the datatype $\mathbb{Z}_U$). With the above-mentioned interpretation rule it follows that the equations in Table 5 are semantic consequences of the axioms for commutative rings (Table 1).

Theorem 3.1. *The DDRSs Nat_1 for $\mathbb{N}_U$ and Int_1 for $\mathbb{Z}_U$ (Table 5) are ground-complete.*

Proof. The AProVE tool [Gie+17] finds that the DDRS Int_1 is terminating (and therefore the DDRS Nat_1 is terminating as well). It remains to be shown that both these DDRSs are ground-confluent. We first consider the DDRS Int_1. Define the set N as follows:

$$\begin{aligned} N &= \{0\} \cup N^+ \cup N^-, \\ N^+ &= \{0 :_u 1\} \cup \{t :_u 1 \mid t \in N^+\}, \\ N^- &= \{-t \mid t \in N^+\}. \end{aligned}$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite rule applies). In order to prove ground-confluence it suffices to show that for each closed term t over Σ_U,

[U1]	$x + 0 = x$	[U5]	$-0 = 0$
[U2]	$x + (y :_u 1) = (x :_u 1) + y$	[U6]	$(-(x :_u 1)) :_u 1 = -x$
		[U7]	$-(-x) = x$
[U3]	$x \cdot 0 = 0$		
[U4]	$x \cdot (y :_u 1) = x + (x \cdot y)$	[U8]	$x + (-y) = -((-x) + y)$
		[U9]	$x \cdot (-y) = -(x \cdot y)$

Table 5: DDRSs Nat_1 for $\mathbb{N}_U$ (left column) and Int_1 for $\mathbb{Z}_U$ that specify natural number and integer arithmetic

either $t \in N$ or t has a rewrite step, so that each normal form is in N. As in the proof of Theorem 2.3, this is sufficient because Int_1 is terminating, all equations of Int_1 are semantic consequences of the axioms for commutative rings, and distinct closed normal forms have distinct values in $\mathbb{Z}$.

We prove this by structural induction on t. The base case is simple: if $t = 0$, then $t \in N$. For the induction step we have to distinguish four cases:

(1) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [U5],
- if $r = r' :_u 1$, then $t \in N$,
- if $r = -(r' :_u 1)$, then t has a rewrite step by equation [U7].

(2) Case $t = r :_u 1$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then $t \in N$,
- if $r = r' :_u 1$, then $t \in N$,
- if $r = -(r' :_u 1)$, then t has a rewrite step by equation [U6].

(3) Case $t = u + r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [U1],
- if $r = r' :_u 1$, then t has a rewrite step by equation [U2],
- if $r = -(r' :_u 1)$, then t has a rewrite step by equation [U8].

(4) Case $t = u \cdot r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [U3],
- if $r = r' :_u 1$, then t has a rewrite step by equation [U4],
- if $r = -(r' :_u 1)$, then t has a rewrite step by equation [U9].

Ground-confluence of the DDRS Nat_1 follows in a similar way by restricting the above proof to the set of nonnegative normal forms. Moreover, a confluence proof for the DDRS Nat_1 was found by the confluence prover CSI [ZFM11] at `http://cocoweb.uibk.ac.at/` (property [CR] and options [2020, TRS, CSI]) with the input file `NAT1.trs` that is available at `https://arxiv.org/src/1608.06212/anc/`. □

The following example shows that with respect to addition of negative normal forms, the DDRS Int_1 is not deterministic:

$$(-011) + 01 \xrightarrow{[\text{U2}]} ((-011)1) + 0 \left[\begin{array}{l} \xrightarrow{[\text{U6}]} (-01) + 0 \xrightarrow{[\text{U1}]} \\ \xrightarrow{[\text{U1}]} (-011)1 \quad \xrightarrow{[\text{U6}]} \end{array} \right] -01. \tag{3.1}$$

[U1]	$x + 0 = x$	[U5]	$-0 = 0$
[U2′]	$x + (y :_u 1) = (x + y) :_u 1$	[U6]	$(-(x :_u 1)) :_u 1 = -x$
		[U7]	$-(-x) = x$
[U3]	$x \cdot 0 = 0$		
[U4′]	$x \cdot (y :_u 1) = (x \cdot y) + x$	[U8]	$x + (-y) = -((-x) + y)$
		[U9]	$x \cdot (-y) = -(x \cdot y)$

Table 6: DDRSs Nat_2 for $\mathbb{N}_U$ (left column) and Int_2 for $\mathbb{Z}_U$ that specify natural number and integer arithmetic

However, the DDRS Nat_1 for $\mathbb{N}_U$ satisfies the following property.

Proposition 3.2. *With respect to addition and multiplication of closed normal forms, the DDRS Nat_1 in Table 5 is* ***deterministic****, that is, for closed normal forms t, t', in each state of rewriting of $t + t'$ and $t \cdot t'$ to their normal form, only one equation (rewrite rule) applies.*

Proof. The case for addition is simple: applicability of equations [U1] or [U2] excludes the other.

The case for multiplication follows by structural induction on t'.

Case $t' = 0$. Equation [U3] defines the only possible rewrite step.

Case $t' = r :_u 1$. Equation [U4] defines the only possible rewrite step, resulting in $t + (t \cdot r)$. By induction, $t \cdot r$ rewrites deterministically to some normal form t_n, say $t \cdot r = t_0$ and $t_0 \to t_1 \twoheadrightarrow t_n$ for some $n > 0$. It is easily seen that t_i is the only redex in $t + t_i$ for $i \leq n$, hence also $t + (t \cdot r)$ rewrites deterministically to $t + t_n$. By the case for addition, the latter term rewrites deterministically. □

3.2. **From the ring of integers to unary view.** In this section we relate the DDRS D_2 for the ring of integers $\mathbb{Z}_r$ to integer arithmetic as defined in the previous section. If we use $t :_u 1$ as an alternative notation for $t + 1$ in D_2 and then delete all equations that contain 1 as a constant (thus $0 :_u 1 = 1$, $x \cdot 1 = 1$, and $(-1) :_u 1 = 0$), we obtain the DDRS Int_2 given in Table 6, which provides an alternative specification of integer arithmetic over the signature Σ_U comparable to the DDRS Int_1 for $\mathbb{Z}_U$ defined in Table 5. Equations [U2′] (replacing [U2]) and [U4′] (replacing [U4]) are new. Clearly, [U1] + [U2′] + [U3] + [U4′] define an alternative DDRS Nat_2 for natural number arithmetic, and these equations are semantic consequences of the axioms for natural number arithmetic (Table 4).

Theorem 3.3. *The DDRSs Nat_2 for $\mathbb{N}_U$ and Int_2 for $\mathbb{Z}_U$ (Table 6) are ground-complete.*

Proof. The AProVE tool [Gie+17] finds that these DDRSs are terminating, so it remains to be shown that both these DDRSs are ground-confluent. We first consider the DDRS Int_2. It immediately follows from the proof of Theorem 3.1 that [U2′] admits a rewrite step if [U2] does, and [U4′] admits a rewrite step if [U4] does. Hence the DDRS Int_2 is ground-confluent.

By restricting this proof to the set of nonnegative normal forms, it follows that the DDRS Nat_2 is ground-confluent. Moreover, a confluence proof for the DDRS Nat_2 has been found by the confluence prover CSI [ZFM11] at `http://cocoweb.uibk.ac.at/` (property

$0' \equiv 1$	$3' \equiv 4$	$6' \equiv 7$
$1' \equiv 2$	$4' \equiv 5$	$7' \equiv 8$
$2' \equiv 3$	$5' \equiv 6$	$8' \equiv 9$

Table 7: Enumeration and successor notation of digits of type $\mathbb{Z}$

[CR] and options [2020, TRS, CSI]) with the input file `NAT2.trs` that is available at `https://arxiv.org/src/1608.06212/anc/`. □

Furthermore, Proposition 2.5 and the above proof imply that the DDRS Int_2 is deterministic with respect to rewriting a sum of two nonnegative closed normal forms. However, Int_2 is not deterministic with respect to multiplication of two nonnegative normal forms:

$$0 \cdot (0 :_u 1) \xrightarrow{[\mathrm{U4}']} (0 \cdot 0) + 0 \begin{bmatrix} \xrightarrow{[\mathrm{U1}]} \; 0 \cdot 0 \; \xrightarrow{[\mathrm{U3}]} \\ \xrightarrow{[\mathrm{U3}]} \; 0 + 0 \; \xrightarrow{[\mathrm{U1}]} \end{bmatrix} 0.$$

4. DDRSs for combining unary, binary and decimal view

In this section we define various DDRSs for the unary, binary and decimal view. In Section 4.1 we fix a signature that comprises these views. In Section 4.2 we define two canonical term algebras that represent the unary view: $\mathbb{N}_{ubd}$ and $\mathbb{Z}_{ubd}$. Their defining DDRSs Nat_{ubd} and Int_{ubd}, respectively, comprise the conversion from numbers in binary or decimal view to unary view (hence the ordering in the subscript).

In Section 4.3 we define DDRSs for a binary view of natural and integer arithmetic, and in Section 4.4 we do the same for a decimal view of natural and integer arithmetic.

4.1. **Digits, a large signature, and two canonical term algebras.**

Digits are elements of the set $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, ordered in the common way:

$$0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9.$$

For the digits $0, 1, \ldots, 8$ we denote with i' the successor digit of i in the given enumeration. In Table 7 the successor notation on digits is specified as a transformation of syntax, and we adopt this notation throughout the paper.

In forthcoming DDRSs we will often add tags of the form

$$[\mathrm{N}n.i]_{i=k}^{\ell} \; t = r$$

with $n, k, \ell \in \mathbb{N}$ (in ordinary, decimal notation) and $k < \ell$, which represents the following $\ell - k + 1$ equations:

$$[\mathrm{N}n.k] \; t[k/i] = r[k/i], \; \ldots, \; [\mathrm{N}n.\ell] \; t[\ell/i] = r[\ell/i],$$

thus with i instantiated from k to ℓ. Occasionally, we will use this notation with two “digit counters”, as in

$$[\mathrm{N}n.i.j]_{i,j=k}^{\ell} \; t = r,$$

for a concise representation of the following $(\ell - k + 1)^2$ equations:

$$\begin{aligned}
&[\mathrm{N}n.k.k]\ t[k/i][k/j] = r[k/i][k/j],\ \ldots,\quad [\mathrm{N}n.k.\ell]\ t[k/i][\ell/j] = r[k/i][\ell/j],\\
&\ldots,\\
&[\mathrm{N}n.\ell.k]\ t[\ell/i][k/j] = r[\ell/i][k/j],\ \ldots,\quad [\mathrm{N}n.\ell.\ell]\ t[\ell/i][\ell/j] = r[\ell/i][\ell/j].
\end{aligned}$$

The signature $\Sigma_{\mathbb{Z}}$ considered henceforth has the following elements:

(1) a sort $\mathbb{Z}$,
(2) for digits the ten constants $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$,
(3) three one-place functions $S, P, - : \mathbb{Z} \to \mathbb{Z}$, "successor", "predecessor", and "minus", respectively,
(4) addition and multiplication (infix) $+, \cdot : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$,
(5) two one-place functions (postfix) $\_ :_b 0,\ \_ :_b 1 : \mathbb{Z} \to \mathbb{Z}$, "binary append zero" and "binary append one", these functions will be used for binary notation,
(6) ten one-place functions (postfix)
$$\_:_d 0,\ \_:_d 1,\ \_:_d 2,\ \_:_d 3,\ \_:_d 4,\ \_:_d 5,\ \_:_d 6,\ \_:_d 7,\ \_:_d 8,\ \_:_d 9 : \mathbb{Z} \to \mathbb{Z},$$
"decimal append zero", ...,"decimal append nine", to be used for decimal notation.

For the unary view, the normal forms we will consider are the constant (digit) 0 and the classical successor terms, that is

$$0, S(0), S(S(0)), ...$$

and all minus instances $-t$ of each nonzero normal form t, e.g. $-(S(S(0)))$. We shall use the following abbreviation, where i is a digit: $S^i(t)$ stands for i applications of the successor function S to t, thus $S^0(t) = t$ and $S^{i'}(t) = S(S^i(t))$.

For the binary view and for the decimal view, we provide one DDRS for each. Normal forms are all appropriate digits, all applications of the respective append functions to a nonzero normal form, and all minus instances $-t$ of each such normal form t that differs from 0. For example,

$$(9 :_d 7) :_d 5 \quad \text{and} \quad ((1 :_b 0) :_b 1) :_b 1$$

represent the decimal number 975 (with the interpretation rule $[\![x :_d i]\!] = 10 \cdot [\![x]\!] + i$), and the binary number 1011, respectively (with the interpretation rule $[\![x :_b i]\!] = 2 \cdot [\![x]\!] + i$).

Finally, we will include in forthcoming DDRSs equations for conversion from the one view to the other.

4.2. **Unary view.** Replacing the $t :_u 1$-occurrences by $S(t)$ in the DDRS Nat_2 defined in Table 6 results in equations [S1] – [S4] in Table 8, which hence define a DDRS for natural number arithmetic in unary view with successor function. We name this DDRS Nat_{ubd} and its CTA $\mathbb{N}_{ubd}$. The equations $[\mathbf{Sb.}i]_{i=0}^{1}$ – $[\mathbf{Sd2.}i]_{i=0}^{9}$ define the conversion of terms that employ constructors from binary or decimal view to unary view.

Integer arithmetic is obtained by replacing all $t :_u 1$-occurrences by $S(t)$ in the DDRS Int_2 defined in Table 6. We name the resuting DDRS Int_{ubd} and its CTA $\mathbb{Z}_{ubd}$. The proof of Theorem 3.3 implies the following result.

Theorem 4.1. *The DDRSs Nat_{ubd} and Int_{ubd} (Table 8) are ground-complete.*

[S1]	$x + 0 = x$	[S5]	$-0 = 0$
[S2]	$x + S(y) = S(x + y)$	[S6]	$S(-S(x)) = -x$
		[S7]	$-(-x) = x$
[S3]	$x \cdot 0 = 0$		
[S4]	$x \cdot S(y) = (x \cdot y) + x$	[S8]	$x + (-y) = -((-x) + y)$
		[S9]	$x \cdot (-y) = -(x \cdot y)$
$[\mathbf{Sb.}i\,]_{i=0}^{1}$	$x\,{:_b}\,i = (x \cdot S(1)) + i$		
$[\mathbf{Sd1.}i]_{i=0}^{8}$	$i' = S(i)$		
$[\mathbf{Sd2.}i]_{i=0}^{9}$	$x\,{:_d}\,i = (x \cdot S(9)) + i$		

Table 8: DDRSs Nat_{ubd} for $\mathbb{N}_{ubd}$ (left column) and Int_{ubd} for $\mathbb{Z}_{ubd}$

Proof. The AProVE tool [Gie⁺17] finds that these DDRSs are terminating. We first prove ground-confluence for the DDRS Int_{ubd}. It is sufficient to consider the proof of Theorem 3.3: the renaming to successor terms is not relevant, and the 'new' equations $[\mathbf{Sb.}i\,]_{i=0}^{1}$ – $[\mathbf{Sd2.}i]_{i=0}^{9}$ rewrite to successor terms and thus preserve ground-confluence.

In a similar way it follows that the DDRS Nat_{ubd} is ground-confluent. Moreover, a ground-confluence proof for the DDRS Nat_{ubd} was found by the ground-confluence prover AGCP [ATK17] at `http://cocoweb.uibk.ac.at/` (property [GCR] and options [2020, TRS, AGCP]) with input `https://arxiv.org/src/1608.06212/anc/NATubd.trs`. □

The equations [S1] – [S4] that define natural number arithmetic with 0 and successor function are very common (see, e.g. [KV03, UK03, WZ95]). Note that the DDRS Nat_{ubd} is deterministic with respect to addition of two normal forms (cf. Proposition 2.5), but not with respect to their multiplication (cf. counterexample (3.1) in Section 3.2).

4.3. **Binary view.** In the left column of Table 9 we define the DDRS Nat_{bud} for a binary view of natural numbers that employs the successor function as an auxiliary function. Leading zeros except for the zero itself are removed by $[\mathrm{b1}.i]_{i=0}^{1}$, and successor terms are rewritten according to [b2] – [b5].

Theorem 4.2. *The DDRS Nat_{bud} for $\mathbb{N}_{bud}$ (Table 9) is ground-complete.*

Proof. The AProVE tool [Gie⁺17] finds that this DDRS is terminating. Ground-confluence follows as in the proof in Appendix A.1 restricted to the set of nonnegative normal forms. □

In the right column of Table 9 minus and predecessor are introduced and the transition from a signature for natural numbers to a signature for integers is made; the rules in this table define the DDRS Int_{bud} and the canonical term algebra $\mathbb{Z}_{bud}$ that is isomorphic to the canonical term algebra $\mathbb{Z}_{ubd}$ of the DDRS Int_{ubd} in Table 8. The DDRS Int_{bud} contains twenty-eight (non-parametric) equations for the specification of numbers, addition, and multiplication. A brief comment on equations [b23] and [b24]:

$$(-x)\,{:_b}\,i$$

$[\text{b1}.i]_{i=0}^{1}$	$0 :_b i = i$	[b13]	$-0 = 0$
		[b14]	$-(-x) = x$
[b2]	$S(0) = 1$		
[b3]	$S(1) = 1 :_b 0$	[b15]	$P(0) = -1$
[b4]	$S(x :_b 0) = x :_b 1$	[b16]	$P(1) = 0$
[b5]	$S(x :_b 1) = S(x) :_b 0$	[b17]	$P(x :_b 0) = P(x) :_b 1$
		[b18]	$P(x :_b 1) = x :_b 0$
[b6]	$x + 0 = x$	[b19]	$P(-x) = -S(x)$
[b7]	$x + 1 = S(x)$		
[b8]	$x + (y :_b 0) = (x + y) + y$	[b20]	$S(-1) = 0$
[b9]	$x + (y :_b 1) = S(x + (y :_b 0))$	[b21]	$S(-(x :_b 0)) = -(P(x) :_b 1)$
		[b22]	$S(-(x :_b 1)) = -(x :_b 0)$
[b10]	$x \cdot 0 = 0$		
[b11]	$x \cdot 1 = x$	[b23]	$(-x) :_b 0 = -(x :_b 0)$
$[\text{b12}.i]_{i=0}^{1}$	$x \cdot (y :_b i) = ((x \cdot y) :_b 0) + (x \cdot i)$	[b24]	$(-x) :_b 1 = -(P(x) :_b 1)$
		[b25]	$x + (-y) = -((-x) + y)$
		[b26]	$x \cdot (-y) = -(x \cdot y)$
$[\textbf{bd1}.i]_{i=1}^{8}$	$i' = S(i)$		
$[\textbf{bd2}.i]_{i=0}^{9}$	$x :_d i = (x \cdot S(9)) + i$		

Table 9: DDRSs Nat_{bud} for $\mathbb{N}_{bud}$ (left column) and Int_{bud} for $\mathbb{Z}_{bud}$

should be equal to $(-(x :_b 0)) + i$, so $(-x) :_b 0 = -(x :_b 0)$, and $(-x) :_b 1$ is determined by

$$-(P(x :_b 0)) \overset{[\text{b24}]}{=} -(P(x) :_b 1).$$

Equations [b21] and [b22] can be explained in a similar way:

$$S(-(x :_b 0)) \text{ should be equal to } -(P(x :_b 0)) = -(P(x) :_b 1),$$
$$S(-(x :_b 1)) \text{ should be equal to } -(P(x :_b 1)) = -(x :_b 0).$$

Normal forms for $\mathbb{Z}_{bud}$ are 0, 1, all applications of $\_ :_b 0$ and $\_ :_b 1$ to a nonzero normal form, and all minus instances $-t$ of each such normal form t that differs from 0.

Note that the equations in Table 9 are semantic consequences of the axioms for commutative rings (Table 1), and that two distinct closed normal forms have distinct values in $\mathbb{Z}$.

Theorem 4.3. *The DDRS Int_{bud} for $\mathbb{Z}_{bud}$ (Table 9) is ground-complete.*

Proof. The AProVE tool [Gie+17] finds that the DDRS Int_{bud} is terminating. In Appendix A.1 we prove that the DDRS Int_{dub} is ground-confluent. $\square$

$1^\star \equiv 9$	$4^\star \equiv 6$	$7^\star \equiv 3$
$2^\star \equiv 8$	$5^\star \equiv 5$	$8^\star \equiv 2$
$3^\star \equiv 7$	$6^\star \equiv 4$	$9^\star \equiv 1$

Table 10: 10's complement notation for decimal digits

4.4. **Decimal view.** We provide DDRSs for the decimal view that are straightforward generalizations of Nat_{bud} and Int_{bud} to the decimal view. In the left column of Table 11 we define the DDRS Nat_{dub} for decimal natural numbers that defines the canonical term algebra $\mathbb{N}_{dub}$, the datatype in which unary and binary view are derived representations. This DDRS consists of twelve (parametric) equations, and another one for conversion from binary view (72 equations in total). The datatype $\mathbb{N}_{dub}$ is isomorphic to the canonical term algebra $\mathbb{N}_{ubd}$ of the DDRS Nat_{ubd} in Table 8. Leading zeros except for the zero itself are removed by $[\text{d1}.i]_{i=0}^{9}$, and successor terms are rewritten according to $[\text{d2}.i]_{i=0}^{8}$ – [d5]. In equation [d8], the notation $+^{10}$ is used for a nested sum:

$$x +^1 y = x + y \quad \text{and for } n = 1, ..., 9, \quad x +^{n+1} y = (x +^n y) + y.$$

Rewriting from binary notation is part of this DDRS, and the equation scheme $[\mathbf{db1}.i]_{i=0}^{1}$ serves that purpose. Clearly, two distinct closed normal forms have distinct values in $\mathbb{N}$, and all equations in Nat_{dub} are semantic consequences of the axioms for natural number arithmetic (Table 4).

Theorem 4.4. *The DDRS Nat_{dub} for $\mathbb{N}_{dub}$ (Table 11) is ground-complete.*

Proof. The AProVE tool [Gie+17] finds that this DDRS is terminating. Ground-confluence follows as in the proof in Appendix A.2, restricted to the set of nonnegative normal forms. □

Before we extend the DDRS Nat_{dub} to the integers, we define "10's complement", notation $i^\star$, for digits $i \in \{1, ..., 9\}$ in Table 10, which can be characterised by the equation scheme

$$i^\star = 10 - i.$$

In Table 11, we define the DDRS Int_{dub} in which minus and predecessor are added. In rule scheme $[\text{d24}.i]_{i=1}^{9}$ we employ the notation $i^\star$. The canonical term algebra thus defined is named $\mathbb{Z}_{dub}$ and is isomorphic to $\mathbb{Z}_{ubd}$ of the specification in Table 8. The DDRS Int_{dub} contains 126 equations in total, including two for conversion from binary view.

The (twenty) equations captured by [d21] – $[\text{d24}.i]_{i=1}^{9}$ can be explained in a similar fashion as was done in the previous section for [b21] – [b24], for example,

$$(-5) :_d 3$$

should be equal to $-(5 :_d 0) + 3 = -(4 :_d 7)$, and this follows immediately from equation [d24.3].

The equations of the DDRS Int_{dub} are semantic consequences of the equations for commutative rings (Table 1). It is also clear that two distinct closed normal forms have distinct values in $\mathbb{Z}$.

Theorem 4.5. *The DDRS Int_{dub} for $\mathbb{Z}_{dub}$ (Table 11) is ground-complete.*

Proof. The AProVE tool [Gie+17] finds that this DDRS is terminating. In Appendix A.2 we prove that the DDRS Int_{dub} is ground-confluent. □

$[\mathrm{d1}.i]_{i=0}^{9}$	$0 :_d i = i$	[d13]	$-0 = 0$
$[\mathrm{d2}.i]_{i=0}^{8}$	$S(i) = i'$	[d14]	$-(-x) = x$
[d3]	$S(9) = 1 :_d 0$	[d15]	$P(0) = -1$
$[\mathrm{d4}.i]_{i=0}^{8}$	$S(x :_d i) = x :_d i'$	$[\mathrm{d16}.i]_{i=0}^{8}$	$P(i') = i$
[d5]	$S(x :_d 9) = S(x) :_d 0$	[d17]	$P(x :_d 0) = P(x) :_d 9$
[d6]	$x + 0 = x$	$[\mathrm{d18}.i]_{i=0}^{8}$	$P(x :_d i') = x :_d i$
$[\mathrm{d7}.i]_{i=1}^{9}$	$x + i = S^i(x)$	[d19]	$P(-x) = -S(x)$
[d8]	$x + (y :_d 0) = x +^{10} y$	$[\mathrm{d20}.i]_{i=0}^{8}$	$S(-i') = -i$
$[\mathrm{d9}.i]_{i=1}^{9}$	$x + (y :_d i) = S^i(x + (y :_d 0))$	[d21]	$S(-(x :_d 0)) = -(P(x) :_d 9)$
[d10]	$x \cdot 0 = 0$	$[\mathrm{d22}.i]_{i=0}^{8}$	$S(-(x :_d i')) = -(x :_d i)$
$[\mathrm{d11}.i]_{i=0}^{8}$	$x \cdot i' = (x \cdot i) + x$	[d23]	$(-x) :_d 0 = -(x :_d 0)$
$[\mathrm{d12}.i]_{i=0}^{9}$	$x \cdot (y :_d i) = ((x \cdot y) :_d 0) + (x \cdot i)$	$[\mathrm{d24}.i]_{i=1}^{9}$	$(-x) :_d i = -(P(x) :_d i^\star)$
		[d25]	$x + (-y) = -((-x) + y)$
$[\mathbf{db1}.i]_{i=0}^{1}$	$x :_b i = (x + x) + i$	[d26]	$x \cdot (-y) = -(x \cdot y)$

Table 11: DDRSs Nat_{dub} for $\mathbb{N}_{dub}$ (left column) and Int_{dub} for $\mathbb{Z}_{dub}$ that specify natural number and integer arithmetic in decimal view, employing $i^\star$ from Table 10

5. DDRSs with digit tree constructors

Having defined DDRSs that employ (postfix) digit append functions in Sections 3 and 4, we now consider the more general *digit tree constructor* functions. For the binary view, this approach is followed by Bouma & Walters in [BW89]; for a view based on any radix (number base), this approach is further continued in Walters [Wal94] and Walters & Zantema [WZ95], where the constructor is called *juxtaposition* because it goes with the absence of a function symbol in order to be close to ordinary decimal and binary notation.

We extend the signature $\Sigma_\mathbb{Z}$ defined in Section 4.1 with the following three functions (infix):

$$\hat{}_u\,,\ \hat{}_b\,,\ \hat{}_d\ : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z},$$

called "unary digit tree constructor function", "binary digit tree constructor function", and "decimal digit tree constructor function", and to be used for unary, binary notation and decimal notation, respectively. The latter two constructors serve to represent positional notation and satisfy the semantic equations

$$[\![x \,\hat{}_b\, y]\!] = 2 \cdot [\![x]\!] + [\![y]\!] \quad \text{and} \quad [\![x \,\hat{}_d\, y]\!] = 10 \cdot [\![x]\!] + [\![y]\!].$$

For integer numbers in decimal view or binary view, normal forms are the relevant digits, all applications of the respective constructor with left argument a nonzero normal form and

[ut1]	$x \mathbin{\hat{_u}} (y \mathbin{\hat{_u}} z) = (x \mathbin{\hat{_u}} y) \mathbin{\hat{_u}} z$	[ut6]	$-0 = 0$
		[ut7]	$-(-x) = x$
[ut2]	$x + 0 = x$		
[ut3]	$x + (y \mathbin{\hat{_u}} 0) = (x + y) \mathbin{\hat{_u}} 0$	[ut8]	$0 \mathbin{\hat{_u}} (-(x \mathbin{\hat{_u}} 0)) = -x$
[ut4]	$x \cdot 0 = 0$	[ut9]	$(x \mathbin{\hat{_u}} 0) \mathbin{\hat{_u}} (-(y \mathbin{\hat{_u}} 0)) = x \mathbin{\hat{_u}} (-y)$
[ut5]	$x \cdot (y \mathbin{\hat{_u}} 0) = (x \cdot y) + x$	[ut10]	$(-(x \mathbin{\hat{_u}} 0)) \mathbin{\hat{_u}} 0 = -x$
		[ut11]	$(-(x \mathbin{\hat{_u}} 0)) \mathbin{\hat{_u}} (y \mathbin{\hat{_u}} 0) = (-x) \mathbin{\hat{_u}} y$
		[ut12]	$(-(x \mathbin{\hat{_u}} 0)) \mathbin{\hat{_u}} (-(y \mathbin{\hat{_u}} 0)) = -((x + y) \mathbin{\hat{_u}} 0)$
		[ut13]	$x + (-y) = -((-x) + y)$
		[ut14]	$x \cdot (-y) = -(x \cdot y)$

Table 12: DDRSs Nat_{ut} for $\mathbb{N}_{ut}$ (left column) and Int_{ut} for $\mathbb{Z}_{ut}$ that specify natural number and integer arithmetic in unary view with unary digit tree constructor

right argument a digit, and all minus instances $-t$ of each such nonzero normal form t, these satisfy $[\![-(t)]\!] = -([\![t]\!])$. E.g.,

$$(9 \mathbin{\hat{_d}} 7) \mathbin{\hat{_d}} 5 \quad \text{and} \quad ((1 \mathbin{\hat{_b}} 0) \mathbin{\hat{_b}} 1) \mathbin{\hat{_b}} 1$$

represent the decimal number 975 and the binary number 1011, respectively, and the normal form that represents the additional inverse of the latter is $-(((1 \mathbin{\hat{_b}} 0) \mathbin{\hat{_b}} 1) \mathbin{\hat{_b}} 1)$. A minor complication with decimal and binary digit tree constructors is that we now have to consider rewritings such as

$$2 \mathbin{\hat{_d}} (1 \mathbin{\hat{_d}} 5) = (2 + 1) \mathbin{\hat{_d}} 5 = 3 \mathbin{\hat{_d}} 5 \quad (= 35),$$

which perhaps are somewhat non-intuitive. For integers in unary view, thus with unary digit tree constructor, this complication is absent (see Section 5.1).

We keep the presentation of the resulting DDRSs (those defining the binary and decimal view are based on Walters [Wal94] and Walters & Zantema [WZ95]) minimal in the sense that equations for conversion from the one view to the other are left out. Of course, it is easy to define such equations. Also, equations for conversion to and from the datatypes defined in Section 4 are omitted, although such equations are also easy to define.

5.1. **Unary view with digit tree constructor.** For naturals in this particular unary view, normal forms are 0 and expressions $t \mathbin{\hat{_u}} 0$ with t a normal form (thus, with association of $\hat{_u}$ to the left). Of course, the phenomenon of "removing leading zeros" does not exist in this unary view. The resulting datatype $\mathbb{N}_{ut}$ is defined by the DDRS Nat_{ut} in Table 12.

The constructor $\hat{_u}$ is an associative operator, as is clear from rule [ut1] (in contrast to digit tree constructors for the binary and decimal case). Moreover, the commutative variants $t \mathbin{\hat{_u}} r$ and $r \mathbin{\hat{_u}} t$ rewrite to the same normal form, which also is implied by the semantics for closed terms:

$$[\![0]\!] = 0, \qquad [\![x + y]\!] = [\![x]\!] + [\![y]\!],$$
$$[\![x \mathbin{\hat{_u}} y]\!] = [\![x]\!] + [\![y]\!] + 1, \qquad [\![x \cdot y]\!] = [\![x]\!] \cdot [\![y]\!].$$

Clearly, all equations in Nat_{ut} are semantic consequences of the axioms for natural number arithmetic (Table 4) and two distinct closed normal forms have distinct values in $\mathbb{N}$.

The extension to integer numbers can be done in a similar fashion as in the previous section, thus obtaining normal forms of the form $-(t)$ with t a nonzero normal form in $\mathbb{N}_{ut}$. However, also terms of the form $x \mathbin{\hat{}_u} (-y)$ and variations thereof have to be considered. We define the DDRS Int_{ut} in Table 12 and we call the resulting datatype $\mathbb{Z}_{ut}$. Adding the interpretation rule $[\![-x]\!] = -[\![x]\!]$, it can be easily checked that also [ut6] – [ut14] are semantic consequences of the axioms for commutative rings, and that two distinct closed normal forms have distinct values in $\mathbb{Z}$.

Theorem 5.1. *The DDRSs Nat_{ut} and Int_{ut} (Table 12) are ground-complete.*

Proof. The AProVE tool [Gie+17] finds that both these DDRSs are terminating. In Appendix B.1 we prove that the DDRS Int_{ut} is ground-confluent.

Ground-confluence of the DDRS Nat_{ut} easily follows by restricting this proof to the set of nonnegative normal forms. Moreover, a ground-confluence proof for this DDRS was found by the ground-confluence prover AGCP [ATK17] at `http://cocoweb.uibk.ac.at/` (property [GCR], options [2020, TRS, AGCP]) with the input file `NATut.trs` available at `https://arxiv.org/src/1608.06212/anc/`. □

5.2. **Binary view with digit tree constructor.** For naturals in binary view with the binary digit tree constructor, the associated datatype $\mathbb{N}_{bt}$ is defined by the DDRS Nat_{bt} in Table 13 (in the left column). In [KW16] it is proven that the associated TRS is terminating.

In [WZ95], Walters & Zantema provide a rewriting system for integer arithmetic with next to juxtaposition and minus, also addition, subtraction and multiplication, and prove its ground-completeness with respect to any radix (number base). In Table 13 we define the DDRS Int_{bt} that defines the datatype $\mathbb{Z}_{bt}$ as a variant of this rewriting system without subtraction (using the binary digit tree constructor). Clearly, two distinct closed normal forms have distinct values in $\mathbb{Z}$. Furthermore, the equations in Table 13 are semantic consequences of the axioms for commutative rings (Table 1).

Theorem 5.2. *The DDRSs Nat_{bt} and Int_{bt} (Table 13) are ground-complete.*

Proof. The AProVE tool [Gie+17] finds that both these DDRSs are terminating (as reported in [KW16]). In Appendix B.2 we prove that the DDRS Int_{bt} is ground-confluent.

Ground-confluence of Nat_{bt} follows in a similar way by restricting this proof to the set of nonnegative normal forms. □

5.3. **Decimal view with digit tree constructor.** For the specification of naturals in decimal view with the decimal digit tree constructor we make use of successor terms in order to avoid (non-parametric) equations such as

$$\begin{aligned}
&1+1=2,\ \ldots,\quad 9+8=1 \mathbin{\hat{}_d} 7,\quad 9+9=1 \mathbin{\hat{}_d} 8,\\
&\ldots,\\
&1\cdot 1=1,\ \ldots,\quad 8\cdot 9=7 \mathbin{\hat{}_d} 2,\quad 9\cdot 9=8 \mathbin{\hat{}_d} 1.
\end{aligned}$$

[bt1]	$0 \mathbin{\hat{}_b} x = x$	[bt13]	$-0 = 0$
[bt2]	$x \mathbin{\hat{}_b} (y \mathbin{\hat{}_b} z) = (x+y) \mathbin{\hat{}_b} z$	[bt14]	$-(-x) = x$
[bt3]	$0 + x = x$	[bt15]	$1 \mathbin{\hat{}_b} (-1) = 1$
[bt4]	$x + 0 = x$	[bt16]	$(x \mathbin{\hat{}_b} 0) \mathbin{\hat{}_b} (-1) = (x \mathbin{\hat{}_b} (-1)) \mathbin{\hat{}_b} 1$
[bt5]	$1 + 1 = 1 \mathbin{\hat{}_b} 0$	[bt17]	$(x \mathbin{\hat{}_b} 1) \mathbin{\hat{}_b} (-1) = (x \mathbin{\hat{}_b} 0) \mathbin{\hat{}_b} 1$
[bt6]	$x + (y \mathbin{\hat{}_b} z) = y \mathbin{\hat{}_b} (x+z)$	[bt18]	$x \mathbin{\hat{}_b} (-(y \mathbin{\hat{}_b} z)) = -((y + (-x)) \mathbin{\hat{}_b} z)$
[bt7]	$(x \mathbin{\hat{}_b} y) + z = x \mathbin{\hat{}_b} (y+z)$	[bt19]	$(-x) \mathbin{\hat{}_b} y = -(x \mathbin{\hat{}_b} (-y))$
[bt8]	$x \cdot 0 = 0$	[bt20]	$1 + (-1) = 0$
[bt9]	$0 \cdot x = 0$	[bt21]	$(-1) + 1 = 0$
[bt10]	$1 \cdot 1 = 1$	[bt22]	$(-1) + (-1) = -(1 \mathbin{\hat{}_b} 0)$
[bt11]	$x \cdot (y \mathbin{\hat{}_b} z) = (x \cdot y) \mathbin{\hat{}_b} (x \cdot z)$	[bt23]	$x + (-(y \mathbin{\hat{}_b} z)) = -(y \mathbin{\hat{}_b} (z + (-x)))$
[bt12]	$(x \mathbin{\hat{}_b} y) \cdot z = (x \cdot z) \mathbin{\hat{}_b} (y \cdot z)$	[bt24]	$(-(x \mathbin{\hat{}_b} y)) + z = -(x \mathbin{\hat{}_b} (y + (-z)))$
		[bt25]	$x \cdot (-y) = -(x \cdot y)$
		[bt26]	$(-x) \cdot y = -(x \cdot y)$

Table 13: DDRSs Nat_{bt} for $\mathbb{N}_{bt}$ (left column) and Int_{bt} for $\mathbb{Z}_{bt}$ for natural number and integer arithmetic in binary view with binary digit tree constructor

The associated datatype $\mathbb{N}_{dt}$ is defined by the DDRS Nat_{dt} in Table 14 (left column). In equations $[\text{dt10}.i]_{i=1}^{9}$ we use the notation

$$\textstyle\sum^i x$$

for $i-1$ repeated applications of $+$ with association to the left, thus

$$\textstyle\sum^1 x = x \quad \text{and for } i = 1, ..., 8, \quad \sum^{i+1} x = (\sum^i x) + x.$$

Observe that equations $[\text{dt2}.i]_{i=0}^{9}$ are instances of their binary counterpart [bt2] (see Table 13), although the AProVE tool [Gie$^+$17] finds that the DDRS Nat_{dt} is terminating if we replace $[\text{dt2}.i]_{i=0}^{9}$ by

$$x \mathbin{\hat{}_d} (y \mathbin{\hat{}_d} z) = (x+y) \mathbin{\hat{}_d} z,$$

and also a ground-confluence proof can be easily given. The reason for this replacement concerns the generalization to integer arithmetic, as discussed below.

The extension to integers is given by the DDRS Int_{dt} in Table 14, which defines the datatype $\mathbb{Z}_{dt}$. Clearly, two distinct closed normal forms have distinct values in $\mathbb{Z}$. Furthermore, the equations in Table 14 are semantic consequences of the axioms for commutative rings (Table 1).

In contrast to the approaches in Walters [Wal94] and Walters & Zantema [WZ95] with juxtaposition, we now make use of successor terms, and the DDRS presented here is composed from rewrite rules for successor, rewrite rules defined in [Wal94] and [WZ95], and combinations thereof.

[dt1]	$0 \,\hat{}_d\, x = x$	[dt12]	$-0 = 0$
$[\text{dt2}.i]_{i=0}^{9}$	$x \,\hat{}_d\, (y \,\hat{}_d\, i) = (x+y) \,\hat{}_d\, i$	[dt13]	$-(-x) = x$
$[\text{dt3}.i]_{i=0}^{8}$	$S(i) = i'$	$[\text{dt14}.i]_{i=0}^{8}$	$S(-i') = -i$
[dt4]	$S(9) = 1 \,\hat{}_d\, 0$	[dt15]	$S(-(x \,\hat{}_d\, 0)) = -(x \,\hat{}_d\, (-1))$
$[\text{dt5}.i]_{i=0}^{8}$	$S(x \,\hat{}_d\, i) = x \,\hat{}_d\, i'$	$[\text{dt16}.i]_{i=0}^{8}$	$S(-(x \,\hat{}_d\, i')) = -(x \,\hat{}_d\, i)$
[dt6]	$S(x \,\hat{}_d\, 9) = S(x) \,\hat{}_d\, 0$	[dt17]	$1 \,\hat{}_d\, (-1) = 9$
$[\text{dt7}.i]_{i=0}^{9}$	$x + i = S^i(x)$	$[\text{dt18}.i]_{i=1}^{8}$	$1 \,\hat{}_d\, (-i') = i^*$
$[\text{dt8}.i]_{i=0}^{9}$	$x + (y \,\hat{}_d\, i) = S^i(y \,\hat{}_d\, x)$	$[\text{dt19}.i]_{i=1}^{8}$	$i' \,\hat{}_d\, (-1) = i \,\hat{}_d\, 9$
[dt9]	$x \cdot 0 = 0$	$[\text{dt20}.i.j]_{i,j=1}^{8}$	$i' \,\hat{}_d\, (-j') = i \,\hat{}_d\, j^*$
$[\text{dt10}.i]_{i=1}^{9}$	$x \cdot i = \sum^i x$	$[\text{dt21}.i]_{i=1}^{9}$	$(x \,\hat{}_d\, 0) \,\hat{}_d\, (-i) = (x \,\hat{}_d\, (-1)) \,\hat{}_d\, i^*$
$[\text{dt11}.i]_{i=0}^{9}$	$x \cdot (y \,\hat{}_d\, i) = ((x \cdot y) \,\hat{}_d\, 0) + (x \cdot i)$	$[\text{dt22}.i.j]_{i,j=0}^{8}$	$(x \,\hat{}_d\, i') \,\hat{}_d\, (-j') = (x \,\hat{}_d\, i) \,\hat{}_d\, (j')^*$
		[dt23]	$(-x) \,\hat{}_d\, y = -(x \,\hat{}_d\, (-y))$
		$[\text{dt24}.i]_{i=0}^{9}$	$x \,\hat{}_d\, (-(y \,\hat{}_d\, i)) = -(((-x)+y) \,\hat{}_d\, i)$
		[dt25]	$x + (-y) = -((-x)+y)$
		[dt26]	$x \cdot (-y) = -(x \cdot y)$

Table 14: DDRSs Nat_{dt} for $\mathbb{N}_{dt}$ (left column) and Int_{dt} for $\mathbb{Z}_{dt}$, for natural number and integer arithmetic with decimal digit tree constructor (using the notations i' from Table 7 and $i^\star$ from Table 10)

The above-mentioned equation $x \,\hat{}_d\, (y \,\hat{}_d\, z) = (x+y) \,\hat{}_d\, z$ is replaced by the ten equations $[\text{dt2}.i]_{i=0}^{9}$ because we failed to find a TRS that employed this equation and that could be proven terminating by the AProVE tool. For uniformity, we also replaced the decimal counterpart of equation [bt18] (see Table 13), that is

$$x \,\hat{}_d\, (-(y \,\hat{}_d\, z)) = -((y + (-x)) \,\hat{}_d\, z)$$

by the ten equations $[\text{dt24}.i]_{i=0}^{9}$.

Theorem 5.3. *The DDRSs Nat_{dt} and Int_{dt} (Table 14) are ground-complete.*

Proof. The AProVE tool [Gie$^+$17] finds that both these DDRSs are terminating. In Appendix B.2 we prove that the DDRS Int_{dt} is ground-confluent.

Ground-confluence of Nat_{dt} follows in a similar way by restricting the proof to the set of nonnegative normal forms. □

We finally note that when we convert the DDRSs from Table 14 to base 2, we obtain alternative DDRSs for the canonical term algebras $\mathbb{N}_{bt}$ and $\mathbb{Z}_{bt}$ that are also ground-complete.

6. Conclusions

This paper is about defining (by means of trial and error) DDRSs for natural number and integer arithmetic rather than about the precise analysis of the various rewriting systems per se. What matters in addition to readability and conciseness of each DDRS is at this stage a proof that it is terminating and ground-confluent (and thus ground-complete), and furthermore that the (intended) normal forms are natural and convincing, while the rewriting systems are comprehensible.

In Section 2 we provided two DDRSs for the datatype $\mathbb{Z}_r$, the ring of integers with the set N of normal forms defined by

$$N = \{0\} \cup N^+ \cup N^-, \quad N^+ = \{1\} \cup \{t+1 \mid t \in N^+\}, \quad N^- = \{-t \mid t \in N^+\}.$$

Each of these DDRSs consists of twelve equations. Perhaps the DDRS D_2 is most attractive: it is comprehensible and deterministic with respect to addition of nonnegative closed normal forms. We leave it as an open question whether $\mathbb{Z}_r$ can be specified by a DDRS with fewer equations (preserving the set N as normal forms). Another open question is to find a DDRS for $\mathbb{Z}_r$ and N that is also deterministic with respect to rewriting $t \cdot t'$ for nonnegative closed normal forms t and t'.[1] One more open question is whether $\mathbb{Z}_r$ can be specified by a complete term rewriting system with the same normal forms.

In Section 3 we provided two DDRSs for natural number and integer arithmetic in unary view, based on the constant 0 and unary append. Both DDRSs for integer arithmetic contain only nine equations and we leave it as an open question whether the resulting canonical term algebra $\mathbb{Z}_U$ can be specified as a DDRS with fewer equations. Concerning their counterparts that define natural number arithmetic (Nat_1 and Nat_2, both containing four equations), the DDRS Nat_1 is deterministic with respect to addition and multiplication of closed normal forms (Proposition 3.2). Furthermore, the DDRSs Nat_1 and Nat_2 are attractive, if only from a didactical point of view:

(1) Positive numbers are directly related to tallying and admit an easy representation and simplifying abbreviations for normal forms, such as 011 for $(0 :_u 1) :_u 1$, or even 11 or $||$ when removal of the leading zero in positive numbers is adopted.

(2) Natural number arithmetic on small numbers can be represented in a comprehensible way that is fully independent of the learning of any positional system for number representation, although names of numbers (zero, one, two, and so on) might be very helpful.[2] Furthermore, notational abbreviations for units of five, like in

$$\cancel{1111}\ \cancel{1111}\ \ 11 \quad \text{or} \quad \cancel{||||}\ \cancel{||||}\ || \quad \text{or} \quad 011111\ 11111\ 11$$

can be helpful because 0111111111111 (thus twelve) is not very well readable or easily distinguishable from 011111111111 (thus eleven).

With respect to negative numbers, similar remarks can be made, but displaying computations according to the DDRSs Int_1 and Int_2 will be more complex and bracketing seems to be unavoidable. Consider for example

$$(-(\,||\,)) + \cancel{||||}\ \cancel{||||}\ || \;=\; (-(\,||\,)\,|\,) + \cancel{||||}\ \cancel{||||}\ | \;=\; (-(\,|\,)) + \cancel{||||}\ \cancel{||||}\ | \;=\; \ldots$$

[1] Note that the alternative for equation [R6] suggested by the DDRS Int_2, does not solve this open question.

[2] In English, Dutch and German, this naming is up to twelve independent of decimal representation, and in French this is up to sixteen.

Although it can be maintained that as a constructor, unary append is a more illustrative notation than the successor function, it is of course only syntactic sugar for that function.

Furthermore, if we add the predecessor function $P(x)$ to the DDRSs defined in Section 3 by the three equations

[P1] $P(0) = -(0 :_u 1)$, [P2] $P(x :_u 1) = x$, [P3] $P(-x) = -(x :_u 1)$

we find that the resulting DDRSs improve on those for unary view defined in [BP16a] in terms of simplicity and number of equations. Finally, if we also add the subtraction function $x - y$ by the single equation

[Sub] $x - y = x + (-y)$

this also improves on the specification of integer arithmetic in unary view in Walters & Zantema [WZ95], which does not employ the unary minus function and contains seventeen rules, and thus eighteen rules when adding the minus function by the rewrite rule

$$-x \ \to \ 0 - x.$$

However, we should mention that the normal forms for negative numbers in [WZ95] are $P(0), P(P(0)), ...,$ instead of $-(0 :_u 1), -((0 :_u 1) :_u 1), ...$

In Section 4 we considered the question how to specify a datatype of integers as an extension of the naturals as specified in [Ber14]. In this case, the unary view leads to satisfactory results, but with high inefficiency. For the binary view and the decimal view based on the unary append functions as discussed in this section, such extensions are provided, but the resulting rewriting systems are at first sight significantly less concise and comprehensible. Some additional notes:

(1) The three DDRSs for integers given in Section 4 each produce an extension datatype for a datatype for the natural numbers. An initial algebra specification of the datatype of integers is obtained from any of the DDRSs given in [Ber14] by
   - taking the reduct to the signature involving unary, binary, and decimal notation only,
   - expanding the signature with a unary additive inverse and a unary predecessor function,
   - adding rewrite rules (in equational form) that allow for the unique normalisation of closed terms involving the minus sign, while making sure that these rewrite rules (viewed as equations) are semantic consequences of the equations for commutative rings.

(2) The DDRSs for the binary view and the decimal view are hardly intelligible unless one knows that the objective is to construct a commutative ring. A decimal normal form is defined as either a digit, or an application of a decimal append function $\_ :_d i$ to a nonzero normal form (for all digits i). This implies the absence of (superfluous) leading zeros, and the closed normal forms thus obtained correspond bijectively to the nonnegative integers (that is, $\mathbb{N}$). Incorporating all minus instances $-(t)$ of each nonzero normal form t yields the class of closed normal forms.

(3) Understanding the concept of a commutative ring can be expected only from a person who has already acquired an understanding of the structure of integers and who accepts the concept of generalization of a structure to a class of structures sharing some but not all of its properties.

   In other words, the understanding that a DDRS for the integers is provided in the binary view and in the decimal view can only be communicated to an audience under the assumption that a reliable mental picture of the integers already exists in the minds of members of the audience. This mental picture, however, can in principle be communicated by taking notice of the DDRS for the unary view first. This conceptual (near) circularity may nevertheless be considered a significant weakness of the approach of defining (and even introducing) the integers as an extension of naturals by means of rewriting.

In Section 5 we discussed some alternatives for the above-mentioned DDRSs based on papers of Bouma & Walters [BW89], Walters [Wal94], and Walters and Zantema [WZ95] in which digit tree constructors are used. In [Wal94], Walters presents a TRS based on juxtaposition as a tree constructor for integer arithmetic with addition and subtraction that is ground-complete and parametric over any radix. In [WZ95], Walters and Zantema extend this TRS with multiplication and prove ground-completeness, using *semantic labelling* for their termination proof, and judge this TRS — named JP (juxtaposition) — to have good efficiency and readability (in comparison with some alternatives discussed in that paper). In [KW16], Kluiving and van Woerkom showed termination of the two DDRSs Nat_{bt} and Int_{bt} for arithmetic over $\mathbb{N}$ and $\mathbb{Z}$ that employ the binary tree constructors (Table 13) with the tool AProVE [Gie$^+$17]. Furthermore, they also proposed a TRS for arithmetic over the natural numbers employing decimal tree constructors and proved termination with AProVE. However, its natural extension to a TRS for integer arithmetic could not be proven terminating, probably due to its size. This finally led us to the DDRSs Nat_{dt} and Int_{dt} (Table 14). We leave it as an open question whether $\mathbb{Z}_{dt}$ (the datatype defined by Int_{dt}) can be specified with fewer equations in such a way that a termination proof can be found with AProVE (or another tool).

A general property of the DDRSs defined in this paper is that the recursion in the definitions of addition and multiplication takes place on the right argument of these operators (as is common), if necessary first replacing negation. We could have used recursion on the left argument instead, obtaining symmetric versions of these DDRSs (for natural number arithmetic with successor function, this is done in e.g. Bouma & Walters [BW89], Walters [Wal90], and Zantema [Zan03]).

Of course, many normal forms in decimal notation have *names* that confirm their base, for example "six hundred eighty-nine" ⟨ae⟩ or "six hundred and eighty-nine" ⟨be⟩. A decimal notation as 689 is so common that one usually does not question whether it represents $(6:_d 8):_d 9$ or $(6 \hat{\ }_d 8) \hat{\ }_d 9$ or some other formally defined notation. Nevertheless, as we have seen, different algorithmic approaches to for example addition may apply, although one would preferably not hamper an (initial) arithmetical method with notation such as $x \hat{\ }_d (y \hat{\ }_d z)$ and rewrite rules such as $x \hat{\ }_d (y \hat{\ }_d z) \to (x+y) \hat{\ }_d z$, and for this reason we have a preference for the DDRSs that employ the various append constructors.

In Table 15 we present the rule count of the DDRSs for decimal representation of natural number and integer arithmetic defined in this paper, and that of the TRSs considered in Walters & Zantema [WZ95] (the TRS named DA is based on digit append constructors and is discussed below). Note that we do *not* count equations that define the informal extra operators such as $+^{10}$ (Table 11) and $\sum^1, ..., \sum^9$ (Table 14), which were used only to get a readable table. For a fair comparison in the case of the decimal append functions we also leave out the equations for conversion to binary view. This shows that our DDRSs are relatively concise, but we note that we did not succeed in getting a more extended overview on specifications for natural number and integer arithmetic, and that it might well be that we missed relevant TRSs.

Name	Nat_{dub}	Nat_{dt}	DA(10)	Int_{dub}	Int_{dt}	JP(10)
rule count	70	71	235	124	276	438
rule schemes	12	11	10	26	26	30
extra operators	$S(x)$	$S(x)$	$s(x), \star_{\delta 0}$	$S(x),\ P(x)$	$S(x)$	$x - y$
Table / reference	11	14	[WZ95]	11	14	[WZ95]

Table 15: Rule count for decimal natural and integer arithmetic, where DA(10) and JP(10) both originate from [WZ95] instantiated for base 10.

For each DDRS it may be taken for a quality criterion if normalising reductions are not excessively long. We leave it as an open question for each of the systems presented to find meaningful upper bounds (in terms of n) of the number of steps of the longest reduction which can be made from a ground term of size n. Especially for the cases of binary and decimal numbers it is interesting to compare these values with the number of steps needed for instance when performing leftmost innermost normalisation where addition and multiplication are performed by means of the standard ("school") algorithms, or even with faster algorithms like Schönhage-Strassen for multiplication.

We have proven that all DDRSs considered in this paper are ground-complete, and both their termination proofs and all associated TRSs used for these proofs can be found at `https://arxiv.org/src/1608.06212/anc/`. It should be noted that handwritten ground-confluence proofs of the size recorded in Appendices A and B are error-prone; however, our proofs can be automated as was shown by van Woerkom in [Woe17], which also contains a general theorem about this type of ground-confluence proofs (Thm.1). Furthermore, with the confluence prover CSI [ZFM11], confluence proofs were found for the DDRSs for natural number arithmetic with unary append Nat_1 (Table 5) and Nat_2 (Table 6), thus these DDRSs are complete. For all other DDRSs defined in this paper, confluence can be disproven by CSI. Finally, the tool AGCP [ATK17] found ground-confluence proofs for the DDRSs for natural number arithmetic with unary append Nat_{ubd} (Table 8) and with unary digit tree constructor Nat_{ut} (Table 12), and also for the DDRSs Nat_1 and Nat_2, but for none of the remaining DDRSs.

We briefly discuss two other, comparable approaches to arithmetic that are also based on some form of digit append constructors for representing numbers. First, in [WZ95] Walters and Zantema introduce a TRS which they named DA (for "digit application") with addition and multiplication on natural numbers. The authors prove termination by recursive path ordering and confluence, and also judge this TRS to have good efficiency and readability. Secondly, in [CMR97], Contejean, Marché and Rabehasaina introduce integer arithmetic based on *balanced ternary numbers*, that is, numbers that can be represented by a digit append function $:_t$ with digits $-1, 0, 1$ and semantics $[\![i]\!] = i$ and $[\![x :_t i]\!] = 3 \cdot [\![x]\!] + i$ (see, e.g., Knuth [Knu97]) and provide a TRS that is confluent and terminating modulo associativity and commutativity of addition and multiplication.

Based on either a DDRS for the natural numbers or a DDRS for the integers one may develop a DDRS for rational numbers in various ways. It is plausible to consider the meadow of rational numbers of [BT07] or the non-involutive meadow of rational numbers (see [BM15]) or the common meadow of rational numbers (see [BP15]) as abstract algebraic structures for rationals in which unary, binary, and decimal notation are to be incorporated in ways possibly based on the specifications presented above. Furthermore, one does well to consider the work discussed in [CMR97] on a TRS for rational numbers, in which arithmetic for rational numbers is specified (this is the main result in [CMR97], for which the above-mentioned work on integer arithmetic is a preliminary): the authors specify rational numbers by means of a TRS that is complete modulo associativity and commutativity of addition and multiplication, taking advantage of Stein's algorithm for computing gcd's of nonnegative integers without any division[3] (see, e.g., [Knu97]).

A survey of equational algebraic specifications for abstract datatypes is provided by Wirsing [Wir91]. In [BT95] one finds the general result that computable abstract datatypes can be specified by means of specifications which are confluent and terminating term rewriting systems. Some general results on algebraic specifications can be found in [BWP84, BT87, CJ98]. More recent applications of equational specifications can be found in [BT07].

Acknowledgement

Many thanks to Wijnand van Woerkom for carefully identifying errors and gaps in an earlier version of this work, and for some very useful suggestions, including those for better rewrite rules for the DDRS Int_{dt} (Table 14). Furthermore, his help in proving termination of some of the larger DDRSs with the AProVE tool and providing these proofs was crucial[4] and led to the webarchive `https://arxiv.org/src/1608.06212/anc/`. Also, many thanks to three reviewers for their careful and comprehensive reports, and very helpful suggestions.

References

[ATK17] Aoto, T., Toyama, Y., and Kimura, Y. (2017). Improving rewriting induction approach for proving ground confluence. In D. Miller (Ed.): *Proceedings of the 2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017*, volume 84 of LIPIcs, pp. 7:1–7:18. Web interface for the ground-confluence prover AGCP: `http://cocoweb.uibk.ac.at/` (choose property [GCR] and options [2020, TRS, AGCP], last accessed 4 January 2021).

[Ber14] Bergstra, J.A. (2014). Four complete datatype defining rewrite systems for an abstract datatype of natural numbers. Electronic report TCS1407v2, University of Amsterdam, Informatics Institute, section Theory of Computer Science (August 2014), `https://ivi.fnwi.uva.nl/tcs/pub/tcsreports/TCS1407v2.pdf`.

[BM15] Bergstra, J.A. and Middelburg, C.A. (2015). Division by zero in non-involutive meadows. *Journal of Applied Logic*, 13(1):1–12 (`https://doi.org/10.1016/j.jal.2014.10.001`). Preprint available at `https://arxiv.org/abs/1406.2092` [math.RA] (2014, 9 June).

[BP15] Bergstra, J.A. and Ponse, A. (2015). Division by zero in common meadows. In R. de Nicola and R. Hennicker (Eds.): *Software, Services, and Systems*, Lecture Notes in Computer Science, Vol. 8950, Springer, pp. 46–61. Revised version: `https://arxiv.org/abs/1406.6878v3` [math.RA] (2019, 14 August).

[BP16] Bergstra, J.A. and Ponse, A. (2016). Fracpairs and fractions over a reduced commutative ring. *Indagationes Mathematicae*, 27:727–748 (`https://doi.org/10.1016/j.indag.2016.01.007`). Preprint available at `https://arxiv.org/abs/1411.4410v2` [math.RA] (2016, 22 Jan).

---

[3] Apart from halving even numbers, which is easy in binary notation, but can otherwise be specified with a shift operation.

[4] For example, we were able to use AProVE's web interface to prove termination of the DDRS Nat_{dt}, but for the DDRS Int_{dt} this was only possible after reducing this DDRS to base 2.

[BP16a] Bergstra, J.A. and Ponse, A. (2016). Three datatype defining rewrite systems for datatypes of integers each extending a datatype of naturals. Available at `https://arxiv.org/abs/1406.3280v4` [cs.LO], 18 July 2016.

[BT95] Bergstra, J.A. and Tucker, J.V. (1995). Equational specifications, complete term rewriting systems, and computable and semicomputable algebras. *Journal of the ACM,* 42(6):1194–1230.

[BT87] Bergstra, J.A. and Tucker, J.V. (1987). Algebraic specifications of computable and semicomputable data types. *Theoretical Computer Science*, 50(2):137–181.

[BT07] Bergstra, J.A. and Tucker, J.V. (2007). The rational numbers as an abstract data type. *Journal of the ACM*, 54(2), Article 7.

[BW89] Bouma, L.G. and Walters, H.R. (1989). Implementing algebraic specifications. In J.A. Bergstra, J. Heering, and P. Klint (Eds.): *Algebraic Specification* (Chapter 5), Addison-Wesley, pp. 199–282.

[BWP84] Broy, M., Wirsing, M., and Pair, C. (1984). A systematic study of models of abstract data types. *Theoretical Computer Science*, 33(2):139–174.

[CMR97] Contejean, E., Marché, C., and Rabehasaina, L. (1997). Rewrite systems for natural, integral, and rational arithmetic. In H. Comon (Ed.): *Rewriting Techniques and Applications, 8th International Conference, RTA-97*, Lecture Notes in Computer Science, Vol. 1232, Springer, pp. 98–112.

[CJ98] Gaudel, M.-C. and James, P.R. (1998). Testing algebraic data types and processes: a unifying theory. *Formal Aspects of Computing,* 10(5-6):436–451.

[Gie+17] Giesl, J., Aschermann, C., Brockschmidt, M., Emmes, F., Frohn, F., Fuhs, C., Hensel, J., Otto, C., Plücker, C., Schneider-Kamp, P., Thomas Ströder, T., Swiderski, S., and Thiemann, R. (2017). Analyzing program termination and complexity automatically with AProVE. *Journal of Automated Reasoning* , 58(1):3–31 (`https://doi.org/10.1007/s10817-016-9388-y`).
Web interface for AProVE: `http://aprove.informatik.rwth-aachen.de/` (last accessed 5 January 2021).

[KV03] Klop, J.W. and Vrijer, R.C. de (2003). *First-order term rewriting systems.* Chapter 2 in [Ter03], pp. 24–59.

[KW16] Kluiving, B. and Woerkom, W.K. van (2016). Number representations and term rewriting. Honours project BSc Computer Science and BSc Artificial Intelligence, University of Amsterdam (January 31, 2016). Available at `https://arxiv.org/abs/1607.04500v1` [cs.LO], 15 Jul 2016.

[Knu97] Knuth, D.E. (1997). *The Art of Computer Programming, Volume 2 (3rd Edition): Seminumerical Algorithms.* Addison-Wesley.

[Ter03] Terese (2003). *Term Rewriting Systems.* Cambridge Tracts in Theoretical Computer Science, Vol. 55, Cambridge University Press.

[UK03] Urso, P. and Kounalis, E. (2003). "Term partition" for mathematical induction. In R. Nieuwenhuis (Ed.): *Rewriting Techniques and Applications, 14th International Conference, RTA-03*, Lecture Notes in Computer Science, Vol. 2706, Springer, pp. 352–366. Extended version available at `https://hal.inria.fr/inria-00072023/document` (2006).

[Wal90] Walters, H.R. (1990). Hybrid implementations of algebraic specifications. In H. Kirchner, W. Wechler (Eds.): *Algebraic and Logic Programming*, Lecture Notes in Computer Science, Vol. 463, Springer, pp. 40–54.

[Wal94] Walters, H.R. (1994). A complete term rewriting system for decimal integer arithmetic. Report CS-R9435, CWI. Available at `http://oai.cwi.nl/oai/asset/5140/5140D.pdf`.

[WZ95] Walters, H.R. and Zantema, H. (1995). Rewrite systems for integer arithmetic. In J. Hsiang (Ed.): *Rewriting Techniques and Applications, 6th International Conference, RTA-95*, Lecture Notes in Computer Science, Vol. 914, Springer, pp. 324–338. Preprint available at `http://oai.cwi.nl/oai/asset/4930/4930D.pdf`.

[Wir91] Wirsing, M. (1991). Algebraic Specification. In: *Handbook of Theoretical Computer Science*, Vol. B, MIT Press, pp. 675–788.

[Woe17] Woerkom, W.K. van (2017). Rascal Tooling for Datatype Defining Rewrite Systems. Honours extension of BSc. thesis Artificial Intelligence, University of Amsterdam, 52 pages (July 2017). Available at `https://scripties.uba.uva.nl/search?id=633785`.

[ZFM11] Zankl, H., Felgenhauer, B., and Middeldorp, A. (2011). CSI - A confluence tool. In N. Bjørner and V. Sofronie-Stokkermans (Eds.): CADE 2011, Lecture Notes in Computer Science, Vol. 6803, Springer, pp. 499–505.
Web interface for CSI: `http://cocoweb.uibk.ac.at/` (last accessed 4 January 2021).

[Zan03] Zantema H. (2003). *Termination.* Chapter 6 in [Ter03], pp. 181–259.

Appendix A. DDRSs with digit append constructors, ground-confluence

In this appendix we prove ground-confluence of the DDRSs Int_{bud} and Int_{dub}. In both proofs we adopt the approach used in the proof of Theorem 2.3.

A.1. Binary view: Int_{bud}, the DDRS for $\mathbb{Z}_{bud}$.

This DDRS is defined in Table 9. Define the set N of closed terms over $\Sigma_{\mathbb{Z}}$ as follows:

$$\begin{aligned} N &= \{0\} \cup N^+ \cup N^-, \\ N^+ &= \{1\} \cup \{t :_b 0, t :_b 1 \mid t \in N^+\}, \\ N^- &= \{-t \mid t \in N^+\}. \end{aligned}$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite rule applies).

In order to prove ground-confluence of this rewriting system, it suffices to show that for each closed term t over $\Sigma_{\mathbb{Z}}$, either $t \in N$ or t has a rewrite step, so that each normal form is in N. We prove this by structural induction on t.

The base cases are simple: if $t \in \{0,1\}$ then $t \in N$, and if $t = i'$ for some $i \in \{1,2,3,4,5,6,7,8\}$, then t has a rewrite step by equation [**bd1.**i]. For the induction step we distinguish eight cases:

(1) Case $t = S(r)$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b2],
- if $r = 1$, then t has a rewrite step by equation [b3],
- if $r = r' :_b 0$, then t has a rewrite step by equation [b4],
- if $r = r' :_b 1$, then t has a rewrite step by equation [b5],
- if $r = -1$, then t has a rewrite step by equation [b20],
- if $r = -(r' :_b 0)$, then t has a rewrite step by equation [b21],
- if $r = -(r' :_b 1)$, then t has a rewrite step by equation [b22].

(2) Case $t = P(r)$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b15],
- if $r = 1$, then t has a rewrite step by equation [b16],
- if $r = r' :_b 0$, then t has a rewrite step by equation [b17],
- if $r = r' :_b 1$, then t has a rewrite step by equation [b18],
- if $r \in N^-$, then t has a rewrite step by equation [b19].

(3) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b13],
- if $r = 1$, then $t \in N$,
- if $r = r' :_b i$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [b14].

(4) Case $t = r :_b 0$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b1.0],
- if $r \in N^+$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [b23].

(5) Case $t = r :_b 1$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b1.1],
- if $r \in N^+$, then $t \in N$,

- if $r \in N^-$, then t has a rewrite step by equation [b24].

(6) Case $t = u + r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b6],
- if $r = 1$, then t has a rewrite step by equation [b7],
- if $r = r' :_b 0$, then t has a rewrite step according to equation [b8],
- if $r = r' :_b 1$, then t has a rewrite step according to equation [b9],
- if $r \in N^-$, then t has a rewrite step according to equation [b25].

(7) Case $t = u \cdot r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [b10],
- if $r = 1$, then t has a rewrite step by equation [b11],
- if $r = r' :_b i$, then t has a rewrite step according to [b12.i],
- if $r \in N^-$, then t has a rewrite step by equation [b26].

(8) Case $t = r :_d i$ for $i \in D$. Now t has a rewrite step by equation [**bd2.**i].

This concludes our proof.

A.2. **Decimal view:** Int_{dub}**, the DDRS for** $\mathbb{Z}_{dub}$**.**
This DDRS is defined in Table 11. Recall that we write D for the set of all digits.

Define the set N of closed terms over $\Sigma_\mathbb{Z}$ as follows:

$$\begin{aligned} N &= \{0\} \cup N^+ \cup N^-, \\ N^+ &= D \setminus \{0\} \cup \{t :_d i \mid t \in N^+, i \in D\}, \\ N^- &= \{-t \mid t \in N^+\}. \end{aligned}$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite rule applies). In order to prove ground-confluence of this rewriting system, it suffices to show that for each closed term t over $\Sigma_\mathbb{Z}$, either $t \in N$ or t has a rewrite step, so that each normal form is in N. We prove this by structural induction on t.

The base cases are trivial: if $t \in D$, then $t \in N$. For the induction step we distinguish eight cases:

(1) Case $t = S(r)$. Assume that $r \in N$ and apply case distinction on r:
- if $r = i$ for $i \in \{0, 1, \ldots, 8\}$, then t has a rewrite step by equation [d2.i],
- if $r = 9$, then t has a rewrite step by equation [d3],
- if $r = r' :_d i$ for $i \in \{0, 1, \ldots, 8\}$, then t has a rewrite step by equation [d4.i],
- if $r = r' :_d 9$, then t has a rewrite step by equation [d5],
- if $r = -i'$ for $i \in \{0, 1, \ldots, 8\}$, then t has a rewrite step by equation [d20.i],
- if $r = -(r' :_d 0)$, then t has a rewrite step by equation [d21],
- if $r = -(r' :_d i')$ for $i \in \{0, 1, \ldots, 8\}$, then t has a rewrite step by equation [d22.i].

(2) Case $t = P(r)$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [d15],
- if $r = i'$ for $i \in \{0, 1, \ldots, 8\}$, then t has a rewrite step by equation [d16.i],
- if $r = r' :_d 0$, then t has a rewrite step by equation [d17],
- if $r = r' :_d i'$ for $i \in \{0, 1, \ldots, 8\}$, then t has a rewrite step by equation [d18.i],
- if $r \in N^-$, then t has a rewrite step by equation [d19].

(3) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [d13],
- if $r \in N^+$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [d14].

(4) Case $t = r :_d 0$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [d1.0],
- if $r \in N^+$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [d23].

(5) Case $t = r :_d i$ for $i \in \{1, 2, \dots, 9\}$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [d1.i],
- if $r \in N^+$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [d24.i].

(6) Case $t = u + r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [d6],
- if $r = i$ for $i \in \{1, 2, \dots, 9\}$, then t has a rewrite step by equation [d7.i],
- if $r = r' :_d 0$, then t has a rewrite step by equation [d8],
- if $r = r' :_d i$ for $i \in \{1, 2, \dots, 9\}$, then t has a rewrite step by equation [d9.i],
- if $r \in N^-$, then t has a rewrite step by equation [d25].

(7) Case $t = u \cdot r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [d10],
- if $r = i'$ for $i \in \{0, 1, \dots, 8\}$, then t has a rewrite step by [d11.i],
- if $r = r' :_d i$ for $i \in D$, then t has a rewrite step by equation [d12.i],
- if $r \in N^-$, then t has a rewrite step by equation [d26].

(8) Case $t = r :_b i$ for $i \in \{0, 1\}$. Now t has a rewrite step by equation [**db1.**i].

This concludes our proof.

Appendix B. DDRSs with digit tree constructors, ground-confluence

In this appendix we prove ground-confluence of the DDRSs Int_{ut}, Int_{bt}, and Int_{dt}, respectively. In all proofs we adopt the approach used in the proof of Theorem 2.3.

B.1. Unary view: the DDRS Int_{ut} for $\mathbb{Z}_{ut}$.

This DDRS is defined in Table 12. Define the signature $\Sigma_{ut} = \{0, -(\_), \_ \mathbin{\hat{}_u} \_, +, \cdot\}$, and the set N of closed terms over Σ_{ut} as follows:

$$\begin{aligned} N &= \{0\} \cup N^+ \cup N^-, \\ N^+ &= \{0 \mathbin{\hat{}_u} 0\} \cup \{t \mathbin{\hat{}_u} 0 \mid t \in N^+\}, \\ N^- &= \{-t \mid t \in N^+\}. \end{aligned}$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite rule applies). In order to prove ground-confluence, it suffices to show that for each closed term t over Σ_{ut}, either $t \in N$ or t has a rewrite step, so that each normal form is in N. We prove this by structural induction on t.

The base case is trivial: if $t = 0$, then $t \in N$. For the induction step we distinguish four cases:

(1) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [ut6],
- if $r = r' \mathbin{\hat{}_u} 0$, then $t \in N$,
- if $r = -(r' \mathbin{\hat{}_u} 0)$, then t has a rewrite step by equation [ut7].

(2) Case $t = v \mathbin{\hat{}_u} r$. Assume that $v, r \in N$ and apply case distinction on r:
- if $r = 0$, then apply case distinction on v:

  – if $v = 0$, then $t \in N$,
  – if $v = v' \mathbin{\hat{}_u} 0$, then $t \in N$,
  – if $v = -(v' \mathbin{\hat{}_u} 0)$, then t has a rewrite step by equation [ut10].
- if $r = r' \mathbin{\hat{}_u} 0$, then apply case distinction on v:
  – if $v = 0$, then t has a rewrite step by equation [ut1],
  – if $v = v' \mathbin{\hat{}_u} 0$, then t has a rewrite step by equation [ut1],
  – if $v = -(v' \mathbin{\hat{}_u} 0)$, then t has a rewrite step by equation [ut11].
- if $r = -(r' \mathbin{\hat{}_u} 0)$, then apply case distinction on v:
  – if $v = 0$, then t has a rewrite step by equation [ut8],
  – if $v = v' \mathbin{\hat{}_u} 0$, then t has a rewrite step by equation [ut9],
  – if $v = -(v' \mathbin{\hat{}_u} 0)$, then t has a rewrite step by equation [ut12].

(3) Case $t = v + r$. Assume that $v, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [ut2],
- if $r = r' \mathbin{\hat{}_u} 0$, then t has a rewrite step by equation [ut3],
- if $r = -(r' \mathbin{\hat{}_u} 0)$, then t has a rewrite step by equation [ut13].

(4) Case $t = v \cdot r$. Assume that $v, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [ut4],
- if $r = r' \mathbin{\hat{}_u} 0$, then t has a rewrite step by equation [ut5],
- if $r = -(r' \mathbin{\hat{}_u} 0)$, then t has a rewrite step by equation [ut14].

This concludes our proof.

B.2. Binary view: Int_{bt}, the DDRS for $\mathbb{Z}_{bt}$.

This DDRS is defined in Table 13. Define the signature $\Sigma_{bt} = \{0, 1, -(\_), \_ \mathbin{\hat{}_b} \_, +, \cdot\}$, and the set N of closed terms over Σ_{bt} as follows:

$$
\begin{aligned}
N &= \{0\} \cup N^+ \cup N^-,\\
N^+ &= \{1\} \cup \{t \mathbin{\hat{}_b} 0, t \mathbin{\hat{}_b} 1 \mid t \in N^+\},\\
N^- &= \{-t \mid t \in N^+\}.
\end{aligned}
$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite rule applies), and that two distinct elements in N have distinct values in $\mathbb{Z}$. In order to prove ground-confluence of this rewriting system, it suffices to show that for each closed term t over Σ_{bt}, either $t \in N$ or t has a rewrite step, so that each normal form is in N. We prove this by structural induction on t.

The base cases are simple: if $t \in \{0, 1\}$, then $t \in N$. For the induction step we distinguish four cases:

(1) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [bt13],
- if $r \in N^+$, then $t \in N$,
- if $r \in N^-$, then t has a rewrite step by equation [bt14].

(2) Case $t = r \mathbin{\hat{}_b} u$. Assume that $r, u \in N$ and apply case distinction on u:
- if $u \in \{0, 1\}$, then apply case distinction on r:
  – if $r = 0$, then t has a rewrite step by equation [bt1],
  – if $r \in N^+$, then $t \in N$,
  – if $r \in N^-$, then t has a rewrite step by equation [bt19].
- if $u = u' \mathbin{\hat{}_b} j$, then t has a rewrite step by equation [bt2],
- if $u = -1$, then apply case distinction on r:

– if $r = 0$, then t has a rewrite step by equation [bt1],
– if $r = 1$, then t has a rewrite step by [bt15],
– if $r = r' \,\hat{_b}\, 0$, then t has a rewrite step by equation [bt16],
– if $r = r' \,\hat{_b}\, 1$, then t has a rewrite step by equation [bt17],
– if $r \in N^-$, then t has a rewrite step by equation [bt19].
- if $u = -(u' \,\hat{_b}\, j)$, then t has a rewrite step by equation [bt18].

(3) Case $t = u + r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [bt4],
- if $r = 1$, then apply case distinction on u:
  – if $u = 0$, then t has a rewrite step by equation [bt3],
  – if $u = 1$, then t has a rewrite step by equation [bt5]
  – if $u = u' \,\hat{_b}\, j$, then t has a rewrite step by equation [bt7],
  – if $u = -1$, then t has a rewrite step by equation [bt21],
  – if $u = -(u' \,\hat{_b}\, j)$, then t has a rewrite step by equation [bt24],
- if $r = r' \,\hat{_b}\, i$, then t has a rewrite step by [bt6],
- if $r = -1$, then apply case distinction on u:
  – if $u = 0$, then t has a rewrite step by equation [bt3],
  – if $u = 1$, then t has a rewrite step by equation [bt20]
  – if $u = u' \,\hat{_b}\, j$ then t has a rewrite step by equation [bt7],
  – if $u = -1$, then t has a rewrite step by equation [bt22],
  – if $u = -(u' \,\hat{_b}\, j)$, then t has a rewrite step by equation [bt24],
- if $r = -(r' \,\hat{_b}\, i)$, then t has a rewrite step by equation [bt23].

(4) Case $t = u \cdot r$. Assume that $u, r \in N$ and apply case distinction on r:
- if $r = 0$, then t has a rewrite step by equation [bt8],
- if $r = 1$, then apply case distinction on u:
  – if $u = 0$, then t has a rewrite step by equation[bt9],
  – if $u = 1$, then t has a rewrite step by equation[bt10],
  – if $u = u' \,\hat{_b}\, j$, then t has a rewrite step by equation [bt12],
  – if $u \in N^-$, then t has a rewrite step by equation [bt26],
- if $r = r' \,\hat{_b}\, i$, then t has a rewrite step by equation [bt11],
- if $r \in N^-$, then t has a rewrite step by equation [bt25].

This concludes our proof.

B.3. **Decimal view: Int_{dt}, the DDRS for $\mathbb{Z}_{dt}$.**
This DDRS is defined in Table 14. Recall that $D = \{0, 1, 2, ..., 9\}$ and define the signature $\Sigma_{dt} = \{+, \cdot, \_ \,\hat{_d}\, \_, -(\_), i \mid i \in D\}$, and the set N of closed terms over Σ_{dt} as follows:

$$N = \{0\} \cup N^+ \cup N^-,$$
$$N^+ = D \setminus \{0\} \cup \{t \,\hat{_d}\, i \mid t \in N^+, i \in D\},$$
$$N^- = \{-t \mid t \in N^+\}.$$

It immediately follows that if $t \in N$, then t is a normal form (no rewrite rule applies). In order to prove ground-confluence of this rewriting system, it suffices to show that for each closed term t over Σ_{dt}, either $t \in N$ or t has a rewrite step, so that each normal form is in N. We prove this by structural induction on t.

The base cases are simple: if $t \in D$, then $t \in N$. For the induction step we have to distinguish five cases:

(1) Case $t = -r$. Assume that $r \in N$ and apply case distinction on r:
   - if $r = 0$ then t has a rewrite step by equation [dt12],
   - if $r \in N^+$, then $t \in N$,
   - if $r \in N^-$, then t has a rewrite step by equation [dt13].

(2) Case $t = S(r)$. Assume that $r \in N$ and apply case distinction on r:
   - if $r = i$ for $i \in D \setminus \{9\}$, then t has a rewrite step by equation [dt3.i],
   - if $r = 9$, then t has a rewrite step by equation [dt4],
   - if $r = r' \,\hat{}_d\, i$ for $i \in D \setminus \{9\}$, then t has a rewrite step by equation [dt5.i],
   - if $r = r' \,\hat{}_d\, 9$, then t has a rewrite step by equation [dt6],
   - if $r = -i'$ for $i \in D \setminus \{9\}$, then t has a rewrite step by equation [dt14.i],
   - if $r = -(r \,\hat{}_d\, 0)$, then t has a rewrite step by equation [dt15],
   - if $r = -(r \,\hat{}_d\, i')$ for $i \in D \setminus \{9\}$, then t has a rewrite step by equation [dt16.i].

(3) Case $t = r \,\hat{}_d\, u$. Assume that $r, u \in N$ and apply case distinction on u:
   - if $u \in D$, then apply case distinction on r:
     - if $r = 0$, then t has a rewrite step by equation [dt1],
     - if $r \in N^+$, then $t \in N$,
     - if $r \in N^-$, then t has a rewrite step by equation [dt23].
   - if $u = u' \,\hat{}_d\, i$ for $i \in D$, then t has a rewrite step by equation [dt2.i],
   - if $u = -i$ for $i \in D \setminus \{0\}$, then apply case distinction on r:
     - if $r = 0$, then t has a rewrite step by equation [dt1],
     - if $r = 1$ and $u = -1$, then t has a rewrite step by equation [dt17],
     - if $r = 1$ and $u = -i'$ for $i \in D \setminus \{9\}$, then t has a rewrite step by equation [dt18.i],
     - if $r = i'$ and $u = -1$ for $i \in D \setminus \{9\}$, then t has a rewrite step by equation [dt19.i],
     - if $r = i'$ and $u = -j'$ for $i, j \in D \setminus \{9\}$, then t has a rewrite step by equation [dt20.$i.j$],
     - if $r = r' \,\hat{}_d\, 0$ and $u = -i$ for $i \in D \setminus \{0\}$, then t has a rewrite step by equation [dt21.i],
     - if $r = r' \,\hat{}_d\, i'$ and $u = -j'$ for $i, j \in D \setminus \{9\}$, then t has a rewrite step by equation [dt22.$i.j$],
     - if $r \in N^-$, then t has a rewrite step by equation [dt23].
   - if $u = -(u' \,\hat{}_d\, i)$, then t has a rewrite step by equation [dt24.i].

(4) Case $t = r + u$. Assume that $r, u \in N$ and apply case distinction on u:
   - if $u = i$ for $i \in D$, then t has a rewrite step by equation [dt7.i],
   - if $u = u' \,\hat{}_d\, i$ for $i \in D$, then t has a rewrite step by equation [dt8.i],
   - if $u \in N^-$, say $u = -u'$, then t has a rewrite step by equation [dt25].

(5) Case $t = r \cdot u$. Assume that $r, u \in N$ and apply case distinction on u:
   - if $u = 0$, then t has a rewrite step by equation [dt9],
   - if $u = i$ for $i \in D \setminus \{0\}$, then t has a rewrite step by equation [dt10.i],
   - if $u = u' \,\hat{}_d\, i$ for $i \in D$, then t has a rewrite step by equation [dt11.i],
   - if $u \in N^-$, then t has a rewrite step by equation [dt26].

This concludes our proof.