The Commutativity Problem of the Map-Reduce Framework: A Transducer-based Approach

Abstract. Map-Reduce is a popular programming model for data parallel computation. In Map-Reduce, the *reducer* produces an output from a list of inputs. Due the scheduling policy and the settings of machines, the input may arrive the reducers with different orders. The *communtative problem* of reducers asks if the output of a reducer independent of the order of its inputs. The problem is in general undecidable due to Rice's theorem and thus is seemingly uninteresting. However, the Map-Reduce model is usually used for data analytics and thus requires very simple data and control flow. By exploiting the simplicity of the required program flows, we propose a commutative decidable language for reducers. We show that the language is expressive enough for common data analytics operations.

1 Introduction

Map-Reduce is a very popular framework for data parallel computation. It has been adopt in various widely used cloud computing frameworks such as Hadoop [4] and Spark [5]. In a typical Map-Reduce program, a *mapper* takes a key-value pair as input and output a list of key-value pairs. The load balance mechanism of the Map-Reduce framework will process the key-value pairs and sends the pairs with the same key to the same *reducer*, in the form of the key and a list of values. The reducer reads the input list and output a key-value pair.

[1]

2 Preliminaries

Let $\mathbb{Z}$ denote the set of integers. Let X denote a finite set of variables ranging over $\mathbb{Z}$ and x, y are variables in X. We assume that X contains a special variable cur, which is used to denote the current data value. Then a guard over X is a formula defined by the rules, $g ::= cur\ o\ c \mid cur\ o\ x \mid g \wedge g$, where $o \in \{=, \neq, <, >, \leq, \geq\}$. Let Y be another set of variables over $\mathbb{Z}$. An arithmetic expression over Y is defined by the rules, $e ::= y \mid c \mid e + e \mid e - e \mid e * e \mid e/e$, where $y \in Y$ and c is a constant over $\mathbb{Z}$. For an expression e, let $vars(e)$ denote the set of variables occurring in e. Let $\mathcal{E}_Y$ denote the set of arithmetic expressions over Y. An assignment η for $X \cup Y$ is a partial function from $(X \setminus \{cur\}) \cup Y$ to $\mathcal{E}_{X \cup Y}$ (the set of arithmetic expressions over $X \cup Y$) such that for each $x \in dom(\eta) \cap X$, $\eta(x) = cur$. Note that the assignments to the special variable cur are disallowed.

A data word is a sequences of data values, that is, sequences $d_1 \dots d_n$, where $d_i \in \mathbb{Z}$ for each i.

A streaming numerical transducer (SNT) $\mathcal{S}$ is a tuple $(Q, X, Y, \delta, q_0, O)$ such that

- Q is a finite set of states,
- X is a finite set of control variables which is used to store some data values that have been met,
- Y is a finite set of data variables, which is used to aggregate some information for the output,
- δ comprises the tuples (q, g, η, q'), where $q, q' \in Q$, g is a guard over X (note that the variables from Y do not occur in g), η is an assignment for $X \cup Y$,
- $q_0 \in Q$ is the initial state,
- O is the output function, which is a partial function from Q to $\mathcal{E}_{(X \setminus \{cur\}) \cup Y}$.

In this paper, we restrict our attention to SNTs $\mathcal{S}$ satisfying the following constraints.

1 (deterministic). For every pair of distinct tuples $(q, g_1, \eta_1, q'_1), (q, g_2, \eta_2, q'_2) \in \delta$, it holds that g_1 and g_2 are mutually exclusive, that is, $g_1 \wedge g_2$ is unsatisfiable.

2 (copyless). For each $(q, g, \eta, q') \in \delta$ and each $y \in Y$, y appears at most once in the collection of expressions $\{\eta(y') \mid y' \in Y\}$.

3 (independently evolving). For each $(q, g, \eta, q') \in \delta$ and each $y \in Y$, $\eta(y)$ contains no variables $y' \in Y$ such that $y' \neq y$.

4 (generalized flat). Each SCC (strongly connected component) of the transition graph of $\mathcal{S}$ is either a single state or a collection of cycles that they share a unique state.

Note that the "copyless" constraint forbids the use of the expressions of the form $y + y$ in the transitions.

The semantics of a SNT $\mathcal{S}$ is given by a transduction as follows: A configuration of $\mathcal{S}$ is a pair (q, β), where $q \in Q$ and β is a valuation of $X \cup Y$, that is, a partial function from $X \cup Y$ to $\mathbb{Z}$. When reading a data word $w = d_1 \dots d_n$, $\mathcal{S}$ runs over w from left to right. Let (q, β) be the configuration of $\mathcal{S}$ reached after running over w, then the output of $\mathcal{S}$ is $\beta(O(q))$, if $O(q)$ is defined and $vars(O(q)) \subseteq dom(\beta)$, and the output is undefined otherwise.

A SNT $\mathcal{S}$ is said to be a SNT$_{\pm}$ if all the expressions occurring in $\mathcal{S}$ use only $+, -$, but not $*, /$.

Example 1 (Max, average). The max transducer over sequences of integers is given by the transitions $(q_0, 1, x < p_1, x := p_1, q_0)$, where $x := p_1$ assigns p_1 to x, and $(q_0, 1, x \geq p_1, \emptyset, q_0)$, and the output function $O(q_0) = x$. The average transducer over sequences of integers is given by the transition $(q_0, 1, true, (sum := sum + p_1, len := len + 1), q_0)$, and $O(q_0) = sum/len$.

Example 2 (Example inspired by Pagerank). The following transducer sum all the data values, except the last position, then it outputs a concatenation of the sum and the last tuple: $(q_0, 1, true, sum := sum + p_1, q_0)$, $(q_0, k, true, (x_i := p_i)_{1 \leq i \leq k}, q_1)$, $O(q_1) = (sum, x_1, \dots, x_k)$.

We focus on the following decision problems of SNT.

(Commutativity). Given a SNT $\mathcal{S}$, decide whether $\mathcal{S}$ is commutative, that is, whether for each data word w and each permutation w' of w, the output of $\mathcal{S}$ over w is equal to that of $\mathcal{S}$ over w'.

(Equivalence). Given two SNTs $\mathcal{S}_1, \mathcal{S}_2$, decide whether $\mathcal{S}_1$ and $\mathcal{S}_2$ are equivalent, that is, whether over each data word w, the output of $\mathcal{S}_1$ on w is defined iff that of $\mathcal{S}_2$ on w is defined, moreover, if the outputs of $\mathcal{S}_1$ and $\mathcal{S}_2$ are defined, then the output of $\mathcal{S}_1$ is equal to that of $\mathcal{S}_2$.

(Non-zero output reachability). Given a SNT $\mathcal{S}$, decide whether $\mathcal{S}$ has a non-zero output, that is, whether there is an input w such that the output of $\mathcal{S}$ on w is non-zero.

Consider the permutation τ_2 and τ_n in [1]. We can define two streaming data string transducers $\mathcal{S}$ and $\mathcal{S}'$ (note that $\mathcal{S}'$ is independent from n) for τ_2 and τ_n. Then the commutativity of a given SNT $\mathcal{T}$ is reduced to the equivalence of $\mathcal{T}$ and $\mathcal{S} \circ \mathcal{T}$ as well as the equivalence of $\mathcal{T}$ and $\mathcal{S}' \circ \mathcal{T}$. Note that an equivalent SNT can be defined for $\mathcal{S} \circ \mathcal{T}$ and $\mathcal{S}' \circ \mathcal{T}$ respectively.

Proposition 1. *The commutativity of SNTs is reduced to the equivalence of SNTs in linear time.*

Proposition 2. *From SNT $\mathcal{S}_1$ and $\mathcal{S}_2$, a SNT $\mathcal{S}_3$ can be constructed in polynomial time such that $\mathcal{S}_1$ is not equivalent to $\mathcal{S}_2$ iff there is a data word w such that the output of $\mathcal{S}_3$ over w is nonzero.*

Therefore, the equivalence problem of SNTs is reduced to the non-zero output reachability problem of SNTs.

3 Closure properties

Boolean operations.

Union, intersection, complement

composition: It seems that SDSITare not closed under composition, similar to that of streaming transducers.

4 Decision problems

In this section, we consider the non-zero output problem of $\text{SNT}_{\pm}$. Recall that we assume that the SNTs are deterministic, copyless, independently evolving and generalized flat.

4.1 Normalization

Suppose $\mathcal{S} = (Q, X, Y, \delta, q_0, O)$ is a SNT. Let c_{min} and c_{max} denote the minimum resp. maximum constant occurring in the transitions of $\mathcal{S}$.

A SNT $\mathcal{S} = (Q, X, Y, \delta, q_0, O)$ is said to be *normalized* if the following two constraints are satisfied.

- For each transition $(q, g, \eta, q') \in \delta$, if $\eta(x) = cur$ for some $x \in X$, then the guard g implies $\bigwedge\limits_{x \in X \setminus \{cur\}} cur \neq x$. Intuitively, when the current data value is stored into some register, it is required that the data value is distinct from all the data values that have already been stored in the register.
- For each transition (q, g, η, q') in $\mathcal{S}$, g includes one of the following formulae as a conjunct: $cur < c_{min}$, or $cur = c$ for $c_{min} \leq c \leq c_{max}$, or $cur > c_{max}$.

Proposition 3. *From each SNT, an equivalent normalized SNT can be constructed (with a possibly exponential blow-up).*

From now on, we assume that all SNTs are normalized.

4.2 Analysis of a single cycle

Let $\mathcal{S} = (Q, X, Y, \delta, q_0, O)$ be a (normalized) SNT such that $X = \{cur, x_1, \dots, x_k\}$ and $Y = \{y_1, \dots, y_l\}$. Moreover, in this and next subsection, we assume that the guards in $\mathcal{S}$ contain no comparisons with constants. Later on, we will consider the more general situation.

Suppose that $C = q_0 \xrightarrow{(g_1, \eta_1)} q_1 \dots q_{n-1} \xrightarrow{(g_n, \eta_n)} q_n$ is a path in $\mathcal{S}$ such that $q_n = q_0$.

Suppose the initial values of the k control variables are $d_1, \dots, d_k$ and the n data values introduced when traversing the cycle once are $d_{k+1}, \dots, d_{k+n}$ (These data values may repeat). Then the guards and the assignments in the cycle induce an equivalence relation $\sim$ on $\{1, \dots, k+n\}$ so that $i \sim j$ iff it can be inferred from the guards and assignments that $d_i = d_j$. Since $\mathcal{S}$ is normalized, we know that for each pair of indices $i, j : 1 \leq i < j \leq k+n$ such that $i \sim j$, it holds that $j \geq k+1$. Let $I_1, \dots, I_{k+r}$ be an enumeration of the equivalence classes of $\sim$ on $\{1, \dots, k+n\}$ such that $\min(I_1) < \dots < \min(I_{k+r})$. Then for each $j : 1 \leq j \leq k$, $\min(I_j) = j$.

In the following, for $\ell = 1, 2, \dots$, we will define the assignment function χ_ℓ with the domain $X \cup Y$ to describe the values of the control and data variables after traversing the cycle for i times.

Suppose the initial values of the k control variables are $d^0_1, \dots, d^0_k$. Moreover, suppose that the r data values $d^1_1, \dots, d^1_r$ are introduced when traversing the cycle for the first time, with one data value for each of $I_1, \dots, I_r$. In addition, suppose that the initial values of $y_1, \dots, y_l$ are $o_1, \dots, o_l$.

The assignment function χ_1 is of the following form,

- there is an injective mapping $\pi : \{1, \dots, k\} \to \{1, \dots, k+r\}$ such that for each $x_j \in X$, if $\pi(j) \leq k$, then $\pi(j) = j$ and $\chi_1(x_j) = d^0_j$, otherwise, $\chi_1(x_j) = d^1_{\pi(j)-k}$,

- for each $y_j \in Y$, $\chi_1(y_j) = \alpha_{j,0} + \alpha_{j,1} o_j + \beta_{j,1} d^0_1 + \cdots + \beta_{j,k} d^0_k + \gamma_{j,1} d^1_1 + \cdots + \gamma_{j,r} d^1_r$ for some constants $\alpha_{j,0}, \alpha_{j,1}, \beta_{j,1}, \ldots, \beta_{j,k}, \gamma_{j,1}, \ldots, \gamma_{j,r}$ such that $\alpha_{j,1} \in \{0, +1, -1\}$ (as a result of the "copyless" and "independently evolving" constraint).

Let J_1 be set of indices j such that $\pi(j) = j$ and $J_2 = \{1, \ldots, k\} \setminus J_1$. Intuitively, the values of the variables from J_1 are unchanged after traversing the cycle once, and the values of the variables from J_2 are changed. Then (J_1, J_2) forms a partition of $\{1, \ldots, k\}$. Moreover, let $J_3 = \{\pi(j) - k \mid j \in J_2\}$ and $J_4 = \{1, \ldots, r\} \setminus J_3$. Then (J_3, J_4) forms a partition of $\{1, \ldots, r\}$.

Let $d^2_1, \ldots, d^2_r$ be the data values introduced when traversing the cycle for the second time. Then χ_2 is defined as follows,

- for each $j \in J_1$, $\chi_2(x_j) = \chi_1(x_j) = d^0_j$, and for each $j \in J_2$, $\chi_2(x_j) = d^2_{\pi(j)-k}$,
- for each $y_j \in Y$, $\chi_2(y_j) = \alpha_{j,0} + \alpha_{j,1}\chi_1(y_j) + \beta_{j,1}\chi_1(x_1) + \cdots + \beta_{j,k}\chi_1(x_k) + \gamma_{j,1} d^2_1 + \cdots + \gamma_{j,r} d^2_r$.

By expanding the expressions $\chi_1(y_j), \chi_1(x_1), \ldots, \chi_1(x_k)$, we get the following expression,

$$\begin{aligned}
\chi_2(y_j) = {} & (\alpha_{j,0} + \alpha_{j,1}\alpha_{j,0}) + \alpha^2_{j,1} o_j + (\alpha_{j,1}\beta_{j,1}) d^0_1 + \cdots + (\alpha_{j,1}\beta_{j,k}) d^0_k + \\
& (\alpha_{j,1}\gamma_{j,1}) d^1_1 + \ldots (\alpha_{j,1}\gamma_{j,r}) d^1_r + \sum_{j' \in J_1} \beta_{j,j'} d^0_{j'} + \sum_{j' \in J_2} \beta_{j,j'} d^1_{\pi(j')-k} + \\
& \gamma_{j,1} d^2_1 + \cdots + \gamma_{j,r} d^2_r \\
= {} & (\alpha_{j,0} + \alpha_{j,1}\alpha_{j,0}) + \alpha^2_{j,1} o_j + \sum_{j' \in J_1} (\beta_{j,j'} + \alpha_{j,1}\beta_{j,j'}) d^0_{j'} + \sum_{j' \in J_2} (\alpha_{j,1}\beta_{j,j'}) d^0_{j'} + \\
& \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} + \alpha_{j,1}\gamma_{j,j'}) d^1_{j'} + \sum_{j' \in J_4} (\alpha_{j,1}\gamma_{j,j'}) d^1_{j'} + \\
& \gamma_{j,1} d^2_1 + \cdots + \gamma_{j,r} d^2_r .
\end{aligned}$$

Let $d^3_1, \ldots, d^3_r$ be the data values introduced when traversing the cycle for the third time. Then χ_3 is defined as follows,

- for each $j \in J_1$, $\chi_3(x_j) = \chi_2(x_j) = d^0_j$, and for each $j \in J_2$, $\chi_3(x_j) = d^3_{\pi(j)-k}$,
- for each $y_j \in Y$, $\chi_3(y_j) = \alpha_{j,0} + \alpha_{j,1}\chi_2(y_j) + \beta_{j,1}\chi_2(x_1) + \cdots + \beta_{j,k}\chi_2(x_k) + \gamma_{j,1} d^3_1 + \cdots + \gamma_{j,r} d^3_r$.

By expanding the expressions $\chi_2(y_j), \chi_2(x_1), \ldots, \chi_2(x_k)$, we get the following expression,

$$\begin{aligned}
\chi_3(y_j) = {} & (\alpha_{j,0} + \alpha_{j,1}\alpha_{j,0} + \alpha^2_{j,1}\alpha_{j,0}) + \alpha^3_{j,1} o_j + \sum_{j' \in J_1} (\beta_{j,j'} + \alpha_{j,1}\beta_{j,j'} + \alpha^2_{j,1}\beta_{j,j'}) d^0_{j'} + \\
& \sum_{j' \in J_2} (\alpha^2_{j,1}\beta_{j,j'}) d^0_{j'} + \sum_{j' \in J_3} (\alpha_{j,1}\beta_{j,\pi^{-1}(j'+k)} + \alpha^2_{j,1}\gamma_{j,j'}) d^1_{j'} + \sum_{j' \in J_4} (\alpha^2_{j,1}\gamma_{j,j'}) d^1_{j'} + \\
& \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} + \alpha_{j,1}\gamma_{j,j'}) d^2_{j'} + \sum_{j' \in J_4} (\alpha_{j,1}\gamma_{j,j'}) d^2_{j'} + \\
& \gamma_{j,1} d^3_1 + \cdots + \gamma_{j,r} d^3_r .
\end{aligned}$$

In general, for $\ell \geq 2$, we have the following expression,

$$\begin{aligned}\chi_\ell(y_j) = {} & (\alpha_{j,0} + \alpha_{j,1}\alpha_{j,0} + \cdots + \alpha_{j,1}^{\ell-1}\alpha_{j,0}) + \alpha_{j,1}^{\ell} o_j + \\ & \sum_{j' \in J_1} (\beta_{j,j'} + \alpha_{j,1}\beta_{j,j'} + \cdots + \alpha_{j,1}^{\ell-1}\beta_{j,j'}) d_{j'}^0 + \\ & \sum_{j' \in J_2} (\alpha_{j,1}^{\ell-1}\beta_{j,j'}) d_{j'}^0 + \sum_{j' \in J_3} (\alpha_{j,1}^{\ell-2}\beta_{j,\pi^{-1}(j'+k)} + \alpha_{j,1}^{\ell-1}\gamma_{j,j'}) d_{j'}^1 + \\ & \sum_{j' \in J_4} (\alpha_{j,1}^{\ell-1}\gamma_{j,j'}) d_{j'}^1 + \cdots + \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} + \alpha_{j,1}\gamma_{j,j'}) d_{j'}^{\ell-1} + \\ & \sum_{j' \in J_4} (\alpha_{j,1}\gamma_{j,j'}) d_{j'}^{\ell-1} + \gamma_{j,1} d_1^\ell + \cdots + \gamma_{j,r} d_r^\ell.\end{aligned}$$

Since $\alpha_{j,1} \in \{0, +1, -1\}$, for $\ell \geq 4$, we observe the following fact.

- If $\alpha_{j,1} = 0$, then

$$\chi_\ell(y_j) = \alpha_{j,0} + \sum_{j' \in J_1} \beta_{j,j'} d_{j'}^0 + \sum_{j' \in J_3} \beta_{j,\pi^{-1}(j'+k)} d_{j'}^{\ell-1} + \gamma_{j,1} d_1^\ell + \cdots + \gamma_{j,r} d_r^\ell.$$

- If $\alpha_{j,1} = +1$, then

$$\begin{aligned}\chi_\ell(y_j) = {} & (\alpha_{j,0}\ell) + o_j + \sum_{j' \in J_1} (\beta_{j,j'}\ell) d_{j'}^0 + \sum_{j' \in J_2} \beta_{j,j'} d_{j'}^0 + \\ & \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} + \gamma_{j,j'}) d_{j'}^1 + \sum_{j' \in J_4} \gamma_{j,j'} d_{j'}^1 + \ldots \\ & \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} + \gamma_{j,j'}) d_{j'}^{\ell-1} + \sum_{j' \in J_4} \gamma_{j,j'} d_{j'}^{\ell-1} + \\ & \gamma_{j,1} d_1^\ell + \cdots + \gamma_{j,r} d_r^\ell.\end{aligned}$$

- If $\alpha_{j,1} = -1$ and ℓ is even, then

$$\begin{aligned}\chi_\ell(y_j) = {} & o_j + \sum_{j' \in J_2} (-\beta_{j,j'}) d_{j'}^0 + \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} - \gamma_{j,j'}) d_{j'}^1 + \sum_{j' \in J_4} (-\gamma_{j,j'}) d_{j'}^1 + \\ & \sum_{j' \in J_3} (-\beta_{j,\pi^{-1}(j'+k)} + \gamma_{j,j'}) d_{j'}^2 + \sum_{j' \in J_4} \gamma_{j,j'} d_{j'}^2 + \cdots + \\ & \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} - \gamma_{j,j'}) d_{j'}^{\ell-1} + \sum_{j' \in J_4} (-\gamma_{j,j'}) d_{j'}^{\ell-1} + \\ & \gamma_{j,1} d_1^\ell + \cdots + \gamma_{j,r} d_r^\ell.\end{aligned}$$

- If $\alpha_{j,1} = -1$ and ℓ is odd, then

$$\begin{aligned}\chi_\ell(y_j) = {} & \alpha_{j,0} - o_j + \sum_{j' \in J_1} \beta_{j,j'} d_{j'}^0 + \sum_{j' \in J_2} \beta_{j,j'} d_{j'}^0 + \\ & \sum_{j' \in J_3} (-\beta_{j,\pi^{-1}(j'+k)} + \gamma_{j,j'}) d_{j'}^1 + \sum_{j' \in J_4} (\gamma_{j,j'}) d_{j'}^1 + \\ & \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} - \gamma_{j,j'}) d_{j'}^2 + \sum_{j' \in J_4} (-\gamma_{j,j'}) d_{j'}^2 + \cdots + \\ & \sum_{j' \in J_3} (\beta_{j,\pi^{-1}(j'+k)} - \gamma_{j,j'}) d_{j'}^{\ell-1} + \sum_{j' \in J_4} (-\gamma_{j,j'}) d_{j'}^{\ell-1} + \\ & \gamma_{j,1} d_1^\ell + \cdots + \gamma_{j,r} d_r^\ell.\end{aligned}$$

From the analysis above, we know that in $\chi_\ell(y_j)$,

- the coefficient of o_j is from $\{0, +1, -1\}$,
- the constant coefficient is either $\alpha_{j,0}$, or $\alpha_{j,0}\ell$, or 0,
- for each data value $d^0_{j'}$, the coefficient of $d^0_{j'}$ is either $\pm\beta_{j,j'}$, or 0, or $\beta_{j,j'}\ell$,
- for each data value $d^i_{j'}$ with $i > 0$, the coefficient of $d^i_{j'}$ is either 0, or $\beta_{j,\pi^{-1}(j'+k)}$, or $\beta_{j,\pi^{-1}(j'+k)} + \gamma_{j,j'}$, or $\pm\gamma_{j,j'}$, or $\pm(\beta_{j,\pi^{-1}(j'+k)} - \gamma_{j,j'})$.

To summarize, we get the following intuition.

- For the control variables with indices from J_1, they can be dealt with the same as the constant coefficients, that is, dealt with as integer counters.
- For the other control variables as well as those newly introduced data values, their coefficients are from a bounded domain and can be dealt with easily.

4.3 Algorithm for a generalized lasso

In this subsection, we present a decision algorithm for a generalized lasso, that is, the transition graph comprises a handle $q_0 q_1 \dots q_m$ and a collection of simple cycles $C_1, \dots, C_n$ which share the unique state q_m. Moreover, we assume that $O(q_m) = a_0 + a_1 x_1 + \dots + a_k x_k + a'_1 y_1 + \dots + a'_l y_l$ and $O(q)$ is undefined for all the other states q.

In the following, we will illustrate the argument for the case $n = 2$, that is, there are only two cycles.

Let $d_1, \dots, d_m$ denote the m data values met when traversing the handle. The guards and assignments of the transitions in the handle induce an equivalence relation on $\{1, \dots, m\}$ such that $i \sim j$ if the guards and assignments imply that $d_i = d_j$. For each $i : 1 \le i \le m$, let $[i]$ denote the equivalence class containing i. Suppose the equivalence relation $\sim$ has s equivalence classes, say $[i_1], \dots, [i_s]$ such that $i_1 < \dots < i_s$. Let $d^0_1, \dots, d^0_s$ denote the s distinct data values introduced when traversing the cycle, corresponding to $[i_1], \dots, [i_s]$ respectively.

We show by induction that for each $i : 1 \le i \le m$ and each variable x_j (resp. y_j), an arithmetic expression e_{i,x_j} (resp. e_{i,y_j}) corresponding to x_j(resp. y_j) after going through the state sequence $q_0 \dots q_i$ can be constructed. Let $(q_i, g_i, \eta_i, q_{i+1})$ be the i-th transition for each $i : 0 \le i < m$.

For each $j : 1 \le j \le k$, if $\eta_1(x_j) = cur$, then $e_{1,x_j} = d^0_1$, otherwise, $e_{1,x_j} = \perp$.
For each $j : 1 \le j \le l$, $e_{1,y_j} = (\eta_1(y_j))[d^0_1/cur]$.
For each $i : 1 < i \le m$,

- for each $j : 1 \le j \le k$, $e_{i,x_j} = d^0_{s'}$ if $\eta_i(x_j) = cur$ and $[i] = [i_{s'}]$, and $e_{i,x_j} = e_{i-1,x_j}$ otherwise,
- for each $j : 1 \le j \le l$, $e_{i,y_j} = \theta_i(\eta_i(y_j))$, where $\theta_i(x_{j'}) = e_{i-1,x_{j'}}$, $\theta_i(y_{j'}) = e_{i-1,y_{j'}}$, and $\theta_i(cur) = d^0_{s'}$, where $[i] = [i_{s'}]$.

Then for each $j : 1 \le j \le k$, $e_{m,x_j} = d^0_{\pi_0(j)}$ for some injective mapping $\pi_0 : \{1, \dots, k\} \to \{1, \dots, s\}$, and for each $j : 1 \le j \le l$, $e_{m,y_j} = c_{0,j} + c_{1,j} d^0_1 + \dots + c_{s,j} d^0_s$ for some integer constants $c_{0,j}, \dots, c_{s,j}$. We use χ_0 to denote the assignment function such that $\chi_0(x_j) = e_{m,x_j}$ and $\chi_0(y_j) = e_{m,y_j}$. Note here we ignore the situations that $e_{m,x_j} = \perp$ for simplicity.

Step I. Decide whether $\chi_0(O(q_m))$ is not identical to zero (it is easy to do so, just check the coefficients of $d^0_1, \ldots, d^0_s$). If the answer is yes, then the decision procedure terminates and returns the answer *true*. Otherwise, the decision procedure continues.

For the cycle C_b (where $b = 1, 2$), the assignment function $\chi_{b,\ell}$ can be defined to describe the values of the control and output variables after traversing the cycle C_b for ℓ times.

Suppose that for $b = 1, 2$, the initial values of the k control variables are $d^0_{\pi_0(1)}, \ldots, d^0_{\pi_0(k)}$. Moreover, suppose that the r_b data values $d^1_1, \ldots, d^1_{r_b}$ are introduced when traversing the cycle for the first time, with one data value for each of $I_1, \ldots, I_{r_b}$. In addition, suppose that the initial values of $y_1, \ldots, y_l$ are $o_1, \ldots, o_l$.

Then for $b = 1, 2$, the assignment function $\chi_{b,1}$ is of the following form,

- there is an injective mapping $\pi_b : \{1, \ldots, k\} \rightarrow \{1, \ldots, k + r_b\}$ such that for each $x_j \in X$, if $\pi_b(j) \leq k$, then $\pi_b(j) = j$ and $\chi_{b,1}(x_j) = d^0_{\pi_0(j)}$, otherwise, $\chi_{b,1}(x_j) = d^1_{\pi_b(j)-k}$,
- for each $y_j \in Y$, $\chi_{b,1}(y_j) = \alpha_{b,(j,0)} + \alpha_{b,(j,1)} o_j + \beta_{b,(j,1)} d^0_{\pi_0(1)} + \cdots + \beta_{b,(j,k)} d^0_{\pi_0(k)} + \gamma_{b,(j,1)} d^1_1 + \cdots + \gamma_{b,(j,r_b)} d^1_{r_b}$ such that $\alpha_{b,(j,1)} \in \{0, +1, -1\}$.

Let $J_{b,1}$ be set of indices $j \in \{1, \ldots, k + r_b\}$ such that $\pi_b(j) = j$ and $J_{b,2} = \{1, \ldots, k\} \setminus J_{b,1}$. Intuitively, the values of the variables from J_1 are unchanged after traversing the cycle once, and the values of the variables from $J_{b,2}$ are changed. Then $(J_{b,1}, J_{b,2})$ forms a partition of $\{1, \ldots, k\}$. Moreover, let $J_{b,3} = \{\pi_b(j) - k \mid j \in J_{b,2}\}$ and $J_{b,4} = \{1, \ldots, r_b\} \setminus J_{b,3}$. Then $(J_{b,3}, J_{b,4})$ forms a partition of $\{1, \ldots, r_b\}$.

We will present the argument for the situation that $J_{1,1} \cap J_{2,1} \neq \emptyset$, $J_{1,1} \setminus J_{2,1} \neq \emptyset$, and $J_{2,1} \setminus J_{1,1} \neq \emptyset$.

A *cycle scheme* $\mathfrak{s}$ is a sequence $\mathfrak{s} = (1, \ell_1)(2, \ell_2) \ldots (((m-1) \bmod 2) + 1, \ell_m)$ such that $\ell_1, \ldots, \ell_m \geq 1$. The number m is called the length of the cycle scheme $\mathfrak{s}$. In principle, for each cycle scheme $\mathfrak{s}$, the expressions $\chi_{\mathfrak{s}}(y_j)$ can be defined to describe the values of y_j after traversing the cycles according to $\mathfrak{s}$.

In the following, we first show how to construct the expressions $\chi_{(1,\ell_1)(2,\ell_2)}$ for a cycle scheme $(1, \ell_1)(2, \ell_2)$ such that $\ell_1, \ell_2 \geq 2$, to describe the values of the control and data variables after traversing C_1 for ℓ_1 times and C_2 for ℓ_2 times.

At first, from the analysis of cycles, we know that

$$\begin{aligned}
\chi_{(1,\ell_1)}(y_j) = {} & (\alpha_{1,(j,0)} + \alpha_{1,(j,1)}\alpha_{1,(j,0)} + \cdots + \alpha^{\ell_1-1}_{1,(j,1)}\alpha_{1,(j,0)}) + \alpha^{\ell_1}_{1,(j,1)} o_j + \\
& \sum_{j' \in J_{1,1}} (\beta_{1,(j,j')} + \alpha_{1,(j,1)}\beta_{1,(j,j')} + \cdots + \alpha^{\ell_1-1}_{1,(j,1)}\beta_{1,(j,j')}) d^0_{\pi_0(j')} + \\
& \sum_{j' \in J_{1,2}} (\alpha^{\ell_1-1}_{1,(j,1)}\beta_{1,(j,j')}) d^0_{\pi_0(j')} + \sum_{j' \in J_{1,3}} (\alpha^{\ell_1-2}_{1,(j,1)}\beta_{1,(j,\pi^{-1}(j'+k))} + \alpha^{\ell_1-1}_{1,(j,1)}\gamma_{1,(j,j')}) d^1_{j'} + \\
& \sum_{j' \in J_{1,4}} (\alpha^{\ell_1-1}_{1,(j,1)}\gamma_{1,(j,j')}) d^1_{j'} + \cdots + \sum_{j' \in J_{1,3}} (\beta_{1,(j,\pi^{-1}(j'+k))} + \alpha_{1,(j,1)}\gamma_{1,(j,j')}) d^{\ell_1-1}_{j'} + \\
& \sum_{j' \in J_{1,4}} (\alpha_{1,(j,1)}\gamma_{1,(j,j')}) d^{\ell_1-1}_{j'} + \gamma_{1,(j,1)} d^{\ell_1}_1 + \cdots + \gamma_{1,(j,r_1)} d^{\ell_1}_{r_1},
\end{aligned}$$

and

$$\begin{aligned}\chi_{(2,\ell_2)}(y_j) = {} & (\alpha_{2,(j,0)} + \alpha_{2,(j,1)}\alpha_{2,(j,0)} + \cdots + \alpha_{2,(j,1)}^{\ell_2-1}\alpha_{2,(j,0)}) + \alpha_{2,(j,1)}^{\ell_2} o_j + \\ & \sum_{j' \in J_{2,1}} (\beta_{2,(j,j')} + \alpha_{2,(j,1)}\beta_{2,(j,j')} + \cdots + \alpha_{2,(j,1)}^{\ell_2-1}\beta_{2,(j,j')}) d_{\pi_0(j')}^0 + \\ & \sum_{j' \in J_{2,2}} (\alpha_{2,(j,1)}^{\ell_2-1}\beta_{2,(j,j')}) d_{\pi_0(j')}^0 + \sum_{j' \in J_{2,3}} (\alpha_{2,(j,1)}^{\ell_2-2}\beta_{2,(j,\pi^{-1}(j'+k))} + \\ & \alpha_{2,(j,1)}^{\ell_2-1}\gamma_{2,(j,j')}) d_{j'}^1 + \sum_{j' \in J_{2,4}} (\alpha_{2,(j,1)}^{\ell_2-1}\gamma_{2,(j,j')}) d_{j'}^1 + \cdots + \\ & \sum_{j' \in J_{2,3}} (\beta_{2,(j,\pi^{-1}(j'+k))} + \alpha_{2,(j,1)}\gamma_{2,(j,j')}) d_{j'}^{\ell_2-1} + \\ & \sum_{j' \in J_{2,4}} (\alpha_{2,(j,1)}\gamma_{2,(j,j')}) d_{j'}^{\ell_2-1} + \gamma_{2,(j,1)} d_1^{\ell_2} + \cdots + \gamma_{2,(j,r_2)} d_{r_2}^{\ell_2}.\end{aligned}$$

The coefficients of $\chi_{(1,\ell_1)(2,\ell_2)}(y_j)$ can be obtained those of $\chi_{(1,\ell_1)}(y_j)$ and $\chi_{(2,\ell_2)}(y_j)$ as follows: For each $i : 1 \leq i \leq \ell_1$ and $j' : 1 \leq j' \leq r_1$, let $d_{j'}^i$ denote the data values introduced when traversing C_1, and for each $i : 1 \leq i \leq \ell_2$ and $j' : 1 \leq j' \leq r_2$, let $d_{(1,\ell_1),j'}^i$ denote the data values introduced when traversing C_2 for ℓ_2 times (after traversing C_1 for ℓ_1 times).

- The coefficient of o_j in $\chi_{(1,\ell_1)(2,\ell_2)}(y_j)$ is $\theta_1\theta_2$, where θ_1 and θ_2 are the coefficient of o_j in $\chi_{(1,\ell_1)}(y_j)$ and $\chi_{(2,\ell_2)}(y_j)$ respectively.
- The constant coefficient of $\chi_{(1,\ell_1)(2,\ell_2)}(y_j)$ is $\alpha_{2,(j,1)}^{\ell_2}\theta_1 + \theta_2$, where θ_1, θ_2 are the constant coefficient of $\chi_{(1,\ell_1)}(y_j)$ and $\chi_{(2,\ell_2)}(y_j)$ respectively.
- For each $j' : 1 \leq j' \leq k$, suppose $\chi_{(1,\ell_1)}(x_{j'}) = d_{j'}^i$, then the coefficient of $d_{j'}^i$ in $\chi_{(1,\ell_1)(2,\ell_2)}(y_j)$ is $\alpha_{2,(j,1)}^{\ell_2}\theta_1 + \theta_2$, where θ_1, θ_2 are the coefficient of $d_{j'}^i$ in $\chi_{(1,\ell_1)}(y_j)$ and $\chi_{(2,\ell_2)}(y_j)$ respectively.
- For each pair (i, j') such that $d_{j'}^i \not\in \{\chi_{(1,\ell_1)}(x_{j''}) \mid 1 \leq j'' \leq k\}$, the coefficient of $d_{j'}^i$ in $\chi_{(1,\ell_1)(2,\ell_2)}(y_j)$ is $\alpha_{2,(j,1)}^{\ell_2}\theta$, where θ is the coefficient of $d_{j'}^i$ in $\chi_{(1,\ell_1)}(y_j)$.
- For each pair (i, j') such that $1 \leq i \leq \ell_2$ and $j' \in J_{2,3} \cup J_{2,4}$, the coefficient of $d_{(1,\ell_1),j'}^i$ in $\chi_{(1,\ell_1)(2,\ell_2)}(y_j)$ is θ, where θ is the coefficient of $d_{j'}^i$ in $\chi_{(2,\ell_2)}(y_j)$, where $d_{(1,\ell_1),j'}^i$ is the data variable corresponding to $d_{j'}^i$ in $\chi_{(2,\ell_2)}(y_j)$, decorated with $(1, \ell_1)$.

Step II. For $b = 1, 2$, let

$$\begin{aligned}\chi_{(b,\ell_b)}(O(q_m)) = a_0 + a_1\chi_{(b,\ell_b)}(x_1) + \ldots a_k\chi_{(b,\ell_b)}(x_k) + \\ a'_1\chi_{(b,\ell_b)}(y_1) + \cdots + a'_l\chi_{(b,\ell_b)}(y_l).\end{aligned}$$

Then $\chi_{(b,\ell_b)}(O(q_m))$ is a linear combination of the variables $d_1^0, \ldots, d_s^0$ and $d_1^1, \ldots, d_{r_b}^1, \ldots, d_1^{\ell_b}, \ldots, d_{r_b}^{\ell_b}$.

For each $j' \in J_{b,1}$, the coefficient of $d_{\pi_0(j')}^0$ in $\chi_{(b,\ell_b)}(O(q_m))$ is

$$a_{j'} + \sum_{1 \leq j \leq l} a'_j (1 + \alpha_{b,(j,1)} + \cdots + \alpha_{b,(j,1)}^{\ell_b-1})\beta_{b,(j,j')}.$$

In general, for each $j' \in J_{1,1}$ and each cycle scheme $\mathfrak{s} = (1, \ell_1)(2, \ell_2) \ldots (1 + (t-1) \bmod 2, \ell_t)$ (where $t \geq 1$), the coefficient of $d^0_{\pi_0(j')}$ in $\chi_{\mathfrak{s}}(O(q_m))$ includes the following expression as a component,

$$a_{j'} + \sum_{1 \leq j \leq l} a'_j(\alpha^{\ell_2}_{2,(j,1)} \cdots \alpha^{\ell_t}_{1,(j,1)})(1 + \alpha_{1,(j,1)} + \cdots + \alpha^{\ell_1 - 1}_{1,(j,1)})\beta_{1,(j,j')}. \quad (*)$$

Note that since $\alpha_{1,(j,1)}, \alpha_{2,(j,1)} \in \{0, +1, -1\}$, the expression $(*)$ is of the form

$$a_{j'} + \sum_{1 \leq j \leq l} a'_j s_j (1 + \alpha_{1,(j,1)} + \cdots + \alpha^{\ell_1 - 1}_{1,(j,1)})\beta_{1,(j,j')},$$

where $s_j \in \{0, +1, -1\}$.

The expression $(*)$ can be rewritten as $\mu_{\mathfrak{s},(1,j')}\ell_1 + \nu_{\mathfrak{s},(1,j')}$ for some integer constant $\mu_{\mathfrak{s},(1,j')}, \nu_{\mathfrak{s},(1,j')}$ (possibly $\mu_{\mathfrak{s},(1,j')} = 0$).

If there are a cycle scheme $\mathfrak{s}$ starting from C_1 and $j' \in J_{1,1}$ such that $\mu_{\mathfrak{s},(1,j')} \neq 0$, then return *true*. Note that in this case, we can let ℓ_1 and $d^0_{\pi_0(j')}$ sufficiently large so that $(\mu_{\mathfrak{s},(1,j')}\ell_1 + \nu_{\mathfrak{s},(1,j')})d^0_{\pi_0(j')}$ dominates $\chi_{\mathfrak{s}}(O(q_m))$ and $\chi_{\mathfrak{s}}(O(q_m))$ becomes non-zero. We would like to remark that although there are infinitely many cycle schemes $\mathfrak{s}$, the constants $\mu_{\mathfrak{s},(1,j')}$ can only have values from a bounded domain. Therefore, it is decidable whether there exist such a desired cycle scheme $\mathfrak{s}$ starting from C_1 and $j' \in J_{1,1}$.

The similar discussion can be applied to C_2 and $J_{2,1}$.

If there are no desired cycle schemes $\mathfrak{s}$ and j' for C_1 and $J_{1,1}$, as well as for C_2 and $J_{2,1}$, then the decision procedure continues.

The constant coefficient in $\chi_{(b,\ell_b)}(O(q_m))$ is

$$a_0 + \sum_{1 \leq j \leq l} a'_j (1 + \alpha_{b,(j,1)} + \cdots + \alpha^{\ell_b - 1}_{b,(j,1)})\alpha_{b,(j,0)}.$$

In general, for each cycle scheme $\mathfrak{s} = (1, \ell_1)(2, \ell_2) \ldots (1 + (t-1) \bmod 2, \ell_t)$ (where $t \geq 1$), the constant coefficient in $\chi_{\mathfrak{s}}(O(q_m))$ includes the following expression as a component,

$$a_0 + \sum_{1 \leq j \leq l} a'_j(\alpha^{\ell_2}_{2,(j,1)} \cdots \alpha^{\ell_t}_{1,(j,1)})(1 + \alpha_{b,(j,1)} + \cdots + \alpha^{\ell_b - 1}_{b,(j,1)})\alpha_{b,(j,0)}. \quad (**)$$

Note that since $\alpha_{1,(j,1)}, \alpha_{2,(j,1)} \in \{0, +1, -1\}$, the expression $(**)$ is of the form

$$a_0 + \sum_{1 \leq j \leq l} a'_j s'_j (1 + \alpha_{b,(j,1)} + \cdots + \alpha^{\ell_b - 1}_{b,(j,1)})\alpha_{b,(j,0)},$$

where $s'_j \in \{0, +1, -1\}$.

The expression $(**)$ can be rewritten as $\mu_{\mathfrak{s},(1,0)}\ell_1 + \nu_{\mathfrak{s},(1,0)}$ for some integer constant $\mu_{\mathfrak{s},(1,0)}, \nu_{\mathfrak{s},(1,0)}$ (possibly $\mu_{\mathfrak{s},(1,0)} = 0$).

If there are a cycle scheme $\mathfrak{s}$ starting from C_1 such that $\mu_{\mathfrak{s},(1,0)} \neq 0$, then return *true*. Note that in this case, we can let ℓ_1 sufficiently large so that $\mu_{\mathfrak{s},(1,0)}\ell_1 + \nu_{\mathfrak{s},(1,0)}$ dominates $\chi_{\mathfrak{s}}(O(q_m))$ and $\chi_{\mathfrak{s}}(O(q_m))$ becomes non-zero. We would like to remark that although there are infinitely many cycle schemes $\mathfrak{s}$, the constants $\mu_{\mathfrak{s},(1,0)}$ can only have values from a bounded domain. Therefore, it is decidable whether there exist such a desired cycle scheme $\mathfrak{s}$ starting from C_1.

A similar discussion for the constant coefficient can be applied to the cycle schemes starting from C_2.

If the decision procedure has not returned yet, then we go to Step III.

Step III. For each cycle scheme $\mathfrak{s} = (1, \ell_1)(2, \ell_2) \ldots (1 + (t-1) \bmod 2, \ell_t)$ or $\mathfrak{s} = (2, \ell_1)(1, \ell_2) \ldots (1 + (t-1) \bmod 2, \ell_t)$, we can ignore all the expressions of the form $c\ \ell_1, \ldots, c\ \ell_m$ (where c is an integer constant) in $\chi_{\mathfrak{s}}(y_j)$, since these expressions will disappear for sure in $\chi_{\mathfrak{s}}(O(q_m))$, according to the analysis above. From this observation, we can show that the constant coefficient as well as the coefficients for $d^0_{\pi_0(j')}$ with $j' \in J_{1,1} \cap J_{2,1}$ in $\chi_{\mathfrak{s}}(y_1), \ldots, \chi_{\mathfrak{s}}(y_l), \chi_{\mathfrak{s}}(O(q_m))$ can be calculated by a $\mathbb{Z}$-VAS (cf. [3]), that is, an integer vector addition system. Moreover, all the other coefficients in $\chi_{\mathfrak{s}}(y_1), \ldots, \chi_{\mathfrak{s}}(y_l), \chi_{\mathfrak{s}}(O(q_m))$ are from a bounded domain, no matter how long the scheme $\mathfrak{s}$ is.

Therefore, in this case, the non-zero output reachability problem is reduced to the non-zero reachability problem of $\mathbb{Z}$-VAS, that is, given an index i, decide whether a vector $\boldsymbol{z}$ where z_i is non-zero can be reached. It is not hard to see that the non-zero reachability problem of $\mathbb{Z}$-VAS can be reduced to the coverability problem of $\mathbb{Z}$-VAS. From the fact that the coverability of $\mathbb{Z}$-VAS is NP-complete ([3]), we conclude that the non-zero output reachability of SNS± whose transition graph is a generalized lasso is decidable.

4.4 Comparison with constants

In this subsection, we consider the situation that the guards may contain comparisons with constants.

5 Discussions

From the analysis of the commutativity of reducers in [6], the commutativity of a reducer in a sequential composition oa map-reduce jobs may depend on some implicit data properties guaranteed by the preceding map-reduce jobs. Therefore, to analyze the commutativity of a reducer in a sequential composition of map-reduce jobs, we may need model both mappers and reducers and do a backward analysis.

References

1. Y. Chen, C. Hong, N. Sinha, and B. Wang. Commutativity of reducers. In *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, pages 131–146, 2015.

2. S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3):16:1–16:30, 2009.
3. C. Haase and S. Halfon. Integer vector addition systems with states. In *RP 2014*, pages 112–124, 2014.
4. Hadoop. https://hadoop.apache.org.
5. Spark. http://spark.apache.org.
6. T. Xiao, J. Zhang, H. Zhou, Z. Guo, S. McDirmid, W. Lin, W. Chen, and L. Zhou. Nondeterminism in mapreduce considered harmful? an empirical study on non-commutative aggregators in mapreduce programs. In *36th International Conference on Software Engineering, ICSE '14, Companion Proceedings*, pages 44–53, 2014.