Determinizing Two-way Alternating Pebble Automata for Data Languages

Tony Tan
Department of Computer Science
Technion – Israel Institute of Technology
Haifa 32000, Israel
Email: `tantony@cs.technion.ac.il`

Abstract

We prove that for every integer $k \geq 1$, two-way alternating k-pebble automata and one-way deterministic k-pebble automata for data languages have the same recognition power.

1 Introduction

Recently there has been a considerable amount of research work in the models of computation for languages over infinite alphabets, or also known as data languages. Various models have been introduced such as finite memory automata [4], pebble automata [6], data automata [2], class memory automata [1] and linear temporal logic with freeze quantifier [3]. Each of these models has its own advantages and disadvantages. For an extensive survey of the progress in this area we refer the reader to [7].

In this paper we continue the study of pebble automata for data languages introduced in [6]. In short, k-PA, which stands for k pebble automata, are finite state automata with k pebbles, numbered from 1 to k. The automaton starts the computation with only pebble k on the input word. The pebbles are placed on/lifted from the input word in the stack discipline according to the strict order of the pebbles: Pebble i can be placed only when pebbles $i+1,\dots,k$ are above the input word. Each pebble is intended to mark one position in the input word and the smallest numbered pebble on the input word, or, equivalently the most recently placed pebble, serves as the head of the automaton. The automaton moves from one state to another depending on the equality tests among data values in the positions currently marked by the pebbles, as well as, the equality tests among the positions of the pebbles.

It is shown in [6] that languages accepted by pebble automata are closed under all boolean operations. However, in general its emptiness problem is undecidable [6]. For a more extensive study of pebble automata we refer the reader to [8] and [9].

Observing the stack discipline imposed on the placement of the pebbles, one can notice the attempt to "descendize" the infinity of data values to finite alphabet. When pebble i is acting as the head pebble, it makes comparison of the data value it sees with the data values seen by pebbles $i+1, \ldots, k$. Thus, pebble i is essentially acting like the standard finite state automaton over the "alphabet" of data values seen by pebbles $i+1, \ldots, k$.

Such observation immediately leads us to conjecture that PA languages are robust. In fact, for each $k \geq 1$, all versions of k-PA: two-way non-deterministic, two-way deterministic, one-way non-deterministic and one-way deterministic, have the same recognition power [6, Theorem 4.6].

In this paper we will prove that for each $k \geq 1$, two-way alternating and one-way deterministic k-PA have the same recognition power. Not surprisingly, our proof follows closely the same proof for the equivalence between two-way alternating and one-way deterministic finite state automata in [5]. The result settles a question left open in [6].

The determinization itself are done inductively from pebble 1 to pebble k. The basis is determinization of the behavior of pebble 1. This step is a straightforward adaptation of the proof in [5]. The induction step is, assuming that pebble $1, \ldots, i$ behave deterministically, we show how to determinize the behavior of pebble $i+1$.

However, one must note that in terms of applications, decidability is more important than robustness. See, for example, [1]. Since the emptiness problem for PA in general is undecidable, it can still be argued that general PA may not be the right kind of model for data languages in terms of applications.

This paper is organized as follows. In Section 2 we review the two-way alternating finite state automata. Especially, we sketch the main idea of the Ladner, Lipton and Stockmeyer's proof that one-way deterministic and two-way alternating finite state automata have the same recognition power. Our proof for the pebble automata adopts essentially their idea. In Section 3 we present the definition of pebble automata for data languages. We present our proof in Section 4.

2 Two-way Alternating Finite State Automata

A two-way alternating finite state automaton over the finite alphabet Σ is a system $\mathcal{M} = \langle Q, q_0, F, \Delta, D, N, U \rangle$, where

- Q, q_0 and $F \subseteq Q$ are the set of states, initial state and the set of final states, respectively;
- Q is partitioned into $D \cup N \cup U$, where $N \cap F = U \cap F = \emptyset$;
- Δ is a set of transitions of the form $(p, \sigma) \to (q, \mathtt{act})$, where $p, q \in Q$, $\sigma \in \Sigma$ and $\mathtt{act} \in \{\mathtt{left}, \mathtt{right}, \mathtt{stay}\}$.

The states in D, N and U are called the *deterministic*, *nondeterministic* and *universal* states, respectively. The states in N and U are the states in which the automaton can perform the disjunctive and conjunctive branching, respectively.

We assume that the automaton $\mathcal{M}$ behaves as follows.

- The input to $\mathcal{M}$ is of the form $\triangleleft w \triangleright$, where $w \in \Sigma^*$ and $\triangleleft, \triangleright \notin \Sigma$ are the left-end and the right-end markers of the input.
- The automaton $\mathcal{M}$ starts the computation with the head is reading the right-end marker $\triangleright$.
- The automaton $\mathcal{M}$ can only enter a final state when the head of the automaton reads the right-end marker $\triangleright$.
- When the automaton $\mathcal{M}$ performs disjunctive and conjunctive branching the head of the automaton is stationery.
  That is, if $(p, \sigma) \to (q, \mathtt{act})$ and $p \in N \cup U$, then $\mathtt{act} = \mathtt{stay}$.

Given a word $w = \sigma_1 \cdots \sigma_n \in \Sigma^*$, a *configuration of* $\mathcal{M}$ *on* $\triangleleft w \triangleright$ is a triple $[q, \triangleleft w \triangleright, l]$, where $l \in \{0, \ldots, n+2\}$ and $q \in Q$. The positions 0 and $n+1$ are positions of the end markers $\triangleleft$ and $\triangleright$, respectively. The *initial* configuration is $\gamma_0 = [q_0, \triangleleft w \triangleright, n+1]$. When $l = n+2$, it means that the head of the automaton "falls off" the right side of the input word and the automaton finishes the computation.

The set of transitions Δ induces the relation $\vdash$ among the configurations as follows. $[q, \lhd w \rhd, l] \vdash [q', \lhd w \rhd, l']$ if there exists a transition $(q, \sigma_l) \rightarrow (q', \texttt{act}) \in \Delta$ and

- if $l = l'$, then $\texttt{act} = \texttt{stay}$;
- if $l = l' - 1$, then $\texttt{act} = \texttt{left}$; and
- if $l = l' + 1$, then $\texttt{act} = \texttt{right}$.

The acceptance criteria is based on the notion of *leads to acceptance* below. For every configuration $\gamma = [q, \lhd w \rhd, l]$,

- if $q \in F$, then γ leads to acceptance;
- if $q \in U$, then γ leads to acceptance if and only if for all configurations γ' such that $\gamma \vdash \gamma'$, γ' leads to acceptance;
- if $q \notin F \cup U$, then γ leads to acceptance if and only if there is at least one configuration γ' such that $\gamma \vdash \gamma'$, and γ' leads to acceptance.

The word w is accepted by $\mathcal{M}$ if the initial configuration γ_0 leads to acceptance.

As usual, a computation of $\mathcal{M}$ on the input $\lhd w \rhd$ can be viewed as a computation tree where each node is labelled with a configuration and

- if a node π is labelled with a configuration $[q, \lhd w \rhd, l]$, where $q \in D \cup N$, then π has only one child labelled with a configuration γ', where $\gamma \vdash \gamma'$;
- if a node π is labelled with a configuration $[q, \lhd w \rhd, l]$, where $q \in U$, then for all configuration γ' such that $\gamma \vdash \gamma'$, there exists a child of π labelled with γ'.

It is shown in [5] that every two-way alternating finite state automaton can be simulated by one-way deterministic finite state automaton. One important notion introduced in [5] is the notion of *closed terms*, which we will describe below.

For each state $q \in Q$, we define a new symbol $\bar{q}$ and let $\bar{Q} = \{\bar{q} : q \in Q\}$. If $S \subseteq Q$, then $\bar{S} = \{\bar{p} : p \in S\}$. We define a *term* to be an object $q \rightarrow S$, where $q \in Q$ and $S \subseteq Q \cup \bar{Q}$. A term $q \rightarrow S$ is *closed*, if $S \subseteq \bar{Q}$. A *partial response* is a set of terms, and a *response* is a set of closed terms. Note that since Q is finite, there are only finitely many closed terms and responses.

A configuration $\gamma = [q, \triangleleft w \triangleright, l]$ *induces* a closed term $q \rightarrow \bar{S}$, for some $S \subseteq Q$, if there exists a computation tree of $\mathcal{M}$ on $\triangleleft w \triangleright$ such that

- the root is labelled with the configuration γ;
- all the leaf nodes are labelled with a configuration $[p, \triangleleft w \triangleright, l+1]$, for some $p \in S$;
- for every $p \in S$, there exists a leaf node labelled with a configuration $[p, \triangleleft w \triangleright, l+1]$;
- every interior node is labelled with a configuration $[s, \triangleleft w \triangleright, j]$, for some $0 \leq j \leq l$ and $s \in Q$.

We define a response $\mathcal{R}(\triangleleft w \triangleright, l)$ as the set of closed terms induced by the configurations $[q, \triangleleft w \triangleright, l]$. In other words, a closed term $p \rightarrow \bar{S} \in \mathcal{R}(\triangleleft w \triangleright, l)$, where $S \subseteq Q$, if and only if there exists a configuration $[p, \triangleleft w \triangleright, l]$ that induces $p \rightarrow \bar{S}$.

Now the main point in the proof in [5] is that given a response $\mathcal{R}(\triangleleft w \triangleright, l)$, we can construct the response $\mathcal{R}(\triangleleft w \triangleright, l+1)$ without simulating the automaton $\mathcal{M}$ on $\triangleleft w \triangleright$. This is done by defining a proof system $\mathcal{S}(\mathcal{R}(\triangleleft w \triangleright, l), \sigma_l)$, where the closed terms in $\mathcal{R}(\triangleleft w \triangleright, l)$ and the transitions $(p, \sigma_l) \rightarrow (q, \mathtt{act}) \in \Delta$ are the axioms. Now, the response $\mathcal{R}(\triangleleft w \triangleright, l+1)$ is precisely the set of closed terms provable in $\mathcal{S}(\mathcal{R}(\triangleleft w \triangleright, l), \sigma_l)$ [5, Claim in pp. 149]. In [5] such set of closed terms is denoted by $\mathrm{CTH}(\mathcal{R}(\triangleleft w \triangleright, l), \sigma_l)$.

The construction of one-way deterministic automaton $\mathcal{M}'$ that accepts the same language as $\mathcal{M}$ is as follows. The states of $\mathcal{M}'$ are exactly the responses. The transitions of $\mathcal{M}'$ are of the form

$$(\mathcal{R}, \sigma_l) \rightarrow (\mathrm{CTH}(\mathcal{R}, \sigma_l), \mathtt{right}),$$

where $\mathcal{R}$ is a response. This is the essence of the proof in [5] that we are going to use in this paper.

3 Definition

We will use the following notation. We always denote by Σ a finite alphabet of *labels* and by $\mathfrak{D}$ an infinite set of *data values*. A Σ-*data word*

$w = \binom{\sigma_1}{a_1}\binom{\sigma_2}{a_2}\cdots\binom{\sigma_n}{a_n}$ is a finite sequence over $\Sigma \times \mathfrak{D}$, where $\sigma_i \in \Sigma$ and $a_i \in \mathfrak{D}$. A Σ-*data language* is a set of Σ-data words. The idea is that the alphabet Σ is accessed directly, while data values can only be tested for equality.

We assume that neither of Σ and $\mathfrak{D}$ contain the left-end marker $\triangleleft$ or the right-end marker $\triangleright$. The input word to the automaton is of the form $\triangleleft w \triangleright$, where $\triangleleft$ and $\triangleright$ mark the left-end and the right-end of the input word.

Finally, the symbol σ, possibly indexed, denotes labels in Σ and the symbol a, possibly indexed, denote data values in $\mathfrak{D}$. We will use the symbols ρ, π to denote the nodes in a computation tree of (alternating) PA on an input word $\triangleleft w \triangleright$.

3.1 Alternating k-PA

Definition 1 (See [6, Definition 2.3]) A *two-way alternating k-pebble automaton* or, in short, k-PA, over Σ is a system $\mathcal{A} = \langle Q, q_0, F, \mu, U, N, D\rangle$ whose components are defined as follows.

- Q, q_0 and F are the set of states, the initial state and the set of final states, respectively;
- $\mu \subseteq \mathcal{C} \times \mathcal{D}$ is the transition relation, where
  - $\mathcal{C}$ is a set whose elements are of the form (i, σ, P, V, q) where $1 \leq i \leq k$, $\sigma \in \Sigma$, $P, V \subseteq \{i+1, \ldots, k\}$ and $q \in Q$; and
  - $\mathcal{D}$ is a set whose elements are of the form $(q, \texttt{act})$, where $q \in Q$ and $\texttt{act} \in \{\texttt{stay}, \texttt{left}, \texttt{right}, \texttt{place-pebble}, \texttt{lift-pebble}\}$.

  Elements of μ will be written as $(i, \sigma, P, V, q) \rightarrow (p, \texttt{act})$.
- Q is partitioned into $U \cup N \cup D$, where
  - $U \subseteq Q - F$ is the set of *universal* states;
  - $N \subseteq Q - F$ is the set of *nondeterministic* states; and
  - D is the set of *deterministic* states.

Given a word $w = \binom{\sigma_1}{a_1}\cdots\binom{\sigma_n}{a_n} \in (\Sigma \times \mathfrak{D})^*$, a *configuration of $\mathcal{A}$ on $\triangleleft w \triangleright$* is a triple $[i, q, \theta]$, where $i \in \{1, \ldots, k\}$, $q \in Q$, and $\theta : \{i, i+1, \ldots, k\} \rightarrow \{0, 1, \ldots, n, n+1\}$, where 0 and $n+1$ are the positions of the end markers

$\triangleleft$ and $\triangleright$, respectively. The function θ defines the position of the pebbles and is called the *pebble assignment*.

The *initial* configuration is $[k, q_0, \theta_0]$ where $\theta_0(k) = n + 1$. That is, in the start of the computation pebble k is positioned in the right-end marker $\triangleright$. This is in contrast with the definition in [6], where pebble k is placed in the left-end marker $\triangleleft$ at the beginning of the computation. Obviously such difference does not change the expressive power.

A transition $(i, \sigma, P, V, p) \to \beta$ *applies to a configuration* $[j, q, \theta]$, if

(1) $i = j$ and $p = q$,

(2) $P = \{l > i : \theta(l) = \theta(i)\}$,

(3) $V = \{l > i : a_{\theta(l)} = a_{\theta(i)}\}$, and

(4) $\sigma_{\theta(i)} = \sigma$.

Note that in a configuration $[i, q, \theta]$, pebble i is in control, serving as the head pebble.

Next, we define the transition relation $\vdash_{\mathcal{A}}$ as follows: $[i, q, \theta] \vdash_{\mathcal{A}} [i', q', \theta']$, if there is a transition $\alpha \to (p, \texttt{act}) \in \mu$ that applies to $[i, q, \theta]$ such that $q' = p$, $\theta'(j) = \theta(j)$, for all $j > i$, and

- if $\texttt{act} = \texttt{stay}$, then $i' = i$ and $\theta'(i) = \theta(i)$;
- if $\texttt{act} = \texttt{left}$, then $i' = i$ and $\theta'(i) = \theta(i) - 1$;
- if $\texttt{act} = \texttt{right}$, then $i' = i$ and $\theta'(i) = \theta(i) + 1$;
- if $\texttt{act} = \texttt{lift-pebble}$, then $i' = i + 1$;
- if $\texttt{act} = \texttt{place-pebble}$, then $i' = i-1$, $\theta'(i-1) = n+1$ and $\theta'(i) = \theta(i)$.

As usual, we denote the reflexive transitive closure of $\vdash_{\mathcal{A}}$ by $\vdash_{\mathcal{A}}^*$. When the automaton $\mathcal{A}$ is clear from the context, we will omit the subscript $\mathcal{A}$. For a subset $\mu' \subseteq \mu$, we will also denote by $\gamma_1 \vdash_{\mu'} \gamma_2$, when the relation $\gamma_1 \vdash \gamma_2$ is obtained by a transition in μ'. For a configuration $[i, q, \theta]$, where $q \in D$, there exists exactly one transition that applies to it.

Similar to the finite state automata, the acceptance criteria is based on the notion of *leads to acceptance* below. For every configuration $\gamma = [i, q, \theta]$,

- if $q \in F$, then γ leads to acceptance;

- if $q \in U$, then γ leads to acceptance if and only if for all configurations γ' such that $\gamma \vdash \gamma'$, γ' leads to acceptance;

- if $q \notin F \cup U$, then γ leads to acceptance if and only if there is at least one configuration γ' such that $\gamma \vdash \gamma'$, and γ' leads to acceptance.

A Σ-data word $w \in (\Sigma \times \mathfrak{D})^*$ is accepted by $\mathcal{A}$, if the initial configuration γ_0 leads to acceptance. The language $L(\mathcal{A})$ consists of all data words accepted by $\mathcal{A}$.

As usual, the computation of $\mathcal{A}$ on w can be viewed as a computation tree, where

- if a node π is labelled with a configuration $[i, q, \theta]$, where $q \in D \cup N$, then π has only one child labelled with a configuration γ', where $\gamma \vdash \gamma'$;

- if a node π is labelled with a configuration $[i, q, \theta]$, where $q \in U$, then for all configuration γ' such that $\gamma \vdash \gamma'$, there exists a child of π labelled with γ'.

4 The Equivalence between Alternating and Deterministic k-PA

In this section we will prove that for all $k \geq 1$, two-way alternating k-PA and one-way deterministic k-PA have the same recognition power. As mentioned earlier, the proof is a direct generalization of the same proof for the equivalence between two-way alternating and one-way deterministic finite state automata in [5].

Let $\mathcal{A} = \langle \Sigma, Q, q_0, F, \mu, U, N, D \rangle$ be a two-way alternating k-PA. We show how to simulate $\mathcal{A}$ with a one-way deterministic k-PA $\mathcal{A}'$. We start by normalizing the behavior of $\mathcal{A}$ as follows.

1. On input word $\triangleleft w \triangleright$, $\mathcal{A}$ starts the computation with pebble k on the right-end marker $\triangleright$.

2. The state Q is partitioned into $Q_1 \cup \cdots \cup Q_k$, where Q_i is the set of states when pebble i is the head pebble.
   Similarly, we denote by U_i, N_i and D_i the set of universal, nondeterministic and deterministic states, respectively and μ_i the set of transitions when pebble i is the head pebble.

3. Each Q_i is further partitioned into $Q_{i,\mathtt{stay}} \cup Q_{i,\mathtt{right}} \cup Q_{i,\mathtt{left}} \cup Q_{i,\mathtt{place}} \cup Q_{i,\mathtt{lift}}$, where
   - if $(i, \sigma, P, V, q) \to (p, \mathtt{stay})$, then $q \in Q_{i,\mathtt{stay}}$;
   - if $(i, \sigma, P, V, q) \to (p, \mathtt{right})$, then $q \in Q_{i,\mathtt{right}}$;
   - if $(i, \sigma, P, V, q) \to (p, \mathtt{left})$, then $q \in Q_{i,\mathtt{left}}$;
   - if $(i, \sigma, P, V, q) \to (p, \mathtt{place\text{-}pebble})$, then $q \in Q_{i,\mathtt{place}}$; and
   - if $(i, \sigma, P, V, q) \to (p, \mathtt{lift\text{-}pebble})$, then $q \in Q_{i,\mathtt{lift}}$.

4. The automaton can only do universal and existential branching while the head pebble is stationery.
   That is, $(i, \sigma, P, V, q) \to (p, \mathtt{act})$ and $q \in U \cup N$, then $\mathtt{act} = \mathtt{stay}$.

5. The automaton places the new pebble on the right-end marker $\triangleright$.

6. The automaton lifts the pebble only when it is on the right-end marker $\triangleright$.

7. When the head pebble is reading the left-end and the right-end markers $\triangleleft$ and $\triangleright$, the automaton does not place new pebble.

8. Only pebble k can enter the final states and it does so only after it reads the right-end marker $\triangleright$.

We will need the following notions. A pebble-i assignment θ is a pebble assignment when the pebbles $i, i+1, \ldots, k$ are on the input word. That is, the domain of θ is $\{i, i+1, \ldots, k\}$.

Let θ be a pebble-i assignment on an input word $w = \binom{\sigma_1}{a_1} \cdots \binom{\sigma_n}{a_n}$. We define $\mathsf{Succ}(\theta) = \theta'$ as follows.

- If $\theta(i) \leq n$, then θ' is a pebble-i assignment, where for each $j = i, i+1, \ldots, k$,
$$\theta'(j) = \begin{cases} \theta(j), & \text{if } j = i+1, \ldots, k, \\ \theta(j)+1, & \text{if } j = i. \end{cases}$$
- If $\theta(i) = n+1$, then θ' is pebble-$(i+1)$ assignment such that for each $j = i+1, \ldots, k$, $\theta'(j) = \theta(j)$.

Similarly, for a pebble-i assignment θ, we can define $\mathsf{Pred}(\theta)$ as follows.

- If $1 \leq \theta(i)$, then θ' is a pebble-i assignment, where for each $j = i, i+1, \ldots, k$,
$$\theta'(j) = \begin{cases} \theta(j), & \text{if } j = i+1, \ldots, k, \\ \theta(j) - 1, & \text{if } j = i. \end{cases}$$

- If $\theta(i) = 0$, then θ' is pebble-$(i+1)$ assignment such that for each $j = i+1, \ldots, k$, $\theta'(j) = \theta(j)$.

In the following subsections we present the determinization of $\mathcal{A}$, starting from pebble 1 and finishing with pebble k, in the following subsections. We will denote by $\mathcal{A}^{(i)}$ the equivalent automaton of $\mathcal{A}$, where the behavior of pebbles $1, \ldots, i$ are one-way and deterministic. By this notation, $\mathcal{A}^{(k)}$ is the equivalent one-way, deterministic version of $\mathcal{A}$.

4.1 Determinizing pebble 1

The determinization follows closely the one described in [5, Section 4]. For completeness, we present it here. The end result of the determinization is such that pebble 1 is placed in the left-end marker $\triangleleft$ and lifted when it reaches the right-end marker $\triangleright$.

We need a few notations. Some of them are repetitions of those that have been introduced in Section 2. For each $q \in Q$, we define a new symbol $\bar{q}$. We denote by $\bar{Q} = \{\bar{q} : q \in Q\}$. If $A \subseteq Q$, then $\bar{A} = \{\bar{p} : p \in A\}$. We define a *term* to be an object of the form $q \to A$ where $q \in Q$ and $A \subseteq Q \cup \bar{Q}$. A term $q \to A$ is *closed*, if $A \subseteq \bar{Q}$. A *partial response* is a set of terms, while a *response* is a set of closed terms.

Let $w = \binom{\sigma_1}{a_1} \cdots \binom{\sigma_n}{a_n}$ be a data word and θ be a pebble-1 assignment. The determinization of pebble 1 depends on the following three concepts: *response* $\mathcal{R}(w, \theta)$, *partial response* $\mathcal{PR}(w, \theta)$ and the *proof system* $\mathcal{S}(\mathcal{R}, \sigma, P, V)$. We will define these concepts one by one starting with the response $\mathcal{R}(w, \theta)$.

The response $\mathcal{R}(w, \theta)$ is defined as follows. For a set $S \subseteq Q$, a closed term $q \to \bar{S}$ belongs to $\mathcal{R}(w, \theta)$ if there exists a computation tree $\mathcal{T}$ of $\mathcal{A}$ on w whose root is labelled with $[1, q, \theta]$ such that

- if $\theta(1) \leq n$, then each leaf is labelled with $[1, p, \mathsf{Succ}(\theta)]$ for some $p \in S$;
- if $\theta(1) = n+1$, then each leaf is labelled with $[2, p, \mathsf{Succ}(\theta)]$ for some $p \in S$;

- each internal node in the computation tree $\mathcal{T}$ is labelled with $[1, q', \theta']$, where $q' \in Q$ and $0 \leq \theta'(1) \leq \theta(1)$; and
- for each $p \in S$, there exists a leaf labelled with $[1, p, \mathsf{Succ}(\theta)]$.

Remark 2 Let w_1, w_2 be data words. Let θ_1 and θ_2 be pebble-1 assignments on $\triangleleft w_1 \triangleright$ and $\triangleleft w_2 \triangleright$, respectively, such that $\theta_1(1) = \theta_2(1) = 0$. That is, on both assignments pebble 1 is reading the left-end marker $\triangleleft$. Then, $\mathcal{R}(w_1, \theta_1) = \mathcal{R}(w_2, \theta_2)$.

Now we define the partial response $\mathcal{PR}(w, \theta)$ as follows. For a set $S \subseteq Q \cup \bar{Q}$, a term $q \to S$ belongs to $\mathcal{PR}(w, \theta)$ if there exists a computation tree $\mathcal{T}$ of $\mathcal{A}$ on w whose root is labelled with $[1, q, \theta]$ such that

- if $\theta(1) \leq n$, then each leaf is labelled with either $[1, p, \mathsf{Succ}(\theta)]$ for some $\bar{p} \in S$ or $[1, p, \theta]$ for some $p \in S$;
- if $\theta(1) = n + 1$, each leaf is labelled with either $[2, p, \mathsf{Succ}(\theta)]$ for some $\bar{p} \in S$ or $[1, p, \theta]$ for some $p \in S$;
- each internal node in the computation tree $\mathcal{T}$ is labelled with $[1, q', \theta']$, where $q' \in Q_1$ and $0 \leq \theta'(1) \leq \theta(1)$;
- if $\theta(1) \leq n$, for each $\bar{p} \in S$, there exists a leaf labelled with $[1, p, \mathsf{Succ}(\theta)]$;
- if $\theta(1) = n + 1$, for each $\bar{p} \in S$, there exists a leaf labelled with $[2, p, \mathsf{Succ}(\theta)]$; and
- for each $p \in S$, there exists a leaf labelled with $[1, p, \theta]$.

We call the tree $\mathcal{T}$ a *witness* for $q \to S \in \mathcal{PR}(w, \theta)$.

We define a *proof system* for $\mathcal{S}(\mathcal{R}, \sigma, P, V)$, where $\sigma \in \Sigma$, $P, V \subseteq \{2, \ldots, k\}$ and a response $\mathcal{R}$, as follows.

1. $$\frac{}{q \to \{q\}}$$

2. $$\frac{q \to B \cup \{p\}, p \to C}{q \to B \cup C}$$

3. $$\frac{(1, \sigma, P, V, q) \to (p_i, \mathtt{stay}) \in \mu_1 \text{ for each } i = 1, \ldots, m \text{ and } q \in U}{q \to \{p_1, \ldots, p_m\}}$$

4. $$\frac{(1,\sigma,P,V,q) \to (p,\texttt{stay}) \in \mu_1 \text{ and } p \notin U}{q \to \{p\}}$$

5. $$\frac{(1,\sigma,P,V,q) \to (p,\texttt{right}) \in \mu_1}{q \to \{\bar{p}\}}$$

6. $$\frac{(1,\sigma,P,V,q) \to (p,\texttt{left}) \in \mu_1 \text{ and } p \to \bar{S} \in \mathcal{R} \text{ and } S \subseteq Q_1}{q \to S}$$

7. $$\frac{(1,\sigma,P,V,q) \to (p,\texttt{lift-pebble}) \text{ if } \sigma = \triangleright \text{ and } P,V = \emptyset}{q \to \{\bar{p}\}}$$

We denote by $\mathrm{TH}(\mathcal{R},\sigma,P,V)$ be the set of terms "provable" using the proof system $\mathcal{S}(\mathcal{R},\sigma,P,V)$.

The following claim is the pebble 1 counter part of a similar claim in [5, pp. 149].

Claim 1 *For every word* $w = \binom{\sigma_1}{a_1}\cdots\binom{\sigma_n}{a_n}$ *and pebble-*1 *assignment* θ *on* $\triangleleft w \triangleright$*,*

$$\mathcal{PR}(w,\theta) = TH(\mathcal{R}(w,\mathsf{Pred}(\theta)),\sigma,P,V),$$

where

- $1 \le \theta(1) \le n+1$*;*
- $P = \{l : \theta(l) = \theta(1)\}$*;*
- $V = \{l : a_{\theta(l)} = a_{\theta(1)}\}$*;*
- $\sigma = \sigma_{\theta(1)}$*.*

Proof. The proof follows closely the similar proof in [5]. First, we show that $\mathcal{PR}(w,\theta) \subseteq \mathrm{TH}(\mathcal{R}(w,\mathsf{Pred}(\theta)),\sigma,P,V)$ inductively on the size of witnesses for terms in $\mathcal{PR}(w,\theta)$. Let $q \to S \in \mathcal{PR}(w,\theta)$. The basis is when the witness for $q \to S \in \mathcal{PR}(w,\theta)$ consists of a single node with the label $[1,q,\theta]$. Then, $S = \{q\}$ and $q \to \{q\}$ is provable using rule 1.

For the induction step, suppose $q \to S \in \mathcal{PR}(w,\theta)$ is witnessed by a tree $\mathcal{T}$ with more than one node. There are five cases to consider:

1. The state q is a universal state, that is, $q \in U_1$. Let

$$\begin{array}{rcl} (1,\sigma,P,V,q) \to (p_1, \mathtt{stay}) & \in & \mu_1; \\ & \vdots & \\ (1,\sigma,P,V,q) \to (p_m, \mathtt{stay}) & \in & \mu_1; \end{array}$$

   In this case, the root of $\mathcal{T}$ is labelled with $[1, q, \theta]$ and its immediate children $\pi_1, \ldots, \pi_m$ are labelled with $[1, p_1, \theta], \ldots, [1, p_m, \theta]$, respectively. The complete subtree rooted at π_i witnesses $p_i \to S_i \in \mathcal{PR}(w, \theta)$, where S_i is the set of states in the labels of the leafs in the subtree. Furthermore, $S_1 \cup \cdots \cup S_m = S$. By the induction hypothesis, $p_i \to S_i \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$. Combining rules 2 and 3, we obtain $q \to S \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$.

2. The state q is a nondeterministic state, that is, $q \in N_1$. Let

$$\begin{array}{rcl} (1,\sigma,P,V,q) \to (p_1, \mathtt{stay}) & \in & \mu_1; \\ & \vdots & \\ (1,\sigma,P,V,q) \to (p_m, \mathtt{stay}) & \in & \mu_1; \end{array}$$

   Or, if q is a deterministic state, i.e. $q \in D_1$, then $m = 1$. This case is just like case 1 above, except that we use rules 4 and 2.

3. $(1, \sigma, P, V, q) \to (p, \mathtt{right}) \in \mu_1$. In this case, $S = \{\bar{p}\}$. By rule 5, we have $q \to \{\bar{p}\} \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$.

4. $(1, \sigma, P, V, q) \to (p, \mathtt{lift\text{-}pebble}) \in \mu_1$, where $\sigma = \triangleright$, $P, V = \emptyset$. In this case, $S = \{\bar{p}\}$. By rule 7, we have $q \to \{\bar{p}\} \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$.

5. $(1, \sigma, P, V, q) \to (p, \mathtt{left}) \in \mu_1$. The child π of the root of $\mathcal{T}$ has the label $[1, p, \mathsf{Pred}(\theta)]$. Every path from π to a leaf of $\mathcal{T}$ must pass through a node with label of the form $[1, r, \theta]$. That is, pebble 1 must return to the position $\theta(1)$ again.

   Let $\Lambda = \{\rho_1, \ldots, \rho_l\}$ be the descendants of π with the properties

   (a) each ρ_i is labelled with $[1, r_i, \theta]$,
   (b) no node between π and ρ_i has a label with the third coordinate θ,
   (c) every path from π to a leaf passes through a node in Λ.

Let $\mathcal{T}'$ be the unique subtree of $\mathcal{T}$ whose root is π and whose set of leaves is Λ. Then, $\mathcal{T}'$ is a witness of $p \to \{\bar{r}_1, \ldots, \bar{r}_l\} \in \mathcal{PR}(w, \mathsf{Pred}(\theta))$. Since this is a closed term, then $p \to \{\bar{r}_1, \ldots, \bar{r}_l\} \in \mathcal{R}(w, \mathsf{Pred}(\theta))$. By rule 6, $q \to \{r_1, \ldots, r_l\} \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$. The complete subtree of $\mathcal{T}$ rooted at ρ_i witnesses $r_i \to S_i \in \mathcal{PR}(w, \theta)$, where S_i is the set of states in the labels of the leafs in the subtree. By the induction hypothesis, $r_i \to S_i \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$. Applying rule 2, we obtain $q \to \bigcup_{1 \le i \le l} S_i \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$. Since $\bigcup_{1 \le i \le l} S_i = S$, this case follows.

Now we prove that $\mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V) \subseteq \mathcal{PR}(w, \theta)$ by induction on the proof length. Suppose $q \to S \in \mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V)$ has a proof length ≥ 1.

- If the last step of the proof (from which $q \to S$ is concluded) is an application of rules 1, 3, 4, 5, or 7, then it is immediate that there is a computation tree that witnesses $q \to S \in \mathcal{PR}(w, \theta)$.
- If the last step of the proof is an application of rule 2, then suppose $q \to A \cup \{p\}$ and $p \to B$ are the antecedents from which $q \to A \cup B$ is concluded ($S = A \cup B$). By the induction hypothesis, there are computation trees $\mathcal{T}$ and $\mathcal{T}'$ which witness $q \to A \cup \{p\}$ and $p \to B$, respectively. If each leaf of $\mathcal{T}$ labelled with $[1, p, \theta]$ is replaced with the tree $\mathcal{T}'$ (whose root is labelled with $[1, p, \theta]$), then the resulting tree witnesses $q \to A \cup B \in \mathcal{PR}(w, \theta)$.
- If the last step of the proof is an application of rule 6, then suppose that $q \to A$ is concluded from

  $$(1, \sigma, P, V, q) \to (p, \texttt{left}) \text{ and } p \to \bar{S} \in \mathcal{R}(w, \mathsf{Pred}(\theta)) \text{ and } S \subseteq Q_1.$$

  Since $p \to \bar{S} \in \mathcal{R}(w, \mathsf{Pred}(\theta))$, then there exists a computation tree $\mathcal{T}'$ such that

  - the root of $\mathcal{T}'$ is labelled with $[1, p, \mathsf{Pred}(\theta)]$;
  - the leaf of $\mathcal{T}'$ is labelled with $[1, r, \theta]$ for some $r \in S$;
  - for each $r \in S$, there is a leaf of $\mathcal{T}'$ labelled with $[1, r, \theta]$.

  Now we can construct a tree $\mathcal{T}$ such that

– the root of $\mathcal{T}$ is labelled with $[1, q, \theta]$;

– the root has only one immediate child π labelled with $[1, p, \mathsf{Pred}(\theta)]$;

– the subtree rooted at π is the tree $\mathcal{T}'$.

The tree $\mathcal{T}$ is a witness of the term $q \to S \in \mathcal{PR}(w, \theta)$.

This completes the proof of the claim. □

We denote by $\mathrm{CTH}(\mathcal{R}, \sigma, P, V)$ the set of closed terms in $\mathrm{TH}(\mathcal{R}, \sigma, P, V)$. Since, by Claim 1, $\mathrm{TH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V) = \mathcal{PR}(w, \theta)$, thus,

$$\mathrm{CTH}(\mathcal{R}(w, \mathsf{Pred}(\theta)), \sigma, P, V) = \mathcal{R}(w, \theta).$$

The determinization of μ_1 is done precisely by means of this equation. Loosely speaking, the set of "states" of the deterministic version of μ_1 are roughly the set of responses $\mathcal{R}(w, \theta)$. There are only finitely many such responses. From the "state" $\mathcal{R}(w, \mathsf{Pred}(\theta))$, if pebble 1 reads the "input" σ, P, V, then it deterministically moves right and enters the state $(\mathcal{R}, \theta)$.

In the following paragraphs we will describe this idea more precisely. But before we do that, we need to make a bit of modification on the behavior of pebble 2.

Let $\tilde{Q}_2$, $\tilde{\mu}_2$, $\tilde{U}_2$, $\tilde{N}_2$, $\tilde{D}_2$ be the modification of Q_2, μ_2, U_2, N_2 ,D_2, respectively, as follows. For a set B, we write 2^B to denote the power set of B.

- $\tilde{Q}_2 = Q_2 \cup 2^{Q_2} \cup 2^{2^{Q_2}}$;
- $\tilde{U}_2 = U_2 \cup (2^{Q_2} - \{\emptyset\})$;
- $\tilde{N}_2 = N_2 \cup 2^{2^{Q_2}}$;
- $\tilde{D}_2 = D_2$.

The set of transitions $\tilde{\mu}_2$ is the set μ_2 plus the following transitions.

1. For every $\sigma \in \Sigma$, $P, V \subseteq \{3, \ldots, k\}$, $S_1, \ldots, S_m \subseteq Q_2$,

   $$(2, \sigma, P, V, \{S_1, \ldots, S_m\}) \to (S_i, \mathtt{stay}) \in \tilde{\mu}_2, \text{ for each } i = 1, 2, \ldots, m.$$

   That is, from the state $\{S_1, \ldots, S_m\} \in \tilde{Q}_2$ pebble 2 performs existential branching.
   Recall that the state $\{S_1, \ldots, S_m\} \in \tilde{Q}_2$ is a nondeterministic state.

2. For every $\sigma \in \Sigma$, $P, V \subseteq \{3, \ldots, k\}$, $S \subseteq Q_2$, we have the following transition in $\tilde{\mu}_2$.

$$(2, \sigma, P, V, S) \to (q, \texttt{stay}) \in \tilde{\mu}_2, \text{ for each } q \in S.$$

   That is, from the state $S \subseteq Q_2$ pebble 2 performs universal branching. Recall that the state $S \in \tilde{Q}_2$ is a universal state.

3. We replace each transition $(2, \sigma, P, V, q) \to (p, \texttt{place-pebble}) \in \mu_2$ with the following transition

$$(2, \sigma, P, V, q) \to ((p, \emptyset), \texttt{place-pebble}) \in \tilde{\mu}_2.$$

   In other words, $\tilde{\mu}_2$ no longer contains the transition $(2, \sigma, P, V, q) \to (p, \texttt{place-pebble})$. Rather, it contains the transition $(2, \sigma, P, V, q) \to ((p, \emptyset), \texttt{place-pebble})$.

All other transitions in μ_2 remain in $\tilde{\mu}_2$.

Now we define the sets of states Q'_1 and the set of transitions μ'_1 for deterministic pebble 1. We use the "prime" sign, as in μ'_1, to indicate that the behavior of pebble 1 (as described by μ'_1) is deterministic. On the other hand, the "tilde" sign, as in $\tilde{\mu}_2$, is used to indicate that the behavior of pebble 2 (as described by $\tilde{\mu}_2$) is still alternating.

- Q'_1 consists of elements of the form $(q, \mathcal{R})$, where $q \in Q_1$ and $\mathcal{R}$ is a response;
- μ'_1 consists of the following transitions. For each $q \in Q_1$,
  1. $(1, \triangleleft, \emptyset, \emptyset, (q, \emptyset)) \to ((q, \mathcal{R}), \texttt{right}) \in \mu'_1$, where $\mathcal{R} = \mathcal{R}(w, \theta)$, for some w and θ such that $\theta(1) = 0$.
     By Remark 2, such $\mathcal{R}(w, \theta)$ is well defined.
  2. For every response $\mathcal{R}$, label $\sigma \in \Sigma$ and $P, V \subseteq \{2, \ldots, k\}$,

$$(1, \sigma, P, V, (q, \mathcal{R})) \to ((q, \mathrm{CTH}(\mathcal{R}, \sigma, P, V)), \texttt{right}) \in \mu'_1.$$

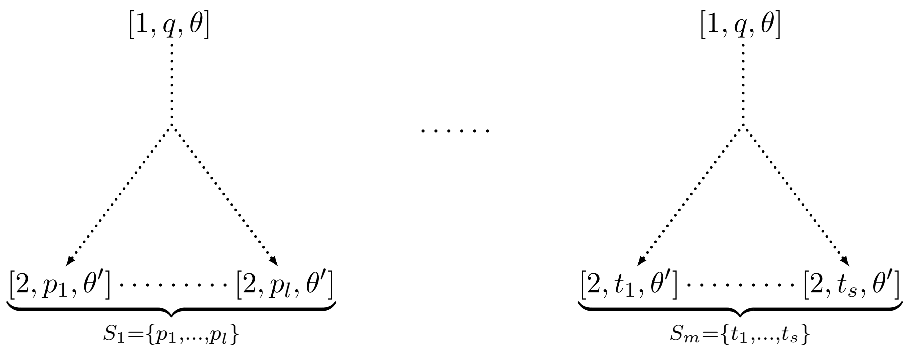
  3. $(1, \triangleright, \emptyset, \emptyset, (q, \mathcal{R})) \to (\{S_1, \ldots, S_m\}, \texttt{lift-pebble})$, where for each $j = 1, \ldots, m$,
     - $q \to \bar{S}_j \in \mathrm{CTH}(\mathcal{R}, \triangleright, \emptyset, \emptyset)$;
     - $S_j \subseteq Q_2$.

Intuitively transitions in item 3 of μ_1' mean the following. Let $\mathcal{R} = \mathcal{R}(w, \theta)$ and θ is pebble-1 assignment, where $\theta(1) = n+1$ and n is the length of w. Let θ' is pebble-2 assignment such that for $i = 2, \ldots, k$, $\theta(i) = \theta'(i)$.

That the closed term $q \to \bar{S}_j$ belongs to $\mathrm{CTH}(\mathcal{R}, \triangleright, \emptyset, \emptyset)$ means that there exists a computation tree $\mathcal{T}$ such that

- the root is labelled with the configuration $[1, q, \theta]$;
- all the non leaf nodes are labelled with 1-configurations, that is, configurations where the head pebble is pebble 1;
- all the leaf is labelled with the configuration $[1, p, \theta']$, for some $p \in S_j$;
- for each $p \in S_j$, there exists a leaf with the configuration $[2, p, \theta']$.

Since $\mathrm{CTH}(\mathcal{R}, \triangleright, \emptyset, \emptyset)$ contains the closed terms $q \to \bar{S}_1, \ldots, q \to \bar{S}_m$, it means that there are only m possible "choices" of sets of states once pebble 1 is lifted, that is, $S_1, \ldots, S_m$. See picture below.



So, once we have deterministically simulated pebble 1, we have to indicate to the automaton that there are m possible "choices" of sets of states for pebble 2, hence, the state $\{S_1, \ldots, S_m\} \in 2^{2^{Q_2}}$. From this state the automaton nondeterministically chooses which set of states pebble 2 enters. Suppose it chooses the set S_j. Then, from S_j the automaton branches conjunctively into each state in S_j. See picture below.

$$
\begin{tikzcd}
 & {[1,(q,\emptyset),\theta_0]} \arrow[d] & & & & {[1,(q,\emptyset),\theta_0]} \arrow[d] & \\
 & {[2,\{S_1,\ldots,S_m\},\theta']} \arrow[d] & & & & {[2,\{S_1,\ldots,S_m\},\theta']} \arrow[d] & \\
 & {[2,S_1,\theta']} \arrow[dl] \arrow[dr] & & \cdots\cdots & & {[2,S_m,\theta']} \arrow[dl] \arrow[dr] & \\
{[2,p_1,\theta']} & \cdots & {[2,p_l,\theta']} & & {[2,t_1,\theta']} & \cdots & {[2,t_s,\theta']} \\
 & \underbrace{\qquad\qquad\qquad\qquad}_{S_1=\{p_1,\ldots,p_l\}} & & & & \underbrace{\qquad\qquad\qquad\qquad}_{S_m=\{t_1,\ldots,t_s\}} &
\end{tikzcd}
$$

We now show that $\mu_1 \cup \mu_2$ and $\mu_1' \cup \tilde{\mu}_2$ are "equivalent." Recall that for a subset $X \subseteq \mu$, recall that $\gamma \vdash_X \gamma'$ denotes that the relation $\gamma \vdash \gamma'$ is obtained by means of a transition in X.

Let $w = \binom{\sigma_1}{a_1} \cdots \binom{\sigma_n}{a_n}$ be a data word and θ be a pebble-2 assignment on $\triangleleft w \triangleright$. For each $i = 0, \ldots, n+1$, we also denote by θ_i a pebble-1 assignment such that

$$\theta_i(j) = \begin{cases} \theta(j), & \text{if } j = 2, \ldots, k, \\ i, & \text{if } j = 1. \end{cases}$$

First, we show that transitions in μ_1 and μ_2 can be "correctly" simulated by transitions in μ_1' and $\tilde{\mu}_2$. Suppose

$$[2, p_1, \theta] \vdash_{\mu_2} [1, p_2, \theta_{n+1}] \vdash_{\mu_1}^* [1, p_3, \theta_{n+1}] \vdash_{\mu_1} [2, p_4, \theta].$$

Thus, this means that there exists a closed term $p_2 \to \bar{S} \in \mathcal{R}(w, \theta_{n+1})$ such that $S \subseteq Q_2$ and $p_4 \in S$.

Now we are going to show that there exists a "deterministic" run by means of the transitions in μ_1' and $\tilde{\mu}_2$ from the configuration $[2, p_1, \theta]$ to the configuration $[2, p_4, \theta]$.

By the construction of $\tilde{\mu}_2$, we have

$$[2, p_1, \theta] \vdash_{\tilde{\mu}_2} [1, (p_2, \emptyset), \theta_0]. \tag{1}$$

Then, by the construction of μ'_1,

$$[1,(p_2,\emptyset),\theta_0] \ \vdash_{\mu'_1} \ [1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_0)),\theta_1]; \tag{2}$$

Furthermore, applying Claim 1 repeatedly, we obtain

$$\begin{array}{rcl}
[1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_0)),\theta_1] & \vdash_{\mu'_1} & [1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_1)),\theta_2] \\
[1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_1)),\theta_2] & \vdash_{\mu'_1} & [1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_2)),\theta_3] \\
 & \vdots & \\
[1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_{n-1})),\theta_n] & \vdash_{\mu'_1} & [1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_n)),\theta_{n+1}]
\end{array}$$

Thus, we obtain

$$[1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_0)),\theta_1] \ \vdash^*_{\mu'_1} \ [1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_n)),\theta_{n+1}] \tag{3}$$

Again, by the construction of μ'_1, we have

$$[1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_n)),\theta_{n+1}] \ \vdash_{\mu'_1} \ [2,\{S_1,\ldots,S_m\},\theta], \tag{4}$$

where for each $j=1,\ldots,m$, $p_2 \to S_j \in \mathrm{CTH}(\triangleleft w \triangleright,\theta_{n+1})$.

Suppose that $S_1 = S$. Again, by the construction of $\tilde{\mu}_2$, we have

$$[2,\{S_1,\ldots,S_m\},\theta] \vdash_{\tilde{\mu}_2} [2,S_1,\theta]. \tag{5}$$

and since $p_4 \in S$,

$$[2,S_1,\theta] \vdash_{\tilde{\mu}_2} [2,p_4,\theta]. \tag{6}$$

Now, combining Equations (1)–(6), we obtain the run

1. $[2,p_1,\theta] \vdash_{\tilde{\mu}_2} [1,(p_2,\emptyset),\theta_0]$;
2. $[1,(p_2,\emptyset),\theta_0] \vdash^*_{\mu'_1} [1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_n)),\theta_{n+1}]$;
3. $[1,(p_2,\mathcal{R}(\triangleleft w \triangleright,\theta_n)),\theta_{n+1}] \vdash_{\mu'_1} [2,\{S_1,\ldots,S_m\},\theta]$;
4. $[2,\{S_1,\ldots,S_m\},\theta] \vdash_{\tilde{\mu}_2} [2,S_1,\theta]$;
5. $[2,S_1,\theta] \vdash_{\tilde{\mu}_2} [2,p_4,\theta]$.

Vice versa, now we show that transitions in μ'_1 and $\tilde{\mu}_2$ can be "correctly" simulated by transitions in μ_1 and μ_2. Suppose we have the following relations:

1. $[2, q, \theta] \vdash_{\tilde{\mu}_2} [1, (p, \emptyset), \theta_0]$;
2. $[1, (p, \emptyset), \theta_0] \vdash_{\mu_1'} [1, (p, \mathcal{R}(\triangleleft w \triangleright, \theta_0)), \theta_1]$;
3. $[1, (p, \mathcal{R}(\triangleleft w \triangleright, \theta_0)), \theta_1] \vdash_{\mu_1'} \cdots \vdash_{\mu_1'} [1, (p, \mathcal{R}(\triangleleft w \triangleright, \theta_n)), \theta_{n+1}]$;
4. $[1, (p, \mathcal{R}(\triangleleft w \triangleright, \theta_n)), \theta_{n+1}] \vdash_{\mu_1'} [2, \{S_1, \ldots, S_m\}, \theta]$;
5. $[2, \{S_1, \ldots, S_m\}, \theta] \vdash_{\tilde{\mu}_2} [2, S_i, \theta]$;
6. $[2, S_i, \theta] \vdash_{\tilde{\mu}_2} [2, s, \theta]$, for each $s \in S_i$.

Now, from the construction of $\tilde{\mu}_2$, Relation (1) implies that the relation below holds.

$$[2, q, \theta] \vdash_{\mu_2} [1, p, \theta_{n+1}].$$

From the construction of μ_1' and Claim 1, Relations (2)–(4) implies that

$$p \to \bar{S}_i \in \mathcal{R}(\triangleleft w \triangleright, \theta_{n+1}), \text{ where } S_i \subseteq Q_2.$$

This means that for each $s \in S_i$,

$$[1, p, \theta_{n+1}] \vdash^*_{\mu_1} [2, s, \theta], \text{ where } s \in S_i.$$

This completes the proof that $\mu_1 \cup \mu_2$ are "equivalent" to $\mu_1' \cup \tilde{\mu}_2$.

4.2 Determinizing pebble i

Now, assuming that the behavior of pebbles $1, \ldots, i-1$ are one-way and deterministic, we will determinize pebble i. The end result of the determinization is such that pebble i is placed in the left-end marker $\triangleleft$ and lifted when it reaches the right-end marker $\triangleright$.

The idea is very similar to the one in Subsection 4.1, with the exception that now during the computation pebble i can place pebble $(i-1)$. The effect of such placement is the state of pebble i changes. Figure 1 below is an example of a sequence of moves of pebble 2 of a two pebble automaton $\mathcal{A}$ on $\binom{\sigma_1}{a_1}\binom{\sigma_2}{a_2}\binom{\sigma_3}{a_3}\binom{\sigma_4}{a_4}$. Recall by our normalization of $\mathcal{A}$ in Section 4, the computation starts with pebble 2 above the right-end marker $\triangleright$. We assume that the behavior of pebble 1 is already determinized in the manner explained in the previous subsection.

For example, the pair (q_2, q_2') in the run of pebble 1 indicates that pebble 2 first arrives at the symbol $\binom{\sigma_3}{a_3}$ when pebble 2 is in the state q_2, upon which

	$\lhd$		$\binom{\sigma_1}{a_1}$		$\binom{\sigma_2}{a_2}$		$\binom{\sigma_3}{a_3}$		$\binom{\sigma_4}{a_4}$		$\rhd$		
pebble 2	q_5	$\leftarrow$	q_4	$\leftarrow$	q_3	$\leftarrow$	(q_2, q_2')	$\leftarrow$	q_1	$\leftarrow$	q_0		
	$\hookrightarrow$	$\rightarrow$	q_6	$\rightarrow$	q_7	$\rightarrow$	q_8	$\rightarrow$	q_9	$\rightarrow$	q_{10}	$\rightarrow$	q_f
pebble 1 (q_2, q_2')	q_{11}	$\rightarrow$	q_{12}	$\rightarrow$	q_{13}	$\rightarrow$	q_{14}	$\rightarrow$	q_{15}	$\rightarrow$	q_{16}		

Figure 1: A sequence of moves of $\mathcal{A}$ on $\binom{\sigma_1}{a_1}\binom{\sigma_2}{a_2}\binom{\sigma_3}{a_3}\binom{\sigma_4}{a_4}$.

pebble 1 is placed. When pebble 1 has finally finished its computation, that is, when it is lifted after reading the right-end marker $\rhd$, $\mathcal{A}$ enters the state q_2' from which pebble 2 continues the computation. This pair (q_2, q_2') can be viewed as a term $q_2 \rightarrow \{q_2'\}$ and has to be included as an "axiom" in the proof system $\mathrm{TH}(\mathcal{R}, \sigma, \emptyset, \emptyset)$. This will be made more precise in the next paragraphs.

Let $Q_1, \ldots, Q_{i-1}$ be the set of states of pebbles $1, \ldots, (i-1)$, respectively, and $\mu_1, \ldots, \mu_{i-1}$ be the set of transitions of pebbles $1, \ldots, (i-1)$, respectively. We assume that the behavior of pebbles $1, \ldots, i-1$, according to $\mu_1, \ldots, \mu_{i-1}$, is deterministic.

Let $w = \binom{\sigma_1}{a_1} \cdots \binom{\sigma_n}{a_n}$ and θ be a pebble-i assignment on w. We define a set of terms $\wp(\mu_i, w, \theta)$ as follows. For $p, q \in Q_i$, the term $p \rightarrow \{q\} \in \wp(\mu_i, w, \theta)$ if and only if there exists $s_1, s_2 \in Q_{i-1}$ such that

1. $(i, \sigma_{\theta(i)}, P, V, p) \rightarrow (s_1, \texttt{place-pebble}) \in \mu_i$, where
   - $P = \{l > i : \theta(l) = \theta(i)\}$;
   - $V = \{l > i : a_{\theta(l)} = a_{\theta(i)}\}$.
2. $[i-1, s_1, \theta_0] \vdash^* [i-1, s_2, \theta_{n+1}]$ is an $(i-1)$-run, where $\theta_0(i-1) = 0$, $\theta_{n+1}(i-1) = n+1$ and $\theta_0(j) = \theta_{n+1}(j) = \theta(j)$, for all $j = i, \ldots, k$.
3. $(i, \rhd, \emptyset, \emptyset, s_2) \rightarrow (q, \texttt{lift-pebble}) \in \mu_{i-1}$.

Since pebbles $1, \ldots, (i-1)$ all behave deterministically, for each $p \in Q_{i,\texttt{place}}$, there exists exactly one $q \in Q_i$ such that the term $p \rightarrow \{q\} \in \wp(\mathcal{A}_i, w, \theta)$.

For a pebble-i assignment θ, we define the response $\mathcal{R}(w, \theta)$ as follows. For a set $S \subseteq Q$, a closed term $q \rightarrow \bar{S}$ belongs to $\mathcal{R}(w, \theta)$ if there exists a computation tree $\mathcal{T}$ of $\mathcal{A}$ on w whose root is labelled with $[i, q, \theta]$ such that

- if $\theta(i) \leq n$, then each leaf is labelled with $[i, p, \mathsf{Succ}(\theta)]$ for some $p \in S$;
- if $\theta(i) = n+1$, then each leaf is labelled with $[i+1, p, \mathsf{Succ}(\theta)]$ for some $p \in S$;
- each internal node in $\mathcal{T}$ is labelled with $[j, q', \theta']$, where
  1. $j \leq i$; and
  2. if $j = i$, then $0 \leq \theta'(i) \leq \theta(i)$.
- for each $p \in S$, there exists a leaf labelled with $[1, p, \mathsf{Succ}(\theta)]$.

Similarly, we define the partial response $\mathcal{PR}(w, \theta)$ as follows. For a set $S \subseteq Q \cup \bar{Q}$, a term $q \to S$ belongs to $\mathcal{PR}(w, \theta)$ if there exists a computation tree $\mathcal{T}$ of $\mathcal{A}$ on w whose root is labelled with $[i, q, \theta]$ such that

- if $\theta(i) \leq n$, then each leaf is labelled with either $[i, p, \mathsf{Succ}(\theta)]$ for some $\bar{p} \in S$ or $[i, p, \theta]$ for some $p \in S$;
- if $\theta(i) = n + 1$, each leaf is labelled with either $[i + 1, p, \mathsf{Succ}(\theta)]$ for some $\bar{p} \in S$ or $[i, p, \theta]$ for some $p \in S$;
- each internal node in $\mathcal{T}$ is labelled with $[j, q', \theta']$, where
  1. $j \leq i$; and
  2. if $j = i$, then $0 \leq \theta'(i) \leq \theta(i)$;
- if $\theta(i) \leq n$, for each $\bar{p} \in S$, there exists a leaf labelled with $[i, p, \mathsf{Succ}(\theta)]$;
- if $\theta(i) = n + 1$, for each $\bar{p} \in S$, there exists a leaf labelled with $[i + 1, p, \mathsf{Succ}(\theta)]$; and
- for each $p \in S$, there exists a leaf labelled with $[i, p, \theta]$.

The following claim is the generalization of Claim 1 and the proof is similar, thus, omitted.

Claim 2 *For every word $w = \binom{\sigma_1}{a_1} \cdots \binom{\sigma_n}{a_n}$ and pebble-i assignment θ on $\triangleleft w \triangleright$,*

$$\mathcal{PR}(w, \theta) = TH(\mathcal{P}, \sigma, P, V),$$

where

- $\mathcal{P} = \mathcal{R}(w, \textsf{\textit{Pred}}(\theta)) \cup \wp(\mu_i, w, \theta)$;

- $1 \leq \theta(i) \leq n + 1$;

- $P = \{l > i : \theta(l) = \theta(i)\}$;

- $V = \{l > i : a_{\theta(l)} = a_{\theta(i)}\}$;

- $\sigma = \sigma_{\theta(i)}$.

We will describe intuitively how to simulate pebble i deterministically in the following paragraph. The "main" states of pebble i will still be of the form $(q, \mathcal{R})$, where $q \in Q_i$ and $\mathcal{R}$ is a response.

Let $w = \binom{\sigma_1}{a_1} \cdots \binom{\sigma_n}{a_n}$ be an input word and θ be a pebble-i assignment such that $1 \leq \theta(i) \leq n$. Let $\mathcal{R}$ be a response. From the configuration $[i, (q, \mathcal{R}), \theta]$, pebble i performs the following.

1. Places pebble $(i-1)$ and simulates it starting from each possible state, in order to obtain the set of terms $\wp(\mu_i, w, \theta)$.

2. Let $\mathcal{P} = \mathcal{R} \cup \wp(\mu_i, w, \theta)$.
   Then, pebble i enters the state $(q, \mathrm{CTH}(\mathcal{P}, \sigma, P, V))$ and moves right, where

   - $\sigma = \sigma_{\theta(i)}$;
   - $P = \{l > i : \theta(l) = \theta(i)\}$;
   - $V = \{l > i : a_{\theta(l)} = a_{\theta(i)}\}$.

The formal description is given below. Let $Q_1, \ldots, Q_i$ be the sets of states of pebbles $1, \ldots, i$, respectively, and $\mu_1, \ldots, \mu_i$ be the sets of transitions of pebbles $1, \ldots, i$, respectively. Recall that the behavior of the pebbles $1, \ldots, (i-1)$, according to $\mu_1, \ldots, \mu_{i-1}$, is deterministic.

Similar to the case of pebble 1, we need to make a bit of modification on the behavior of pebble $(i+1)$. Let $\tilde{Q}_{i+1}$, $\tilde{\mu}_{i+1}$, $\tilde{U}_{i+1}$, $\tilde{N}_{i+1}$, $\tilde{D}_{i+1}$ be the modification of Q_{i+1}, μ_{i+1}, U_{i+1}, N_{i+1} ,D_{i+1}, respectively, as follows.

- $\tilde{Q}_{i+1} = Q_{i+1} \cup 2^{Q_{i+1}} \cup 2^{2^{Q_{i+1}}}$;

- $\tilde{U}_{i+1} = U_{i+1} \cup 2^{Q_{i+1}} - \{\emptyset\}$;

- $\tilde{N}_{i+1} = N_{i+1} \cup 2^{2^{Q_{i+1}}}$;

- $\tilde{D}_{i+1} = D_{i+1}$.

The set of transitions $\tilde{\mu}_{i+1}$ is the set μ_{i+1} plus the following transitions:

1. For every label $\sigma \in \Sigma$, sets $P, V \subseteq \{i+2, \ldots, k\}$, and sets $S_1, \ldots, S_m \subseteq Q_{i+1}$,

   $$(i+1, \sigma, P, V, \{S_1, \ldots, S_m\}) \to (S_j, \texttt{stay}) \in \tilde{\mu}_{i+1}, \text{ for each } j = 1, \ldots, m.$$

   That is, from the state $\{S_1, \ldots, S_m\} \in \tilde{Q}_{i+1}$ pebble $(i+1)$ performs existential branching.
   Recall that the state $\{S_1, \ldots, S_m\} \in \tilde{Q}_{i+1}$ is a nondeterministic state.

2. For every σ, P, V, $S \subseteq Q_{i+1}$, we have the following transition in $\tilde{\mu}_{i+1}$.

   $$(i+1, \sigma, P, V, S) \to (q, \texttt{stay}) \in \tilde{\mu}_{i+1}, \text{ for each } q \in S.$$

   That is, from the state $S \in \tilde{Q}_{i+1}$ pebble $(i+1)$ performs universal branching.
   Recall that the state $S \in \tilde{Q}_{i+1}$ is a universal state.

3. We replace each transition $(i+1, \sigma, P, V, q) \to (p, \texttt{place-pebble}) \in \mu_{i+1}$ with the following transition in $\tilde{\mu}_{i+1}$

   $$(i+1, \sigma, P, V, q) \to ((p, \emptyset), \texttt{place-pebble}) \in \tilde{\mu}_{i+1}.$$

   That is, $\tilde{\mu}_{i+1}$ no longer contains $(i+1, \sigma, P, V, q) \to (p, \texttt{place-pebble})$, rather it contains $(i+1, \sigma, P, V, q) \to ((p, \emptyset), \texttt{place-pebble})$.

All other transitions in μ_{i+1} remain in $\tilde{\mu}_{i+1}$.

Now, we define the sets of states $Q'_1, \ldots, Q'_i$ and the sets of transitions $\mu'_1, \ldots, \mu'_i$ such that the behavior of pebbles $1, \ldots, i$, according to $\mu'_1, \ldots, \mu'_i$, is deterministic. We start with defining the sets of states $Q'_1, \ldots, Q'_i$.

1. Q'_i consists of elements of the forms

   - $(q, \mathcal{PR})$ where $q \in Q_i$ and $\mathcal{PR}$ is a *partial response*;
   - $(q, X, \mathcal{PR})$ where $q \in Q_i$, $X \subseteq Q_{i,\texttt{place}}$ and $\mathcal{PR}$ is a *partial response.*

The intuitive meaning of the state $(q, \mathcal{PR})$ is like in the previous subsection. The purpose of the state $(q, X, \mathcal{PR})$ is for simulating pebble $(i-1)$ in order to compute the set $\wp$. The set X is supposed to contain the states of pebble i from which the automaton has yet to simulate pebble $(i-1)$.

2. For each $j = 1, \ldots, i-1$, the states in Q'_j are of the form

$$((q, X, \mathcal{PR}, s), p)$$

where $q \in Q_i$, $X \subseteq Q_{i,\mathtt{place}}$, $\mathcal{PR}$ is a partial response, $s \in Q_{i,\mathtt{place}}$ and $p \in Q_j$.
The intuitive meaning of these states is as follows.

- The triple $(q, X, \mathcal{PR})$ is to remember the state of pebble i while simulating pebble $(i-1)$.
- The component $s \in Q_{i,\mathtt{place}}$ is to remember the starting state of the simulation of pebble $(i-1)$.
- The last component $p \in Q_j$ is the current state of the simulation.

The sets of transitions $\mu'_1, \ldots, \mu'_i$ are defined as follows.

1. The sets $\mu'_1, \ldots, \mu'_{i-1}$, are defined as follows.

   (a) For each $j = 1, \ldots, i-2$, for each transition

   $$(j, \sigma, P, V, p) \to (t, \mathtt{act}) \in \mu_j,$$

   we have the transition

   $$(j, \sigma, P, V, ((q, X, \mathcal{PR}, s), p)) \to (((q, X, \mathcal{PR}, s), t), \mathtt{act}) \in \mu'_j.$$

   (b) For each transition

   $$(i-1, \sigma, P, V, p) \to (t, \mathtt{act}) \in \mu_{i-1},$$

   where $\mathtt{act} \neq \mathtt{lift\text{-}pebble}$, we have the transition

   $$(i-1, \sigma, P, V, ((q, X, \mathcal{PR}, s), p)) \to (((q, X, \mathcal{PR}, s), t), \mathtt{act}) \in \mu'_{i-1}.$$

(c) For each transition

$$(i-1, \triangleright, \emptyset, \emptyset, p) \to (t, \texttt{lift-pebble}) \in \mu_{i-1}$$

we have the transition

$$(i-1, \triangleright, \emptyset, \emptyset, ((q, X, \mathcal{PR}, s), p)) \to$$
$$((q, X, \mathcal{PR} \cup \{s \to \{t\}\}), \texttt{lift-pebble}) \in \mu'_{i-1}.$$

2. μ'_i consists of the following transitions.

(a) For each $q \in Q_i$, $(i, \triangleleft, \emptyset, \emptyset, (q, \emptyset)) \to ((q, \mathcal{R}), \texttt{right}) \in \mu'_i$, where $\mathcal{R} = \mathcal{R}(w, \theta)$, for some w and θ such that $\theta(i) = 0$.
By Remark 2, such $\mathcal{R}(w, \theta)$ is well defined.

(b) For state $q \in Q_i$, every response $\mathcal{R}$, label $\sigma \in \Sigma$ and $P, V \subseteq \{i+1, \ldots, k\}$,

$$(i, \sigma, P, V, (q, \mathcal{R})) \to ((q, Q_{i,\texttt{place}}, \mathcal{R}), \texttt{stay}) \in \mu'_i.$$

The purpose of this transition is to start computing the set of terms $\wp$.

(c) For every state $q \in Q_i$, every partial response $\mathcal{PR}$, every nonempty set $X \subseteq Q_{i,\texttt{place}}$, every label $\sigma \in \Sigma$ and every sets $P, V \subseteq \{i+1, \ldots, k\}$,

$$(i, \sigma, P, V, (q, X, \mathcal{PR})) \to$$
$$(((q, X - \{s\}, \mathcal{PR}, s), t), \texttt{place-pebble}) \in \mu'_i,$$

where $X \neq \emptyset$, $s \in X$ and $(i, \sigma, P, V, s) \to (t, \texttt{place-pebble})$.
The purpose of these transitions is to simulate pebble $(i-1)$ from the state s, where s is the state of pebble i before pebble $(i-1)$ is placed for the simulation.
Note that this is a place-pebble transition, so the state $((q, X - \{s\}, \mathcal{PR}, s), t) \in Q'_{i-1}$.

(d) For every state $q \in Q_i$, every partial response $\mathcal{PR}$, every label $\sigma \in \Sigma$ and every sets $P, V \subseteq \{i+1, \ldots, k\}$,

$$(i, \sigma, P, V, (q, \emptyset, \mathcal{PR})) \quad \to \quad ((q, \mathrm{CTH}(\mathcal{PR}, \sigma, P, V)), \texttt{right}) \in \mu'_i.$$

The purpose of these transitions is as follows. Now that the automaton has finished simulating pebble $(i-1)$ from all possible states, as indicated by the fact that $X = \emptyset$, pebble i computes $\mathrm{CTH}(\mathcal{PR}, \sigma, P, V)$, enters the state $(q, \mathrm{CTH}(\mathcal{PR}, \sigma, P, V))$ and moves right.

(e) $(i, \triangleright, \emptyset, \emptyset, (q, \mathcal{R})) \to (\{S_1, \dots, S_m\}, \texttt{lift-pebble})$, where for each $j = 1, \dots, m$,

- $q \to \bar{S}_j \in \mathrm{CTH}(\mathcal{R}, \triangleright, \emptyset, \emptyset)$;
- $S_j \subseteq Q_{i+1}$.

The purpose of these transitions is the same as their pebble 1 counterpart. Recall also that no new pebble is placed when the head pebble is reading the right-end marker $\triangleright$, thus, it is not necessary to compute the set of terms $\wp$.

The proof that $\mu_1 \cup \cdots \cup \mu_i \cup \mu_{i+1}$ and $\mu'_1 \cup \cdots \cup \mu'_i \cup \tilde{\mu}_{i+1}$ are equivalent is similar to the corresponding proof for the case of pebble 1, thus, omitted.

4.3 Determinizing $\mathcal{A}$

For the final step, we define the deterministic k-PA $\mathcal{A}' = \langle Q', q'_0, \mu', F' \rangle$ that accepts the same language as $\mathcal{A} = \langle Q, q_0, \mu, F \rangle$. By the induction step explained in the previous subsection, we assume that the behavior of pebbles $1, \dots, k-1$ is deterministic.

- $Q' = Q'_1 \cup \cdots \cup Q'_{k-1} \cup Q'_k \cup \{q_{acc}, q_{rej}\}$, where each $Q'_1, \dots, Q'_{k-1}, Q'_k$ are the modification of the set of states $Q_1, \dots, Q_{k-1}, Q_k$ like in the previous subsection;
- $q'_0 = (q_0, \emptyset)$;
- $F' = \{q_{acc}\}$;
- $\mu' = \mu'_1 \cup \cdots \cup \mu'_{k-1} \cup \mu'_k$, where each $\mu'_1, \dots, \mu'_{k-1}, \mu'_k$ are the modification of the set of transitions $\mu_1, \dots, \mu_{k-1}, \mu_k$ like in the previous subsection, plus the following transitions.
  The transition

$$(k, \triangleright, \emptyset, \emptyset, (q_0, \mathcal{R})) \to (q_{acc}, \texttt{right}) \in \mu'_k,$$

if there exists a set $S \subseteq F$ such that $q_0 \to \bar{S} \in \mathrm{CTH}(\mathcal{R}, \triangleright, \emptyset, \emptyset)$, and the transition

$$(k, \triangleright, \emptyset, \emptyset, (q_0, \mathcal{R})) \to (q_{rej}, \mathtt{right}) \in \mu'_k,$$

if there does not exists a set $S \subseteq F$ such that $q_0 \to \bar{S} \in \mathrm{CTH}(\mathcal{R}, \triangleright, \emptyset, \emptyset)$.

The proof that $\mathcal{A}$ and $\mathcal{A}'$ are equivalent is similar to the corresponding proof for the case of pebble 1, thus, omitted.

References

[1] Henrik Björklund and Thomas Schwentick. On Notions of Regularity for Data Languages. In *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory (FCT 2007)*, Springer, pp. 88–99, *LNCS* 4639.

[2] Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin and Claire David. Two-Variable Logic on Words with Data. In *Proceedings of the 21th IEEE Symposium on Logic in Computer Science (LICS 2006)*, pp. 7–16.

[3] Stéphane Demri and Ranko Lazic. LTL with the Freeze Quantifier and Register Automata. In *Proceedings of the 21th IEEE Symposium on Logic in Computer Science (LICS 2006)*, pp. 17–26.

[4] Michael Kaminski and Nissim Francez. Finite-memory automata. *Theoretical Computer Science* **134** (1994) 329–363.

[5] Richard Ladner, Richard Lipton and Larry Stockmeyer. Alternating Pushdown and Stack Automata. *SIAM Journal of Computing* **13**:1 (1984) 135–155.

[6] Frank Neven, Thomas Schwentick and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Transactions on Computational Logic* **5**:3 (2004) 403–435.

[7] Luc Segoufin. Automata and Logics for Words and Trees over an Infinite Alphabet. In *Proceedings of the 20th International Workshop/15th Annual Conference of the EACSL Computer Science Logic (CSL 2006)*, pp. 41–57, *LNCS* 4207.

[8] Tony Tan. Graph Reachability and Pebble Automata over Infinite Alphabets. To appear in the *Proceedings of the 24th IEEE Symposium on Logic in Computer Science (LICS 2009)*.

[9] Tony Tan. On Pebble Automata for Data Languages with Decidable Emptiness Problem. Technical Report CS-2009-05, Department of Computer Science, Technion – Israel Institute of Technology, 2009.