Automata Theory and Computability

Piyush Kumar (23801)

April 25, 2025

Assignemnt 5 *(Turing Machines and Decidability)*

1. Is the following question decidable: Given a Turing machine M and a state q of M, does M ever enter state q on *some* input? Justify your answer.

Solution: *No. The halting problem can be reduced to this.*

Given any Turing machine M and input x, we can construct a new Turing machine $P_{M,x}$ that erases its input and runs M on x. Thus $P_{M,x}$ halts on all inputs iff M halts on x. Let t be the accept state of $P_{M,x}$.

This map $M\#x \mapsto P_{M,x}\#t$ is computable, which gives a reduction from **HP** to

$$\{M\#q \mid M \text{ enters state } q\}.$$

Thus, if it were decidable whether $P_{M,x}$ enters state t on some input, we could decide the halting problem. ■

■

2. An *enumeration* machine N is a two-tape Turing machine with the following distinctions:
   - The machine is not given any input; both its tapes are blank to begin with.
   - The rst tape is a write-only tape, on which the machine can only write symbols of Σ. The second tape is a usual two-way read/write tape on which it can write any element of Γ.
   - The machine has no accept/reject state, but instead it has a special enumeration state e by which it signals that it has written something interesting on its rst tape. Thus the contents of the first tape are said to have been "enumerated" whenever the machine enters the state e. After entering e, the contents of the first tape are "automatically" erased and the first tape head is rewound to the left end of the tape. The machine then resumes working from there.
   - The language $L(N)$ is defined to be the set of strings in Σ^* enumerated by N.

   (a) Prove that enumeration machines and Turing machines are equal in computation power: i.e. the class of languages they define is precisely the r.e. languages.
   (b) Prove that an r.e. language is recursive i there is an enumeration machine that enumerates it in *increasing* order.

Solution: We address each part of the problem.

Part (a): Equivalence of Enumeration Machines and Turing Machines in computational power

We need to show that the class of languages enumerable by enumeration machines is exactly the class of recursively enumerable (r.e.) languages. This requires proving two inclusions:

1. If a language is enumerable by an enumeration machine, it is recursively enumerable.
2. If a language is recursively enumerable, it is enumerable by an enumeration machine.

1. Language enumerated by N is r.e.

Let L be a language enumerated by an enumeration machine N. We construct a Turing machine M that accepts L.

The Turing machine M will have an input tape where the string w is initially placed. M will simulate the enumeration machine N. M can use additional tapes to simulate N's two tapes. Let's say M uses Tape 2 to simulate N's write-only Tape 1 and Tape 3 to simulate N's read/write Tape 2.

M starts with input w on Tape 1. Tapes 2 and 3 are blank, simulating N's initial state. M simulates the transitions of N. When N writes a symbol on its Tape 1, M writes that symbol on its Tape 2. When N moves its Tape 1 head, M moves its Tape 2 head accordingly (though it's write-only, the simulation needs to track the head position). When N writes on its Tape 2 or moves its Tape 2 head, M performs the equivalent operations on its Tape 3.

Crucially, when N enters the enumeration state e, M performs the following actions:

- M copies the contents of its Tape 2 (the simulated Tape 1 of N) to a separate work tape or uses Tape 1.
- M compares the copied string with the input string w on Tape 1.

- If the strings match, M enters an accept state and halts.
- If the strings do not match, M erases its Tape 2 and resets the Tape 2 head to the beginning, simulating N's automatic tape erasure and rewind. M then continues the simulation of N from the state N would transition to after entering e.

If $w \in L(N)$, then N will eventually enumerate w. When N enumerates w, M will detect the match and accept w. If $w \notin L(N)$, N will never enumerate w, and M will continue simulating N indefinitely without entering an accept state. This behavior precisely matches the definition of a Turing machine accepting a recursively enumerable language.

Thus, any language enumerable by an enumeration machine is recursively enumerable.

2. Any r.e. language can be enumerated by N.

Let L be a recursively enumerable language. By definition, there exists a Turing machine M that accepts L. We construct an enumeration machine N that enumerates L.

The enumeration machine N will use its second tape (the read/write tape) to simulate the behavior of M and to systematically generate candidate strings from Σ^*.

N will operate as follows:

1. N generates all possible strings in Σ^* in a systematic order. A common method is to generate strings in increasing order of length, and for strings of the same length, in lexicographical order: $\epsilon, s_1, s_2, s_3, \ldots$. This can be done on its second tape.
2. N needs to simulate M on each generated string to check if M accepts it. Since M might not halt on strings not in L, N cannot simply simulate M sequentially on each string to completion. Instead, N employs a dovetailing simulation.
3. N maintains a list of generated strings and the current configuration (state, tape contents, head position) of the simulation of M for each string on its second tape.
4. N proceeds in stages $k = 1, 2, 3, \ldots$. In stage k:
   - N generates the first k strings in the systematic order: $s_1, s_2, \ldots, s_k$.
   - For each string s_i ($1 \leq i \leq k$), N simulates M on input s_i for k steps. This simulation is performed on N's second tape, keeping track of the state, tape contents, and head position for each of the k simulations.
   - If the simulation of M on s_i halts in an accepting state within k steps, N writes the string s_i onto its first (write-only) tape. After writing s_i, N enters the enumeration state e. (The automatic erasure and rewind of Tape 1 occur here). After entering e, N returns to its simulation task for the current stage k.

Since M accepts L, for every string $s \in L$, M will halt in an accepting state after some finite number of steps. Let this number be k_s. In any stage $k \geq k_s$, N will simulate M on s for at least k_s steps, the simulation will reach an accepting state, and N will enumerate s. Thus, every string in L is eventually enumerated by N.

If a string $s \notin L$, M on input s either halts in a rejecting state or loops indefinitely. In neither case does M enter an accepting state. Therefore, N will never enumerate s.

This construction shows that any recursively enumerable language can be enumerated by an enumeration machine.

From both inclusions, we conclude that the class of languages defined by enumeration machines is precisely the class of recursively enumerable languages.

Part (b): Relationship between recursive languages and ordered enumeration

We need to prove that an r.e. language L is recursive if and only if there is an enumeration machine that enumerates L in increasing order. Increasing order means strings are ordered by length, and strings of the same length are ordered lexicographically.

1. If an r.e. language L is enumerated in increasing order by an enumeration machine, then L is recursive.

Assume L is an r.e. language enumerated in increasing order by an enumeration machine N_{ord}. We construct a Turing machine $M_{decider}$ that decides L.

$M_{decider}$ takes an input string w. $M_{decider}$ simulates N_{ord} and observes the strings that are enumerated. Since N_{ord} enumerates strings in increasing order, $M_{decider}$ can use this property to decide whether $w \in L$.

$M_{decider}$ simulates N_{ord}. N_{ord} will enumerate strings $s_1, s_2, s_3, \ldots$ such that $|s_i| \leq |s_{i+1}|$, and if $|s_i| = |s_{i+1}|$, then s_i comes before s_{i+1} lexicographically.

$M_{decider}$ compares each enumerated string s with its input w:

- If $s = w$, $M_{decider}$ accepts w and halts.
- If s comes after w in the increasing order (i.e., $|s| > |w|$, or $|s| = |w|$ and s is lexicographically greater than w), then $M_{decider}$ knows that w cannot be in L. This is because if w were in L, it would have been enumerated before s due to the increasing order. In this case, $M_{decider}$ rejects w and halts.

Since L is r.e., every string in L is eventually enumerated. If $w \in L$, N_{ord} will eventually enumerate w, and $M_{decider}$ will accept. If $w \notin L$, N_{ord} will eventually enumerate a string that comes after w in the increasing order. Since the enumeration includes all strings of a shorter length before any string of a longer length, and strings of the same length are in lexicographical order, $M_{decider}$ will encounter a string that allows it to definitively say $w \notin L$. For example, if w has length k, $M_{decider}$ simulates N_{ord} until it enumerates a string of length greater than k, or it has enumerated all strings of length k lexicographically less than w and the next enumerated string of length k is lexicographically greater than w. In either case, $M_{decider}$ halts and rejects.

Since $M_{decider}$ always halts on any input w (either by accepting if w is enumerated, or by rejecting if a string after w in the increasing order is enumerated), $M_{decider}$ is a decider for L. Therefore, L is a recursive language.

2. If an r.e. language L is recursive, then there is an enumeration machine that enumerates it in increasing order.

Assume L is a recursive language. By definition, there exists a Turing machine $M_{decider}$ that decides L; that is, $M_{decider}$ halts on all inputs, accepting if the input is in L and rejecting otherwise. We construct an enumeration machine N_{ord} that enumerates L in increasing order.

N_{ord} will use its second tape to generate strings in increasing order and simulate $M_{decider}$.

1. N_{ord} systematically generates all strings in Σ^* in increasing order (length-wise, then lexicographically): $\epsilon, s_1, s_2, s_3, \ldots$.
2. For each generated string s_i, N_{ord} simulates the Turing machine $M_{decider}$ with input s_i on its second tape.
3. Since $M_{decider}$ is a decider, this simulation will always halt in a finite number of steps, either in an accepting or a rejecting state.
4. If the simulation of $M_{decider}$ on s_i halts in an accepting state, N_{ord} writes the string s_i onto its first (write-only) tape and enters the enumeration state e.
5. If the simulation of $M_{decider}$ on s_i halts in a rejecting state, N_{ord} does not enumerate s_i and moves to the next string in the increasing order.

Since N_{ord} generates strings in increasing order and only enumerates a string if the decider $M_{decider}$ accepts it, N_{ord} will enumerate exactly the strings in L. Because the strings are generated and processed in increasing order, they will be enumerated in increasing order. Since $M_{decider}$ always halts, N_{ord} will process every string in the infinite sequence Σ^* in order, ensuring that all strings in L are eventually enumerated in the correct order.

Thus, if an r.e. language is recursive, there is an enumeration machine that enumerates it in increasing order.

Combining both implications, we conclude that an r.e. language is recursive if and only if there is an enumeration machine that enumerates it in increasing order.

■

3. Show that neither the language

$$TOTAL = \{M \mid M \text{ halts on all inputs}\}$$

nor its complement is r.e.

Both proofs are by reducing $\neg\, HP$ to the given language. The reduction for $\neg\, TOTAL$ is easier, so we present that proof first.

$\neg\, HP \leq \neg\, TOTAL$. Given a Turing machine M and input x, we can construct a new Turing machine $P_{M,x}$ that erases its input and runs M on x. Thus if M halts on x, then $P_{M,x}$ halts on all inputs. If M does not halt on x, then $P_{M,x}$ does not halt on any input.

Let φ be the map that sends $M\#x$ to $P_{M,x}$. Then φ is a computable function, and

$$M\#x \in \neg\, HP \iff \varphi(M\#x) \in \neg\, TOTAL.$$

This proves the claim. □

The reduction for $TOTAL$ is more involved.

$TOTAL$. Given a Turing machine M and input x, we can construct a new Turing machine $Q_{M,x}$ that does the following:

1. For input w, simulate M on x for $|w|$ steps.
2. If M halts on x within $|w|$ steps, then enter a looping state.
3. If M does not halt on x within $|w|$ steps, then halt.

This is easy to achieve by storing the input w on one tape, and simulating M on x on another tape, blanking one letter from w after each step of the simulation. This is a computable function.

If M halts on x in n steps, then $Q_{M,x}$ enters an infinite loop for inputs longer than n. But if M does not halt on x, then $Q_{M,x}$ halts on every input. Thus

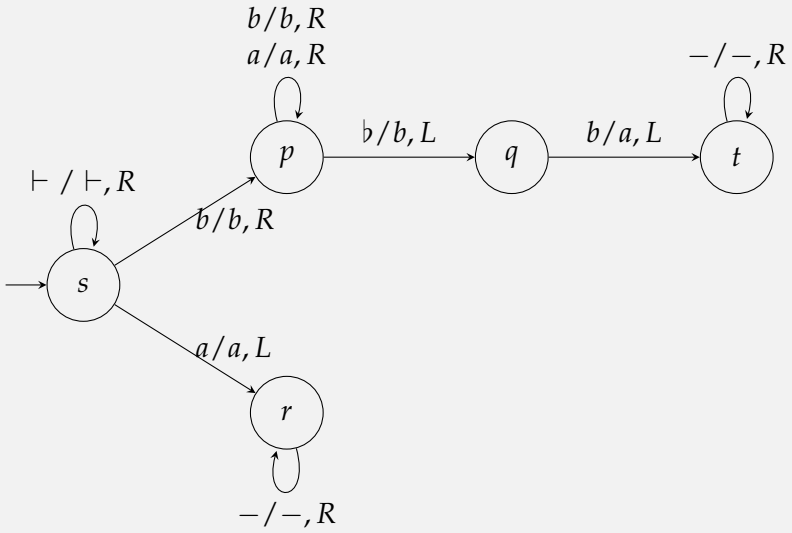
$$M\#x \in nHP \iff Q_{M,x} \in TOTAL.$$

This proves the claim. □

Since $\neg HP$ is not recursively enumerable, neither $TOTAL$ nor its complement is recursively enumerable.

■

4. Consider the TM M below, with input alphabet $\{a, b\}$.



(a) Give any string in $Valcomp_{M,baab}$.

(b) Recall the notion of matching triples of symbols used in class. Give the entire set of matching triples for M.

(c) Justify the claim that for two valid configurations c_1 and c_2 of M, which are of the same length, we have: $c_1 \equiv c_2$ if and only if for each position in c_1, the triple of symbols in c_1 and the corresponding triple in c_2 match.

Solution:

(a)

$$\begin{array}{cccccccccccccccccccccccccc}
\vdash & b & a & a & b & b & \flat & \# & \vdash & b & a & a & b & b & \flat & \# & \vdash & b & a & a & b & b & \flat & \# & \vdash & b & a & a & b & b & \flat & \# \\
s & - & - & - & - & - & - & - & - & s & - & - & - & - & - & - & - & p & - & - & - & - & - & - & p & - & - & - & - & - & p & - & - & \#
\end{array}$$

$$\begin{array}{cccccccccccccccccccccccccc}
\vdash & b & a & a & b & b & \flat & \# & \vdash & b & a & a & b & b & \flat & \# & \vdash & b & a & a & b & b & b & \# & \vdash & b & a & a & b & a & b & \# \\
- & - & - & - & - & p & - & - & - & - & - & - & - & - & p & - & - & - & - & - & - & q & - & - & - & - & - & - & - & t & \#
\end{array}$$

(b)

$$\left\langle \begin{matrix} x & y & z \\ - & - & - \end{matrix}, \begin{matrix} x & y & z \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & z \\ - & - & - \end{matrix}, \begin{matrix} x & y & z \\ u & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & z \\ - & - & - \end{matrix}, \begin{matrix} x & y & z \\ - & - & u \end{matrix} \right\rangle \quad \text{for any } x, y, z \in \Gamma, u \in Q$$

Corresponding to the transition $\delta(s, \vdash) = (s, \vdash, R)$, we have the matching triples

$$\left\langle \begin{matrix} \vdash & x & y \\ s & - & - \end{matrix}, \begin{matrix} \vdash & x & y \\ - & s & - \end{matrix} \right\rangle \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transition $\delta(s,a) = (r,a,L)$, we have the matching triples

$$\left\langle \begin{matrix} a & x & y \\ s & - & - \end{matrix}, \begin{matrix} a & x & y \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & a & y \\ - & s & - \end{matrix}, \begin{matrix} x & a & y \\ r & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & a \\ - & - & s \end{matrix}, \begin{matrix} x & y & a \\ - & r & - \end{matrix} \right\rangle \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transition $\delta(s,b) = (p,b,R)$, we have the matching triples

$$\left\langle \begin{matrix} b & x & y \\ s & - & - \end{matrix}, \begin{matrix} b & x & y \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & b & y \\ - & s & - \end{matrix}, \begin{matrix} x & b & y \\ - & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & b \\ - & - & s \end{matrix}, \begin{matrix} x & y & b \\ - & - & - \end{matrix} \right\rangle \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transition $\delta(s,\flat) = (r,\flat,R)$, we have the matching triples

$$\left\langle \begin{matrix} \flat & x & y \\ s & - & - \end{matrix}, \begin{matrix} \flat & x & y \\ - & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & \flat & y \\ - & s & - \end{matrix}, \begin{matrix} x & \flat & y \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & \flat \\ - & - & s \end{matrix}, \begin{matrix} x & y & \flat \\ - & - & - \end{matrix} \right\rangle \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transitions $\delta(p,x) = (p,x,R)$ for all $x \in \Gamma \setminus \{\flat\}$, we have the matching triples

$$\left\langle \begin{matrix} x & y & z \\ p & - & - \end{matrix}, \begin{matrix} x & y & z \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} y & x & z \\ - & p & - \end{matrix}, \begin{matrix} y & x & z \\ - & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} y & z & x \\ - & - & p \end{matrix}, \begin{matrix} y & z & x \\ - & - & - \end{matrix} \right\rangle \quad \text{for any } x \in \Gamma \setminus \{\flat\} \text{ and } y, z \in \Gamma$$

Corresponding to the transition $\delta(p,\flat) = (q,b,L)$, we have the matching triples

$$\left\langle \begin{matrix} \flat & x & y \\ p & - & - \end{matrix}, \begin{matrix} b & x & y \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & \flat & y \\ - & p & - \end{matrix}, \begin{matrix} x & b & y \\ q & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & \flat \\ - & - & p \end{matrix}, \begin{matrix} x & y & b \\ - & q & - \end{matrix} \right\rangle \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transition $\delta(q,b) = (t,a,R)$, we have the matching triples

$$\left\langle \begin{matrix} b & x & y \\ q & - & - \end{matrix}, \begin{matrix} a & x & y \\ - & t & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & b & y \\ - & q & - \end{matrix}, \begin{matrix} x & a & y \\ - & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & b \\ - & - & q \end{matrix}, \begin{matrix} x & y & a \\ - & - & - \end{matrix} \right\rangle, \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transition $\delta(q,a) = (r,a,R)$, we have the matching triples

$$\left\langle \begin{matrix} a & x & y \\ q & - & - \end{matrix}, \begin{matrix} a & x & y \\ - & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} x & a & y \\ - & q & - \end{matrix}, \begin{matrix} x & a & y \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} x & y & a \\ - & - & q \end{matrix}, \begin{matrix} x & y & a \\ - & - & - \end{matrix} \right\rangle \quad \text{for any } x, y \in \Gamma$$

Corresponding to the transitions $\delta(r,x) = (r,x,R)$ for all $x \in \Gamma$, we have the matching triples

$$\left\langle \begin{matrix} x & y & z \\ r & - & - \end{matrix}, \begin{matrix} x & y & z \\ - & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} y & x & z \\ - & r & - \end{matrix}, \begin{matrix} y & x & z \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} y & z & x \\ - & - & r \end{matrix}, \begin{matrix} y & z & x \\ - & - & - \end{matrix} \right\rangle \quad \text{for any } x, y, z \in \Gamma$$

Similarly, corresponding to the transitions $\delta(t,x) = (t,x,R)$ for all $x \in \Gamma$, we have the matching triples

$$\left\langle \begin{matrix} x & y & z \\ t & - & - \end{matrix}, \begin{matrix} x & y & z \\ - & t & - \end{matrix} \right\rangle, \left\langle \begin{matrix} y & x & z \\ - & t & - \end{matrix}, \begin{matrix} y & x & z \\ - & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} y & z & x \\ - & - & t \end{matrix}, \begin{matrix} y & z & x \\ - & - & - \end{matrix} \right\rangle \quad \text{for any } x, y, z \in \Gamma$$

$$\left\langle \begin{matrix} a\ a\ b \\ -\ r\ - \end{matrix}, \begin{matrix} a\ a\ b \\ -\ -\ r \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ a \\ -\ r\ - \end{matrix}, \begin{matrix} b\ b\ a \\ -\ -\ r \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash a\ \flat \\ s\ -\ - \end{matrix}, \begin{matrix} \vdash a\ \flat \\ -\ s\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ \flat\ \flat \\ -\ p\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ q\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} \vdash b\ a \\ -\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ s\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ \flat \\ -\ -\ p \end{matrix}, \begin{matrix} b\ b\ b \\ -\ q\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ b \\ -\ -\ q \end{matrix}, \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ \flat\ \flat \\ s\ -\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ r\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} b\ b\ a \\ r\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ \flat \\ -\ -\ r \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ \flat \\ q\ -\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ t\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ a \\ -\ -\ q \end{matrix}, \begin{matrix} b\ b\ a \\ -\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ b\ a \\ -\ -\ p \end{matrix}, \begin{matrix} a\ b\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ b \\ -\ -\ q \end{matrix}, \begin{matrix} a\ b\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash b\ \flat \\ s\ -\ - \end{matrix}, \begin{matrix} \vdash b\ \flat \\ -\ s\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ b \\ -\ -\ t \end{matrix}, \begin{matrix} b\ a\ b \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ \flat\ \flat \\ -\ p\ - \end{matrix}, \begin{matrix} b\ b\ \flat \\ q\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ b\ b \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ b \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ a \\ p\ -\ - \end{matrix}, \begin{matrix} b\ a\ a \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ b \\ s\ -\ - \end{matrix}, \begin{matrix} a\ a\ b \\ -\ r\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ a \\ -\ p\ - \end{matrix}, \begin{matrix} b\ a\ a \\ -\ -\ p \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ a \\ -\ -\ q \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash a\ a \\ -\ -\ r \end{matrix}, \begin{matrix} \vdash a\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} b\ b\ \flat \\ p\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat\ \flat\ \flat \\ -\ t\ - \end{matrix}, \begin{matrix} \flat\ \flat\ \flat \\ -\ -\ t \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash \flat\ \flat \\ s\ -\ - \end{matrix}, \begin{matrix} \vdash \flat\ \flat \\ -\ s\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ a\ b \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ b \\ -\ -\ q \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ a \\ s\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash a\ b \\ -\ -\ - \end{matrix}, \begin{matrix} \vdash a\ b \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ \flat\ \flat \\ p\ -\ - \end{matrix}, \begin{matrix} b\ b\ \flat \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ a \\ -\ -\ p \end{matrix}, \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \flat\ \flat\ \flat \\ r\ -\ - \end{matrix}, \begin{matrix} \flat\ \flat\ \flat \\ -\ r\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ \flat\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} b\ \flat\ \flat \\ p\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ a \\ -\ -\ - \end{matrix}, \begin{matrix} b\ a\ a \\ s\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} b\ a\ \flat \\ p\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ a\ b \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ b \\ s\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ \flat\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} b\ \flat\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash b\ b \\ -\ s\ - \end{matrix}, \begin{matrix} \vdash b\ b \\ -\ -\ p \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ \flat \\ -\ r\ - \end{matrix}, \begin{matrix} b\ a\ \flat \\ -\ -\ r \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} b\ b\ a \\ s\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \vdash a\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} \vdash a\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ \flat \\ -\ p\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ -\ p \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ a \\ -\ -\ - \end{matrix}, \begin{matrix} b\ a\ a \\ r\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} b\ b\ \flat \\ s\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash a\ b \\ -\ -\ r \end{matrix}, \begin{matrix} \vdash a\ b \\ -\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b\ a\ \flat \\ s\ -\ - \end{matrix}, \begin{matrix} b\ a\ \flat \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ a \\ p\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ \flat\ \flat \\ -\ t\ - \end{matrix}, \begin{matrix} b\ b\ \flat \\ -\ -\ t \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash b\ b \\ -\ -\ - \end{matrix}, \begin{matrix} \vdash b\ b \\ -\ -\ q \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ a \\ -\ -\ q \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \vdash b\ a \\ -\ -\ p \end{matrix}, \begin{matrix} \vdash b\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ b \\ -\ r\ - \end{matrix}, \begin{matrix} b\ b\ b \\ -\ -\ r \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat\ \flat\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} \flat\ \flat\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ a \\ -\ -\ q \end{matrix}, \begin{matrix} a\ a\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} b\ a\ \flat \\ s\ -\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b\ a\ a \\ -\ -\ - \end{matrix}, \begin{matrix} b\ a\ a \\ p\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ \flat\ \flat \\ s\ -\ - \end{matrix}, \begin{matrix} b\ \flat\ \flat \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ b \\ -\ p\ - \end{matrix}, \begin{matrix} a\ b\ b \\ -\ -\ p \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ b \\ p\ -\ - \end{matrix}, \begin{matrix} b\ a\ b \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ \flat \\ r\ -\ - \end{matrix}, \begin{matrix} b\ b\ \flat \\ -\ r\ - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ b\ b \\ p\ -\ - \end{matrix}, \begin{matrix} a\ b\ b \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ b \\ r\ -\ - \end{matrix}, \begin{matrix} b\ b\ b \\ -\ r\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ a \\ -\ r\ - \end{matrix}, \begin{matrix} b\ a\ a \\ -\ -\ r \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ \flat \\ r\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ a \\ -\ -\ - \end{matrix}, \begin{matrix} b\ b\ a \\ -\ -\ p \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ a\ \flat \\ -\ p\ - \end{matrix}, \begin{matrix} a\ a\ \flat \\ -\ -\ p \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat\ \flat\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} \flat\ \flat\ \flat \\ t\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ b \\ -\ q\ - \end{matrix}, \begin{matrix} a\ a\ b \\ -\ -\ t \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ b \\ -\ -\ - \end{matrix}, \begin{matrix} b\ b\ b \\ r\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ a \\ -\ p\ - \end{matrix}, \begin{matrix} a\ b\ a \\ -\ -\ p \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ b\ b \\ -\ -\ p \end{matrix}, \begin{matrix} a\ b\ b \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash b\ a \\ -\ -\ q \end{matrix}, \begin{matrix} \vdash b\ a \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ a\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} a\ a\ \flat \\ p\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ b \\ s\ -\ - \end{matrix}, \begin{matrix} a\ b\ b \\ -\ r\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ a\ b \\ -\ p\ - \end{matrix}, \begin{matrix} b\ a\ b \\ -\ -\ p \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \flat\ \flat\ \flat \\ -\ -\ - \end{matrix}, \begin{matrix} \flat\ \flat\ \flat \\ r\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ \flat \\ -\ -\ t \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash a\ b \\ s\ -\ - \end{matrix}, \begin{matrix} \vdash a\ b \\ -\ s\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ \flat\ \flat \\ r\ -\ - \end{matrix}, \begin{matrix} a\ b\ \flat \\ -\ r\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ \flat \\ -\ p\ - \end{matrix}, \begin{matrix} b\ \flat\ \flat \\ -\ -\ p \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a\ a\ b \\ -\ -\ r \end{matrix}, \begin{matrix} a\ a\ b \\ -\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} b\ b\ a \\ s\ -\ - \end{matrix}, \begin{matrix} b\ b\ a \\ -\ p\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ \flat\ \flat \\ -\ r\ - \end{matrix}, \begin{matrix} a\ \flat\ \flat \\ -\ -\ r \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ b \\ -\ -\ - \end{matrix}, \begin{matrix} a\ b\ b \\ s\ -\ - \end{matrix} \right\rangle, \left\langle \begin{matrix} a\ b\ b \\ -\ r\ - \end{matrix}, \begin{matrix} a\ b\ b \\ -\ -\ r \end{matrix} \right\rangle,$$

$$\left\langle \begin{matrix} a & b & a & a & b & a \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & - & ' & - & q \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & \flat & \vdash & b & b \\ - & - & p' & - & q & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & b & \vdash & b & b \\ - & - & - & ' & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & a & a & b & a \\ r & - & - & ' & - & r \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & \flat & \flat & \vdash & \flat & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & - & ' & s & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a & a & a & a \\ - & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & b & \vdash & b & b \\ s & - & - & ' & - & s \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ r & - & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b & b & b & b \\ - & p & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b & b & b & b \\ - & - & - & ' & - & q \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & b & \vdash & a & b \\ - & q & - & ' & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a & a & a & a \\ r & - & - & ' & - & r \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & a & b & a & a & b \\ r & - & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & a & \vdash & b & a \\ - & - & - & ' & - & q \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & \flat & b & b & \flat \\ s & - & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a & a & a & a \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ s & - & - & ' & - & r \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & \flat & \flat & \vdash & \flat & \flat \\ - & s & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ - & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & \flat & b & b & \flat \\ - & r & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & \flat & a & a & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & a & a & a & a & a \\ - & p & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & \flat & b & a & \flat \\ - & p & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & \flat & a & a & b \\ - & - & p' & - & q & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & a & b & b & a \\ - & p & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b & b & b & b \\ - & - & r' & - & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & a & b & a & a & b \\ - & - & - & ' & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b & b & b & b \\ p & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & \flat & \flat & b & \flat & \flat \\ - & r & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & \flat & a & a & \flat \\ r & - & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ - & - & - & ' & r & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & b & a & b & b & a \\ - & - & - & ' & - & q \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & a \\ - & - & q' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & \flat & b & b & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & b & a & a & b \\ - & - & p' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & a & b & b & a \\ - & - & - & ' & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \flat & \flat & \flat & \flat & \flat & \flat \\ - & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & a & b & a & a \\ s & - & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat & \flat & \flat & \flat & \flat & \flat \\ t & - & - & ' & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & - & ' & r & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & - & ' & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & a & a & \vdash & a & a \\ s & - & - & ' & - & s \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & \flat & \flat & \vdash & \flat & \flat \\ - & - & - & ' & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & \flat & a & a & \flat \\ - & - & - & ' & s & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & a & \flat & \vdash & a & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & a & a & b & a & a \\ - & - & p' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ s & - & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b & b & b & a \\ - & - & q' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & a & b & a & a \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ - & r & - & ' & - & r \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & b & \flat & b & b & \flat \\ - & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ - & t & - & ' & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & a & a & b & a \\ s & - & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & a & a & b & a \\ - & r & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & a & a & b & a \\ - & - & - & ' & p & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \vdash & a & \flat & \vdash & a & \flat \\ - & s & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & \flat & \flat & a & \flat & \flat \\ p & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & \flat & \flat & b & \flat & \flat \\ t & - & - & ' & t & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & \flat & \flat & a & \flat & \flat \\ - & - & - & ' & s & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & b & a & a & b \\ - & - & - & ' & r & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & \flat & \flat & b & \flat & \flat \\ - & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & \flat & \flat & b & \flat & \flat \\ r & - & - & ' & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & \flat & b & a & \flat \\ - & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & \flat & b & a & \flat \\ - & - & - & ' & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat & \flat & \flat & \flat & \flat & \flat \\ s & - & - & ' & - & r \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & a & \flat & b & a & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & \flat & \flat & b & \flat & \flat \\ - & - & t' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & a & b & \vdash & a & b \\ - & - & t' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & b & a & a & b \\ - & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat & \flat & \flat & \flat & \flat & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & a & a & b & a & a \\ - & - & q' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & b & a & a & b \\ p & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & b & \vdash & b & b \\ - & - & p' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & \flat & b & a & \flat \\ p & - & - & ' & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a & a & a & a \\ - & - & - & ' & s & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & \flat & \flat & b & \flat & \flat \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & a & \vdash & b & a \\ - & s & - & ' & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & \flat & a & a & \flat \\ s & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ r & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b & b & a & b \\ - & - & p' & - & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & b & b & a & b & b \\ - & - & r' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat & a & b & \flat \\ q & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & b & a & b & b \\ r & - & - & ' & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & a & b & b & a \\ - & - & p' & - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b & b & b & b \\ - & - & p' & - & - & - \end{matrix} \right\rangle,$$

$$\left\langle \begin{matrix} a & a & b \\ - & p & - \end{matrix}, \begin{matrix} a & a & b \\ - & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a \\ p & - & - \end{matrix}, \begin{matrix} a & a & a \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b \\ - & - & - \end{matrix}, \begin{matrix} b & b & b \\ s & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & a & b \\ - & s & - \end{matrix}, \begin{matrix} \vdash & a & b \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & a & a \\ - & s & - \end{matrix}, \begin{matrix} \vdash & a & a \\ - & - & r \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & a & \flat \\ - & r & - \end{matrix}, \begin{matrix} a & a & \flat \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} b & \flat & \flat \\ - & - & - \end{matrix}, \begin{matrix} b & \flat & \flat \\ s & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & b \\ - & - & t \end{matrix}, \begin{matrix} a & a & b \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & b \\ - & q & - \end{matrix}, \begin{matrix} a & a & b \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & a \\ - & - & - \end{matrix}, \begin{matrix} b & a & a \\ - & - & q \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & a & \flat \\ p & - & - \end{matrix}, \begin{matrix} a & a & \flat \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b \\ - & q & - \end{matrix}, \begin{matrix} b & a & b \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & a \\ - & - & r \end{matrix}, \begin{matrix} b & b & a \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & a \\ r & - & - \end{matrix}, \begin{matrix} b & b & a \\ - & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & a \\ - & - & - \end{matrix}, \begin{matrix} b & a & a \\ - & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & a & \flat \\ r & - & - \end{matrix}, \begin{matrix} b & a & \flat \\ - & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & a \\ r & - & - \end{matrix}, \begin{matrix} b & a & a \\ - & r & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a \\ - & r & - \end{matrix}, \begin{matrix} a & a & a \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b \\ s & - & - \end{matrix}, \begin{matrix} b & b & b \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & a \\ - & - & - \end{matrix}, \begin{matrix} a & b & a \\ r & - & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & b & a \\ p & - & - \end{matrix}, \begin{matrix} a & b & a \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & a \\ - & - & q \end{matrix}, \begin{matrix} a & b & a \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & \flat & \flat \\ - & t & - \end{matrix}, \begin{matrix} b & \flat & \flat \\ - & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} a & \flat & \flat \\ - & - & - \end{matrix}, \begin{matrix} a & \flat & \flat \\ r & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & a \\ s & - & - \end{matrix}, \begin{matrix} a & a & a \\ - & r & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} a & b & \flat \\ - & - & p \end{matrix}, \begin{matrix} a & b & b \\ - & q & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & b \\ - & - & q \end{matrix}, \begin{matrix} \vdash & b & a \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & a \\ p & - & - \end{matrix}, \begin{matrix} b & b & a \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & b \\ - & - & - \end{matrix}, \begin{matrix} a & b & b \\ r & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & b \\ - & - & - \end{matrix}, \begin{matrix} a & b & b \\ - & - & q \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & b & b \\ - & - & - \end{matrix}, \begin{matrix} b & b & b \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & \flat & \flat \\ - & - & - \end{matrix}, \begin{matrix} a & \flat & \flat \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat & \flat & \flat \\ - & - & - \end{matrix}, \begin{matrix} \flat & \flat & \flat \\ - & s & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & b & \flat \\ - & s & - \end{matrix}, \begin{matrix} \vdash & b & \flat \\ - & - & p \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & b \\ - & - & - \end{matrix}, \begin{matrix} b & b & b \\ - & p & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & b & \flat \\ - & - & - \end{matrix}, \begin{matrix} b & b & \flat \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & b \\ - & - & - \end{matrix}, \begin{matrix} a & b & b \\ - & p & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \vdash & a & a \\ - & - & - \end{matrix}, \begin{matrix} \vdash & a & a \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat & \flat & \flat \\ p & - & - \end{matrix}, \begin{matrix} b & \flat & \flat \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} a & b & \flat \\ p & - & - \end{matrix}, \begin{matrix} a & b & \flat \\ - & p & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} \vdash & b & a \\ s & - & - \end{matrix}, \begin{matrix} \vdash & b & a \\ - & s & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & \flat \\ - & - & p \end{matrix}, \begin{matrix} b & a & b \\ - & q & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & a & b \\ - & r & - \end{matrix}, \begin{matrix} b & a & b \\ - & - & r \end{matrix} \right\rangle, \left\langle \begin{matrix} a & a & \flat \\ - & - & - \end{matrix}, \begin{matrix} a & a & \flat \\ - & - & - \end{matrix} \right\rangle, \left\langle \begin{matrix} b & b & \flat \\ p & - & - \end{matrix}, \begin{matrix} b & b & \flat \\ - & p & - \end{matrix} \right\rangle,$$
$$\left\langle \begin{matrix} b & b & b \\ - & q & - \end{matrix}, \begin{matrix} b & a & b \\ - & - & t \end{matrix} \right\rangle, \left\langle \begin{matrix} \flat & \flat & \flat \\ - & r & - \end{matrix}, \begin{matrix} \flat & \flat & \flat \\ - & - & r \end{matrix} \right\rangle$$

(c)

Let the turing machine be given by $M = (Q, \Sigma, \Gamma, s, \delta, \vdash, \flat, t, r)$

Suppose two configurations, c_1 and c_2 of the same length n are such that $c_1 \Rightarrow^1 c_2$. Then by definition, $c_1(i), c_1(i+1), c_1(i+2)$ and $c_2(i), c_2(i+1), c_2(i+2)$ are matching triples for all $i \in \{1, 2, \ldots, n-2\}$.

For the converse, we will proceed by induction on n. Note that $n \geq 1$ since the tape must always begin with $\vdash$. If $n = 1$ or $n = 2$, we can pad the configurations with $(\flat, -)$ to make them of length 3. For $n = 3$, the result holds trivially since $c_1(1), c_1(2), c_1(3)$ and $c_2(1), c_2(2), c_2(3)$ match if and only if $c_1 \Rightarrow^1 c_2$. Now suppose the result holds for some $n \geq 3$. Let c_1 and c_2 be configurations of length $n+1$ such that all corresponding triples of c_1 and c_2 match.

Suppose neither $c_1(n+1)$ nor $c_2(n+1)$ contains the tape head. Let $c_1' = c_1(1) \ldots c_1(n)$ and $c_2' = c_2(1) \ldots c_2(n)$. Since all corresponding triples in c_1 and c_2 are matching, so do all corresponding triples in c_1' and c_2'. Then by the induction hypothesis, $c_1' \Rightarrow^1 c_2'$. Therefore $c_1 \Rightarrow^1 c_2$, since the last symbol in each configuration is unaffected by the transition (because the tape head is not at either end).

Now suppose $c_1(n+1)$ contains the tape head, and $c_1(n+1) = (a, p)$. If the transition is of the form $\delta(p, a) = (q, b, R)$, then $c_2(n-1)c_2(n)c_2(n+1)$ does not contain the tape head since the triples $c_1(n-1)c_1(n)c_1(n+1)$ and $c_2(n-1)c_2(n)c_2(n+1)$ match. But since c_2 is a valid configuration, $c_2(i)$ contains the tape head for some $i \in \{1, 2, \ldots, n-2\}$. Suppose $i = 1$. Then since $c_1(1)c_1(2)c_1(3)$ and $c_2(1)c_2(2)c_2(3)$ are matching triples, this is a contradiction because none of

$c_1(1)$, $c_1(2)$ and $c_1(3)$ contain the tape head. Suppose $i > 1$. Then since $c_1(i-1)c_1(i)c_1(i+1)$ and $c_2(i-1)c_2(i)c_2(i+1)$ are matching triples, this is a contradiction because none of $c_1(i-1)$, $c_1(i)$ and $c_1(i+1)$ contain the tape head. Therefore the transition must be of the form $\delta(p,a) = (q,b,L)$. Now since all corresponding triples in c_1 and c_2 are matching, the first $n-1$ symbols of c_1 and c_2 are the same. Moreover, since $c_1(n-1)c_1(n)c_1(n+1)$ and $c_2(n-1)c_2(n)c_2(n+1)$ are matching triples, $c_1(n)$ and $c_2(n)$ contain the same tape symbol and $c_2(n)$ contains the state q. Moreover, $c_2(n+1) = (b,-)$. Therefore $c_1 \Rightarrow^1 c_2$.

Finally, suppose that $c_2(n+1)$ contains the tape head. Since c_1 is a valid configuration, $c_1(i)$ contains the tape head for some $i \in \{1,2,\ldots,n+1\}$. Note that $i \neq n+1$ since $c_1(n-1)c_1(n)c_1(n+1)$ and $c_2(n-1)c_2(n)c_2(n+1)$ are matching triples. Similarly, if $1 < i < n$ then $c_2(n+1)$ cannot contain the tape head since $c_1(i-1)c_1(i)c_1(i+1)$ and $c_2(i-1)c_2(i)c_2(i+1)$ are matching triples. Finally if $i = 1$, then again $c_2(n+1)$ cannot contain the tape head since $c_1(1)c_1(2)c_1(3)$ and $c_2(1)c_2(2)c_2(3)$ are matching triples. Therefore $i = n$. Now since all corresponding triples in c_1 and c_2 are matching, the first $n-1$ symbols of c_1 and c_2 are the same. Moreover, since $c_1(n-1)c_1(n)c_1(n+1)$ and $c_2(n-1)c_2(n)c_2(n+1)$ are matching triples, $c_1(n+1)$ and $c_2(n+1)$ contain the same tape symbol and $c_2(n+1)$ contains the consecutive state, while $c_1(n)$ contains the updated tape symbol. Therefore $c_1 \Rightarrow^1 c_2$.

■

5. Is it decidable whether the complement of a given CFL is a CFL? Justify your answer.

Solution: We aim to prove that the problem of deciding whether the complement of a context-free language (CFL) is also a CFL is undecidable. Formally, given a CFL L, we need to determine if $\overline{L}$, the complement of L, is a CFL. We will show this by reducing the halting problem, which is known to be undecidable, to this decision problem.

Proof by Reduction We proceed by constructing a CFL L' whose complement $\overline{L'}$'s CFL status depends on whether a Turing machine M halts on input x.

Construction of the CFL L' Consider a Turing machine M and an input x. A computation of M on x can be represented as a string:

$$c_0 \# c_1^R \# c_2 \# c_3^R \# \cdots \# c_N$$

where:

- c_i is the i-th configuration of M on x, starting from the initial configuration c_0.
- # is a separator symbol.
- c_i^R denotes the reversal of configuration c_i, applied to odd-numbered configurations (i.e., i is odd).
- If the final index N is odd, c_N is replaced by c_N^R.

There exist two pushdown automata (PDAs):

- P_1: Checks transitions from even-numbered to odd-numbered configurations (e.g., $c_0 \Rightarrow c_1$, so $c_0 \# c_1^R$).

- P_2: Checks transitions from odd-numbered to even-numbered configurations (e.g., $c_1 \Rightarrow c_2$, so $c_1^R \# c_2$).

Define:

- $L_1 = L(P_1)$, the language of strings where even-to-odd transitions are valid.
- $L_2 = L(P_2)$, the language of strings where odd-to-even transitions are valid.

Since P_1 and P_2 are PDAs, L_1 and L_2 are CFLs. Let:

$$L = L_1 \cap L_2$$

L is the set of all valid computations of M on x, with odd-numbered configurations reversed. L is in bijection with $\text{Valcomp}_{M,x}$, the set of all valid computations in standard form $c_0 \# c_1 \# \cdots \# c_N$.

Now, let Σ be the alphabet of L, which includes tape symbols, state symbols, and #. Introduce a new symbol $\$ \notin \Sigma$, and define the new alphabet $\Gamma = \Sigma \cup \{\$\}$. Construct a new language:

$$L' = L \cup \Gamma^* \$ \Gamma^*$$

where $\Gamma^* \$ \Gamma^*$ is the set of all strings over Γ containing at least one \$. Since L_1, L_2, and $\Gamma^* \$ \Gamma^*$ (a regular language) are CFLs, we can redefine L' as:

$$L' = L_1 \cup \Gamma^* \$ \Gamma^*$$

which is a CFL, as the union of two CFLs is a CFL.

Analysis of the Complement $\overline{L'}$ The complement of L' is:

$$\overline{L'} = \overline{L \cup \Gamma^* \$ \Gamma^*} = \overline{L} \cap \overline{\Gamma^* \$ \Gamma^*}$$

Since $\Gamma^* \$ \Gamma^*$ contains all strings with at least one \$, its complement is:

$$\overline{\Gamma^* \$ \Gamma^*} = \Sigma^*$$

(i.e., all strings over Γ with no \$, which are exactly the strings over Σ). Thus:

$$\overline{L'} = \overline{L} \cap \Sigma^* = \overline{L}$$

since $\overline{L} \subseteq \Sigma^*$. Therefore, $\overline{L'}$ is the set of all strings over Σ that are not valid computations of M on x.

Case Analysis Based on Halting We analyze whether $\overline{L'}$ is a CFL based on whether M halts on x.

Case 1: M Halts on x If M halts on x, L is non-empty and infinite, containing all prefixes of the halting computation (e.g., c_0, $c_0 \# c_1^R$, ..., up to the halting configuration). $\overline{L'} = \overline{L}$ contains all strings over Σ that are not valid computations.

L is not a CFL when M halts, because it encodes valid computations with dependencies across configurations (similar to $\{a^n b^n c^n \mid n \geq 0\}$, which is not context-free due to the pumping lemma). Since CFLs are not closed under complementation, and L is neither empty nor co-finite, $\overline{L}$ (and thus $\overline{L'}$) is not a CFL.

Case 2: M Does Not Halt on x If M does not halt on x, $L = \emptyset$, as there are no valid halting computations. Then:

$$L' = \emptyset \cup \Gamma^*\$\Gamma^* = \Gamma^*\$\Gamma^*$$

$$\overline{L'} = \overline{\Gamma^*\$\Gamma^*} = \Sigma^*$$

Σ^* is a regular language, hence a CFL. Thus, $\overline{L'}$ is a CFL in this case.

Reduction to the Halting Problem We have:

- L' is a CFL in both cases (as a union of CFLs).
- $\overline{L'}$ is a CFL if and only if M does not halt on x:
  - If M halts, $\overline{L'}$ is not a CFL.
  - If M does not halt, $\overline{L'} = \Sigma^*$, which is a CFL.

Suppose we have a decision procedure to determine whether the complement of a CFL is a CFL. Then we can decide the halting problem as follows:

1. Given M and x, construct $L' = L \cup \Gamma^*\$\Gamma^*$, which is a CFL.
2. Use the decision procedure to check if $\overline{L'}$ is a CFL:
   - If yes, M does not halt on x.
   - If no, M halts on x.

This procedure decides whether M halts on x. However, the halting problem is undecidable. Therefore, the assumption that we can decide whether the complement of a CFL is a CFL leads to a contradiction. Hence, this decision problem is undecidable.

■