Consider the infinite word $w_0 = b^\omega$. Clearly, w_0 is accepted by $\mathcal{A}$, so $\mathcal{A}$ has an accepting run on w_0. Thus, w_0 has a finite prefix u_0 such that $\delta(q_0, u_0) \in \alpha$. Consider now the infinite word $w_1 = u_0 a b^\omega$. Clearly, w_1 is also accepted by $\mathcal{A}$, so $\mathcal{A}$ has an accepting run on w_1. Thus, w_1 has a finite prefix $u_0 b u_1$ such that $\delta(q_0, u_0 a u_1) \in \alpha$. In a similar fashion we can continue to find finite words u_i such that $\delta(q_0, u_0 a u_1 a \ldots a u_i) \in \alpha$. Since Q is finite, there are i, j, where $0 \le i < j$, such that $\delta(q_0, u_0 a u_1 a \ldots a u_i) = \delta(q_0, u_0 a u_1 a \ldots a u_i a \ldots a u_j)$. It follows that $\mathcal{A}$ has an accepting run on

$$u_0 a u_1 a \ldots a u_i (a u_{i+1} \ldots u_{j-1} a u_j)^\omega.$$

But the latter word has infinitely many occurrences of a, so it is not in L. $\square$

Note that the complementary language $(a \vee b)^\omega \setminus L = ((a \vee b)^* a)^\omega$ (the set of infinite words in which a occurs infinitely often) is accepted by the deterministic Büchi automaton of Figure 4.6. The automaton remembers the last symbol it read (q_0 corresponds to a and q_1 corresponds to b).

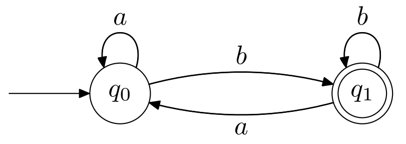

Figure 4.6: A deterministic Büchi automaton accepting $((a \vee b)^* a)^\omega$

Despite these discouraging observations, Büchi automata turn out to be closed under complement. The proof of this remarkable result is contained in the next section.

4.1.3 Closure under complement

Fix a Büchi automaton $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \alpha)$ with n states. We prove the existence of a Büchi automaton $comp(\mathcal{A})$ such that $\mathcal{L}_\omega(comp(\mathcal{A})) = comp(\mathcal{L}_\omega(\mathcal{A}))$ in three steps.

1. We define a mapping *dag* which assigns to each word $w \in \Sigma^\omega$ a directed acyclic graph $dag(w)$. We also define an *odd ranking* of $dag(w)$ as a labelling of the nodes of $dag(w)$ by natural numbers satisfying certain properties.

2. We prove that w is rejected by $\mathcal{A}$ if and only if $dag(w)$ admits an odd ranking.

3. We construct a Büchi automaton $comp(\mathcal{A})$ which accepts w if and only if $dag(w)$ admits an odd ranking.

These steps are carried out in the next three sections. We use the automaton of Figure 4.7 as running example.

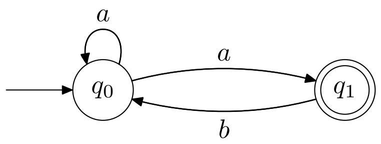

Figure 4.7: Running example for the complementation proof

The mapping dag, rankings, and good rankings

Let $w = \sigma_1\sigma_2\ldots$ be a word of Σ^ω. The directed acyclic graph $dag(w)$ has nodes in $Q \times \mathbb{N}$ and edges labelled by letters of Σ; it is inductively defined as follows:

- $dag(w)$ contains a node $\langle q, 0\rangle$ for every initial state $q \in Q_0$.
- If $dag(w)$ contains a node $\langle q, i\rangle$ and $q' \in \delta(\sigma_{i+1}, q)$, then $dag(w)$ also contains a node $\langle q', i+1\rangle$ and an edge $\langle\langle q, i\rangle, \langle q', i+1\rangle\rangle$ labelled by σ_{i+1}.
- $dag(w)$ contains no other nodes or edges.

Loosely speaking, $dag(w)$ is the result of bundling together all possible runs of $\mathcal{A}$ on the word w. Formally, $q_0q_1q_2\ldots$ is a run of $\mathcal{A}$ if and only if $\langle q_0, 0\rangle\langle q_1, 1\rangle\langle q_2, 2\rangle\ldots$ is a path of $dag(w)$. Figure 4.8 shows the initial parts of $dag(aba^\omega)$ and $dag((ab)^\omega)$ for our running example. Nodes corresponding to the final state q_1 are drawn with a double circle.

We introduce some basic terminology. A node $\langle q, i\rangle$ is *accepting* if q is an accepting state. A path is accepting if it contains infinitely many accepting nodes. If there is a path from $\langle q, i\rangle$ to $\langle q', j\rangle$, then $\langle q', j\rangle$ is *reachable* from $\langle q, i\rangle$. This path may possibly contain no edges, and so in particular a node is a successor of itself. If the path contains exactly one edge, i.e., if $j = i+1$, then we say that $\langle q', j\rangle$ is a *child* of $\langle q, i\rangle$.

A *ranking* of $dag(w)$ is a mapping that associates to each node a natural number in the range $[0 \ldots 2n]$, called a *rank*, satisfying the following two properties:

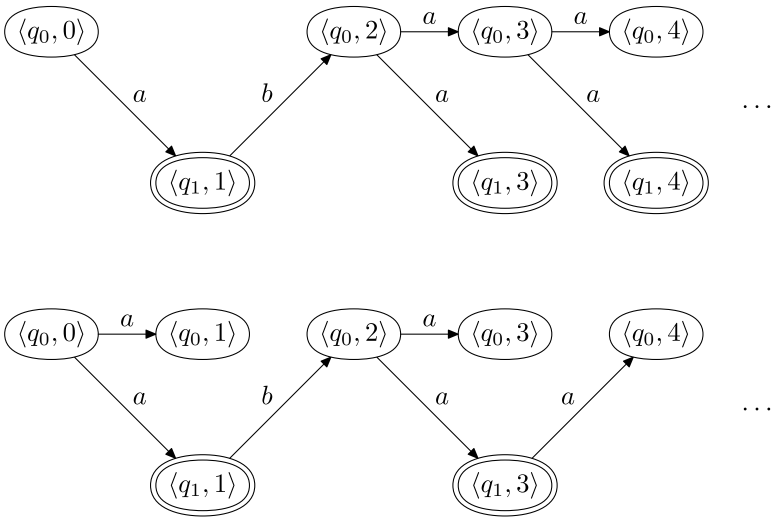

Figure 4.8: The (initial parts of) $dag(aba^\omega)$ and $dag((ab)^\omega)$

- the rank of a node is greater than or equal to the rank of its children, and
- the rank of an accepting node is even.

The ranks of the nodes in an infinite path form a non-increasing sequence, and so there is a node such that all its (infinitely many) successors have the same rank; we call this number the *stable rank* of the path. A ranking is *odd* if all infinite paths have odd stable rank. Figure 4.9 shows rankings for $dag(aba^\omega)$ and $dag((ab)^\omega)$. The ranking for $dag(aba^\omega)$ is odd, since the only infinite path has stable rank 1. The ranking for $dag((ab)^\omega)$ is not odd, because the only infinite path has stable rank 0.

Acceptance and good rankings

We prove that w is rejected by $\mathcal{A}$ if and only if $dag(w)$ admits an odd ranking. The 'if' half is easy, while the 'only if' part is more involved.

Exercise 23. Show: If $dag(w)$ admits an odd ranking, then w is rejected by $\mathcal{A}$.

Proposition 4.1.5 *If w is rejected by $\mathcal{A}$, then $dag(w)$ admits an odd ranking.*

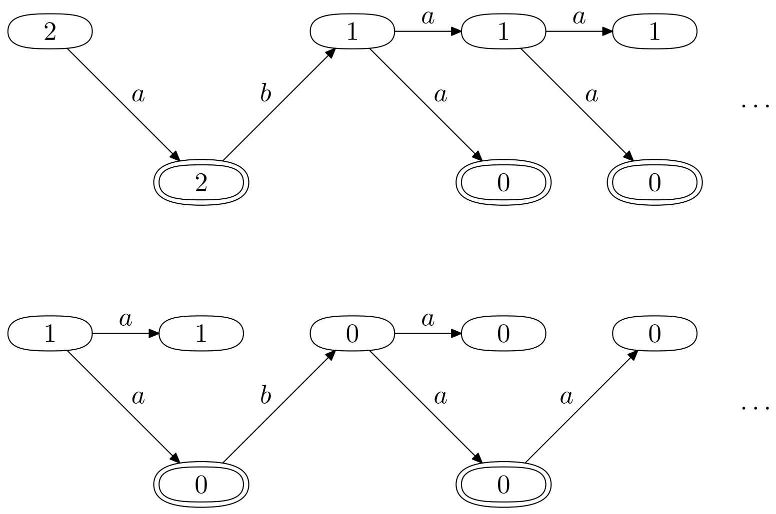

Figure 4.9: Rankings for $dag(aba^\omega)$ and $dag((ab)^\omega)$

Proof: In the sequel, given two DAGs D, D', we denote by $D' \subseteq D$ the fact that D' can be obtaind from D through deletion of some nodes and their adjacent edges.

Assume that $\mathcal{A}$ rejects w. We describe an odd ranking for $dag(w)$. We say that a node $\langle q, l\rangle$ is *endangered* in a (possibly finite) DAG $D \subseteq dag(w)$ iff only finitely many nodes of D are reachable from $\langle q, l\rangle$. The node $\langle q, l\rangle$ is *safe* in D iff all the nodes in D that are reachable from $\langle q, l\rangle$ are not accepting. In particular, safe nodes are not accepting. We inductively define an infinite sequence $D_0 \supseteq D_1 \supseteq D_2 \supseteq \ldots$ of DAGs by:

- $D_0 = dag(w)$;
- $D_{2i+1} = D_{2i} \setminus \{\langle q, l\rangle \mid \langle q, l\rangle \text{ is endangered in } D_{2i}\}$;
- $D_{2i+2} = D_{2i+1} \setminus \{\langle q, l\rangle \mid \langle q, l\rangle \text{ is safe in } D_{2i+1}\}$.

Figure 4.10 shows D_0, D_1, and D_2 for $dag(aba^\omega)$. D_3 is the empty dag.

Consider the function f that assigns to each node of $dag(w)$ a natural number as follows:

$$f(\langle q, l\rangle) = \begin{cases} 2i & \text{if } \langle q, l\rangle \text{ is endangered in } D_{2i} \\ 2i+1 & \text{if } \langle q, l\rangle \text{ is safe in } D_{2i+1} \end{cases}$$

We prove that f is an odd ranking. The proof is divided into three parts:

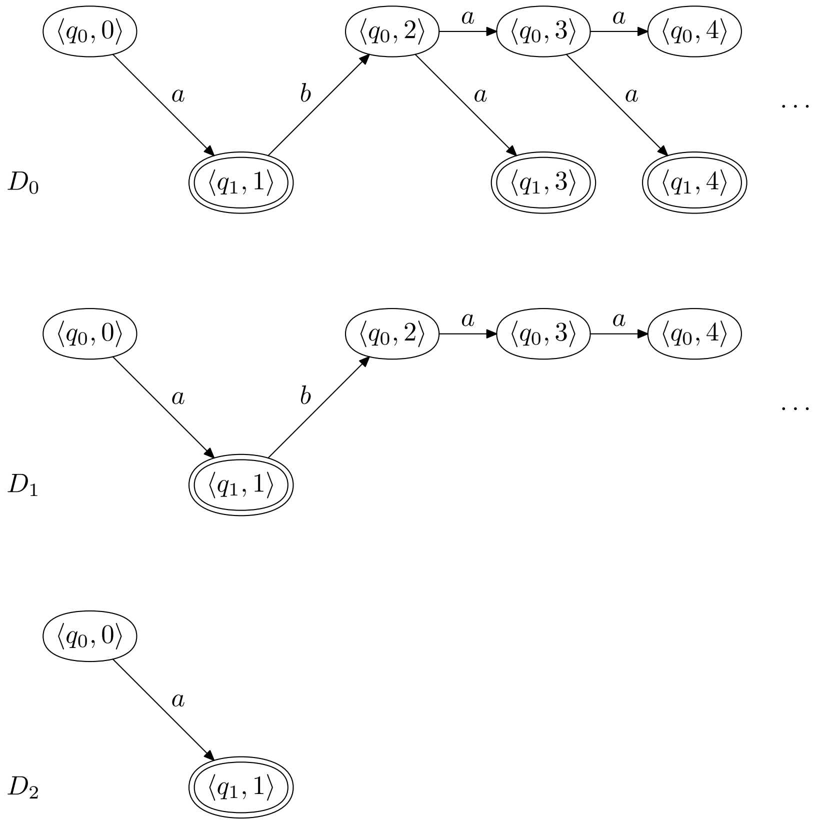

Figure 4.10: The DAGs D_0, D_1, D_2 for $dag(aba^\omega)$

(1) f assigns all nodes a number in the range $[0 \ldots 2n]$.

(2) If $\langle q', l' \rangle$ is a child of $\langle q, l \rangle$, then $f(\langle q', l' \rangle) \leq f(\langle q, l \rangle)$.

(3) If $\langle q, l \rangle$ is an accepting node, then $f(\langle q, l \rangle)$ is even.

We leave parts (2) and (3) as exercises.

Exercise 24. Prove that f satisfies condition (2) above. Hint: prove first that for every node $\langle q, l \rangle$ of $dag(w)$, if $\langle q, l \rangle \notin D_i$, then $f(q, l) < i$.

Exercise 25. Prove that f satisfies condition (3) above. For part (1), we

show that for every $i \geq 0$ there exists a number l_i such that for all $l \geq l_i$, the DAG D_{2i} contains at most $n - i$ nodes of the form $\langle q, l \rangle$. This implies that D_{2n} is finite, and so that D_{2n+1} is empty, which in turn implies that f assigns all nodes a number in the range $[0 \ldots 2n]$.

The proof is by an induction on i. The case where $i = 0$ follows from the definition of G_0. Indeed, in $dag(w)$ all levels $l \geq 0$ have at most n nodes of the form $\langle q, l \rangle$. Assume now that the hypothesis holds for i; we prove it for $i + 1$. Consider the DAG D_{2i}. We distinguish between two cases. First, if D_{2i} is finite, then D_{2i+1} is empty; D_{2i+2} is empty as well, and we are done. Otherwise, we claim that there must be some safe node in D_{2i+1}. To see this, assume, by way of contradiction, that D_{2i} is infinite and no node in D_{2i+1} is safe. Since D_{2i} is infinite, D_{2i+1} is also infinite. Moreover, since D_{2i+1} is obtained by removing all endangered nodes from D_{2i}, every node of D_{2i+1} has at least one child. Let $\langle q_0, l_0 \rangle$ be an arbitrary node of D_{2i+1}. Since, by the assumption, it is not safe, there exists an accepting node $\langle q_0', l_0' \rangle$ reachable from $\langle q_0, l_0 \rangle$. Let $\langle q_1, l_1 \rangle$ be a child of $\langle q_0', l_0' \rangle$. By the assumption, $\langle q_1, l_1 \rangle$ is also not safe. Hence, there exists an accepting node $\langle q_1', l_1' \rangle$ reachable from $\langle q_1, l_1 \rangle$. Let $\langle q_2, l_2 \rangle$ be a child of $\langle q_1', j_1' \rangle$. By the assumption, $\langle q_2, l_2 \rangle$ is also not safe. We can thus construct an infinite sequence of nodes $\langle q_j, l_j \rangle$, $\langle q_j', l_j' \rangle$ such that for all i the node $\langle q_j', l_j' \rangle$ is accepting, reachable from $\langle q_j, l_j \rangle$, and $\langle q_{j+1}, l_{j+1} \rangle$ is a child of $\langle q_j', l_j' \rangle$. Such a sequence, however, corresponds to a path in $dag(w)$ visiting infinitely many accepting nodes, which contradicts the assumption that $\mathcal{A}$ rejects w.

So, let $\langle q, l \rangle$ be a safe node in D_{2i+1}. We claim that we can take $l_{i+1} = l$. That is, we claim that for all $j \geq l$, there are at most $n - (i + 1)$ nodes of the form $\langle q, j \rangle$ in D_{2i+2}. Since $\langle q, l \rangle$ is in D_{2i+1}, it is not endangered in D_{2i}. Thus, there are infinitely many nodes in D_{2i} that are reachable from $\langle q, l \rangle$. Hence, by König's Lemma, D_{2i} contains an infinite path $\langle q, l \rangle, \langle q_1, l + 1 \rangle, \langle q_2, l + 2 \rangle, \ldots$. For all $k \geq 1$, D_{2i} contains infinitely many nodes reachable from $\langle q_k, l+k\rangle$, and thus $\langle q_k, l+k\rangle$ is not endangered in D_{2i}. Therefore, the path $\langle q, l\rangle, \langle q_1, l+1\rangle, \langle q_2, l+2\rangle, \ldots$ exists also in D_{2i+1}. Recall that $\langle q, l\rangle$ is safe. Hence, being reachable from $\langle q, l\rangle$, all the nodes $\langle q_k, l+k\rangle$ in the path are safe as well. Therefore, they are not in D_{2i+2}. It follows that for all $j \geq l$ the number of nodes of the form $\langle q, j\rangle$ in D_{2i+2} is strictly smaller than their number in D_{2i}, which, by the induction hypothesis, is $n-i$. So there are at most $n-(i+1)$ nodes of the form $\langle q, j\rangle$ in D_{2i+2}. □

The automaton $comp(\mathcal{A})$

By Exercise 23 and Proposition 4.1.5, an automaton that complements $\mathcal{A}$ can proceed on an input word w by guessing an odd ranking for the run DAG of $\mathcal{A}$ on w. We now define such an automaton $comp(\mathcal{A})$ formally. We need some preliminaries.

A *level ranking* for $\mathcal{A}$ and w is a function $g : Q \to [0..2n] \cup \{\perp\}$, such that if $g(q)$ is odd, then $q \notin \alpha$. Let $\mathcal{R}$ be the set of all level rankings. Any ranking r of $dag(w)$ can be decomposed into an infinite sequence $lr_1, lr_2, \ldots$ of level rankings by defining $lr_i(q) = r(\langle q, i\rangle)$ if $\langle q, i\rangle$ is a node of $dag(w)$, and $lr_i(q) = \perp$ otherwise. This allows us to look at a ranking as an infinite word over the alphabet of level rankings. For example, if we represent a level ranking lr of our running example by the column vector

$$\begin{bmatrix} lr(q_0) \\ lr(q_1) \end{bmatrix},$$

then the rankings of Figure 4.9 correspond to the words

$$\begin{bmatrix} 2 \\ \perp \end{bmatrix}\begin{bmatrix} \perp \\ 2 \end{bmatrix}\begin{bmatrix} 1 \\ \perp \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\omega}$$

$$\begin{bmatrix} 1 \\ \perp \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\left(\begin{bmatrix} 0 \\ \perp \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)^{\omega}$$

For two level rankings g and g', we say that g' *covers* g if for all q and q' in Q, if $g(q) \geq 0$ and $q' \in \delta(q, \sigma)$, then $0 \leq g'(q') \leq g(q)$.

The automaton $comp(\mathcal{A}) = (\overline{\Sigma}, \overline{Q}, \overline{q}_0, \overline{\delta}, \overline{\alpha})$ is formally defined as follows:

- $\overline{\Sigma} = \Sigma$, $\overline{Q} = \mathcal{R} \times 2^Q$, $\overline{\alpha} = \mathcal{R} \times \emptyset$.
- $\overline{Q}_0 = \langle lr_0, \emptyset\rangle$, where $lr_0(q) = 2n$ if $q \in Q_0$, and $lr_0(q) = \perp$ otherwise.

- $\langle lr', P' \rangle \in \overline{\delta}(\langle lr, P \rangle, a)$ iff $\langle lr', P' \rangle$ satisfies the following three conditions:
  - lr' covers lr;
  - if $P \neq \emptyset$, then $P' = \{q' \in Q \mid \ q' \in \delta(q, a)$ for some $q \in P$ and $lr'(q')$ is even$\}$;
  - if $P = \emptyset$, then $P' = \{q' \in Q \mid lr'(q')$ is even$\}$.

Thus, when *comp*$(\mathcal{A})$ reads the lth letter in the input, for $l \geq 1$, it guesses the level ranking for level l in the run DAG. This level ranking should cover the level ranking of level $l-1$. In addition, in the P component, *comp*$(\mathcal{A})$ keeps track of states whose corresponding nodes at the current level have even ranks. Paths that traverse such nodes should eventually reach a node with an odd rank. When all the paths of the DAG have visited a node with an odd rank, the set P becomes empty, and is initiated by new obligations for visits in odd ranks according to the current level ranking. The acceptance condition $\mathcal{R} \times \{\emptyset\}$ then checks that there are infinitely many levels in which all the obligations have been fulfilled.

Exercise 26. Construct the automata *comp*$(\mathcal{A})_1$ and *comp*$(\mathcal{A})_2$ for $(\{a\}, \{q\}, \{q\}, \delta, \{q\})$ and $(\{a\}, \{q\}, \{q\}, \delta, \emptyset)$, where $\delta(q, a) = \{q\}$.

The size of *comp*$(\mathcal{A})$

Let n be the number of states of $\mathcal{A}$. Since a level ranking is a mapping $lr\colon Q \rightarrow \{\perp\} \cup [0, 2n]$, there are at most $(2n+2)^n$ level rankings. So *comp*$(\mathcal{A})$ has at most $(2n+2)^n \cdot 2^n \in n^{O(n)}$ states. Since $n^{O(n)} = 2^{O(n \cdot \log n)}$, we have introduced an extra $\log n$ factor in the exponent with respect to the subset construction for automata on finite words. We show that this factor is unavoidable. We exhibit a family $\{L_n\}_{n \geq 1}$ of infinitary languages such that L_n is accepted by an automaton with $n+2$ states and any Büchi automaton accepting the complement of L_n has at least $n! \in n^{O(n)}$ states. The family is due to Michel [**?**].

Let $\Sigma_n = \{1, \ldots, n, \#\}$. We associate to a word $w \in \Sigma_n^\omega$ the following directed graph $G(w)$: the nodes of $G(w)$ are $1, \ldots, n$ and there is an edge from i to j if w contains infinitely many occurrences of the word ij. Define L_n as the language of infinite words $w \in A^\omega$ for which $G(w)$ has at least a cycle and define *comp*$(L)_n$ as the complement of L_n.

Exercise 27. Show that for all $n \geq 1$, L_n is recognized by a Büchi automaton with $n+2$ states.

Proposition 4.1.6 *For all $n \geq 1$, if a Büchi automaton recognizes $comp(L)_n$, then it has at least $n!$ states.*

Proof: We need some preliminaries. Given a permutation $\tau = \langle \tau(1), \ldots, \tau(n) \rangle$ of $\langle 1, \ldots, n \rangle$, we identify τ and the word $\tau(1) \ldots \tau(n)$. We make two observations:

(a) $(\tau\#)^\omega \in comp(L)_n$ for every permutation τ.
The edges of $G((\tau\#)^\omega)$ are $\langle \tau(1), \tau(a_2) \rangle$, $\langle \tau(a_2), \tau(a_3) \rangle$, ..., $\langle \tau(a_{n-1}), \tau(a_n) \rangle$, and so $G((\tau\#)^\omega)$ is acyclic.

(b) If a word w contains infinitely many occurrences of two different permutations τ and τ' of $1 \ldots n$, then $w \in L_n$.
Since τ and τ' are different, there are i and j in $\{1, \ldots, n\}$ such that i precedes j in τ and j precedes i in τ'. Since w contains infinitely many occurrences of τ, $G(w)$ has a path from i to j. Since it contains infinitely many occurrences of τ', $G(w)$ has a path from j to i. So $G(w)$ contains a cycle, and so $w \in L_n$.

Now, let $\mathcal{A}$ be a Büchi automaton recognizing $comp(L)_n$, and let τ, τ' be two arbitrary permutations of $(1, \ldots, n)$. By (a), there exist runs ρ and ρ' of A accepting $(\tau\#)^\omega$ and $(\tau'\#)^\omega$, respectively. We prove that the intersection of $\inf(\rho)$ and $\inf(\rho')$ is empty. This implies that A contains at least as many final states as permutations of $(1, \ldots, n)$, which proves the Proposition.

We proceed by contradiction. Assume $q \in \inf(\rho) \cap \inf(\rho')$. We build an accepting run ρ'' by "combining" ρ and ρ' as follows:

(0) Starting from the initial state of ρ, go to q following the run ρ.

(1) Starting from q, follow ρ' until having gone through a final state, and having read at least once the word τ'; then go back to q (always following ρ').

(2) Starting from q, follow ρ until having gone through a final state, and having read at least once the word τ; then go back to q (always following ρ).

(3) Go to (1).

The word accepted by ρ'' contains infinitely many occurrences of both τ and τ'. By (b), this word belongs to L_n, contradicting the assumption that $\mathcal{A}$ recognizes $comp(L)_n$. $\square$