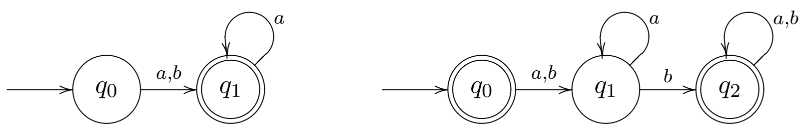
*Example* 3.8. In the preceding example, only the change of final states is necessary to recognize the complement: $0$ instead of $01$. Nonetheless, the completion is often necessary as in the following case where $q_2$ stores the garbage which is a part of the complement.




**Theorem 3.9.** *(McNaughton's Theorem) A Büchi automaton can be transformed effectively into an equivalent deterministic Muller automaton.*

*Proof.* The classical subset construction does not work in the case of infinite words, since passing infinitely often through a macrostate $M$ that contains a final state $f$ does not imply that we pass infinitely often through this final state $f$ – we may pass through the states in $M \setminus \{f\}$. Therefore, an improved version of this construction, called *Safra determinization*, is presented.

Assume the following run, where $u_i, v_i$ are words, the states here are actually the macrostates, i.e. subsets of states, with the macrostate transitions; $F_i$ are subsets of final states:

$$\begin{array}{ccccccccccc} R_0 & \overset{u_1}{\rightsquigarrow} & Q_1 & \overset{v_1}{\rightsquigarrow} & R_1 & \overset{u_2}{\rightsquigarrow} & & \overset{u_i}{\rightsquigarrow} & Q_i & \overset{v_i}{\rightsquigarrow} & R_i \\ & & \cup\mid & & \mid\mid & & \cdots & & \cup\mid & & \mid\mid \\ & & F_1 & \overset{v_1}{\rightsquigarrow} & G_1 & & & & F_i & \overset{v_i}{\rightsquigarrow} & G_i \end{array}$$

We call the situation $R_i = G_i$ a *breakpoint*. Clearly, any state in $R_i$ is reachable from $R_0$, moreover, passing $i$ times through $F$. Particularly, if we choose $R_0 = \{q_0\}$ and if we have such a sequence for all $i > 0$, then there is a successful run on $u_1 v_1 u_2 v_2 \cdots$. Indeed, by choosing corresponding runs for every $i$ and $q \in R_i$, we get an infinite tree with finite branching and we conclude by König's Lemma.

Thus, an infinite sequence of breakpoints can serve to detect a successful run, we will show it to be complete as well.

Obviously, the idea of remembering the subset of $Q_i$ has to be applied recursively. Here, the notion of Safra tree arises, these will be the states of the new deterministic automaton. It is a tree with macrostates as nodes and the properties as follows. The key point is that an own thread of macrostates is split off whenever final states are encountered on all the levels. We describe the action of a letter on a Safra tree.

To proceed to the next Safra tree in the run, we apply the subset construction to each macrostate in the tree, particularly the root always contains the momentarily reachable states. And then, for each macrostate in the tree that contains some final states, we introduce his new (youngest, i.e. rightmost) son containing these final states. Note that in such a way, sons are subsets of their fathers.

In this way, the trees would grow without any bounds. Therefore, we introduce two merges. Firstly, we merge horizontally: we erase a state from macrostate if it is contained in its older brother. And if a macrostate is then left empty, we delete it. Thus, brothers are disjoint $(*)$. Secondly, we merge vertically: if the union of sons is the whole father, we delete the sons (including their subtrees) – this is our breakpoint. Thus, the union of sons is a proper subset of the father $(**)$.

We show the tree is of size at most $|Q|$ (the set of the original states). Each macrostate contains at least one state that is not included in his sons (by $(**)$), we call the set of these states the remainder. Thus, the remainders are nonempty. The remainders for different macrostates are disjoint by $(*)$ and $(**)$. Hence, the number of remainders is bounded by $|Q|$.

Given the Büchi automaton $B$, we construct the Muller automaton $M$ as follows. The states are the Safra trees, where we distinguish at each node whether it is marked or not (the marked macrostates will be the breakpoints). Each node (carrying a subset of states and the flag of being marked) is identified with a number from $\{1, \ldots, 2m\}$. The numbers are inherited from the previous tree in the run, some nodes may be deleted, some nodes are new, these have to get their number. However, the continuity of subsequentness in the run must be preserved, i.e. if in a successive sequence $(s_i)$ of Safra trees there is a node named $k$ in each of them (say with respective subsets $R_i$ for $s_i$), we need a thread of states $q_i \in R_i$ that form a run. Therefore, a new node must not get the number of any node from the previous tree (not even from the just deleted nodes), that is why we need numbers up to twice the maximum size of a tree. For a new node, we choose the least momentarily free number.

The construction begins with the tree with only one node $\{q_0\}$, we inductively follow with adding the youngest sons with final states, applying the subset construction, applying the horizontal merge and then the vertical merge where the parents of the deleted sons get marked (as they are breakpoints).

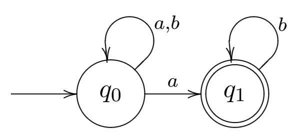
A set of Safra trees $S$ is set to be in the system $\mathcal{F}$ if there is a node $k$ in each Safra tree in $S$ and at least one of these occurrences is marked.

The proof of equivalence of both automata:

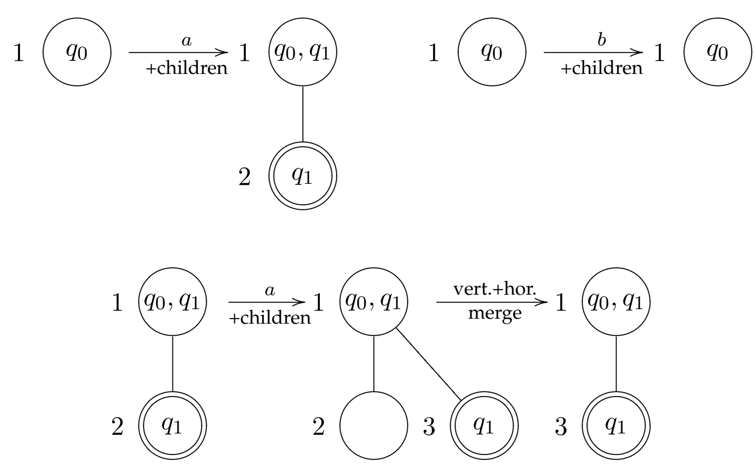
Firstly, if $M$ accepts $w$, we have a run where from some point onwards some node $k$ is always present and it is marked infinitely often, we get a sequence of breakpoints discussed above, whence the correctness.

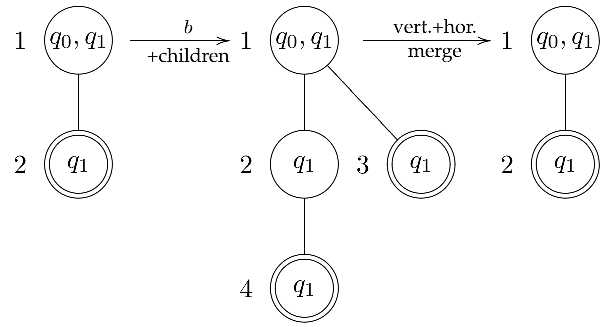
Secondly, if $B$ accepts $w$, we have a run passing some $q \in F$ infinitely often. Consider the corresponding Safra tree run. If the root is marked infinitely often (let us call that stabilization), $M$ accepts. Otherwise, after the last occurrence in the root, it must be put into its son. If it stabilizes, we are done. Otherwise, by horizontal merge it moves to the older brothers . If it stabilizes in some of them, we are done. If it does not, then (as there are only finitely many of them), there must be a last occurrence in this level in some son and we proceed by induction as we did with the root. By the finiteness of the trees, there must be a grandson of the root where we stabilize. This implies acceptance, whence the completeness. □

*Example* 3.10. As to the automaton from the Example 3.4 that recognizes words with finitely many $a$'s, the determinization results in the same automaton as the determinization of this automaton on *finite* words, see the Example 3.6. Therefore, we illustrate the whole process on its modification that requires moreover at least one occurrence of $a$.

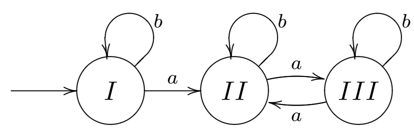



For each tree, we depict the corresponding subset transition together with adding the children, where final states occur. The next step is the horizontal and then the vertical merge.






We have obtained new states which are trees with the respective nodes: $I$:1, $II$:1, 2, $III$:1, 3. The transitions for the tree $III$ are obtained in the same way as those for $II$. We sum up the transitions by depicting the new *deterministic* automaton.




What is the Muller condition? For the marked state 2, there is only one tree containing it – the tree $II$. Similarly for 3 and $III$. Therefore, $\mathcal{F} = \{\{II\}, \{III\}\}$.

**Proposition 3.11.** *A Muller automaton can effectively be transformed into an equivalent Büchi automaton.*

*Proof.* At the beginning, the Büchi automaton guesses non-deterministically which $F \in \mathcal{F}$ is correct for the Muller automaton. Then it reads until the last symbol from $Q \setminus F$ (guessed again) and then checks if only $F$ states follow and if they are all infinitely many times encountered. For the latter, in the state we keep a subset of $F$ and when we encounter a state from $F$ we add it to it. When the subset is equal to $F$ – this is our final state – we make our set empty and start again. Clearly, we accept iff the whole $F$ is visited infinitely many times. □

**Proposition 3.12.** *$\mathcal{R}(A^\omega)$ is closed under complement.*

*Proof.* Given $L \in \mathcal{R}(A^\omega)$, we transform the corresponding Büchi automaton to the deterministic Muller automaton $(Q, A, \Delta, i, \mathcal{F})$ by the McNaughton's theorem. We make this automaton complete (by adding a new state where the undefined transitions will end). And for final states we take $\mathcal{P}(Q) \setminus F$. As