§4. **The generalization of Shelah – Stup theorem.**

A *signature* is a finite set of predicate symbols; each symbol has its own dimension (= number of arguments).

A *structure* of signature $\sigma = (\mathbf{P}_1, \ldots, \mathbf{P}_n)$ is a tuple $\langle \mathbf{D}, \bar{\mathbf{P}}_1, \ldots \bar{\mathbf{P}}_n \rangle$, where $\mathbf{D}$ is a non-empty set (the universum of the structure) and $\bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_n$ are relations on $\mathbf{D}$ such that the dimension of $\bar{\mathbf{P}}_i$ equals to the dimension of $\mathbf{P}_i$ for each $\mathbf{i} \leq \mathbf{n}$.

Let us define the notion of *tree counterpart* of a structure $\langle \mathbf{D}; \bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_n \rangle$. This is a structure of signature $\bar{\sigma} = (\mathbf{P}_1, \ldots, \mathbf{P}_n, \mathbf{Q})$ where $\mathbf{Q}$ is a new binary predicate symbol. Its universum is the set $\mathbf{D}^*$ of all the words over the alphabet $\mathbf{D}$. Similar to the case of binary tree, we call $\mathbf{D}^*$ the *tree of* $\mathbf{D}$*-branching, or the tree of branching* $\mathbf{D}$ (e.g., binary tree is the tree of branching $\{\mathbf{L}, \mathbf{R}\}$). The elements of $\mathbf{D}$ are called *directions.* Let us define the interpretations $\bar{\bar{\mathbf{P}}}_1, \ldots, \bar{\bar{\mathbf{P}}}_n, \bar{\bar{\mathbf{Q}}}$ of predicate symbols $\mathbf{P}_1, \ldots, \mathbf{P}_n, \mathbf{Q}$ in the tree counterpart. The predicate symbol $\mathbf{Q}$ is interpreted as the relation "to be a son", i.e., $\bar{\bar{\mathbf{Q}}}(\mathbf{u}, \mathbf{v})$ is true if $\mathbf{u} = \mathbf{vd}$ for some $\mathbf{d} \in \mathbf{D}$. The predicate symbol $\mathbf{P}_i$ is interpreted as the relation $\bar{\bar{\mathbf{P}}}_i$ which is true on a tuple $\langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$ (where $\mathbf{k}$ is the dimension of $\mathbf{P}_i$) if there exist $\mathbf{v} \in \mathbf{D}^*$ and $\mathbf{d}_1, \ldots, \mathbf{d}_k \in \mathbf{D}$ such that $\mathbf{v}_1 = \mathbf{vd}_1$, $\mathbf{v}_2 = \mathbf{vd}_2, \ldots, \mathbf{v}_k = \mathbf{vd}_k$ and $\bar{\mathbf{P}}_i(\mathbf{d}_1, \ldots, \mathbf{d}_k)$ is true. In other words, $\bar{\bar{\mathbf{P}}}_i(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ is true if $\mathbf{v}_1, \ldots, \mathbf{v}_k$ "are in the same fan" and their last letters satisfy $\bar{\mathbf{P}}_i$.

Let us give the definition of the monadic theory of a structure. In §**1**, this definition was given for the structure $\langle$binary tree; $\mathbf{L}$, $\mathbf{R}\rangle$. Let $\mathbf{S} = \langle \mathbf{M}; \bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_n \rangle$ be a structure of signature $\sigma = (\mathbf{P}_1, \ldots, \mathbf{P}_n)$. The monadic language contains both individual and set variables, its atomic formulae are of the form $\mathbf{x} \in \mathbf{Q}$ and $\mathbf{P}_i(\mathbf{x}_1, \ldots, \mathbf{x}_k)$, where $\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_k$ are individual variables, $\mathbf{Q}$ is a set variable and $\mathbf{P}_i$ is a $\mathbf{k}$-ary predicate symbol in σ. Both individual and set variables can be bounded by quantifiers in formulae of the language. The monadic theory of $\mathbf{S}$ consists of all closed formulae of the language true in the structure $\mathbf{S}$.

Shelah and Stup [?] proved the following generalization of Rabin's theorem.

Theorem of Shelah and Stup. *If the monadic theory of a structure is decidable, then the monadic theory of its tree counterpart is decidable also.*

Theorem of Rabin is a consequence of Theorem of Shelah and Stup, as theory **S2S** is equivalent to the monadic theory of the tree counterpart of the structure $\langle \{\mathbf{L}, \mathbf{R}\}$; unary relation "to be equal to $\mathbf{L}$"$\rangle$.

Our goal is to strengthen Theorem of Shelah and Stup by enriching the signature of tree counterpart. Let us include in the signature of tree counterpart the unary relation $\bar{\bar{\mathbf{R}}}(\mathbf{u})$ which means that $\mathbf{u} = \mathbf{wdd}$ for some $\mathbf{w} \in \mathbf{D}^*$ and $\mathbf{d} \in \mathbf{D}$, i.e., the last letter of $\mathbf{u}$ and the last but one letter of $\mathbf{u}$ are equal. Thus, the tree counterpart of a structure $\langle \mathbf{D}; \bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_n \rangle$ in the new sense is the structure $\langle \mathbf{D}^*; \bar{\bar{\mathbf{P}}}_1, \ldots, \bar{\bar{\mathbf{P}}}_n, \bar{\bar{\mathbf{Q}}}, \bar{\bar{\mathbf{R}}} \rangle$ of signature $(\mathbf{P}_1, \ldots, \mathbf{P}_n, \mathbf{Q}, \mathbf{R})$.

Note that if $\mathbf{k}$ is the dimension of $\mathbf{P}_i$, then, in enriched language, we can express that vertices $\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}$ are in the same fan and their last letters and their common last but one letter satisfy $\bar{\mathbf{P}}_i$, as well as other relations of this type.

Theorem 6. *If the monadic theory of a structure is decidable, then the monadic theory of its tree counterpart in the new sense is decidable also.*

Let us mention that this theorem cannot be strengthened by means of including in the signature the unary relation "the last letter of $\mathbf{u}$ and the last but two letter of $\mathbf{u}$ are equal". Moreover, we even cannot replace $\bar{\bar{\mathbf{R}}}$ with this relation. Namely, if the initial structure is $\mathbf{S} = (\mathbb{N}; <)$, the monadic theory of which is known to be decidable, and we enrich the tree counterpart of $\mathbf{S}$ (in the first sense) by the unary relation "the last letter and the last but two letter are equal", then we get a structure with undecidable monadic theory. Indeed, in the monadic theory of the resulting structure we can simulate any machine, having two counters, i.e., for a given machine with two counters and its input we can construct a monadic formula which is true in the resulting structure iff the given machine halts on the given input. It is well known that machines with two counters can simulate Turing machines.

Proof of Theorem 6. The proof is similar to our proof of Rabin's theorem.

For every notion used in that proof we give its analog.

A Σ-tree of $\mathbf{D}$-branching (where Σ is an alphabet) is any mapping from $\mathbf{D}^*$ into Σ. Analogous to the notion of an automaton on binary Σ-trees, we could give the notion of an automaton on Σ-trees of $\mathbf{D}$-branching. However, we need a notion of automaton different from that natural generalization. Namely, in the automaton of the new type, the state labeling a son $\mathbf{u}$ of a vertex $\mathbf{v}$ can depend not only on automaton state in vertex $\mathbf{v}$, on the label of $\mathbf{v}$ in the tree and on the last letter of $\mathbf{u}$ but also on the last letter of $\mathbf{v}$ (if $\mathbf{v}$ is not empty). Let us give the formal definition. Compared with automata on binary Σ-trees, only the notion of a transition and the notion of a table of transitions are changed. Now, a *transition* is a quadruple $\langle \mathbf{s}, \mathbf{a}, \mathbf{d}, \mathbf{f} \rangle$, where $\mathbf{s}$ is a state, $\mathbf{a} \in \mathbf{\Sigma}$, $\mathbf{d} \in \mathbf{D}$ and $\mathbf{f}$ is a function from $\mathbf{D}$ into the set of automaton states. A set $\mathbf{T}$ of transitions is called monadic definable if there is a monadic formula $\varphi(\mathbf{s}, \mathbf{a}, \mathbf{d}, \mathbf{f})$ such that a transition $\langle \mathbf{s}, \mathbf{a}, \mathbf{d}, \mathbf{f} \rangle$ belongs to $\mathbf{T}$ iff $\varphi(\mathbf{s}, \mathbf{a}, \mathbf{d}, \mathbf{f})$ is true in $\mathbf{D}$.

To make this definition meaningful we enrich the monadic language as follows:

(1) we include variables ranging Σ and all elements from Σ as constants;

(2) we include variables ranging $\mathbf{S}$ (the set of states) and all elements from $\mathbf{S}$ as constants;

(3) we include variables ranging the set of functions from $\mathbf{D}$ into $\mathbf{S}$; for any such variable $\mathbf{f}$, for any individual variable $\mathbf{x}$ (i.e. ranging $\mathbf{D}$) and for any $\mathbf{s} \in \mathbf{S}$ we allow the expression $\mathbf{f}(\mathbf{x}) = \mathbf{s}$;

(4) we include equality relation;

(5) we allow to bound new variables by quantifiers.

Let us note that if the monadic theory of a structure is decidable, then the theory of that structure in enriched monadic language is also decidable, since given a closed formula in enriched monadic language we can construct an equivalent formula in monadic language.

So, we require the table of transitions to be monadic definable.

As earlier, let us call a *transition of automaton* $\mathfrak{A}$ any transition in the table of transitions of $\mathfrak{A}$. Let us call an automaton $\mathfrak{A}$ *correct* if its transitions having the initial state $\mathbf{s}_0$ at the bottom don't depend on the last letter of vertex, i.e., if for all $\mathbf{a} \in \mathbf{\Sigma}$, $\mathbf{d}, \mathbf{d}' \in \mathbf{D}$ and $\mathbf{f} \colon \mathbf{D} \longrightarrow \mathbf{S}$,
$\langle \mathbf{s}_0, \mathbf{a}, \mathbf{d}, \mathbf{f} \rangle$ is a transition of $\mathfrak{A} \iff \langle \mathbf{s}_0, \mathbf{a}, \mathbf{d}', \mathbf{f} \rangle$ is a transition of $\mathfrak{A}$.

For a labelled tree $\mathbf{G}$ and for a vertex $\mathbf{x}$, we denote by $\mathbf{G}(\mathbf{x})$ the label of $\mathbf{x}$ in $\mathbf{G}$. Let us define the notion of a run of a correct automaton $\mathfrak{A}$ on a Σ-tree $\mathbf{T}$ of $\mathbf{D}$-branching. This is a $\mathbf{S}$-tree $\mathbf{H}$ of $\mathbf{D}$-branching such that

1) $\mathbf{H}(\lambda) = \mathbf{s}_0$;

2) for any $\mathbf{x} \in \mathbf{D}^*$, the transition $\langle \mathbf{H}(\mathbf{x}), \mathbf{T}(\mathbf{x}), \mathbf{d}, \mathbf{f} \rangle$ belongs to the table of transitions of $\mathfrak{A}$, where $\mathbf{d}$ and $\mathbf{f}$ are defined as follows: $\mathbf{d}$ is the last letter of $\mathbf{x}$ (and any element of $\mathbf{D}$ if $\mathbf{x} = \lambda$) and $\mathbf{f}$ is the function from $\mathbf{D}$ into $\mathbf{S}$ such that $\mathbf{f}(\mathbf{d}') = \mathbf{H}(\mathbf{x}\mathbf{d}')$; the transition $\langle \mathbf{H}(\mathbf{x}), \mathbf{T}(\mathbf{x}), \mathbf{d}, \mathbf{f} \rangle$ is called the *transition of* $\mathbf{H}$ *in* $\mathbf{x}$ (when $\mathbf{T}$ is clear from the context).

Now, similar to the case of binary tree, we have to prove that any monadic definable set of Σ-trees of $\mathbf{D}$-branching can be recognized by a correct automaton on Σ-trees of $\mathbf{D}$-branching. The proof runs as the proof of Theorem 1. As it was earlier, only one point is nontrivial: to prove that the complement of any recognizable set of Σ-trees of $\mathbf{D}$-branching is recognizable.

A correct semiautomaton on Σ-trees of $\mathbf{D}$-branching is a pair $\langle$a correct automation $\mathfrak{L}$ on Σ-trees of $\mathbf{D}$-branching; a (nondeterministic) automaton $\mathfrak{B}$ on ω-words over the set of $\mathfrak{L}$'s states$\rangle$. The semiautomation accepts a Σ-tree $\mathbf{T}$ of $\mathbf{D}$-branching if there is a run of $\mathfrak{L}$ on $\mathbf{T}$ such that all the sequences of $\mathfrak{L}$'s states lying along infinite paths in that run are rejected by $\mathfrak{B}$.

Let us prove first that the complement of any recognizable (by a correct automaton) set of Σ-trees of $\mathbf{D}$-branching is recognizable by a correct semiautomaton. Again we define the notion of a strategy with memory. The only difference with old definition is that now the strategy distributes all possible (in a vertex) copy-transitions among elements of $\mathbf{D}$ and does not divide them into left and right. The analog of Theorem 4 is proved in the similar way. The only difficulty arises in the proof of the analog of Theorem 5, i.e., in the proof that for any correct automaton $\mathfrak{A}$ and for any strategy set $\mathbf{M}$, the set of Σ-trees of $\mathbf{D}$-branching on which there is a rejecting strategy for $\mathfrak{A}$ based on $\mathbf{M}$, is semirecognizable. Let us present the proof of this statement in detail. To understand the following better, we advise the reader to recall the notions of a strategy with memory, of a strategy set and of a probable path.

Let us remind that in binary case the states of the semiautomaton recognizing the set of Σ-trees on which there is a rejecting strategy, were pairs of disjoint sets of copy-transitions of the initial automaton. In our case, if $\mathbf{D}$ is infinite, then the number of such pairs is infinite. Therefore, we have to take another semiautomaton. Let us define its first component $\mathfrak{L}$ (a correct automaton on Σ-trees of $\mathbf{D}$-branching). The states of $\mathfrak{L}$ are subsets of $\mathbf{M} \times \mathbf{M}$ (recall that $\mathbf{M}$ is the strategy set). The initial state is the set $\{\langle \mathbf{m}_0, \mathbf{m}_0 \rangle\}$, where $\mathbf{m}_0$ is the initial copy of initial state of automaton $\mathfrak{A}$. Let us define the table of transitions of $\mathfrak{L}$. Let $\langle \mathbf{A}, \mathbf{a}, \mathbf{d}, \mathbf{F} \rangle$ be a transition, where $\mathbf{A} \subset \mathbf{M} \times \mathbf{M}$, $\mathbf{a} \in \mathbf{\Sigma}$, $\mathbf{d} \in \mathbf{D}$, and $\mathbf{F}\colon \mathbf{D} \longrightarrow \mathbf{P}(\mathbf{M} \times \mathbf{M})$. Let us define in what case it belongs to the table of transitions. Let $\mathbf{G}$ be a function from $\mathbf{M} \times \mathbf{\Sigma} \times \mathbf{D} \times \mathbf{S}^{\mathbf{D}}$ into $\mathbf{D}$ (the meaning is that $\mathbf{G}$ attaches a direction to every copy-transition). Denote by $\mathbf{G}_{\mathbf{A},\mathbf{a},\mathbf{d}}$ the following function from $\mathbf{D}$ into $\mathbf{P}(\mathbf{M} \times \mathbf{M})$. Its value on $\mathbf{d}' \in \mathbf{D}$ consists of all pairs $\langle \mathbf{m}, \mathbf{f}(\mathbf{d}') \rangle$ such that $\langle \mathbf{m}, \mathbf{a}, \mathbf{d}, \mathbf{f} \rangle$ is a transition of $\mathfrak{A}$ for which $\mathbf{G}(\langle \mathbf{m}, \mathbf{a}, \mathbf{d}, \mathbf{f} \rangle) = \mathbf{d}'$ and $\mathbf{m}$ belongs to the second projection of $\mathbf{A}$.

Formally, $\mathbf{G}_{\mathbf{A},\mathbf{a},\mathbf{d}}(\mathbf{d}') = \Big\{\langle \mathbf{m}, \mathbf{f}(\mathbf{d}')\rangle \mid \langle \mathbf{m}, \mathbf{a}, \mathbf{d}, \mathbf{f}\rangle$ is a transition of $\mathfrak{A}$, $\mathbf{G}(\langle \mathbf{m}, \mathbf{a}, \mathbf{d}, \mathbf{f}\rangle) = \mathbf{d}'$ and there is $\mathbf{m}' \in \mathbf{M}$ such that $\langle \mathbf{m}', \mathbf{m}\rangle \in \mathbf{A}\Big\}$.

So, a quadruple $\langle \mathbf{A}, \mathbf{a}, \mathbf{d}, \mathbf{F}\rangle$ is a transition of $\mathfrak{L}$ if there is a function $\mathbf{G}$ such that $\mathbf{G}_{\mathbf{A},\mathbf{a},\mathbf{d}}(\mathbf{d}') \subseteq \mathbf{F}(\mathbf{d}')$ for all $\mathbf{d}' \in \mathbf{D}$. Clearly, $\mathfrak{L}$ is correct.

Let us prove that the property of a quadruple $\langle \mathbf{A}, \mathbf{a}, \mathbf{d}, \mathbf{F}\rangle$ "to be a transition of $\mathfrak{L}$" is monadic definable. Indeed, this property can be expressed by the formula $\varphi(\mathbf{A}, \mathbf{a}, \mathbf{d}, \mathbf{F})$ which states that for every copy-transition of $\mathfrak{A}$ of the form $\langle \mathbf{m}, \mathbf{a}, \mathbf{d}, \mathbf{f}\rangle$, where $\mathbf{m}$ belongs to the second projection of $\mathbf{A}$, there is a $\mathbf{d}' \in \mathbf{D}$ such that the pair $\langle \mathbf{m}, \mathbf{f}(\mathbf{d}')\rangle$ belongs to $\mathbf{F}(\mathbf{d}')$.

Let us construct the second component of the semiautomaton, i.e., the automaton $\mathfrak{B}$ on ω-words over the set of states of $\mathfrak{L}$. Let us define when $\mathfrak{B}$ accepts a sequence $\alpha = (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_k, \ldots)$ of states of $\mathfrak{L}$. Let us call a sequence $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k, \ldots$ of states of $\mathfrak{A}$ *possible for* α if there exist copies $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_k, \ldots$ of the states $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k, \ldots$, respectively, such that $\langle \mathbf{m}_0, \mathbf{m}_1\rangle \in \mathbf{A}_1$, $\langle \mathbf{m}_1, \mathbf{m}_2\rangle \in \mathbf{A}_2, \ldots, \langle \mathbf{m}_{k-1}, \mathbf{m}_k\rangle \in \mathbf{A}_k, \ldots$. So, $\mathfrak{B}$ accepts α if there is a sequence $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k, \ldots$ having final limit macrostate which is possible for α. It is easy to construct an automaton on ω-words satisfying this requirement.

Let us prove that the defined semiautomaton accepts a Σ-tree $\mathbf{T}$ of $\mathbf{D}$-branching iff there is a rejecting strategy based on $\mathbf{M}$ for $\mathfrak{A}$ on $\mathbf{T}$. Assume that there is a rejecting strategy based on $\mathbf{M}$ for $\mathfrak{A}$ on a Σ-tree $\mathbf{T}$ of $\mathbf{D}$-branching. Let us define the following run of the semiautomaton on $\mathbf{T}$: in every vertex $\mathbf{x}$, take the transition $\langle \mathbf{A}, \mathbf{a}, \mathbf{d}, \mathbf{G}_{\mathbf{A},\mathbf{a},\mathbf{d}}\rangle$ where $\mathbf{d}$ is the last letter of $\mathbf{x}$, $\mathbf{a}$ is the label of $\mathbf{x}$ and $\mathbf{G}$ is the function from $\mathbf{M} \times \mathbf{\Sigma} \times \mathbf{D} \times \mathbf{S}^{\mathbf{D}}$ into $\mathbf{D}$ indicating how the rejecting strategy distributes copy-transitions possible in $\mathbf{x}$ among directions (on impossible in $\mathbf{x}$ copy-transitions $\mathbf{G}$ is defined in any way). Obviously, any sequence $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k, \ldots$ that is possible for a sequence $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_k, \ldots$ of states of the semiautomaton lying along an infinite path in this run, is a probable path of the rejecting strategy and therefore has non-final limit macrostate. Hence, the defined run is an accepting run.

Conversely, if there is an accepting run $\mathbf{H}$ of the semiautomaton on a Σ-tree of $\mathbf{D}$-branching, then there is a rejecting strategy on that tree (in a vertex $\mathbf{x}$, the rejecting strategy distributes transitions among directions according to the function $\mathbf{G}$ for which $\forall \mathbf{d}'\ \mathbf{G}_{\mathbf{A},\mathbf{a},\mathbf{d}}(\mathbf{d}') \subseteq \mathbf{F}(\mathbf{d}')$, where $\langle \mathbf{A}, \mathbf{a}, \mathbf{d}, \mathbf{F}\rangle$ is the transition of $\mathbf{H}$ in $\mathbf{x}$).

Just as it was done in Theorem 1, we can prove that any set recognizable by a correct semiautomaton on Σ-trees of $\mathbf{D}$-branching is recognizable by a correct automaton on Σ-trees of $\mathbf{D}$-branching.

Thus, we have completed the proof of the analog of Theorem 1. To complete the proof of Theorem 6 we have to construct an algorithm deciding whether a correct automaton $\mathfrak{A}$ on Σ-trees of $\mathbf{D}$-branching recognizes the empty set, where Σ consists of a single letter. There is unique Σ-tree of $\mathbf{D}$-branching (for one-element Σ), call it simply the *tree of* $\mathbf{D}$-*branching*; we may think, that its vertices have no labels. Therefore we may delete the second component from transitions of $\mathfrak{A}$. The algorithm recognizing emptiness is more complicated compared with binary case. We are going to construct it.

Let us consider arbitrary automata on the tree of $\mathbf{D}$-branching (not only correct ones). Moreover, let us consider automata with dead ends from a set Δ, which are defined in usual way. Let $\mathbf{d}_0$ be a direction. Let us define the notion of a $\mathbf{d}_0$-run of an automaton with dead ends from Δ on a $\mathbf{P}(\Delta)$-tree of $\mathbf{D}$-branching. Informally speaking, a $\mathbf{d}_0$-run is an ordinary run of the automaton with dead ends on $\mathbf{P}(\Delta)$-tree of $\mathbf{D}$-branching provided the last letter of the empty word is defined to be $\mathbf{d}_0$. Formally, a $\mathbf{d}_0$-run of an automaton $\mathfrak{A}$ with dead ends from Δ on a $\mathbf{P}(\Delta)$-tree of $\mathbf{D}$-branching is any subtree $\mathbf{H}$ of the tree $\mathbf{D}^*$ labelled by letters from $\mathbf{S} \bigcup \mathbf{\Delta}$ (where $\mathbf{S}$ is the set of states of $\mathfrak{A}$) such that

(1) $\mathbf{H}(\lambda)$ is the initial state (recall that $\mathbf{H}(\mathbf{x})$ denotes the label of vertex $\mathbf{x}$ in $\mathbf{H}$);

(2) if $\mathbf{x}$ is a vertex such that $\mathbf{H}(\mathbf{x}) \in \mathbf{S}$, then the vertex $\mathbf{xd}$ is in $\mathbf{H}$ for any $\mathbf{d} \in \mathbf{D}$ and the triple $\langle \mathbf{H}(\mathbf{x}), \mathbf{d}', \mathbf{f} \rangle$ is a transition of $\mathbf{H}$, where $\mathbf{d}' = \mathbf{d}_0$ if $\mathbf{x} = \lambda$ and $\mathbf{d}'$ is the last letter of $\mathbf{x}$ otherwise and $\mathbf{f}(\mathbf{d}) = \mathbf{H}(\mathbf{xd})$ for all $\mathbf{d} \in \mathbf{D}$;

(3) if $\mathbf{x}$ is a vertex such that $\mathbf{H}(\mathbf{x}) \in \Delta$, then $\mathbf{x}$ has no descendants in $\mathbf{H}$.

The notion of an accepting $\mathbf{d}_0$-path is defined similar to the binary case. Let us say that an automaton $\mathfrak{A}$ $\mathbf{d}_0$-*accepts* a tree if there is an accepting $\mathbf{d}_0$-run of $\mathfrak{A}$ on that tree.

A *simple* $\mathbf{P}(\Delta)$-tree is any $\mathbf{P}(\Delta)$-tree of $\mathbf{D}$-branching such that for any vertex $\mathbf{x}$, the label of $\mathbf{x}$ depends only on the last letter of $\mathbf{x}$ (if $\mathbf{x}$ is not the root) and the root of the tree is labelled by the empty set. Obviously, every function from $\mathbf{D}$ into $\mathbf{P}(\Delta)$ uniquely defines a simple $\mathbf{P}(\Delta)$-tree.

For any given automaton $\mathfrak{A}$ with dead ends, we construct a monadic formula $\varphi(\mathbf{g}, \mathbf{d})$ in the language of $\mathbf{D}$, where $\mathbf{d} \in \mathbf{D}$ and $\mathbf{g}\colon \mathbf{D} \longrightarrow \mathbf{P}(\Delta)$, which is true in $\mathbf{D}$ iff $\mathfrak{A}$ $\mathbf{d}$-accepts the simple $\mathbf{P}(\Delta)$-tree defined by $\mathbf{g}$. Assume that this is already done. Then to decide whether the given correct automaton (without dead ends) accepts the tree of $\mathbf{D}$-branching we can construct the formula $\varphi(\mathbf{g},\mathbf{d})$ for that automaton, substitute the function $\mathbf{f}(\mathbf{d}) \equiv \emptyset$ for $\mathbf{g}$ and any direction for $\mathbf{d}$ and then decide whether the resulting formula is true (making use of decidability of monadic theory of $\mathbf{D}$).

So let us start with the construction of the formula $\varphi(\mathbf{g},\mathbf{d})$. Let $\mathbf{h}$ be a function from $\mathbf{D}$ into $\mathbf{P}(\Delta \bigcup \mathbf{S})$ and let $\mathbf{g}$ be a function from $\mathbf{D}$ into $\mathbf{P}(\Delta)$. Say that $\mathbf{h}$ *extends* $\mathbf{g}$ if $\mathbf{h}(\mathbf{d}) \bigcap \Delta = \mathbf{g}(\mathbf{d})$ for all $\mathbf{d} \in \mathbf{D}$.

We use the induction on the number of states.

Consider the case of an automaton $\mathfrak{A}$ having single state. Denote that state by $\mathbf{s}$. Consider two subcases.

First subcase: the macrostate $\{\mathbf{s}\}$ is not final.

Say that a function $\mathbf{h}$ from $\mathbf{D}$ into $\mathbf{P}(\Delta \bigcup \{\mathbf{s}\})$ is *downward closed* if $\mathbf{s} \in \mathbf{h}(\mathbf{d})$ for each $\mathbf{d} \in \mathbf{D}$ such that there is a transition $\langle \mathbf{s},\mathbf{d},\mathbf{f}\rangle$ of $\mathfrak{A}$ for which $\forall \mathbf{d}' \in \mathbf{D}\ \mathbf{f}(\mathbf{d}') \in \mathbf{h}(\mathbf{d}')$.

Lemma 6. *The following conditions are equivalent.*

(1) *The automaton* $\mathfrak{A}$ $\mathbf{d}_0$*-accepts the simple* $\mathbf{P}(\Delta)$*-tree defined by the function* $\mathbf{g}$.

(2) $\mathbf{s} \in \mathbf{h}(\mathbf{d}_0)$ *for any function* $\mathbf{h}$ *which is downward closed and extends* $\mathbf{g}$.

Proof. Let us prove first that (2) involves (1). Let (2) be fulfilled.

$$\text{Set } \mathbf{h}(\mathbf{d}) = \begin{cases} \mathbf{g}(\mathbf{d}) \bigcup \{\mathbf{s}\} & \text{if } \mathfrak{A}\ \mathbf{d} - \text{accepts the simple tree} \\ & \text{defined by } \mathbf{g}; \\ \mathbf{g}(\mathbf{d}) & \text{else.} \end{cases}$$

Obviously, $\mathbf{h}$ extends $\mathbf{g}$ and $\mathbf{h}$ is downward closed, therefore $\mathbf{s} \in \mathbf{h}(\mathbf{d}_0)$, i.e., $\mathfrak{A}$ $\mathbf{d}_0$-accepts the simple tree defined by $\mathbf{g}$.

Conversely, let (2) be false, that is, there is a downward closed function $\mathbf{h}$ extending $\mathbf{g}$ for which $\mathbf{s} \notin \mathbf{h}(\mathbf{d}_0)$. Let us prove that any $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$ has an infinite path or a finite path ending in a non-allowed dead end.

For any $\mathbf{x} \in \mathbf{D}^*$, let us define

$$\mathbf{l}(\mathbf{x}) = \begin{cases} \text{the last letter of } \mathbf{x} & \text{if } \mathbf{x} \neq \lambda, \\ \mathbf{d}_0 \quad \text{if } \mathbf{x} = \lambda. \end{cases}$$

Let $\mathbf{H}$ be a $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. Using the induction on $\mathbf{i}$, we define the path $\mathbf{x}_0 = \lambda$, $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots$ such that $\mathbf{H}(\mathbf{x}_i) \notin \mathbf{h}(\mathbf{l}(\mathbf{x}_i))$

for all $\mathbf{i}$. For $\mathbf{i} = 0$ we have $\mathbf{H}(\mathbf{x}_0) = \mathbf{H}(\lambda) = \mathbf{s} \notin \mathbf{h}(\mathbf{l}(\lambda)) = \mathbf{h}(\mathbf{d}_0)$. Let $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_i$ have already been defined and $\mathbf{H}(\mathbf{x}_i) \notin \mathbf{h}(\mathbf{l}(\mathbf{x}_i))$. If $\mathbf{H}(\mathbf{x}_i)$ is a dead end, then that dead end is not allowed as $\mathbf{h}$ extends $\mathbf{g}$. In this case the desired path is already constructed. Otherwise (when $\mathbf{H}(\mathbf{x}_i) = \mathbf{s}$) we have $\mathbf{s} \notin \mathbf{h}(\mathbf{l}(\mathbf{x}_i))$. As $\mathbf{h}$ is downward closed, there is a $\mathbf{d}' \in \mathbf{D}$ such that $\mathbf{H}(\mathbf{x}_i\mathbf{d}') \notin \mathbf{h}(\mathbf{d}')$. Set $\mathbf{x}_{i+1} = \mathbf{x}_i\mathbf{d}'$. Lemma 6 is proved.

Obviously, the property of a function $\mathbf{g}$ and of an element $\mathbf{d}$ stated in item (2) of Lemma 6 is monadic definable, thus the first subcase is completed.

Second subcase: the macrostate $\{\mathbf{s}\}$ is final.

Let us call a function $\mathbf{h}\colon \mathbf{D} \longrightarrow \mathbf{P}(\Delta \bigcup \{\mathbf{s}\})$ *upward closed* if $\mathbf{s} \in \mathbf{h}(\mathbf{d})$ implies that there is a transition $\langle \mathbf{s}, \mathbf{d}, \mathbf{f} \rangle$ of $\mathfrak{A}$ such that $\forall \mathbf{d}' \in \mathbf{D},\ \mathbf{f}(\mathbf{d}') \in \mathbf{h}(\mathbf{d}')$.

Now we need the following

Lemma 7. *The following assertions are equivalent.*

(1) *The automaton* $\mathfrak{A}$ $\mathbf{d}_0$*-accepts the simple* $\mathbf{P}(\Delta)$*-tree defined by the function* $\mathbf{g}$.

(2) $\mathbf{s} \in \mathbf{h}(\mathbf{d}_0)$ *for some function* $\mathbf{h}$ *which is upward closed and extends* $\mathbf{g}$.

Proof. Assume that (1) is true. Let us prove that (2) is true. Let us fix an accepting $\mathbf{d}_0$-run $\mathbf{H}$ of $\mathfrak{A}$ on the simple $\mathbf{P}(\Delta)$-tree defined by $\mathbf{g}$. Define the function $\mathbf{l}(\mathbf{x})$ just as it was done in Lemma 6.
Set

$$\mathbf{h}(\mathbf{d}) = \begin{cases} \{\mathbf{s}\} \bigcup \mathbf{g}(\mathbf{d}) & \text{if there is } \mathbf{x} \in \mathbf{D}^* \text{ such that } \mathbf{d} = \mathbf{l}(\mathbf{x}) \text{ and } \mathbf{H}(\mathbf{x}) = \mathbf{s}, \\ \mathbf{g}(\mathbf{d}) & \text{else.} \end{cases}$$

Obviously, $\mathbf{h}$ extends $\mathbf{g}$, $\mathbf{h}$ is upward closed and $\mathbf{s} \in \mathbf{h}(\mathbf{d}_0)$.

Conversely, let $\mathbf{s}$ be in $\mathbf{h}(\mathbf{d}_0)$ for some upward closed function $\mathbf{h}$ extending $\mathbf{g}$. Let us construct an accepting $\mathbf{d}_0$-run $\mathbf{H}$ of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. For every vertex $\mathbf{x} \in \mathbf{D}^*$, let us define whether $\mathbf{x}$ belongs to $\mathbf{H}$ and if this is the case then let us define the label of $\mathbf{x}$. This will be done by induction on the length of $\mathbf{x}$ in such a way that the label of $\mathbf{x}$ belongs to $\mathbf{h}(\mathbf{l}(\mathbf{x}))$.

Base of induction: the root $\mathbf{x} = \lambda$ belongs to $\mathbf{H}$ and is labelled by $\mathbf{s}$.

Induction step. Assume that for any vertex $\mathbf{x}$ of length not exceeding $\mathbf{n}$, we have made our decision in such a way that the label of $\mathbf{x}$ belongs to $\mathbf{h}(\mathbf{l}(\mathbf{x}))$. Let $\mathbf{x}$ be a vertex of length $\mathbf{n}$. If $\mathbf{x}$ doesn't belong to $\mathbf{H}$ or $\mathbf{x}$ is labelled by a dead end, then all the sons of $\mathbf{x}$ don't belong to $\mathbf{H}$. Assume that $\mathbf{x}$ is labelled by $\mathbf{s}$. Since $\mathbf{s} \in \mathbf{h}(\mathbf{l}(\mathbf{x}))$ and $\mathbf{h}$ is upward closed, there is a transition $\langle \mathbf{s}, \mathbf{l}(\mathbf{x}), \mathbf{f} \rangle$ of $\mathfrak{A}$ such that $\forall \mathbf{d}' \in \mathbf{D}\ \mathbf{f}(\mathbf{d}') \in \mathbf{h}(\mathbf{d}')$. Include all the sons of $\mathbf{x}$ in $\mathbf{H}$ and label the son $\mathbf{x}\mathbf{d}'$ by $\mathbf{f}(\mathbf{d}')$.

Thus, the run $\mathbf{H}$ is defined. All its finite paths have allowed dead ends because if $\mathbf{x}$ is a leaf of $\mathbf{H}$, then the label of $\mathbf{x}$ is a dead end from $\mathbf{g}(\mathbf{l}(\mathbf{x}))$ (as $\mathbf{h}$ extends $\mathbf{g}$). As the macrostate $\{\mathbf{s}\}$ is final, all infinite paths in $\mathbf{H}$ have final limit macrostate, therefore, the run $\mathbf{H}$ is an accepting run. Lemma 7 is proved.

Obviously, the property of a function $\mathbf{g}$ and of an element $\mathbf{d}$ stated in item (2) of Lemma 7 is monadic definable, therefore the case of an automaton with unique state is completed.

Let us turn to automata having more than one state. Let an automaton $\mathfrak{A}$ have $\mathbf{n}$ states $0, 1, \ldots, \mathbf{n}-1$ and let 0 be the initial state. Let Δ be the set of dead ends of $\mathfrak{A}$. Consider two subcases.

First subcase: the macrostate $\mathbf{S} = \{0, 1, \ldots, \mathbf{n}-1\}$ is not final. We use the automaton $\mathfrak{B}_i$, where $\mathbf{i} \in \mathbf{S}$, introduced in §2. Recall that the set of states of $\mathfrak{B}_i$ is $\mathbf{S} \setminus \{\mathbf{i}+1\}$, where the addition is modulo $\mathbf{n}$, the initial state of $\mathfrak{B}_i$ is $\mathbf{i}$; $\mathfrak{B}_i$ has all the transitions of $\mathfrak{A}$ except for those having $(\mathbf{i}+1)$ on the bottom, the set of dead ends of $\mathfrak{B}_i$ is $\Delta \bigcup (\mathbf{i}+1)$; finally, the macrostates of $\mathfrak{B}_i$ are obtained from macrostates of $\mathfrak{A}$ by deleting $(\mathbf{i}+1)$.

Let us say that a function $\mathbf{h}$ from $\mathbf{D}$ into $\mathbf{P}(\mathbf{S} \bigcup \boldsymbol{\Delta})$ is *downward closed* if $\mathbf{i} \in \mathbf{h}(\mathbf{d})$ for any $\mathbf{i} \in \mathbf{S}$ and for any $\mathbf{d} \in \mathbf{D}$ such that $\mathfrak{B}_i$ $\mathbf{d}$-accepts the simple $\mathbf{P}(\Delta \bigcup \{\mathbf{i}+1\})$-tree defined by the function $\mathbf{h}_i(\mathbf{d}') = \mathbf{h}(\mathbf{d}') \bigcap (\Delta \bigcup \{\mathbf{i}+1\})$.

Lemma 8. *The following assertions are equivalent.*

(1) *The automaton* $\mathfrak{A}$ $\mathbf{d}_0$*-accepts the simple* $\mathbf{P}(\Delta)$*-tree defined by* $\mathbf{g}$.

(2) $0 \in \mathbf{h}(\mathbf{d}_0)$ *for any downward closed function* $\mathbf{h}$ *extending* $\mathbf{g}$.

Proof. Assume that (2) is true. Let us prove that (1) is true. Define $\mathfrak{A}_i$ to be the automaton obtained from $\mathfrak{A}$ by declaring $\mathbf{i}$ to be the (single) initial state. Set
$\mathbf{h}(\mathbf{d}) = \mathbf{g}(\mathbf{d}) \bigcup \{\mathbf{i} \in \mathbf{S} \mid \mathfrak{A}_i\ \mathbf{d}\text{-accepts the simple tree defined by } \mathbf{g}\}$.

Obviously $\mathbf{h}$ extends $\mathbf{g}$ and is downward closed. Hence $0 \in \mathbf{h}(\mathbf{d}_0)$, that is, the automaton $\mathfrak{A}_0 = \mathfrak{A}$ $\mathbf{d}_0$-accepts the simple tree defined by $\mathbf{g}$.

Conversely, assume that there is a downward closed function $\mathbf{h}$ extending $\mathbf{g}$ such that $0 \notin \mathbf{h}(\mathbf{d}_0)$. Let us prove that (1) is false. Let us define the function $\mathbf{l}(\mathbf{x})$ just as it was done in the proof of Lemma 6. Let $\mathbf{H}$ be an arbitrary $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. Let us prove that $\mathbf{H}$ has an infinite path having non-final limit macrostate or a finite path with a non-allowed dead end. We construct that path step by step; on each step we add a finite or an infinite continuation to existing path.

Denote by $\mathbf{P}_j$ (finite or infinite) path obtained on $\mathbf{j}$-th step. On $\mathbf{j}$-th step we want to satisfy one of three following conditions:

(1) $\mathbf{P}_j$ is infinite and has non-final limit macrostate; in this case the construction in completed;

(2) $\mathbf{P}_j$ is finite and its last vertex $\mathbf{x}_j$ is labelled in $\mathbf{H}$ by a non-allowed dead end; in this case the construction is completed also;

(3) $\mathbf{P}_j$ is finite and its last vertex $\mathbf{x}_j$ is labelled by the state $\mathbf{i}$, where $\mathbf{i} = (\mathbf{j} \bmod \mathbf{n})$, and $\mathbf{i} \notin \mathbf{h}(\mathbf{l}(\mathbf{x}_j))$.

At the start $\mathbf{x}_0 = \lambda$ and $\mathbf{H}(\mathbf{x}_0) = \mathbf{H}(\lambda) = 0 \notin \mathbf{h}(\mathbf{d}_0)$, thus for $\mathbf{j} = 0$ third alternative holds.

Assume now that the step $\mathbf{j}$ is made and that assertion (3) is true. Denote by $\mathbf{x}_j$ the last vertex of $\mathbf{P}_j$ and by $\mathbf{i} = (\mathbf{j} \bmod \mathbf{n})$ its label in $\mathbf{H}$. As $\mathbf{i} \notin \mathbf{h}(\mathbf{l}(\mathbf{x}_j))$ and $\mathbf{h}$ is downward closed, the automaton $\mathfrak{B}_i$ doesn't $\mathbf{l}(\mathbf{x}_j)$-accept the simple $\mathbf{P}(\Delta \bigcup \{\mathbf{i}+1\})$-tree defined by the function $\mathbf{h}_i(\mathbf{d}') = \mathbf{h}(\mathbf{d}') \bigcap (\Delta \bigcup \{\mathbf{i}+1\})$. Let us define a subtree $\mathbf{H}'$ of the tree of $\mathbf{D}$-branching labelled by elements of $\mathbf{S}$. The tree $\mathbf{H}'$ consists of all the vertices $\mathbf{y}$ such that no proper prefix of the word $\mathbf{x}_j\mathbf{y}$ is labelled by $\mathbf{i}+1$ in $\mathbf{H}$. A vertex $\mathbf{y}$ has in $\mathbf{H}'$ the label equal to the label of $\mathbf{x}_j\mathbf{y}$ in $\mathbf{H}$. It is easy to verify that $\mathbf{H}'$ is a $\mathbf{l}(\mathbf{x}_j)$-run of $\mathfrak{B}_i$ on the simple $\mathbf{P}(\Delta \bigcup \{\mathbf{i}+1\})$-tree defined by $\mathbf{h}_i$. We know that $\mathbf{H}'$ is not an accepting run. Hence, there is a finite path $\mathbf{S}'$ in $\mathbf{H}'$ ending in a vertex $\mathbf{y}$ such that $\mathbf{H}'(\mathbf{y}) \in \Delta \bigcup \{\mathbf{i}+1\} \setminus \mathbf{h}_i(\mathbf{l}(\mathbf{y}))$ or an infinite path $\mathbf{S}'$ in $\mathbf{H}'$ having non-final limit macrostate. In the first case either $\mathbf{H}'(\mathbf{y}) \in \Delta \setminus \mathbf{g}(\mathbf{l}(\mathbf{y}))$, or $\mathbf{H}'(\mathbf{y}) = \mathbf{i}+1 \notin \mathbf{h}(\mathbf{l}(\mathbf{y}))$. Thus, in this case we can define the last vertex of the new path $\mathbf{P}_{j+1}$ to be $\mathbf{x}_j\mathbf{y}$. In the second case the path $\mathbf{x}_j\mathbf{S}'$ is infinite path in $\mathbf{H}$ having non-final limit macrostate.

The construction of the path is completed. It remains to note that if we have made infinitely many steps then the limit macrostate of the defined path is equal to $\mathbf{S}$ and $\mathbf{S}$ is non-final macrostate. Lemma 8 is proved.

By induction hypothesis, for each $\mathbf{i} \in \mathbf{S}$ there is a formula $\varphi_i(\mathbf{h}_i, \mathbf{d})$ which is true iff $\mathfrak{B}_i$ $\mathbf{d}$-accepts the simple $\mathbf{P}(\Delta \bigcup \{\mathbf{i}+1\})$-tree defined by $\mathbf{h}_i$. Thus, the property of $\mathbf{h}$ to be downward closed is monadic definable, therefore, the property of $\mathbf{g}$ and $\mathbf{d}_0$ stated in item (2) of Lemma 8 is monadic definable.

Second subcase: The macrostate $\mathbf{S}$ is final.

Let us call a function $\mathbf{h}\colon \mathbf{D} \longrightarrow \mathbf{P}(\Delta \bigcup \mathbf{S})$ *upward closed* if $\mathfrak{B}_i$ $\mathbf{d}_0$-accepts the simple $\mathbf{P}(\Delta \bigcup \{\mathbf{i}+1\})$-tree defined by the function $\mathbf{h}_i(\mathbf{d}') = \mathbf{h}(\mathbf{d}') \bigcap (\Delta \bigcup \{\mathbf{i}+1\})$ for any $\mathbf{i} \in \mathbf{S}$ and for any $\mathbf{d}_0 \in \mathbf{D}$ such that $\mathbf{i} \in \mathbf{h}(\mathbf{d}_0)$.

Lemma 9. *The following assertions are equivalent.*

(1) *The automaton* $\mathfrak{A}$ $\mathbf{d}_0$*-accepts the simple* $\mathbf{P}(\Delta)$*-tree defined by* $\mathbf{g}$.

(2) *There is an upward closed function* $\mathbf{h}$ *extending* $\mathbf{g}$ *such that* $0 \in \mathbf{h}(\mathbf{d}_0)$.

Proof. Assume that (1) is true. Let us prove that (2) is true. Let us fix an accepting $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. Set $\mathbf{h}(\mathbf{d}) = \Big\{\mathbf{H}(\mathbf{x}) \mid \mathbf{l}(\mathbf{x}) = \mathbf{d}\Big\} \bigcup \mathbf{g}(\mathbf{d})$, where $\mathbf{l}(\mathbf{x})$ is defined just as it was done in Lemma 6. It is easy to verify that $\mathbf{h}$ is upward closed, extends $\mathbf{g}$ and $0 \in \mathbf{h}(\mathbf{d}_0)$.

Conversely, assume that there is an upward closed function $\mathbf{h}$ extending $\mathbf{g}$ such that $0 \in \mathbf{h}(\mathbf{d}_0)$. Let us define an accepting $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. As $0 \in \mathbf{h}(\mathbf{d}_0)$ and $\mathbf{h}$ is upward closed, there is an accepting $\mathbf{d}_0$-run $\mathbf{H}_0$ of $\mathfrak{B}_0$ on the simple tree defined by $\mathbf{h}_0$.

Obviously $\mathbf{H}_0$ is also a $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple $\mathbf{P}(\Delta)$-tree defined by $\mathbf{g}$. The $\mathbf{d}_0$-run $\mathbf{H}_0$ is not an accepting $\mathbf{d}_0$-run of $\mathfrak{A}$ only if some its leaves are labelled by 1.

For every vertex $\mathbf{x}$ in $\mathbf{H}_0$ labelled by 1 we make the following. We know that $1 \in \mathbf{h}(\mathbf{l}(\mathbf{x}))$. Therefore there is an accepting $\mathbf{l}(\mathbf{x})$-run $\mathbf{H}_1$ of $\mathfrak{B}_1$ on the simple tree defined by $\mathbf{h}_1$. Let us glue to $\mathbf{x}$ the labelled tree $\mathbf{H}_1$.

The resulting labelled tree may be regarded as a $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. This tree is not an accepting $\mathbf{d}_0$-run only if some its leaves are labelled by 2. We repeat the above actions and so on. After ω steps we obtain a $\mathbf{d}_0$-run of $\mathfrak{A}$ on the simple tree defined by $\mathbf{g}$. Any dead end in that run is allowed. Every infinite path in that run either from some place comes into an accepting run of $\mathfrak{B}_i$ (for some $\mathbf{i} \in \mathbf{S}$), or for each $\mathbf{i} \in \mathbf{S}$ has infinitely many occurrences of $\mathbf{i}$ and therefore has final limit macrostate. Lemma 9 is proved.

It remains to note that the property of $\mathbf{d}_0$ and $\mathbf{g}$ stated in item (2) of Lemma 9 is monadic definable.