NESTED WORD AUTOMATA

RASEEK C

Reference

1 . “Adding Nesting Structure to Words” , RAJEEV ALUR, P. Madhusudan

2. https://www.cis.upenn.edu/~alur/nw.html

Contents

Definition

A *nested word automaton* (NWA) A over an alphabet Σ is a structure $(Q, q_0, Q_f, P, p_0, P_f, \delta_c, \delta_i, \delta_r)$ consisting of

—a finite set of (linear) states Q,
—an initial (linear) state $q_0 \in Q$,
—a set of (linear) final states $Q_f \subseteq Q$,
—a finite set of hierarchical states P,
—an initial hierarchical state $p_0 \in P$,
—a set of hierarchical final states $P_f \subseteq P$,
—a call-transition function $\delta_c : Q \times \Sigma \mapsto Q \times P$,
—an internal-transition function $\delta_i : Q \times \Sigma \mapsto Q$, and
—a return-transition function $\delta_r : Q \times P \times \Sigma \mapsto Q$.

A Run Of Automaton

- i/p: nested word $n = (a_1 \ldots a_\ell, \rightsquigarrow)$
- sequence $q_i \in Q$, for $0 \leq i \leq \ell$, of states corresponding to linear edges
- sequence $p_i \in P$, for calls i, of states corresponding to nesting edges
- for each position $1 \leq i \leq \ell$,

  —if i is a call, then $\delta_c(q_{i-1}, a_i) = (q_i, p_i)$;

  —if i is an internal, then $\delta_i(q_{i-1}, a_i) = q_i$;

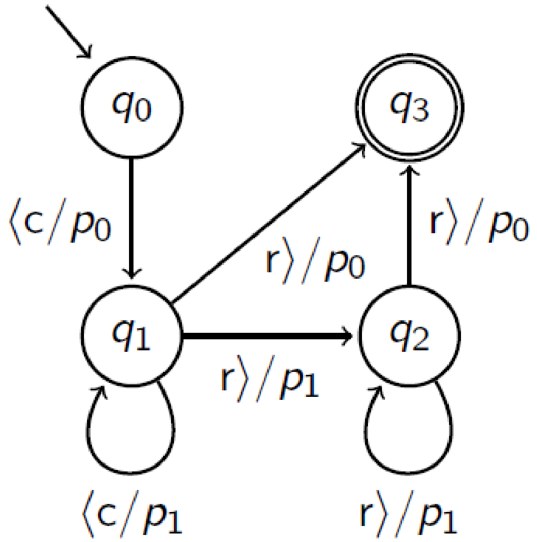
  —if i is a return with call-predecessor j, then $\delta_r(q_{i-1}, p_j, a_i) = q_i$, and if i is a pending return, then $\delta_r(q_{i-1}, p_0, a_i) = q_i$.

- accepts the nested word n if in this run, $q_\ell \in Q_f$ and for pending calls i, $p_i \in P_f$

Example

- Consider $\mathcal{L}2 = \{c^n r^n \mid n > 0\}$.

We construct an NWA for $\mathcal{L}_2' := \{(\langle c)^n\ (r\rangle)^n \mid n > 0\}$.



$P = \{p_0, p_1\}$, $P_f \subseteq \{p_0\}$

Non-deterministic Nested Word Automata

—a finite set of (linear) states Q,

can have Multiple Initial States

—a set of (linear) initial states $Q_0 \subseteq Q$,

—a set of (linear) final states $Q_f \subseteq Q$,

—a finite set of hierarchical states P,

can have Multiple hierarchical States

—a set of initial hierarchical states $P_0 \subseteq P$,

—a set of final hierarchical states $P_f \subseteq P$,

—a call-transition relation $\delta_c \subseteq Q \times \Sigma \times Q \times P$,

can have Multiple transitions on same symbol `

—an internal-transition relation $\delta_i \subseteq Q \times \Sigma \times Q$, and

—a return-transition relation $\delta_r \subseteq Q \times P \times \Sigma \times Q$.

A Run Of Non-Dterministic Automaton

- i/p: nested word $n = (a_1 \ldots a_\ell, \rightsquigarrow)$
- sequence $q_i \in Q$, for $0 \leq i \leq \ell$, of states corresponding to linear edges
- sequence $p_i \in P$, for calls i, of states corresponding to nesting edges
- for each position $1 \leq i \leq \ell$,

  —if i is a call, then $(q_{i-1}, a_i, q_i, p_i) \in \delta_c$;

  —if i is an internal, then $(q_{i-1}, a_i, q_i) \in \delta_i$;

  —if i is a matched return with call-predecessor j then $(q_{i-1}, p_j, a_i, q_i) \in \delta_r$, and if i is a pending return then $(q_{i-1}, p_0, a_i, q_i) \in \delta_r$ for some $p_0 \in P_0$.

- accepts the nested word n if in this run, $q_\ell \in Q_f$ and for pending calls i, $p_i \in P_f$
- The automaton A accepts the nested word n if A has some accepting run over n .

Determinization

Consider the NNWA $\mathcal{A} = \langle Q, Q_0, Q_f, P, P_0, P_f, \delta_i, \delta_c, \delta_r \rangle$.
We construct the DNWA $\mathcal{B} = \langle Q', q'_0, Q'_f, P', p'_0, P'_f, \delta'_i, \delta'_c, \delta'_r \rangle$:

—The states of B are $Q' = 2^{Q \times Q}$.

—The initial state is the set $Q_0 \times Q_0$ of pairs of initial states.

—A state $S \in Q'$ is accepting iff it contains a pair of the form (q, q') with $q' \in Q_f$.

—The hierarchical states of B are $P' = \{p'_0\} \cup (Q' \times \Sigma)$.

—The initial hierarchical state is p'_0.

Determinization (contd)

- Consider a nested word n with k pending calls , represented as

$$n = n_1 \langle c_1 \, n_2 \langle c_2 \cdots n_k \langle c_k \, n_{k+1}$$

- each n_i is a nested word with no pending calls
- The initial nested word n_1 can have pending returns, and the nested words n2,…, n_{k+1} are well-matched
- After reading n, B will be in state S_{k+1}, where (S_i , c_i) will be the hierarchical state for each <c_i .
- S_i contains the pair (q, q') iff $q \xrightarrow{n_i}_{\mathcal{A}} q'$
- B accepts n if $S_{k+1} \in Q'_f$.

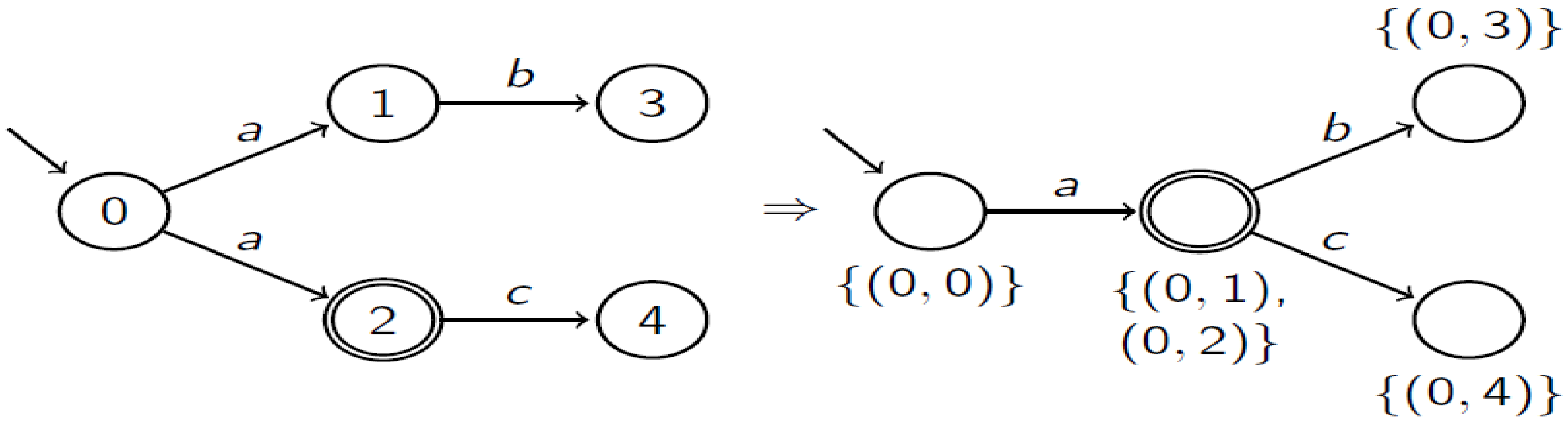
$$\text{i.e., } \exists q, q' ((q, q') \in S_{k+1}) \wedge (q \xrightarrow{n_{k+1}}_{\mathcal{A}} q') \wedge (q' \in Q_f)$$

Internal Transitions

- Consider a nested word n with k pending calls , represented as

$$n = n_1 \langle c_1 n_2 \langle c_2 \cdots n_k \langle c_k n_{k+1}$$

- The internal-transition function δ'_i is given by: for $S \in Q'$ and $a \in \Sigma$, $\delta'_i(S, a)$ consists of pairs (q, q'') such that there exists $(q, q') \in S$ and an internal transition $(q', a, q'') \in \delta_i$.

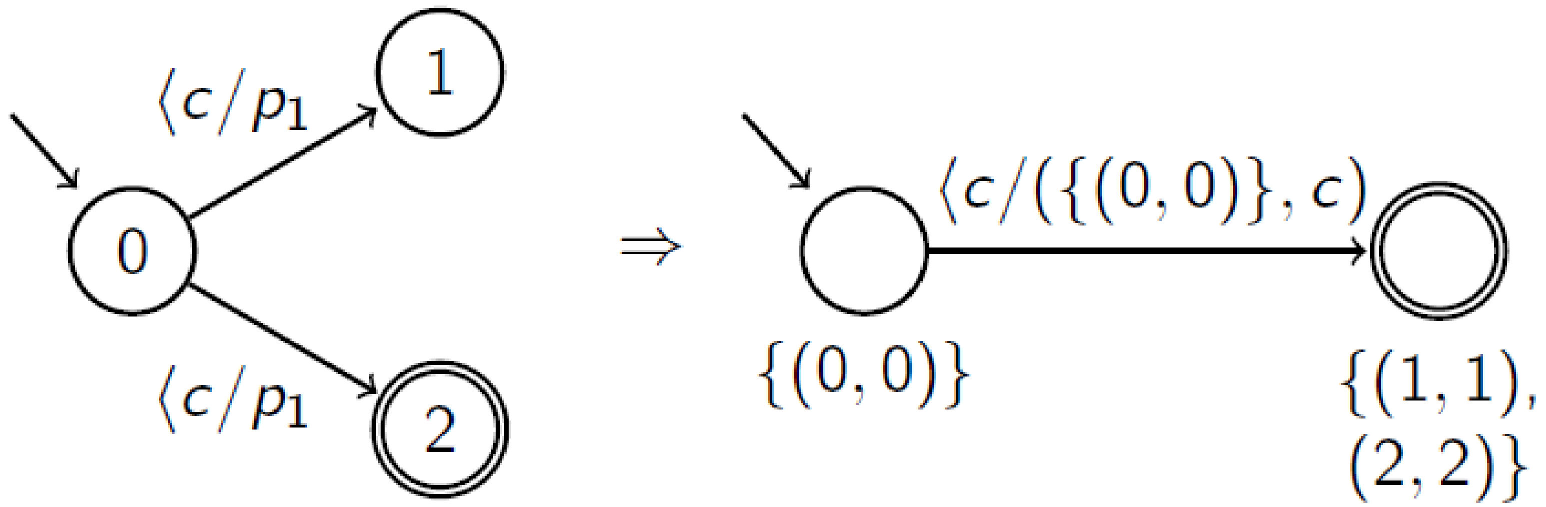


Call Transitions

- Consider a nested word n with k pending calls , represented as

$$n = n_1 \langle c_1 n_2 \langle c_2 \cdots n_k \langle c_k n_{k+1}$$

- The call-transition function δ'_c is given by: for $S \in Q'$ and $a \in \Sigma$, $\delta'_c(S, a) = (S', (S, a))$, where S' consists of pairs (q'', q'') such that there exists $(q, q') \in S$ and a hierarchical state $p \in P$ and a call transition $(q', a, q'', p) \in \delta_c$.



Return Transitions

- Consider a nested word n with k pending calls , represented as

$$n = n_1 \langle c_1 n_2 \langle c_2 \cdots n_k \langle c_k n_{k+1}$$

- Two cases

$k = 0$ no matching call, like internal transition

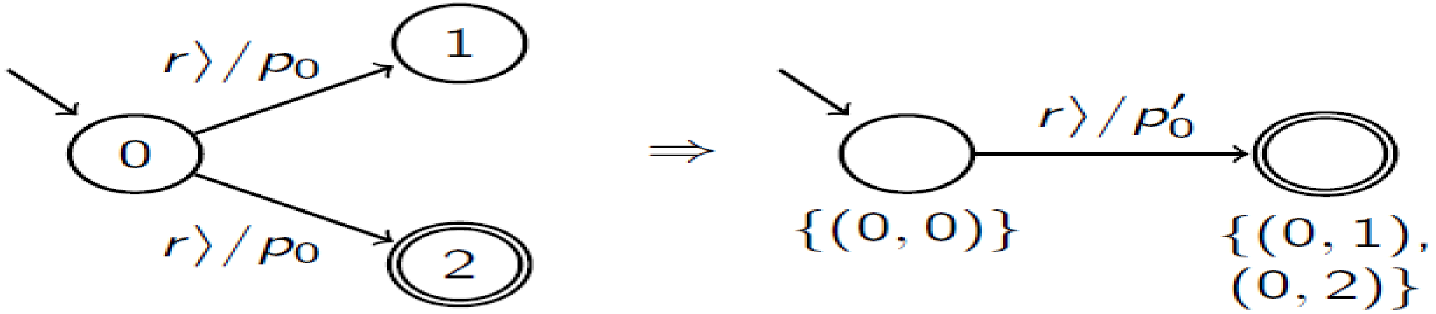
$$\delta'_r(S_{k+1}, p'_0, r) = \{(q, q'') \mid (q, q') \in S_{k+1} \wedge \exists p \in P_0 . q'' \in \delta_r(q', p, r)\}$$

$k > 0$ subword $n_k \langle c_k n_{k+1} r \rangle$, hierarchical state $= (S_k, c_k)$

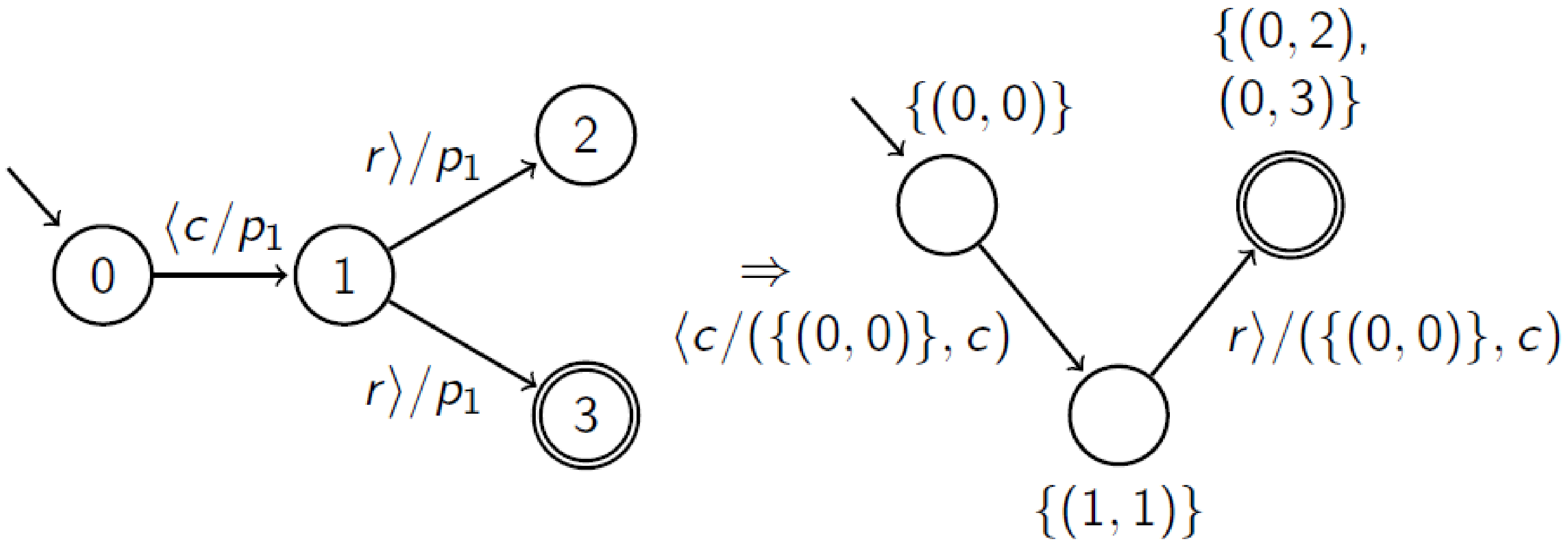
$$\delta'_r(S_{k+1}, (S_k, c_k), r) = \{(q, q'') \mid (q, q') \in S_k \wedge (q_1, q_2) \in S_{k+1} \wedge \exists p \in P . (q_1, p) \in \delta_c(q', c_k) \wedge q'' \in \delta_r(q_2, p, r)\}$$

Return Transitions (contd)

- Case 1 : Example



- Case 2 : Example



Closure Properties

- Nested word Automata are closed under the following operations
  - Union
  - Intersection
  - Complement
  - Concatenation
  - Reversal
  - Prefixes
  - Suffixes
  - Homomorphism

Boolean Closure

- *If L_1 and L_2 are regular languages of nested words over Σ, then $L_1 \cup L_2$, $L_1 \cap L_2$, and $NW(\Sigma) \setminus L_1$ are also regular languages.*
- Let $A_j = (Q^j, q_0^j, Q_f^j, P^j, p_0^j, \delta_c^j, \delta_i^j, \delta_r^j)$, for $j = 1, 2$ be a NWA accepting L_j
- Define the product of these two automata as follows
- The set of linear states - $Q^1 \times Q^2$; The initial state - (q_0^1, q_0^2)
- The set of hierarchical states - $P^1 \times P^2$; The initial hierarchical state (p_0^1, p_0^2)
- Transition functions are defined in obvious way. For example, return transition function in product can be defined as $\delta_r((q_1, q_2), (p_1, p_2), a) = (\delta_r^1(q_1, p_1, a), \delta_r^2(q_2, p_2, a))$
- Final state for $L_1 \cup L_2$: $(Q_f^1 \times Q_2) \cup (Q_1 \times Q_f^2)$
- Final state for $L_1 \cap L_2$: $Q_f^1 \times Q_f^2$
- Complement of NWA, A $(Q, q_0, Q_f, P, p_0, \delta_c, \delta_i, \delta_r)$ is $(Q, q_0, Q \setminus Q_f, P, p_0, \delta_c, \delta_i, \delta_r)$

Concatenation Closure

- If L_1 and L_2 are regular languages of nested words, then so are $L_1.L_2$ and L_1^*

.Proof:

- Let A1 and A2 are the NWAs, with disjoint sets accepting L_1 and L_2 respectively.

- The NWA simulates A1, and at some point, instead of going to final state of A1, switches to the initial state of A2.

- While simulating A2, at a return, if the state labeling the incoming nesting edge is a state of A1, then it is treated like the initial state of A2.

Kleene Closure

- Let $A = (Q, Q_0, Q_f, \delta_c^l, \delta_i, \delta_r)$ be a NNWA that accepts L.

A * can be modelled as

- Simulates A step by step, when A changes its state to final state, A* can nondeterministically update its state to an initial state.
- Upon this switch A* must treat the unmatched nesting edges as if they are pending
- Initial and final states are Q0.
- *(Internal).* For each internal transition $(q, a, p) \in \delta_i$, A^* contains the internal transitions (q, a, p) and (q', a, p'), and if $p \in Q_f$, then the internal transitions (q, a, r') and (q', a, r') for each $r \in Q_0$.

Kleene Closure (contd)

- *(Call).* For each (linear) call transition $(q, a, p) \in \delta_c^l$, A^* contains the call transitions (q, a, p) and (q', a, p), and if $p \in Q_f$, then the call transitions (q, a, r') and (q', a, r'), for each $r \in Q_0$.
- *(Return).* For each return transition $(q, r, a, p) \in \delta_r$, A^* contains the return transitions (q, r, a, p) and (q, r', a, p'), and if $p \in Q_f$, then the return transitions (q, r, a, s') and (q, r', a, s'), for each $s \in Q_0$. For each return transition $(q, r, a, p) \in \delta_r$ with $r \in Q_0$, A^* contains the return transitions (q', s, a, p') for each $s \in Q \cup Q'$, and if $p \in Q_f$, also the return transitions (q', s, a, t') for each $s \in Q \cup Q'$ and $t \in Q_0$.

Closure under Word Operations

- (Closure under word operations). *If L is a regular language of nested words then all the following languages are regular: the set of reversals of all the nested words in L; the set of all prefixes of all the nested words in L; the set of all suffixes of all the nested words in L.*
- **Reverse** of a nested word n : $w\_nw(b_\ell \ldots b_2 b_1)$, where for each $1 \le i \le \ell$, $b_i = a_i$ if i is an internal, $b_i = \langle a_i$ if i is a return, and $b_i = a_i\rangle$ if i is a call.
- Consider a NWA $A = (Q, Q_0, Q_f, P, P_0, P_f, \delta_c \delta_i, \delta_r)$
- Define A^R to be $(Q, Q_f, Q_0, P, P_f, P_0, \delta_c^R, \delta_i^R, \delta_r^R)$ where $(q, a, q', p) \in \delta_c$ iff $(q', p, a, q) \in \delta_r^R$, $(q, p, a, q') \in \delta_r$ iff $(q', a, q, p) \in \delta_c^R$, and $(q, a, q') \in \delta_i$ iff $(q' a, q) \in \delta_i^R$.

Closure under Word Operations (Prefix)

- Consider a NNWA $A = (Q, Q_0, Q_f, \delta_c^l, \delta_i, \delta_r)$
- Then , an automaton B can be defined as
- states :
- $(q, q', 1)$ if there exists a nested word n which takes A from state q to state $q' \in Q_f$
  $(q, q', 2)$ if there exists a nested word n without any pending returns, which takes A from state q to state $q' \in Q_f$
  $(q, q', 3)$ if there exists a well-matched nested word n which takes A from state q to state q'
- Initial states : $\{(q, q', 1) \ / q \in Q_0 \text{ and } q' \in Q_f \}$
- All states are final
- The state of B keeps track the current state of A along with a target state where the run of A can end
- Initially, the target state is required to be final state & this target is propagated along therun

Closure under Word Operations (Prefix- contd)

- At a call, B can propagate
  - either the current target across the linear edge requiring that the current state can reach the target without using pending returns
  - or the current target across the nesting edge, and across the linear edge, guess a new target state requiring that the current state can reach this target using a well-matched word
- The third component of the state is used to keep track of the constraint on whether pending calls and/or returns are allowed
- Reachability information necessary for effectively constructing the automaton B are computed

Closure under Word Operations (Prefix- contd)

- *(Internal).* For every internal transition $(q, a, p) \in \delta_i$, for $x = 1, 2, 3$, for every $q' \in Q$, if both (q, q', x) and (p, q', x) are states of B, then there is an internal transition $((q, q', x), a, (p, q', x))$.
- *(Call).* Consider a linear call transition $(q, a, p) \in \delta_c^l$ and $q' \in Q$ and $x = 1, 2, 3$, such that (q, q', x) is a state of B. Then for every state r such that $(p, r, 3)$ is a state of B and there exists $b \in \Sigma$ and state $r' \in Q$ such that (r', q', x) is a state of B and $(r, q, b, r') \in \delta_r$, there is a call transition $((q, q', x), a, (p, r, 3))$. In addition, if $x = 1, 2$ and $(p, q', 2)$ is a state of B, then there is a call transition $((q, q', x), a, (p, q', 2))$.
- *(Return).* For every return transition $(q, p, a, r) \in \delta_r$, for $x = 1, 2, 3$, for $q' \in Q$, if (p, q', x) and (r, q', x) are states of B, then there is a return transition $((q, q, 3), (p, q', x), a, (r, q', x))$. Also, for every return transition $(q, p, a, r) \in \delta_r$ with $p \in Q_0$, for every $q' \in Q_f$, if $(q, q', 1)$ and $(r, q', 1)$ and $(p, q', 1)$ are states of B then there is a return transition $((q, q', 1), (p, q', 1), a, (r, q', 1))$.
- The automaton B accepts a nested word n if there exists a nested word n` such that the concatenation of n and n` is accepted by A.