

# 1 State-Quantified Hyperautomata

## 1.1 Intuition

Standard Finite-Word Hyperautomata (NFH) operate on a static assignment of words to variables, determined entirely before the run begins. This limits the automaton to verifying properties where the witness words are pre-selected. We introduce the *State-Quantified NFH* (SQ-NFH) to allow for *dynamic variable extension*. In this model, the automaton can query the input hyperword  $S$  during execution to append new words to its variables. This capability is formalized by a labeling function  $L$  that enforces extensions: if a state is labeled with a variable index, the machine must pick a word from  $S$  and append it to that variable's buffer before processing any transitions. The rest of the definitions are as defined in regular NFH.

## 1.2 Formal Definition

**Definition 1** (Configuration). Let  $\Sigma$  be a finite alphabet. A configuration is a tuple  $c = (q, u_1, \dots, u_k)$ , where  $q$  is a state and  $u_i \in \Sigma^*$  represents the current content of the buffer for variable  $x_i$ . For a step  $j$  in a run, we denote the *ingress configuration* as  $c_j^{in}$  and the *egress configuration* as  $c_j^{out}$ .

**Definition 2** (SQ-NFH). A State-Quantified Non-deterministic Finite Hyper-automaton is a tuple  $\mathcal{A} = \langle \Sigma, X, Q, Q_0, F, \delta, \alpha, L \rangle$ , where:

- $\Sigma$  is the finite alphabet.
- $X = \{x_1, \dots, x_k\}$  is a finite set of word variables.
- $Q$  is a finite set of states.
- $Q_0 \subseteq Q$  is the set of initial states.
- $\delta \subseteq Q \times (\Sigma \cup \{\#\})^k \times Q$  is the transition relation.
- $F \subseteq Q$  is the set of accepting states.
- $\alpha = Q_1 x_1 Q_2 x_2 \dots Q_k x_k$  is a quantifier prefix, where  $Q_i \in \{\forall, \exists\}$  for every  $1 \leq i \leq k$ .
- $L \subseteq Q \times \{1, \dots, k\}$  is the extension labeling relation. If  $(q, i) \in L$ , variable  $x_i$  must be extended at state  $q$ .

**Semantics.** The semantics of an SQ-NFH are defined with respect to a hyperword  $S$ . Let  $v : X \rightarrow S$  be an assignment of the word variables of  $\mathcal{A}$  to words in  $S$ . We denote by  $v[x \rightarrow w]$  the assignment obtained from  $v$  by assigning the word  $w \in S$  to  $x \in X$ . We denote the satisfaction relation as  $S \models_v (\alpha, \mathcal{A})$ , where  $\alpha$  is a quantifier prefix, and define it as follows.

1. If  $\alpha = \exists x_i \beta$ , then  $S \models_v (\alpha, \mathcal{A})$  iff there exists  $w \in S$  such that  $S \models_{v[x_i \mapsto w]} (\beta, \mathcal{A})$ .
2. If  $\alpha = \forall x_i \beta$ , then  $S \models_v (\alpha, \mathcal{A})$  iff for all  $w \in S$ ,  $S \models_{v[x_i \mapsto w]} (\beta, \mathcal{A})$ .
3. If  $\alpha = \epsilon$ , then  $S \models_v (\epsilon, \mathcal{A})$  iff there exists a sequence of configuration pairs  $\rho = (c_0^{in}, c_0^{out}), \dots, (c_m^{in}, c_m^{out})$  satisfying:
  - **Initialization:**  $c_0^{in} = (q_0, v(x_1), \dots, v(x_k))$  for some  $q_0 \in Q_0$ .
  - **Quantification:** For every  $0 \leq j \leq m$ , let  $c_j^{in} = (q_j, u_1, \dots, u_k)$ . The egress configuration  $c_j^{out} = (q_j, u'_1, \dots, u'_k)$  satisfies:
    - If  $(q_j, i) \in L$ , then  $u'_i = u_i \cdot w$  for some  $w \in S$ .
    - If  $(q_j, i) \notin L$ , then  $u'_i = u_i$ .
  - **Transition:** For every  $0 \leq j < m$ , let  $c_j^{out} = (q_j, u'_1, \dots, u'_k)$  and  $c_{j+1}^{in} = (q_{j+1}, u''_1, \dots, u''_k)$ . There exists a transition vector  $\vec{\sigma} = (\sigma_1, \dots, \sigma_k) \in (\Sigma \cup \{\#\})^k$  such that  $(q_j, \vec{\sigma}, q_{j+1}) \in \delta$  and for all  $i$ :
    - If  $\sigma_i \in \Sigma$ , then  $u'_i = \sigma_i \cdot u''_i$ .
    - If  $\sigma_i = \#$ , then  $u'_i = u''_i$  (Asynchronous).
  - **Acceptance:**  $c_m^{out} = (q_m, \epsilon, \dots, \epsilon)$  with  $q_m \in F$ .

**Note.** When discussing an SQ-NFH accepting a hyperword, the quantifier prefix  $\alpha$  covers all variables in the automaton. Therefore, we can write  $S \models \mathcal{A}$  without specifying an explicit assignment  $v$ .

**Definition 3** (Hyperlanguage of SQ-NFH). The hyperlanguage recognized by an SQ-NFH  $\mathcal{A}$  with quantifier prefix  $\alpha$ , denoted  $\mathcal{L}(\mathcal{A})$ , is the set of hyperlanguages  $S$  that satisfy the automaton:

$$\mathcal{L}(\mathcal{A}) = \{S \subseteq \Sigma^* \mid S \models \mathcal{A}\}$$

## 2 SQ-NFH for $H_{chain}^{\$}$

### 2.1 Formal Definition

**Definition 4** (Automaton for  $H_{chain}^{\$}$ ). Let  $\Sigma = \{0, 1, \dots, n, \$\}$ . We define the SQ-NFH  $\mathcal{A} = \langle \Sigma, X, Q, q_0, F, \delta, \alpha, L \rangle$  as follows:

- $X = \{x_1, x_2\}$ ,  $\alpha = \exists x_1 \forall x_2$ ,  $F = \{q_{acc}\}$ .
- $L = \{(q, 1) \mid q \in Q_{ext}\}$ .
- $\delta$  is defined visually below.

### 2.2 Automaton Diagram

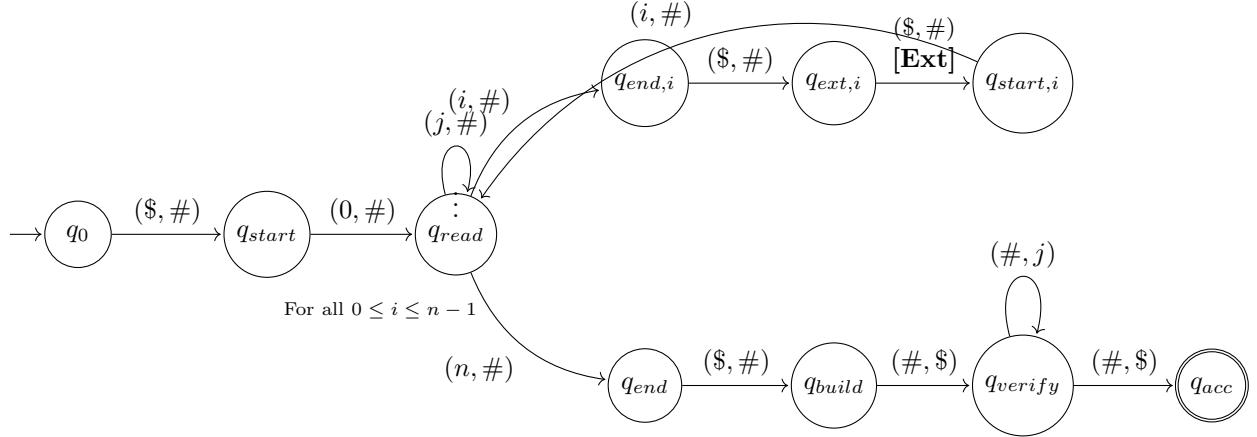


Figure 1: Visual representation of the SQ-NFH. The upper loop represents the dynamic extension phase for  $x_1$  (Part 1), verifying the chain links. The bottom path represents the validation phase for  $x_2$  (Part 2), ensuring well-formedness.