Automata over Infinite Alphabets

Andrzej Murawski and Nikos Tzevelekos

Lecture 5: Automata with History/Class Storage

Expressivity beyond FRAs

Fresh-Register Automata are great, but can we do more?

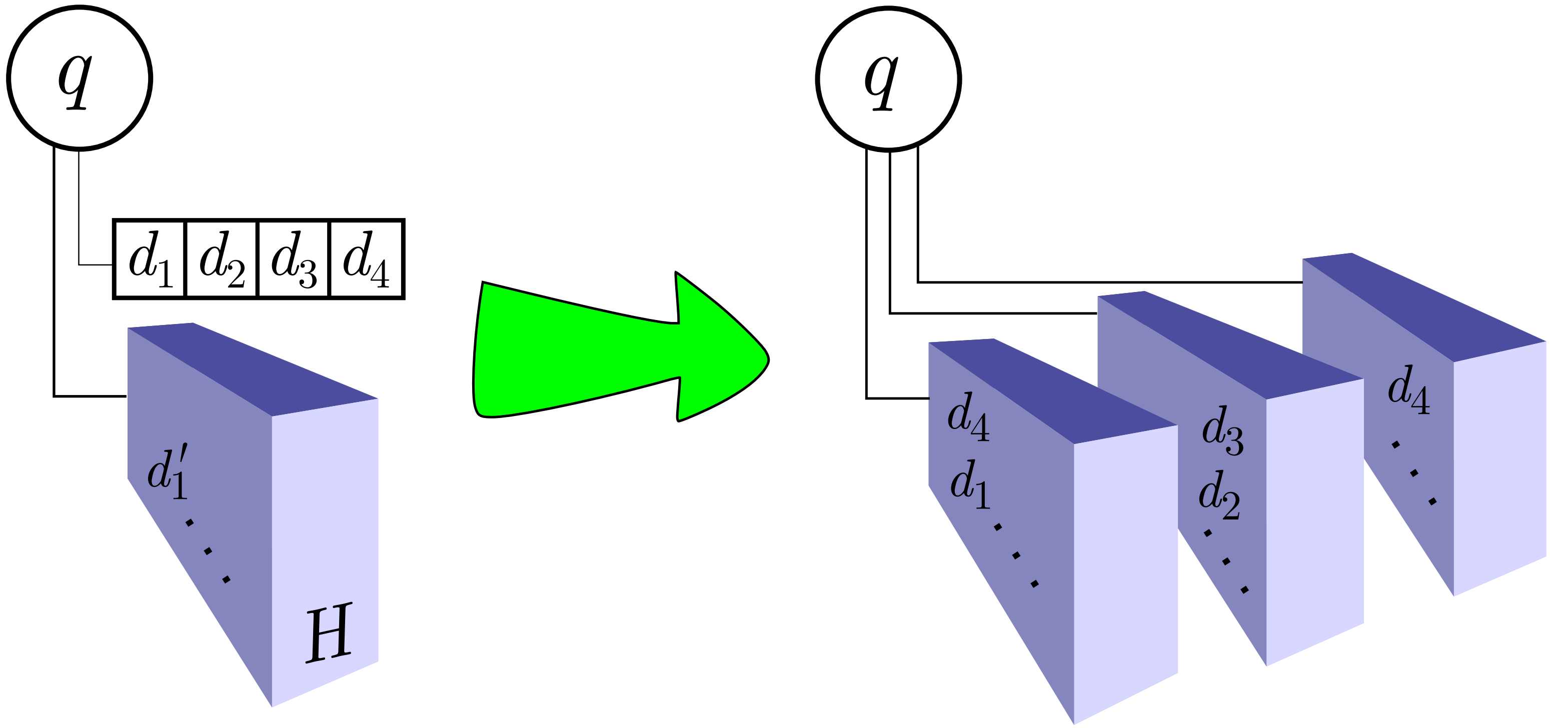
For example:

- (Non) Closure under concatenation ($\mathcal{L}_{\text{fresh}} \cdot \mathcal{L}_{\text{fresh}}$) and Kleene star ($\mathcal{L}_d^*$): what if we could reset the history?
- Why *only one* history?

In this lecture we examine automata that manipulate whole histories/classes containing names instead of single registers.

We present in particular *History-Register Automata*, which are an inclusive representative of such models similar in spirit to RAs.

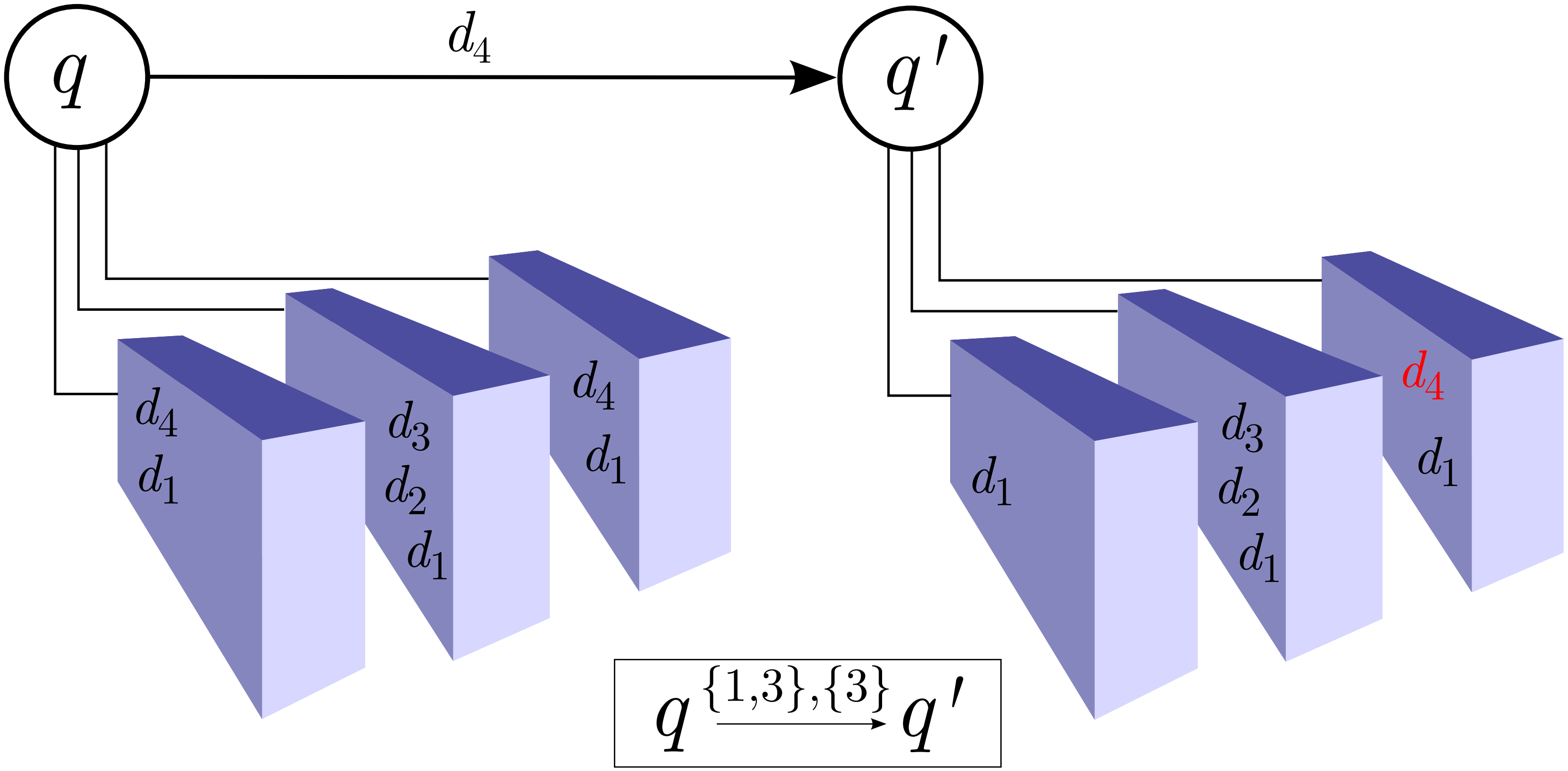
History-Register Automata: in pictures



Two kinds of transitions:

- $q \xrightarrow{X,Y} q'$: *accept a name that is stored exactly in the histories listed in X and* **transfer** *it to the histories listed in Y*
- $q \xrightarrow{\mathsf{rs}(i)} q'$: **reset** *history i (without accepting any input letter)*

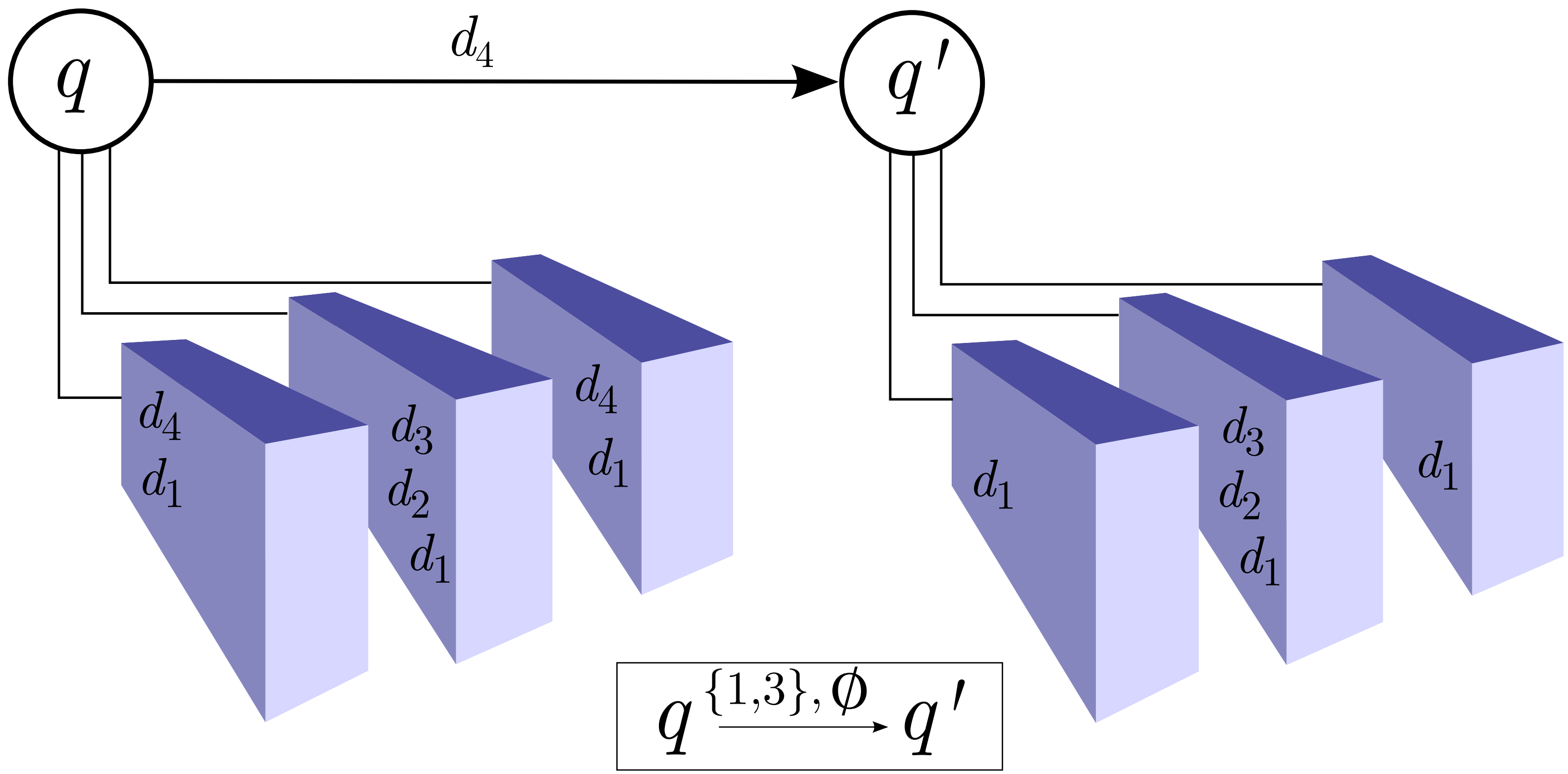
History-Register Automata: transitions



Two kinds of transitions:

- $q \xrightarrow{X,Y} q'$: *accept a name that is stored exactly in the histories listed in X and* **transfer** *it to the histories listed in Y*
- $q \xrightarrow{\mathsf{rs}(i)} q'$: **reset** *history i (without accepting any input letter)*

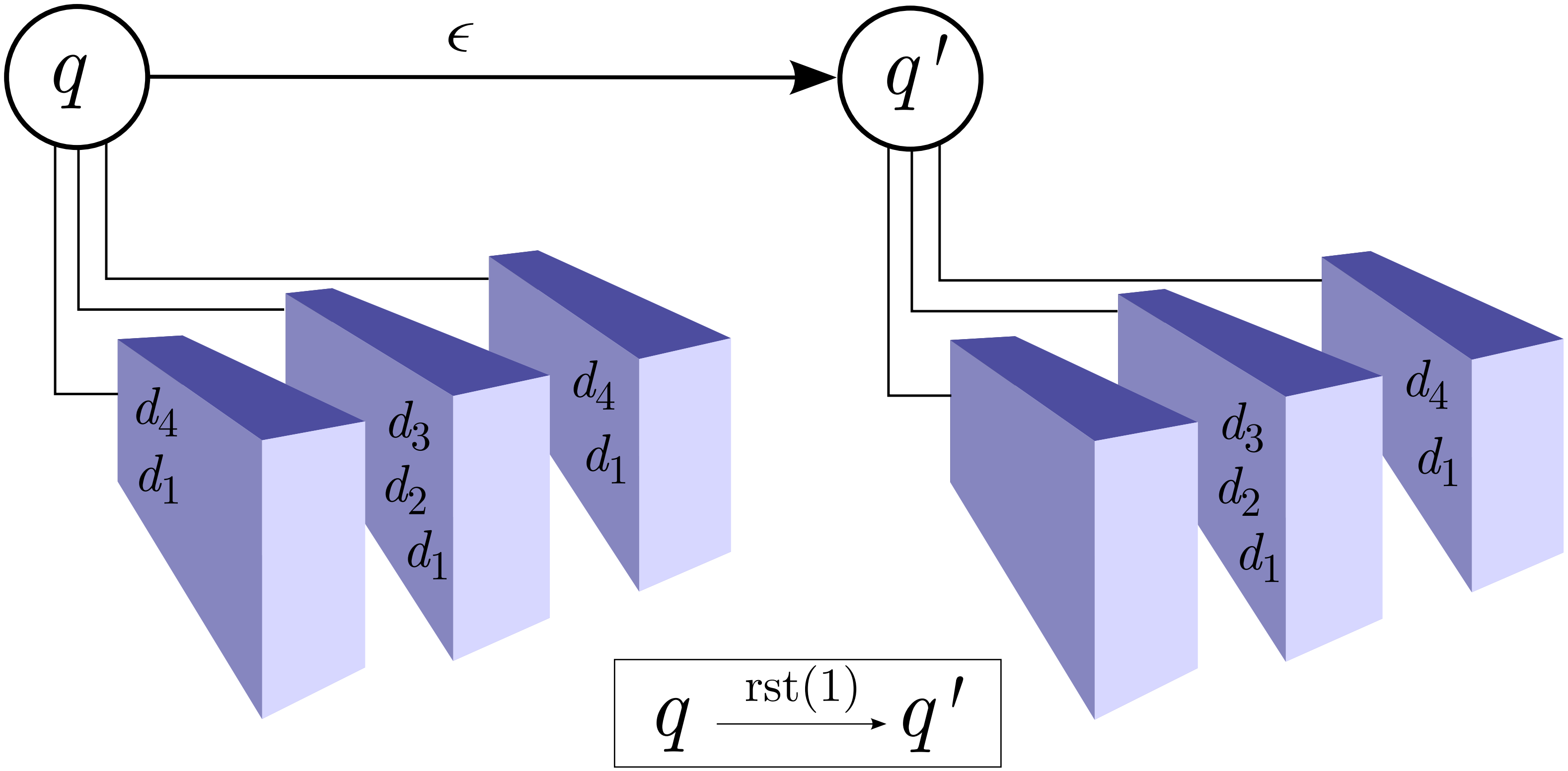
History-Register Automata: transitions



Two kinds of transitions:

- $q \xrightarrow{X,Y} q'$: *accept a name that is stored exactly in the histories listed in* X *and* **transfer** *it to the histories listed in* Y
- $q \xrightarrow{\mathsf{rs}(i)} q'$: **reset** *history* i *(without accepting any input letter)*

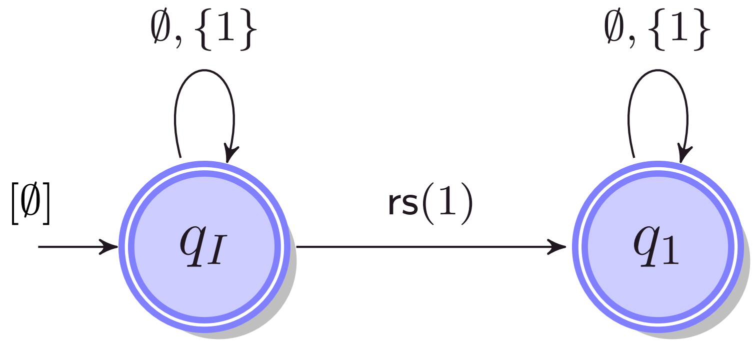
History-Register Automata: transitions

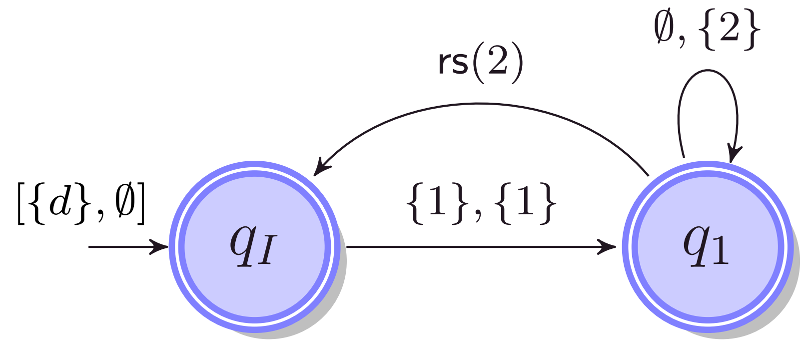


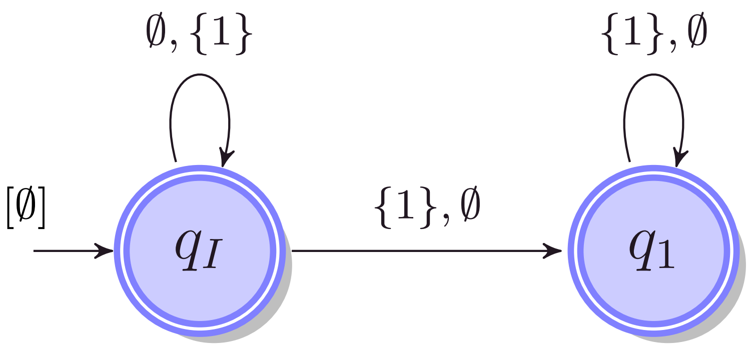
Two kinds of transitions:

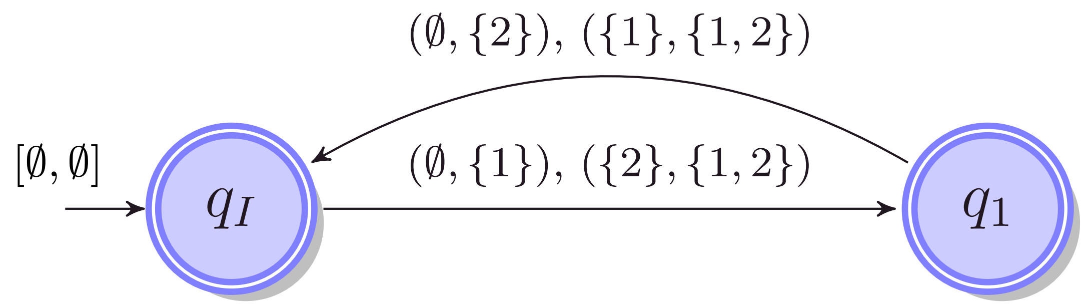
- $q \xrightarrow{X,Y} q'$: *accept a name that is stored exactly in the histories listed in X and* **transfer** *it to the histories listed in Y*
- $q \xrightarrow{\mathsf{rs}(i)} q'$: **reset** *history i (without accepting any input letter)*

Examples









$$\mathcal{L}_{\text{fresh}} \cdot \mathcal{L}_{\text{fresh}} = \{\, ww' \mid w, w' \in \mathcal{L}_{\text{fresh}} \,\}$$

$$\mathcal{L}_d^* = \{\, d\, w_1 \cdots d\, w_n \in \mathcal{D}^* \mid n \geq 0 \land \forall_{1 \leq i \leq n}\, dw_i \in \mathcal{L}_{\text{fresh}} \,\}$$

$$\mathcal{L}_{\mathsf{prod/cons}} = \{\, ww' \in \mathcal{D}^* \mid w, w' \in \mathcal{L}_{\text{fresh}} \land \nu(w') \subseteq \nu(w) \,\}$$

$$\mathcal{L}_{\text{fresh}} \parallel \mathcal{L}_{\text{fresh}} = \{\, d_1 d_1' \cdots d_n d_n'' \in \mathcal{D}^* \mid n \geq 0 \land d_1 \cdots d_n, d_1' \cdots d_n' \in \mathcal{L}_{\text{fresh}} \,\}$$

History-register automata

An n-**History-Register Automaton (**n**-HRA)** is a tuple

$$\mathcal{A} = \langle Q, q_I, H_I, \delta, F \rangle$$

- Q is a finite set of states,
- $q_I \in Q$ is the initial state,
- $F \subseteq Q$ is the set of final states,
- $H_I \in His_n$ is the initial n-history assignment,
- and $\delta \subseteq Q \times Op_n \times Q$ is the transition relation,

where

$$Op_n = \{ (X, Y) \mid X, Y \subseteq [n] \} \cup \{ \mathsf{rs}(i) \mid i \in [n] \}$$
$$His_n = \{ H : [n] \to \mathcal{P}_{\text{fin}}(\mathcal{D}) \}$$

Semantics of HRAs

Let $\mathcal{A} = \langle Q, q_I, H_I, \delta, F\rangle$ be an n-HRA. Let us set:

$$Conf_{\mathcal{A}} = \{\, (q, H) \in\, Q \times His_n \,\}$$

i.e. configurations are now pairs of a state q and an n-history assignment.

A labelled transition $(q_1, H_1) \xrightarrow{d} (q_2, H_2)$ between configurations needs to satisfy the following condition (for some $X, Y \in [n]$):

- $(q_1 \xrightarrow{X,Y} q_2) \in \delta$, and $H_1^{-1}(d) = X$, and $H_2(i) = \begin{cases} H_1(i) \cup \{d\} & \text{if } i \in Y \\ H_1(i) \setminus \{d\} & \text{otherwise} \end{cases}$

moreover, an (un)labelled transition $(q_1, H_1) \xrightarrow{\epsilon} (q_2, H_2)$ can occur when:

- $(q_1 \xrightarrow{\mathsf{rs}(i)} q_2) \in \delta$, and $H_2 = H_1[i \mapsto \emptyset]$.

The **configuration graph** of $\mathcal{A}$ is formed by all possible configuration transitions, and $\mathcal{L}(\mathcal{A}) = \{\, w \in \mathcal{D}^* \mid (q_I, H_I) \overset{w}{\twoheadrightarrow} (q, H) \wedge q \in F \,\}$.

What about registers?

Registers can be modelled by histories with at most one name: we can enforce this using resets. An extra history then models, well, the history...

Theorem. *For any r-fresh-register automaton $\mathcal{A} = \langle Q, q_I, \tau_I, \delta, F \rangle$ there exists an n-HRA $\mathcal{A}' = \langle Q,' q'_I, H_I, \delta', F' \rangle$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

First attempt:

- take $n = r + 1$; model each register i by a corresponding history i, and use history $r + 1$ to store all encountered names
- $q \xrightarrow{i} q'$ becomes $q \xrightarrow{\{i,r+1\},\{i,r+1\}} q'$
- $q \xrightarrow{i^{\circledast}} q'$ becomes $q \xrightarrow{\mathsf{rs}(i)} \cdot \xrightarrow{\emptyset,\{i,r+1\}} q'$
- $q \xrightarrow{i^{\bullet}} q'$ becomes $q \xrightarrow{\mathsf{rs}(i)} \cdot \xrightarrow{\emptyset,\{i,r+1\}} q'$ **and** $q \xrightarrow{\mathsf{rs}(i)} \cdot \xrightarrow{\{r+1\},\{i,r+1\}} q'$

but then, e.g. $q_1 \xrightarrow{1^{\bullet}} q_2 \xrightarrow{1^{\bullet}} q_3$ could accept some dd!

this is because we reset too soon in $q \xrightarrow{\mathsf{rs}(i)} \cdot \xrightarrow{\{r+1\},\{i,r+1\}} q'$

HRAs can model FRAs

We need an extra history so that we can first store the new name and then reset i (for $q \xrightarrow{i^\bullet} q'$). This gives the following.

Theorem. *For any r-fresh-register automaton $\mathcal{A} = \langle Q, q_I, \tau_I, \delta, F\rangle$ there exists an $(r+2)$-HRA $\mathcal{A}' = \langle Q,' q_I', H_I, \delta', F'\rangle$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Idea:

- states are pairs (q, f), where $f : [r] \xrightarrow{\cong} [r+1]$ remembers where each register of the RA is modelled within the HRA's $r+1$ first histories
- this f needs to be updated accordingly after each transition
- the $(r+2)$th history is used to store all encountered names

HRAs can model FRAs: the construction

Theorem. *For any r-fresh-register automaton $\mathcal{A} = \langle Q, q_I, \tau_I, \delta, F\rangle$ there exists an $(r+2)$-HRA $\mathcal{A}' = \langle Q,' q'_I, H_I, \delta', F'\rangle$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Proof. We take $Q' = (Q \times ([r] \overset{\cong}{\to} [r+1])) + Q^{\mathsf{rs}}$, $q'_I = (q_I, \mathsf{id}_{[r]})$ and

$$F' = F \times ([r] \overset{\cong}{\to} [r+1]). \text{ Moreover, } H_I(i) = \begin{cases} \{\tau_I(i)\} & \text{if } \tau_I(i) \neq \# \\ \emptyset & \text{o.w. if } i \leq r+1 \\ \nu(\tau_I) & \text{if } i = r+2 \end{cases}$$

Finally, we include in δ' the following transitions:

- for $q \xrightarrow{i} q'$ in δ, we add $(q, f) \xrightarrow{\{f(i),r+2\},\{f(i),r+2\}} (q', f)$,
- for $q \xrightarrow{i^{\circledast}} q'$ in δ, we add $(q, f) \xrightarrow{\emptyset,\{\hat{f},r+2\}} \cdot \xrightarrow{\mathsf{rs}(f(i))} (q', f[i \mapsto \hat{f}])$,
- for $q \xrightarrow{i^{\bullet}} q'$ in δ, we add $(q, f) \overset{\{r+2\},\{\hat{f},r+2\}}{\xrightarrow{\emptyset,\{\hat{f},r+2\}}} \cdot \xrightarrow{\mathsf{rs}(f(i))} (q', f[i \mapsto \hat{f}])$,

where, in each case, $\hat{f}$ is the unique element in $[r+1] \setminus f([r])$.
Also, Q^{rs} supplies all the intermediate states (denoted with "$\cdot$" above).

HRAs can model FRAs: tying up the proof

Theorem. *For any r-fresh-register automaton $\mathcal{A} = \langle Q, q_I, \tau_I, \delta, F \rangle$ there exists an $(r+2)$-HRA $\mathcal{A}' = \langle Q,' q_I', H_I, \delta', F' \rangle$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

(ctd). It now suffices to show that $\mathcal{A}$ and $\mathcal{A}'$ are bisimilar, but for a special notion of bisimulation that "consumes" ϵ-transitions. Put simply, we consider:

$$\mathcal{G}'_{\mathcal{A}'} = \{\, \kappa_1 \xrightarrow{d} \kappa_2 \mid \kappa_1 \xrightarrow{d} \cdot \xrightarrow{\epsilon} \kappa_2 \in \mathcal{G}_{\mathcal{A}'} \,\}$$

as the configuration graph of $\mathcal{A}'$. Then, taking:

$$\begin{aligned} R = \{\, ((q,\tau,H),((q,f),H')) \mid H'(r+2) = H \wedge H'(\hat{f}) = \emptyset \\ \wedge \forall i.\ \tau(i) = d \implies H'(f(i)) = \{d\} \\ \wedge \tau(i) = \# \implies H'(f(i)) = \emptyset \,\} \end{aligned}$$

we show that R is a bisimulation. $\square$

Closure properties

HRAs are already of the "M-type", so we can readily close them under the same operations as RAs. So, if $\mathcal{A}_1$ and $\mathcal{A}_2$ are HRAs:

- $\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$: here we can simply make use of ϵ-transitions.
- $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$: we can do a similar construction as with RAs.
- $\mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$: we can again do a similar construction, simplified in that we can reset the histories of $\mathcal{A}_1$ when we move on to $\mathcal{A}_2$.
- $\mathcal{L}(\mathcal{A}_1)^*$: similar to the above.
- $\overline{\mathcal{L}(\mathcal{A}_1)}$: we do *not* have closure under complement.
  For example, the language

$$\mathcal{L}_{\text{twice}} = \{\, w \in \mathcal{D}^* \mid \forall_{d \in \nu(w)}\, d \text{ occurs exactly twice in } w \,\}$$

cannot be recognised by an HRA, but its complement can [Exercise].

HRAs cannot count exactly

Theorem. *$\mathcal{L}_{\text{twice}} = \{\, w \in \mathcal{D}^* \mid \forall_{d \in \nu(w)}\, d$ occurs exactly twice in $w \,\}$ is not HRA-recognisable.*

Proof. We argue by contradiction. Suppose there is an HRA $\mathcal{A}$ with k states and m histories such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}_{\text{twice}}$. Then, in particular, $w_{\text{bad}} = d_1 \cdots d_k d_1 \cdots d_k \in \mathcal{L}(\mathcal{A})$ for some pairwise distinct $d_1, \cdots, d_k$.

We examine the transition path of $\mathcal{A}$ that accepts w_{bad}, call it $p = p_1 p_2$.

- The name-accepting transitions of p_1 must have labels $(\emptyset, X)$.
- Let q be a state appearing twice in p_1 (p_1 has length $> k$), so $p_1 = p_{11}(q)p_{12}(q)p_{13}$. We have that the new path:

  $$p' = p'_1 p_2 \;\text{ where }\; p'_1 = p_{11}(q)p_{12}(q)p_{12}(q)p_{13}$$

  is also accepting for $\mathcal{A}$, accepting, say, some w'_{bad}. Indeed, we saw that the transitions in p_1 cannot "block", and we have that p_2 can still be taken (if anything, it will have more names to use now).

Contradiction: we have $|p'_1| > |p_2|$ and therefore $w'_{\text{bad}} \notin \mathcal{L}_{\text{twice}}$. □

Universality and Emptiness

Universality undecidable: Since HRAs extend RAs, this follows from undecidability for RAs.

Emptiness: We can reduce to a decidable weak counter-machine model called **Vector Addition Systems with States (VASS)**.

Ideas:

- To decide emptiness, we need to determine whether we can reach a final configuration from an initial one.
- We can forget about *what word* is accepted and *what names* are in the histories, as long as we remember *how many* names are exactly in each history set $X \subseteq [n]$.
- We can therefore simply count the names in each $X \subseteq [n]$, which gives us a *counter machine* model (with *transfers*).
- Since the counters cannot be tested for emptiness, the latter model is decidable for state reachability.

Weaker forms of HRAs

There is too much liberty in HRAs: the complexity of emptiness checking is *Very Large* (non-primitive recursive).

We therefore examine restrictions thereof:

- Ban resets!

Non-reset HRAs: closure properties

These are HRAs that do not have rs(i) transitions.

Closure under:

- intersection carries over.
- union and concatenation is proven as in the case of RAs.
- complement is still not possible (same counterexample).
- Kleene star is lost:
    - while we do not have a general "boundedness" result, as in the case of FRAs, we can show that the property fails for a specific HRA-recognisable language. Taking

      $$\mathcal{L}_d = \{\, d\, d_1 \cdots d_n \mid d, d_1, \cdots, d_n \text{ pairwise distinct} \,\}$$

      for some fixed d, we can show that $\mathcal{L}_d^*$ is not recognisable by any non-reset HRA.

Kleene star failure

Theorem. *Taking* $\mathcal{L}_d = \{\, d\, d_1 \cdots d_n \mid d, d_1, \cdots, d_n$ *pairwise distinct* $\}$, $\mathcal{L}_d^*$ *is not recognisable by any non-reset HRA.*

Proof. Suppose $\mathcal{L}_d^* = \mathcal{L}(\mathcal{A})$ for some non-reset HRA $\mathcal{A}$ with m histories. We pick $w_{\text{bad}} \in \mathcal{L}(\mathcal{A})$ of the form:

$$w_{\text{bad}} = d\, w_1 d\, w_2 \cdots d\, w_{2^{2^m}}$$

such that no w_i shares any name with the initial history of $\mathcal{A}$ and:

- each w_i is of non-zero even length, say $w_i = w_{i1} w_{i2}$ with $|w_{i1}| = |w_{i2}|$,
- for each $i < j$ there is $d_{ij} \in \mathcal{D}$ that appears in w_{i1} and w_{j2} but does not appear in $w_{i2}\, w_{i+1} \cdots w_{j1}$.

Consider the path $p = p_1 \cdots p_{2^{2^m}}$ accepting w_{bad}. For each i, $p_i = p_{i0} p_{i1} p_{i2}$, where p_{i0} accepts d, and p_{ik} accepts w_{ik} $(k = 1, 2)$.

Let Q_i be the set of all the X's that appear in p_{i1} in target position $(q \xrightarrow{\ldots, X} q')$. Then, because w_{bad} has 2^{2^m} components, there are $\hat{i} < \hat{j}$ such that $Q_{\hat{i}} = Q_{\hat{j}}$.

Kleene star failure

Theorem. *Taking* $\mathcal{L}_d = \{\, d\, d_1 \cdots d_n \mid d, d_1, \cdots, d_n$ *pairwise distinct* $\}$, $\mathcal{L}_d^*$ *is not recognisable by any non-reset HRA.*

(ctd). Since, $d_{\hat{i}\hat{j}}$ appears in $w_{\hat{i}1}$ and then only appears again in $w_{\hat{j}2}$, it must be that in $p_{\hat{i}1}$ it is stored in some $\hat{X}$, and remains there until $p_{\hat{j}2}$.

But note that, at that the start of $p_{\hat{j}2}$, $\hat{X}$ already contains a name, say d', because $p_{\hat{j}1}$ also contains $\hat{X}$ in target position. This means that $\mathcal{A}$ can accept d' instead of $d_{\hat{i}\hat{j}}$ in $w_{\hat{j}2}$, and therefore accept some $w'_{\text{bad}} \notin \mathcal{L}_d^*$, a contradiction.

It remains to show that such w_{bad} exists. We take:

$$
\begin{aligned}
w_{\text{bad}} = {} & d\ d_{12}\, d_{13} \cdots d_{12^{2^m}}\ \Box\ \Box \cdots \Box && (2^{2^m} - 1\ \text{ boxes}) \\
& d\ \Box\ d_{23} \cdots d_{22^{2^m}}\ d_{12}\, \Box \cdots \Box && (2^{2^m} - 1\ \text{ boxes}) \\
& \cdots \\
& d\ \Box\ \Box\ \cdots\ \Box\ \ d_{12^{2^m}} \cdots d_{(2^{2^m}-1)2^{2^m}} && (2^{2^m} - 1\ \text{ boxes})
\end{aligned}
$$

and fill in the boxes with fresh names. $\Box$

Non-reset HRAs extend FRAs

Theorem. *For any r-FRA $\mathcal{A} = \langle Q, q_I, \tau_I, \delta, F\rangle$ there exists a non-reset $3r$-HRA $\mathcal{A}' = \langle Q,' q_I', H_I, \delta', F'\rangle$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

The proof is much harder than the one we saw using resets...

Ideas:

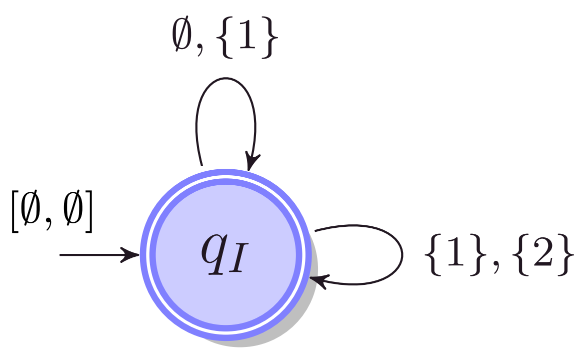
- We model each register i by 3 histories: i_r, i_b, i_y
- All histories contain disjoint names, and no name is "lost"
- The name of register i will be stored in one of i_r, i_b, i_y:
  - in history i_r if we guess that we are going to read it
  - in one of i_b, i_y if we guess that we are going to rewrite it
- We need to remember (in the state) in which of i_r, i_b, i_y is the name of register i stored, and update that info in every transition
- History i_r will always contain at most 1 name, while histories i_b, i_y will contain several (some of them garbage)

Local acceptance conditions

For emptiness of HRAs to be decidable, it is important that we cannot check histories for emptiness arbitrarily (cf. full-powered counter machines). But what about allowing just one such final check?

$$\mathcal{L}_{\text{twice}} = \{\, w \in \mathcal{D}^* \mid \forall d \in \nu(w).\ d \text{ occurs exactly twice in } w \,\}$$

The above could be accepted by the following HRA

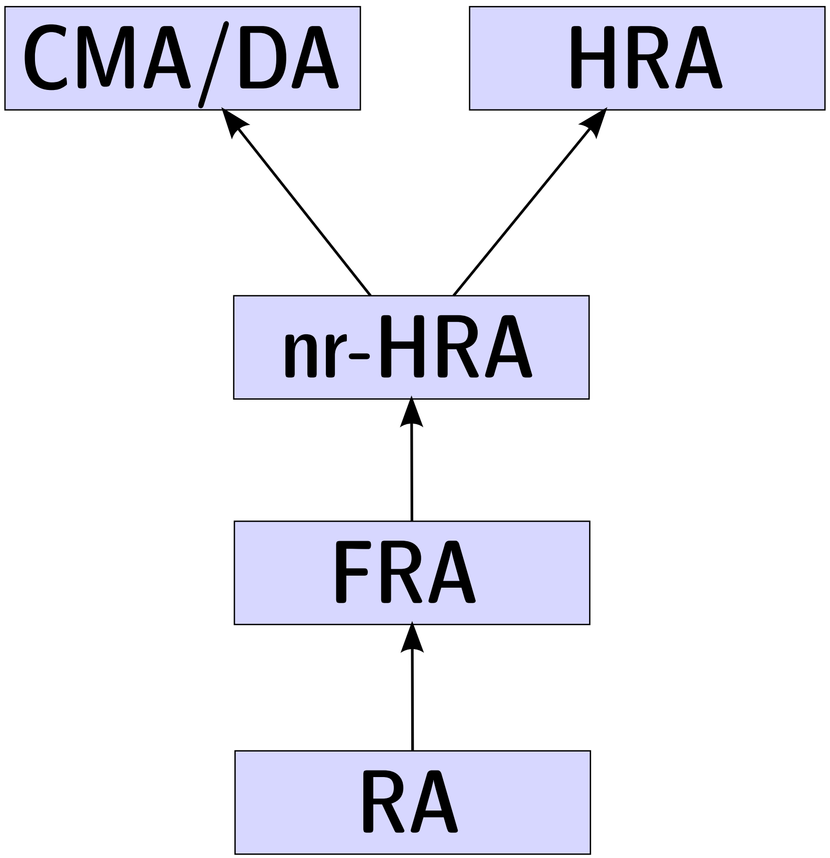


if we had imposed a **local acceptance condition**: “history 1 empty”

A model equivalent to non-reset HRAs + local acceptance conditions is **Data Automata/ Class Memory Automata** (cf. References)

The expressivity picture

H'



Summary and References

Automata with History/Class Storage

- History-Register Automata: definitions and examples
- Simulation of registers via histories and resets
- Closures and non-closures
- VASS and emptiness decidability
- Non-reset HRAs
- Local acceptance: DA/CMA

References and further directions

- H. Björklund, T. Schwentick: On notions of regularity for data languages. Theor. Comput. Sci. 411(4-5): 702-715 (2010)
- M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, L. Segoufin: Two-variable logic on data words. ACM Trans. Comput. Log. 12(4): 27 (2011)
- C. Cotton-Barratt, A. S. Murawski, C.-H. Luke Ong: Weak and Nested Class Memory Automata. LATA 2015: 188-199
- N. Tzevelekos, R. Grigore: History-Register Automata. FoSSaCS 2013: 17-33

Exercises

1. Show that the language accepted by HRAs are closed under union, concatenation and Kleene star.
2. Show that there is no HRA accepting the language:

$$\mathcal{L}^{\text{exact}}_{\text{prod/cons}} = \{\, ww' \in \mathcal{D}^* \mid w, w' \in \mathcal{L}_{\mathrm{fresh}} \wedge \nu(w') = \nu(w) \,\}$$

3. Design an HRA that accepts the complement of $\mathcal{L}_{\text{twice}}$.
4. Design an HRA with a local acceptance condition that accepts the language $\mathcal{L}^{\text{exact}}_{\text{prod/cons}}$ defined above.