Alternation Elimination by Complementation

Christian Dax, Felix Klaedtke

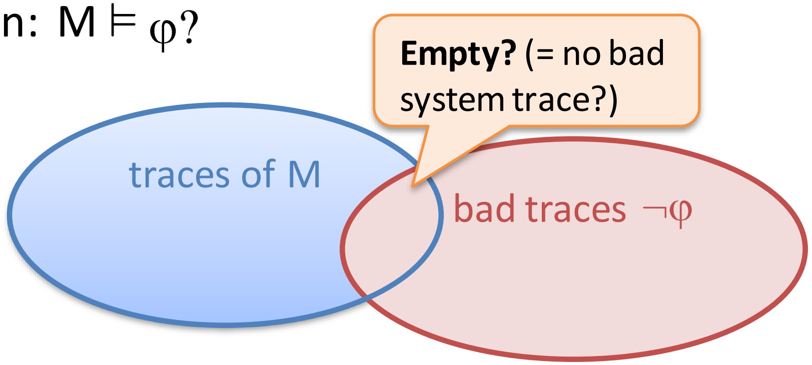
ETH Zurich

Recent results and ongoing work

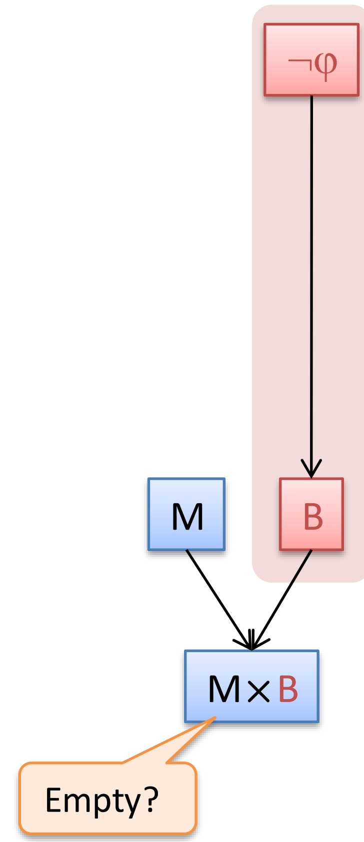
ETH Zurich, August 12th, 2008

Motivation: Finite-State Model Checking

- Consider the problem:
  - Given: finite-state system M (system traces)
  - Given: specification as temporal formula $\varphi \Rightarrow \neg\varphi$ (bad traces)
  - Question: $M \models \varphi$?



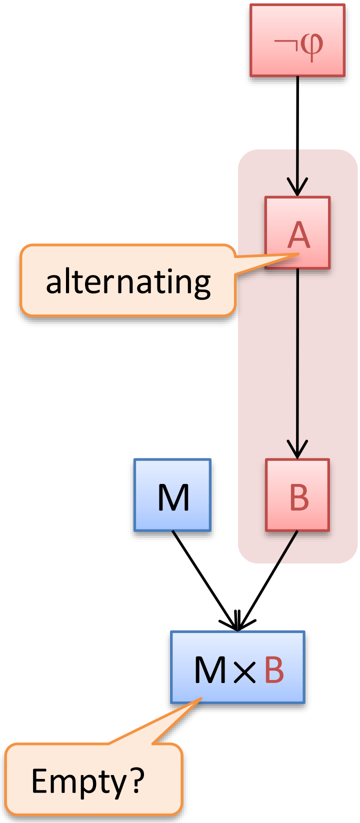
- Automata-based approach:
  1. View M as nondeterministic automaton
  2. Translate $\neg\varphi$ to nondeterministic automaton B
  3. Represent intersection via product automaton $M \times B$
  4. Check emptiness of $M \times B$



Motivation: Alternation Elimination

- Translation via alternating automaton:
  1. **Direct/efficient:** formula to alternating automaton
  2. **Complex/crucial:** alternating to nondeterministic automaton
  3. **Easy/efficient:** emptiness check

- This talk: focus on step 2.

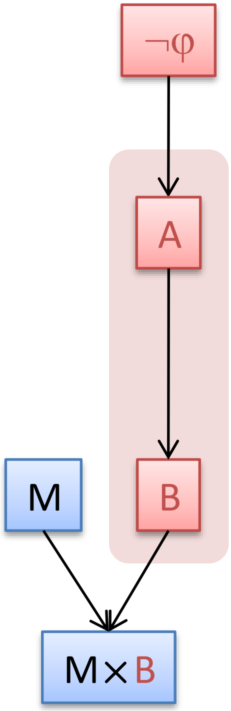


Outline

1. Background: automata

2. From alternating to nondeterministic automata

3. From PSL logic + past operators to nondeterministic automata (includes ongoing work)



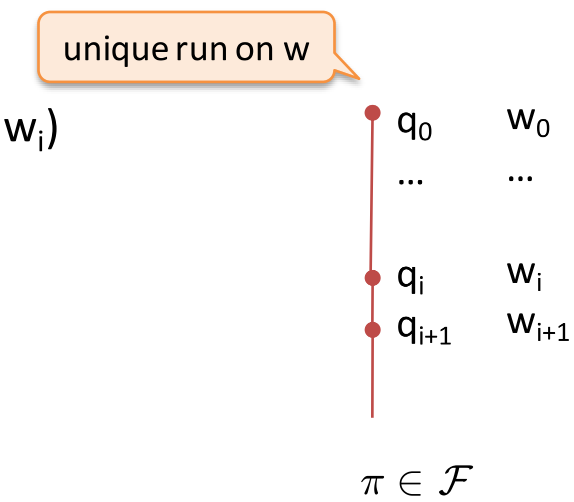
Background: Automata

Deterministic Automata (DA)

- A **DA** is a tuple $(Q, \Sigma, \delta, q_0, \mathcal{F})$
  - δ: $Q \times \Sigma \to Q$ transition function
  - $\mathcal{F} \subseteq Q^\omega$ set of sequences over Q that are accepting

  - Remark: Büchi/co-Büchi condition given as $F \subseteq Q$
    Büchi: $\mathcal{F}_F = \{\pi \in Q^\omega \mid \pi$ visits F-states ∞-often$\}$
    co-Büchi: $\mathcal{F}_F = \{\pi \in Q^\omega \mid \pi$ does not visit F-states ∞-often$\}$

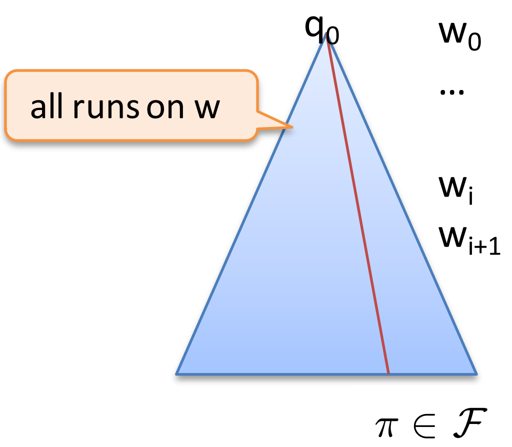
- For a word $w = w_0 w_1 \ldots$
  - A **run** $q_0 q_1 \ldots$ is a sequence of states with $q_{i+1} = \delta(q_i, w_i)$
  - w is **accepted** :⇔ the run $\pi = q_0 q_1 \ldots$ on w is in $\mathcal{F}$



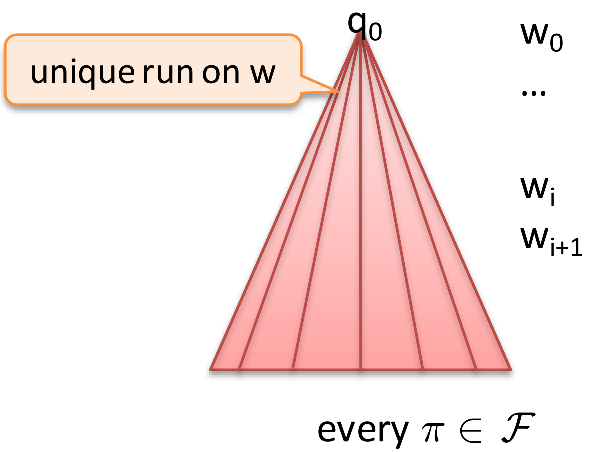
- **Syntax**: 'automaton as relation over words'
  - $\mathcal{A}(w)$:⇔ word w is accepted by automaton $\mathcal{A}$

Nondeterministic/Universal Automata (NA/UA)

- An **NA/UA** is a tuple $(Q, \Sigma, \delta, q_0, \mathcal{F})$
  - $\delta: Q \times \Sigma \rightarrow 2^Q$ transition function



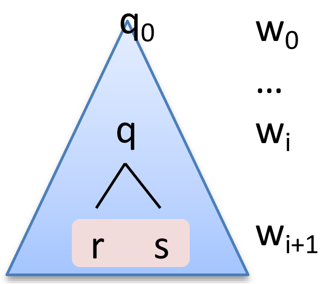
- For a word $w = w_0 w_1 \ldots$
  - A **nondeterministic run** $q_0 q_1 \ldots$ is a sequence of states with $q_{i+1} \in \delta(q_i, w_i)$
  - w is **accepted** :⇔ there is a run on w that is in $\mathcal{F}$



  - A **universal run** is a Q-labeled tree
    - the root is labeled by q_0, and
    - a q-labeled node in level i has children labeled by $\delta(q, w_i)$
  - w is **accepted** :⇔ every path in the run is in $\mathcal{F}$

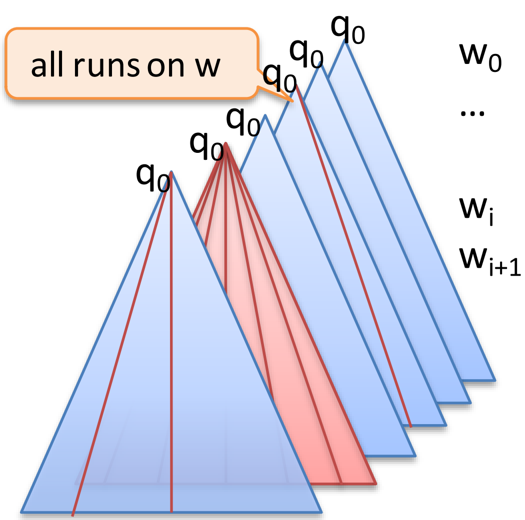
Alternating Automata (AA)

- An **AA** is a tuple $(Q, \Sigma, \delta, q_0, \mathcal{F})$
  - $\delta: Q \times \Sigma \rightarrow \mathcal{B}^+(Q)$ transition function
  - Here, we assume that $\delta(q, a)$ is in DNF, for all (q, a)



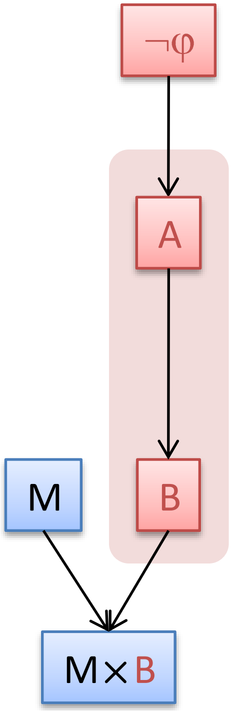
$\delta(q, w_i) = (r \wedge s) \vee (s \wedge t)$

- For a word $w = w_0 w_1 \ldots$
  - A **alternating run** is a Q-labeled tree, where
    - the root is labeled by q_0, and
    - a q-labeled node in level i has children that are labeled by one of the monomials of $\delta(q, w_i)$
  - w **accepted** :⇔ there is a run s.t. every path is in $\mathcal{F}$



every $\pi \in \mathcal{F}$

From Alternating to Nondeterministic Automata

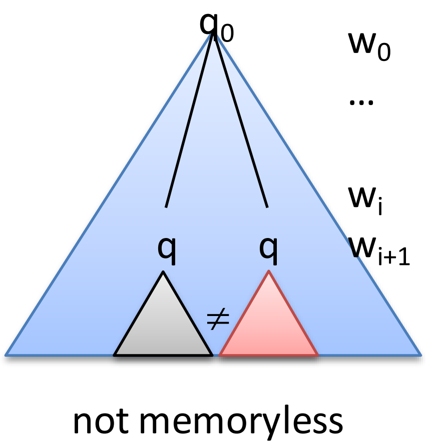


Related Work

- We use building blocks that appeared in
  - Vardi (POPL ’88, ICALP ‘98),
  - Miyano-Hayashi (TCS ’92),
  - Lange-Stirling (LICS ’01),
  - Kupferman-Piterman-Vardi (CONCUR ’01),
  - Gastin-Oddoux (CAV ’01, MFCS ‘03),
  - Dax-Hofmann-Lange (FSTTCS ’06).

- We unify and generalize building blocks:
  - The papers mentioned above solve particular translation problems.
  - We identify and refine the main ingredients of these translations.
  - We present one scheme that unifies + simplifies constructions and proofs.

Step 1 of 2: Run as Word



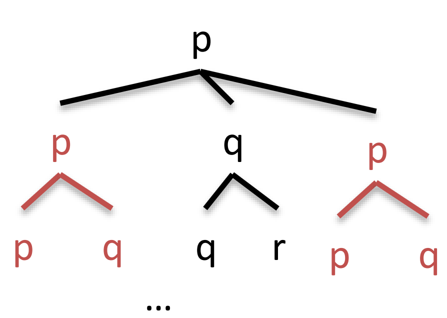
- Memoryless automata
  - We use that Rabin, parity, … automata are memoryless.
  - A **run is memoryless** :⇔ equally labeled nodes in the same level have equally labeled subtrees
  - An **AA is memoryless** :⇔ every accepted word has a memoryless accepting run

- Memoryless run as word:
  - Merge equally-labeld nodes in same level
  - Encode memoryless run as word $\mathbf{r_0 r_1 r_2 \ldots} \in \mathbf{(Q \to 2^Q)}^\omega$
  - $r_i(q)$: 'labels of children of q-labeled node in level i'
  - Example:

$r_0(p) = \{p, q\}$

$r_1(p) = \{p, q\}, r_1(q) = \{q, r\}$

$r_2(p) = \ldots, r_2(q) = \ldots, r_2(r) = \ldots$

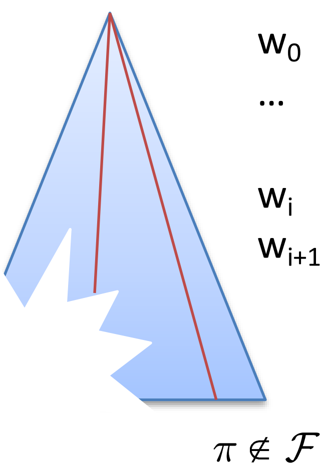


Step 2 of 2: Alternation Elimination

- Let $\mathcal{A}$ = (Q, Σ, δ, q_0, $\mathcal{F}$) be an AA and Γ := Q $\to 2^Q$
- A word w is accepted
  - $\Leftrightarrow$ there is a run on w s.t. every path is in $\mathcal{F}$
  - $\Leftrightarrow \exists$ r: r $\in$ runs(w) $\wedge \forall \pi \in$ r: $\pi \in \mathcal{F}$
  - $\Leftrightarrow \exists$ r: $\neg$ (r $\notin$ runs(w) $\vee \exists \pi \in$ r: $\pi \notin \mathcal{F}$) ★
  - $\Leftrightarrow \exists$ r: $\neg$ $\mathcal{B}$(w, r)

'refuter's strategy'

- It is easy to build an NA $\mathcal{B}$ over $\Sigma \times \Gamma$ for ★
  - $\mathcal{B}$:= (Q, $\Sigma \times \Gamma$, η, q_0, $Q^\omega \setminus \mathcal{F}$)
  - η(q, (a,r)) := $\begin{cases} r(q) & r(q) \text{ is monomial in } \delta(q, a) \\ \{acc\text{-}sink\} & \text{otherwise} \end{cases}$



- Finally: complement the NA $\mathcal{B}$ and project it on Σ.

Some Instances

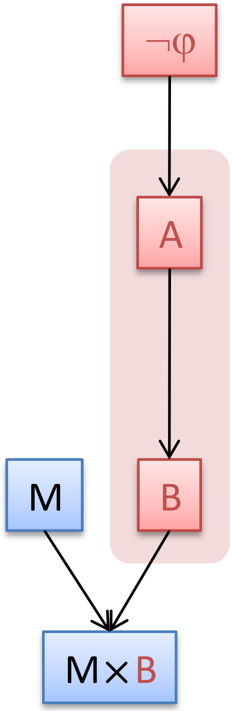
- Remark: scheme also works for 2-way automata
  - 2-way automata can move the read-only head in both directions.

- Number of states of resulting 1-way NBAs

	1-Weak Büchi LTL (+ Past)	**Büchi** PSL (+ Past)	**Parity** μLTL (+ Past)	**Rabin**
1-way	$O(n2^{n})$	$O(2^{2n})$	$O(2^{nk \log n})$	$\mathbf{O(2^{nk \log nk})}$
2-way	$\mathbf{O(n2^{3n})}$	$O(2^{n*n})$	$O(2^{nk*nk})$	
2-way + loop-free	$O(n2^{2n})$	$\mathbf{O(2^{4n})}$	-- in progress --	-- in progress --

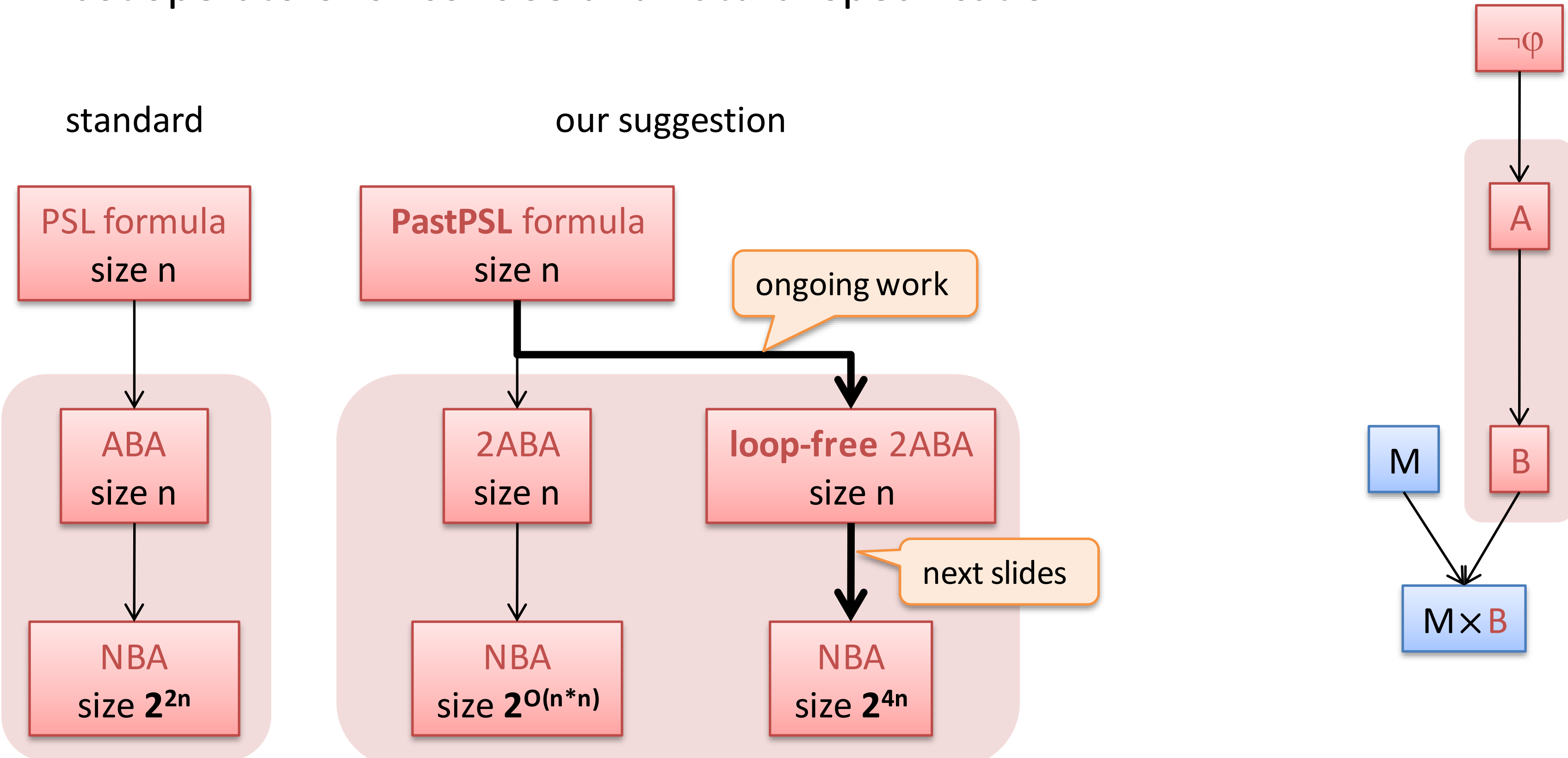
From PSL with Past to Nondeterministic Büchi Automata (NBAs)

(includes ongoing work)



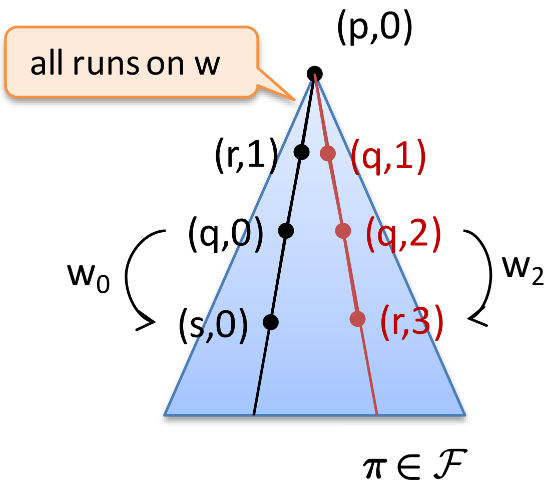
Motivation: Property Specification Language (PSL)

- PSL is an IEEE standard and increasingly used in hardware industry
- linear-time fragment of PSL $\approx$ LTL + regular expressions + syntactic sugar
- Past operators for concise and natural specification



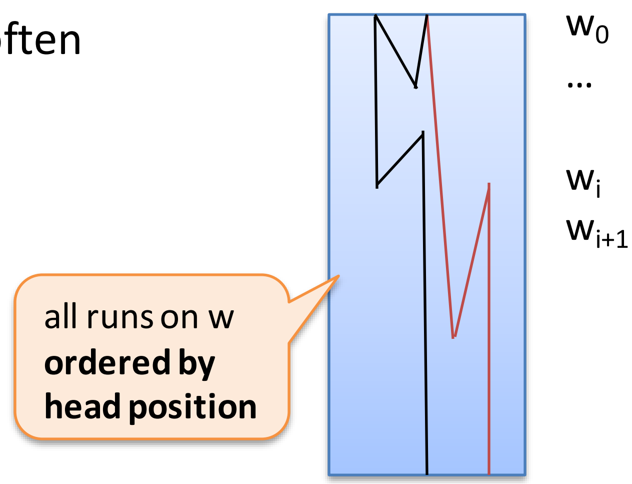
Background: 2-Way Nondet. Büchi Automata (2NBA)

- A 2NBA is a tuple $(Q, \Sigma, \delta, q_0, F)$
  - $\delta: Q \times \Sigma \rightarrow 2^{Q \times \{-1, 0, 1\}}$ transition function
  - Additional info where to move the read-only head



- For a word $w = w_0 w_1 \ldots$
  - A **configuration** (q, j) is a pair in $Q \times$ 'head positions'
  - A run $(q_0, j_0)\ (q_1, j_1) \ldots$ is a sequence of configurations with $(q_{i+1}, j_{i+1} - j_i) \in \delta(q_i, w\_j_i)$
  - w accepted $\Leftrightarrow$ ex. run on w that visits F-states ∞-often

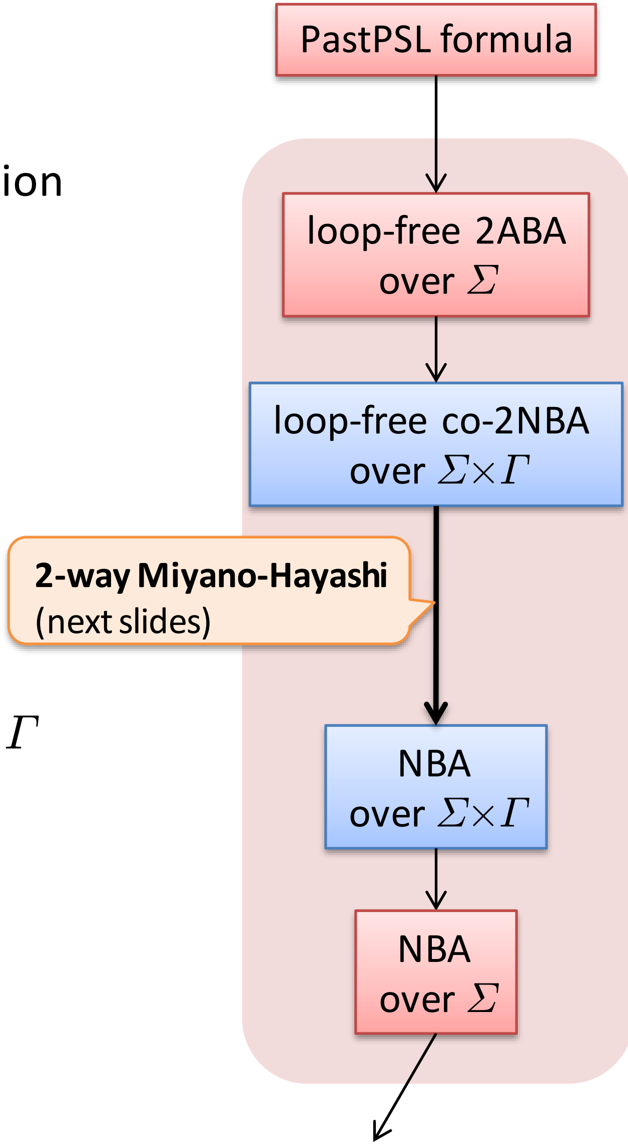
- For AAs: $Q \times$ 'head positions'-labeled run-trees



Outline: From PSL to NBA

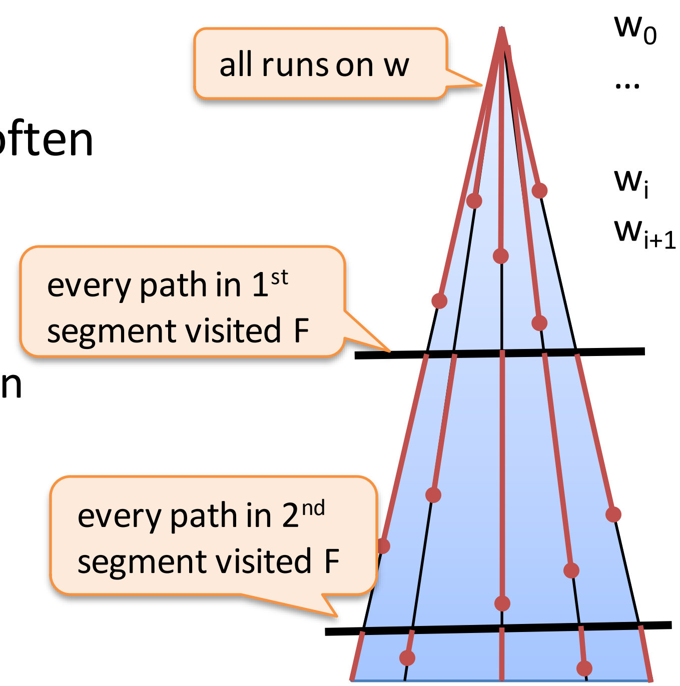
- Loop-freeness
  - A **run is loop-free** :⇔ for every path, no configuration occurs twice on the path
  - An **AA is loop-free** :⇔ every run is loop-free

- PastPSL to 1-way NBA
  1. PastPSL formula $\rightarrow$ 2-way ABA (ongoing work)
  2. Construction scheme:
     - Lemma: if AA is loop-free then $\mathcal{B}$ is loop-free.
     - Construct loop-free 2-way co-NBA $\mathcal{B}$ over $\Sigma\times\Gamma$
     - Complement with **2-way Miyano-Hayashi**
     - Project resulting 1-way NBA on Σ

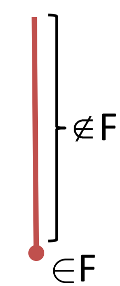


1-Way Miyano-Hayashi Complementation

- A co-NBA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ accepts a word w
  :⇔ ex. run on w that does not visit F-states ∞-often



- NBA for the complement
  - w rejected ⇔ each run of $\mathcal{A}$ on w visits F ∞-often
  - $\mathcal{B} := (2^Q \times 2^Q, \Sigma, \eta, (\{q_0\},\emptyset), 2^Q \times \{\emptyset\})$
  - $\eta((R, \emptyset), a) := (\delta(R,a), \delta(R,a) \setminus F)$
  - $\eta((R, S), a) := (\delta(R,a), \delta(S,a) \setminus F)$
  - Subset-construction with R-component:
    compute all runs in parallel (**black** lines)
  - States of S-component have to visit F (**red** lines)
  - $2^Q \times \{\emptyset\}$ is visited ∞-often ⇔ every run visits F ∞-often

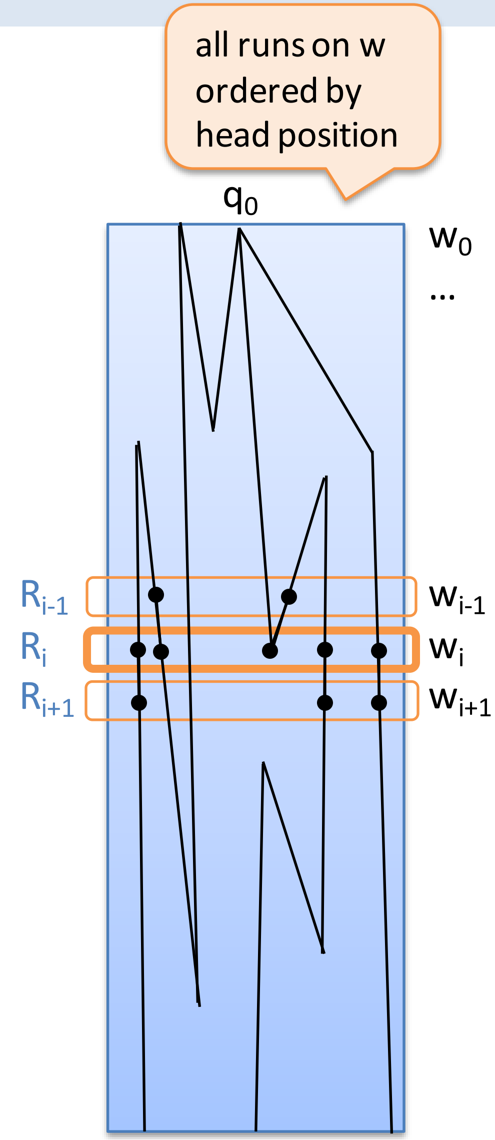


2-Way Miyano-Hayashi Complementation

- A loop-free co-2NBA $\mathcal{A}$ = (Q, Σ, δ, q_0, F) accepts w :⇔ ex. run on w that does not visit F-states ∞-often

- 1-way NBA for the complement
  - w rejected ⇔ every run of $\mathcal{A}$ on w visits F ∞-often

1. Guess sequence $R_0R_1\ldots \in (2^Q)^\omega$ that represents all runs on w ordered by head positions (**black** lines).
2. Check locally that guess is correct:
   if p $\in$ R_i and (q, d) $\in$ δ(p, w_i) then q $\in$ R_{i+d}

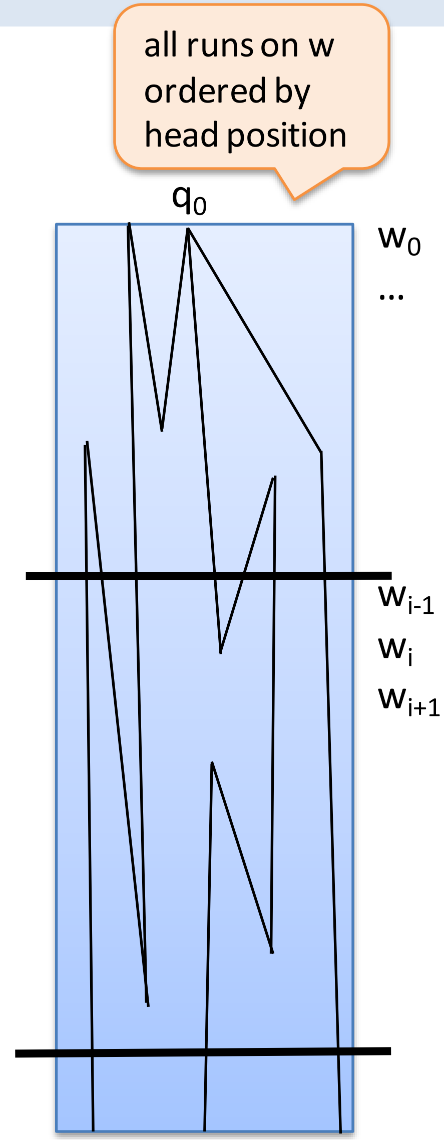


2-Way Miyano-Hayashi Complementation

- A loop-free co-2NBA $\mathcal{A}$ = (Q, Σ, δ, q_0, F) accepts w
  :⇔ ex. run on w that does not visit F-states ∞-often

- 1-way NBA for the complement
  - w rejected ⇔ every run of $\mathcal{A}$ on w visits F ∞-often

1. Guess sequence $R_0R_1 \ldots \in (2^Q)^\omega$ that represents all runs on w ordered by head positions (**black** lines).
2. Check locally that guess is correct:
   if $p \in R_i$ and $(q, d) \in \delta(p, w_i)$ then $q \in R_{i+d}$
3. Guess breakpoints:
   - partitioning of the R-sequence in segments
   - each run starting at the previous breakpoint visits F before reaching the next breakpoint

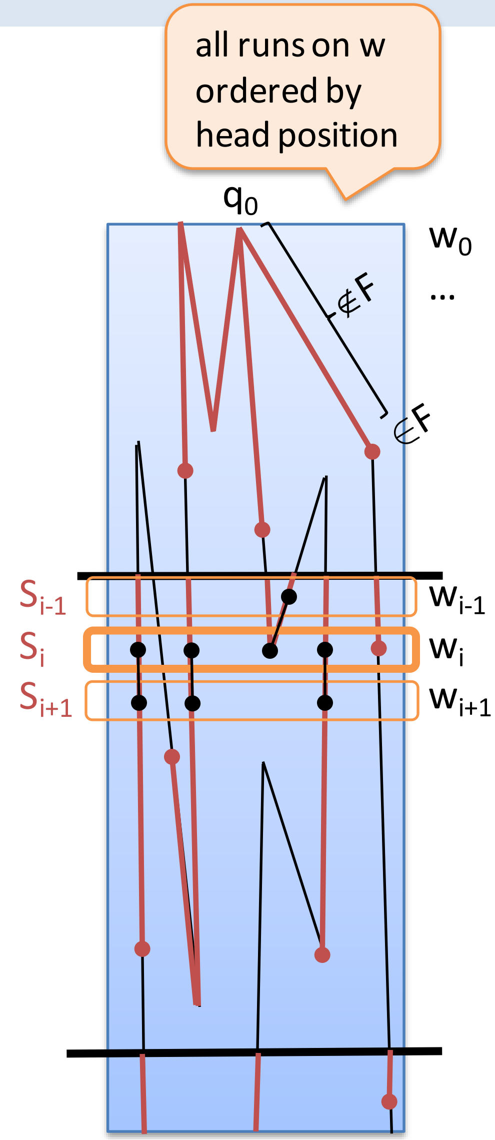


2-Way Miyano-Hayashi Complementation

- A loop-free co-2NBA $\mathcal{A}$ = (Q, Σ, δ, q_0, F) accepts w
  :⇔ ex. run on w that does not visit F-states ∞-often

- 1-way NBA for the complement
  - w rejected ⇔ every run of $\mathcal{A}$ on w visits F ∞-often

4. Guess sequence $S_0 S_1 \ldots \in (2^{Q \setminus F})^\omega$ that represents all runs from q_0 or a breakpoint to an F-state (**red** lines).
5. Check locally that guess is correct:
   if $p \in S_i$, $(q, d) \in \delta(p, w_i)$ and $q \notin F$ then either $q \in S_{i+d}$ or $S_{i+d} = \emptyset$ (breakpoint).

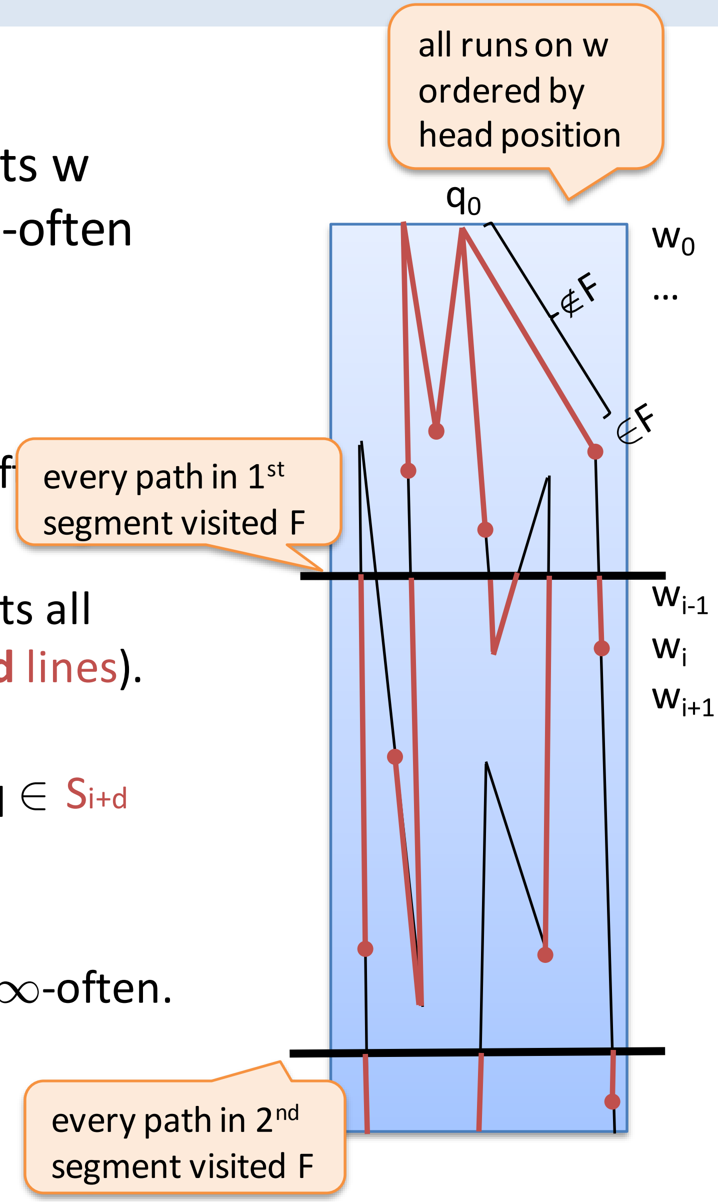


2-Way Miyano-Hayashi Complementation

- A loop-free co-2NBA $\mathcal{A}$ = (Q, Σ, δ, q_0, F) accepts w
  :⇔ ex. run on w that does not visit F-states ∞-often

- 1-way NBA for the complement
  - w rejected ⇔ every run of $\mathcal{A}$ on w visits F ∞-of

4. Guess sequence $S_0 S_1 \ldots \in (2^{Q \setminus F})^\omega$ that represents all runs from q_0 or a breakpoint to an F-state (**red** lines).
5. Check locally that guess is correct:
   if p $\in$ S_i, (q, d) $\in$ δ(p, w_i) and q $\notin$ F then either q $\in$ S_{i+d} or S_{i+d} = $\emptyset$ (breakpoint).
6. Check that pattern '$S_i = \emptyset$, $S_{i+1} = R_{i+1} \setminus F$' occurs ∞-often.



Conclusion

- Construction scheme for translating AAs to NAs
  - Requires complementation construction for NA with co-acceptance condition
  - Requires AA to accept by memoryless runs
  - 3 novel translations
  - Previous translations can be seen as instances: unifies and simplifies constructions and proofs
- Novel complementation construction for loop-free co-2NBAs
  - 1-way Miyano-Hayashi and constructions by Gastin-Oddoux are special cases
  - Efficient automata constructions for PastPSL possible
- Ongoing and future work
  - Scheme for automata that do not accept by memoryless runs
  - Translations for PSL and μLTL with past operators
  - Practical experiences of translating 2-way AAs to NAs