Simulation Reduction of Finite Nondeterministic Word and Tree Automata

Tomáš Vojnar

FIT, Brno University of Technology, Czech Republic

Plan of the Lecture

- Mediated Simulation Reduction for Finite Word Automata
- Simulation-based Reduction of Finite Tree Automata
- Computing Simulations on Tree Automata and Labelled Transition Systems

Mediated Simulation Reduction for Finite Word Automata

How to reduce NFA?

- Computing minimal deterministic automata is not a good way:
  - requires determinisation – costly, may run out of memory even before one can begin with the actual minimisation,
  - the result can still be bigger than the original nondeterministic automaton.

How to reduce NFA?

- Computing minimal deterministic automata is not a good way:
  - requires determinisation – costly, may run out of memory even before one can begin with the actual minimisation,
  - the result can still be bigger than the original nondeterministic automaton.

- A well-known way of reducing the size of nondeterministic automata without determinizing them is quotienting w.r.t. forward/backward (bi)simulation equivalence.

Simulation-based NFA Reduction

- Forward simulation F for word automata:
  - qFr implies that
    - if $q \xrightarrow{a} q'$, then $r \xrightarrow{a} r'$ with $q'Fr'$, and
    - $q \in \mathcal{F} \implies r \in \mathcal{F}$ where $\mathcal{F}$ are the final states.
  - F implies inclusion of languages accepted from states.

Simulation-based NFA Reduction

- Forward simulation F for word automata:
  - qFr implies that
    - if $q \xrightarrow{a} q'$, then $r \xrightarrow{a} r'$ with $q' F r'$, and
    - $q \in \mathcal{F} \implies r \in \mathcal{F}$ where $\mathcal{F}$ are the final states.
  - F implies inclusion of languages accepted from states.

- Backward simulation B for word automata:
  - qBr implies that
    - if $q' \xrightarrow{a} q$, then $r' \xrightarrow{a} r$ with $q' B r'$, and
    - $q \in \mathcal{I} \implies r \in \mathcal{I}$ where $\mathcal{I}$ are the initial states.
  - B implies inclusion of languages accepted at states.

Simulation-based NFA Reduction

- Forward simulation F for word automata:
  - qFr implies that
    - if $q \xrightarrow{a} q'$, then $r \xrightarrow{a} r'$ with $q'Fr'$, and
    - $q \in \mathcal{F} \implies r \in \mathcal{F}$ where $\mathcal{F}$ are the final states.
  - F implies inclusion of languages accepted from states.

- Backward simulation B for word automata:
  - qBr implies that
    - if $q' \xrightarrow{a} q$, then $r' \xrightarrow{a} r$ with $q'Br'$, and
    - $q \in \mathcal{I} \implies r \in \mathcal{I}$ where $\mathcal{I}$ are the initial states.
  - B implies inclusion of languages accepted at states.

- A simulation S is a pre-order (reflexive and transitive). For quotienting, one needs a simulation equivalence, which can be obtained by taking the symmetric closure $S \cap S^{-1}$.

Bisimulation-based NFA Reduction

- One can also quotient wrt. forward/backward bisimulations.
  - Forward bisimulation F for word automata:
    - qFr implies that
      - if $q \xrightarrow{a} q'$, then $r \xrightarrow{a} r'$ with $q'Fr'$,
      - symmetrically, if $r \xrightarrow{a} r'$, then $q \xrightarrow{a} q'$ with $q'Fr'$, and
      - $q \in \mathcal{F} \Leftrightarrow r \in \mathcal{F}$.
  - Backward simulation B for word automata:
    - qBr implies that
      - if $q' \xrightarrow{a} q$, then $r' \xrightarrow{a} r$ with $q'Br'$,
      - symmetrically, if $r' \xrightarrow{a} r$, then $q' \xrightarrow{a} q$ with $q'Br'$, and
      - $q \in \mathcal{I} \Leftrightarrow r \in \mathcal{I}$.

Bisimulation-based NFA Reduction

- One can also quotient wrt. forward/backward bisimulations.
  - Forward bisimulation F for word automata:
    - qFr implies that
      - if $q \xrightarrow{a} q'$, then $r \xrightarrow{a} r'$ with $q'Fr'$,
      - symmetrically, if $r \xrightarrow{a} r'$, then $q \xrightarrow{a} q'$ with $q'Fr'$, and
      - $q \in \mathcal{F} \Leftrightarrow r \in \mathcal{F}$.
  - Backward simulation B for word automata:
    - qBr implies that
      - if $q' \xrightarrow{a} q$, then $r' \xrightarrow{a} r$ with $q'Br'$,
      - symmetrically, if $r' \xrightarrow{a} r$, then $q' \xrightarrow{a} q$ with $q'Br'$, and
      - $q \in \mathcal{I} \Leftrightarrow r \in \mathcal{I}$.
- Bisimulations are equivalences, so no need to make a symmetric closure.
- Rough time complexity for m transitions and n states:
  - computing simulation: $\mathcal{O}(m.n)$, computing bisimulation: $\mathcal{O}(m.log\ n)$.

Bisimulation-based NFA Reduction

- One can also quotient wrt. forward/backward bisimulations.
  - Forward bisimulation F for word automata:
    - qFr implies that
      - if $q \xrightarrow{a} q'$, then $r \xrightarrow{a} r'$ with $q'Fr'$,
      - symmetrically, if $r \xrightarrow{a} r'$, then $q \xrightarrow{a} q'$ with $q'Fr'$, and
      - $q \in \mathcal{F} \Leftrightarrow r \in \mathcal{F}$.
  - Backward simulation B for word automata:
    - qBr implies that
      - if $q' \xrightarrow{a} q$, then $r' \xrightarrow{a} r$ with $q'Br'$,
      - symmetrically, if $r' \xrightarrow{a} r$, then $q' \xrightarrow{a} q$ with $q'Br'$, and
      - $q \in \mathcal{I} \Leftrightarrow r \in \mathcal{I}$.
- Bisimulations are equivalences, so no need to make a symmetric closure.
- Rough time complexity for m transitions and n states:
  - computing simulation: $\mathcal{O}(m.n)$, computing bisimulation: $\mathcal{O}(m.log\ n)$.
- The use of forward and backward (bi)simulation can be efficiently combined in coarser (and hence better reducing) mediated equivalences.

Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

- Quotienting corresponds to merging some states,
  - which is the same as allowing "jumps" (ϵ-transitions) between the states.

Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

- Quotienting corresponds to merging some states,
  - which is the same as allowing "jumps" (ϵ-transitions) between the states.

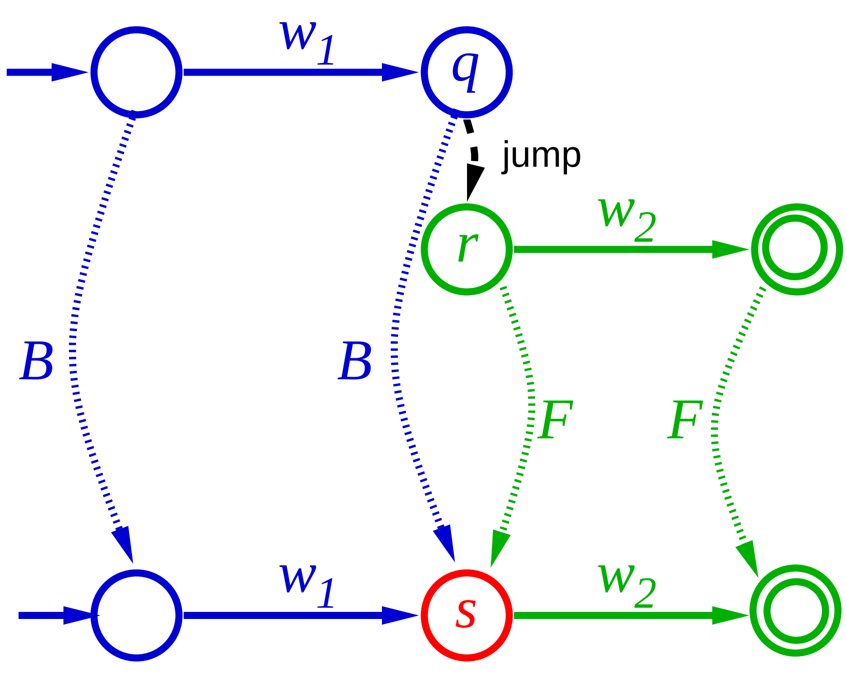
- A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:

Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS’08, CIAA’08, MEMICS’08],
later [L. Clemente for Büchi automata – ICALP’11]

- Quotienting corresponds to merging some states,
  - which is the same as allowing “jumps” (ϵ-transitions) between the states.

- A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:

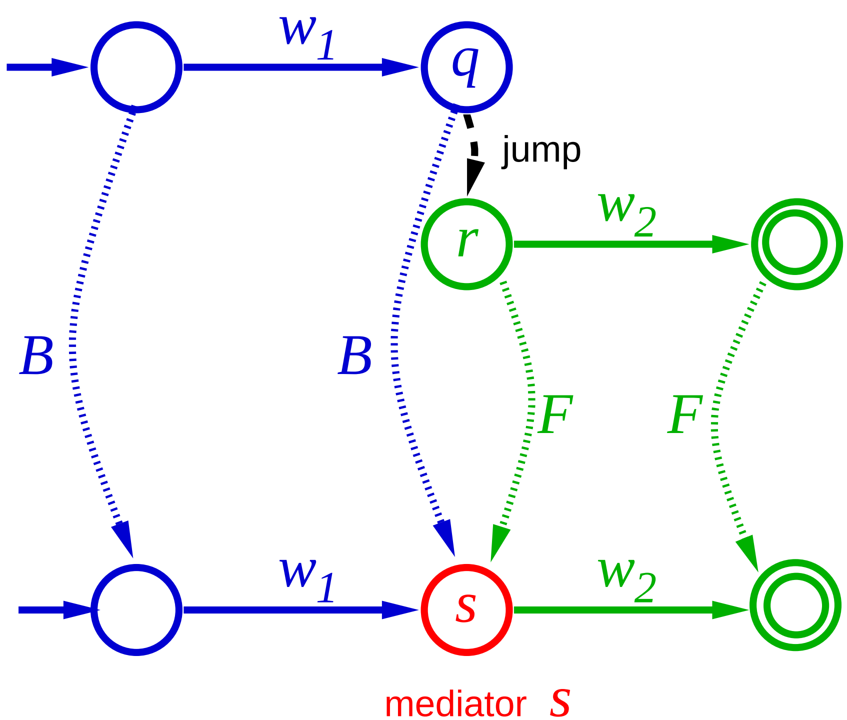


Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08], later [L. Clemente for Büchi automata – ICALP'11]

- Quotienting corresponds to merging some states,
  - which is the same as allowing "jumps" (ϵ-transitions) between the states.

- A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:



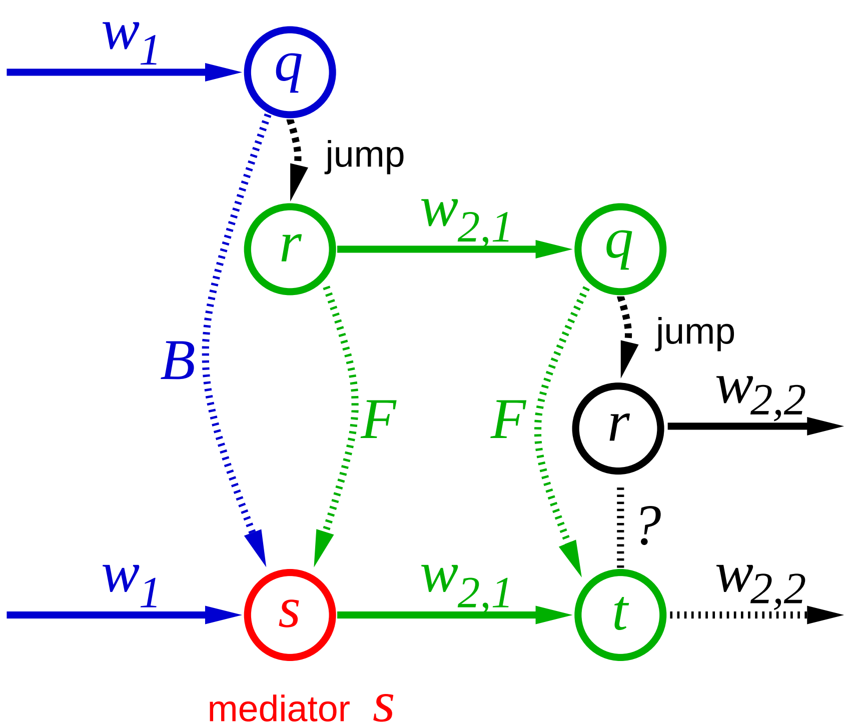
- Can we allow a jump if there is a mediator?

Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

- Quotienting corresponds to merging some states,
  - which is the same as allowing "jumps" (ϵ-transitions) between the states.

- A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:



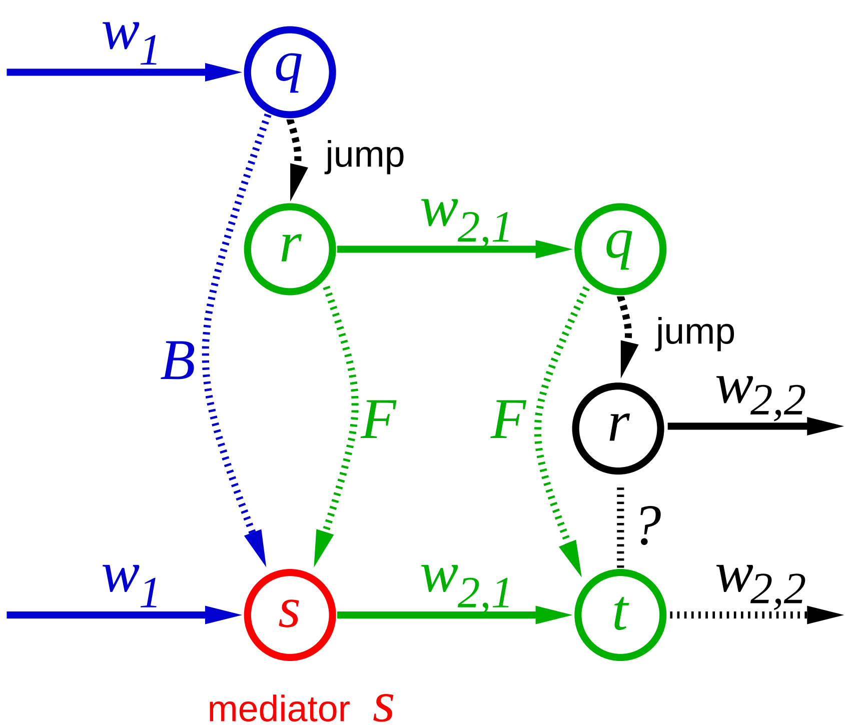
- Can we allow a jump if there is a mediator? NO, in general, we cannot.

Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

- Quotienting corresponds to merging some states,
  - which is the same as allowing “jumps” (ϵ-transitions) between the states.

- A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:



- Can we allow a jump if there is a mediator? NO, in general, we cannot.

- A fix: we take as the mediated preorder M the maximal transitive fragment of $B \circ F^{-1}$ that contains F^{-1}.

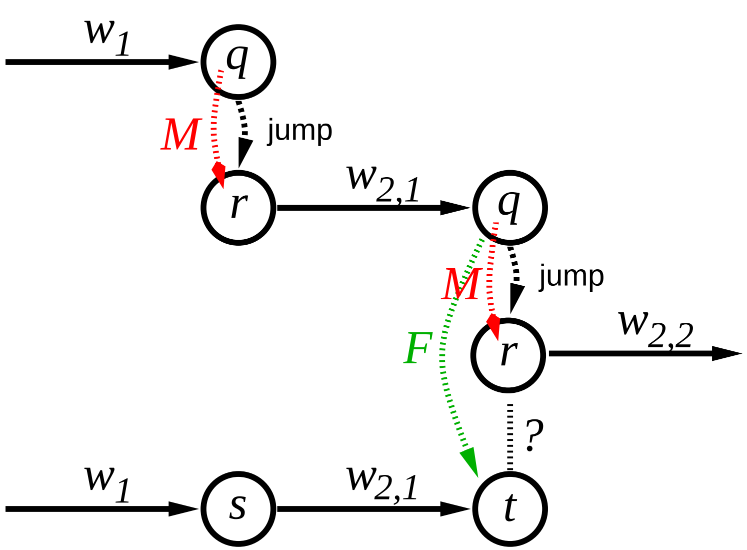
Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

❖ Quotienting corresponds to merging some states,

- which is the same as allowing "jumps" (ϵ-transitions) between the states.

❖ A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:



❖ Can we allow a jump if there is a mediator? NO, in general, we cannot.

❖ A fix: we take as the mediated preorder M the maximal transitive fragment of $B \circ F^{-1}$ that contains F^{-1}.

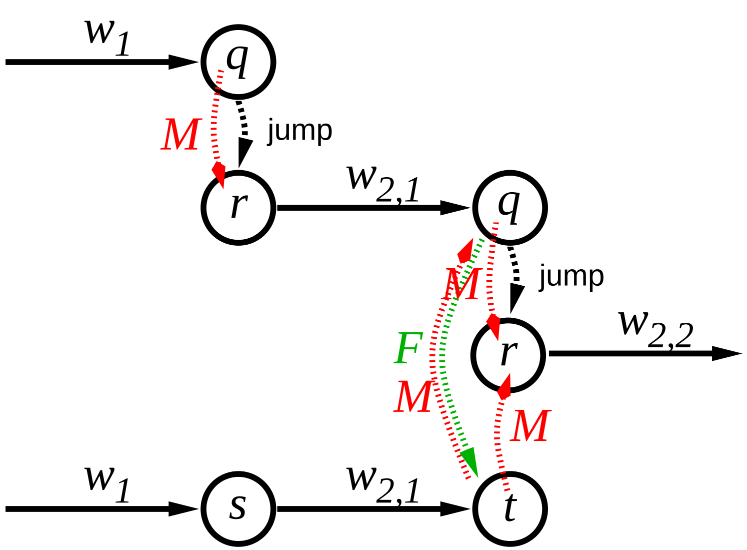
Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

❖ Quotienting corresponds to merging some states,

- which is the same as allowing "jumps" (ϵ-transitions) between the states.

❖ A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:



❖ Can we allow a jump if there is a mediator? NO, in general, we cannot.

❖ A fix: we take as the mediated preorder M the maximal transitive fragment of $B \circ F^{-1}$ that contains F^{-1}.

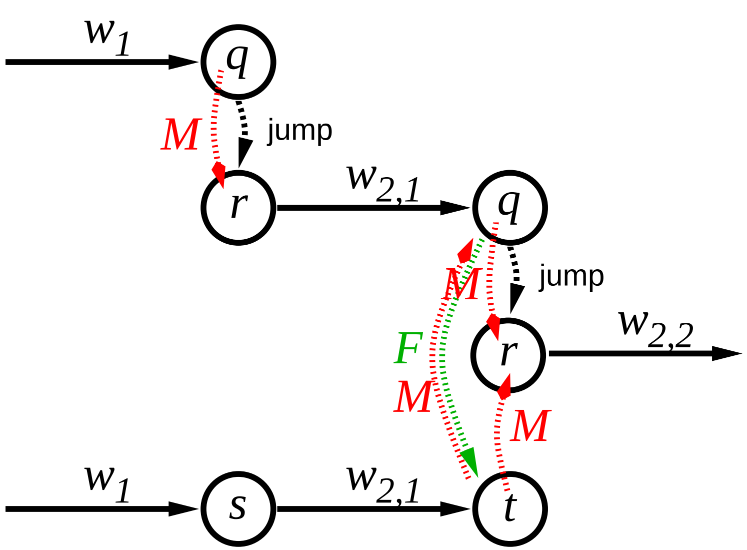
Mediated Simulation Reduction

[Abdulla, Bouajjani, Holík, Kaati, V. – TACAS'08, CIAA'08, MEMICS'08],
later [L. Clemente for Büchi automata – ICALP'11]

❖ Quotienting corresponds to merging some states,

- which is the same as allowing "jumps" (ϵ-transitions) between the states.

❖ A mediated preorder allows a jump from a state q to a state r only if there exists a mediator state s such that qBs and rFs:



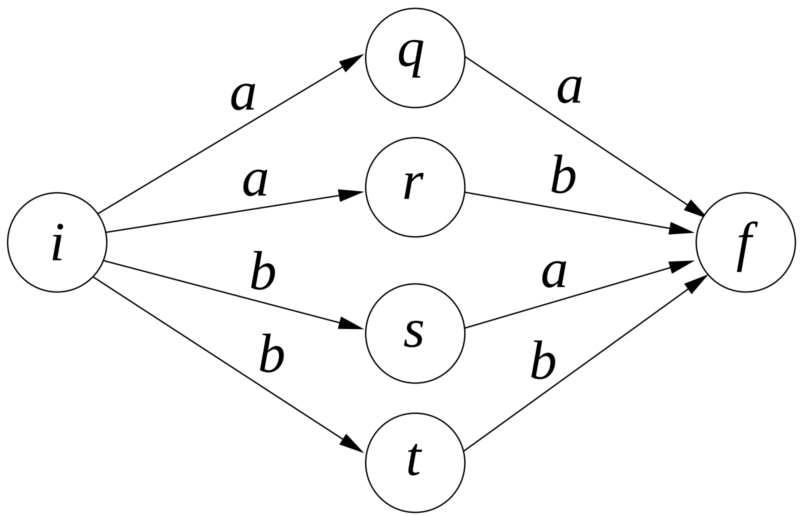
❖ Can we allow a jump if there is a mediator? NO, in general, we cannot.

❖ A fix: we take as the mediated preorder M the maximal transitive fragment of $B \circ F^{-1}$ that contains F^{-1}.

❖ We can merge states according to the mediated equivalence $\sim_M = M \cap M^{-1}$.

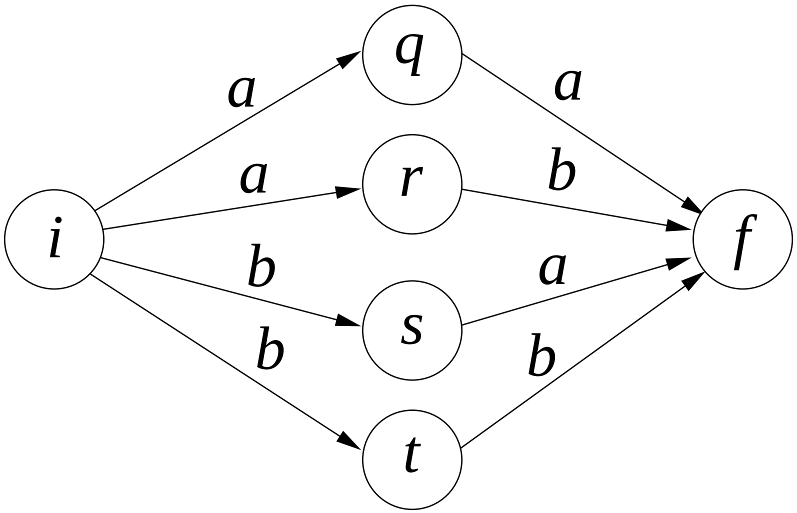
Mediated Reduction: An Example

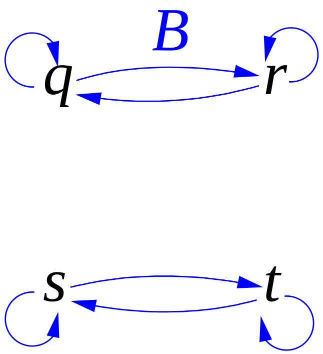
- Consider the following NFA with $\mathcal{I} = \{i\}$ and $\mathcal{F} = \{f\}$:



Mediated Reduction: An Example

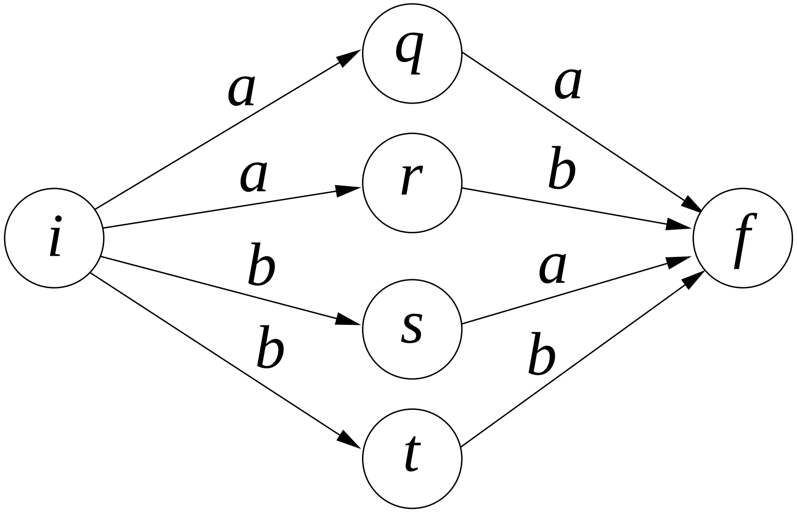
- Consider the following NFA with $\mathcal{I} = \{i\}$ and $\mathcal{F} = \{f\}$:

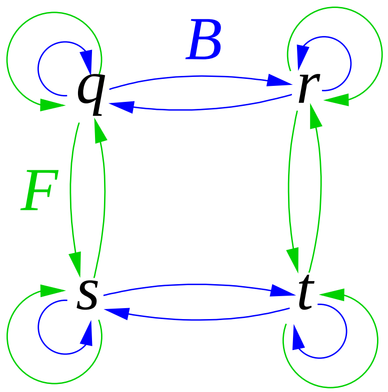




Mediated Reduction: An Example

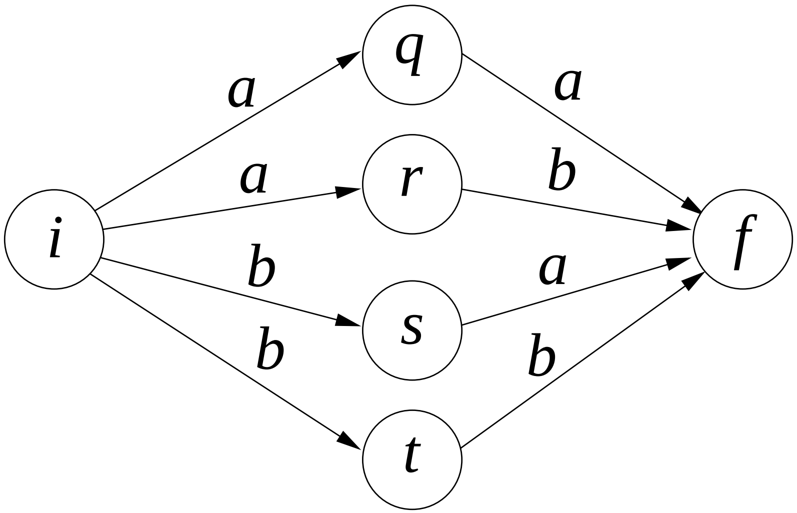
❖ Consider the following NFA with $\mathcal{I} = \{i\}$ and $\mathcal{F} = \{f\}$:

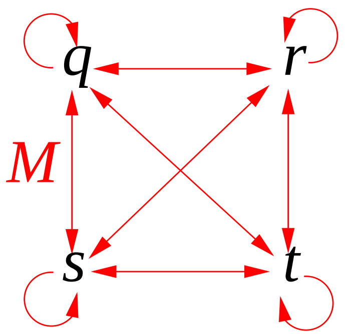




Mediated Reduction: An Example

❖ Consider the following NFA with $\mathcal{I} = \{i\}$ and $\mathcal{F} = \{f\}$:



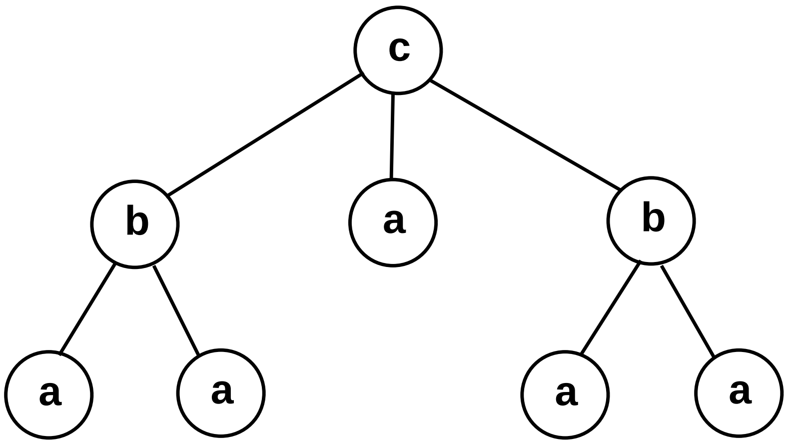


Mediated Simulation Reduction for Finite Tree Automata

Tree Automata

- A bottom-up tree automaton: $A = (Q, \Sigma, F, \Delta)$ where
  - Q is a finite set of states,
  - $F \subseteq Q$ is a set of final states,
  - Σ a ranked alphabet with a rank function $\# : \Sigma \rightarrow \mathbb{N}$,
  - Δ is a set of tree transition rules of the form as in the following example:

$$\Delta = \{ (r,q,r) \xrightarrow{c} s, \quad (q,q) \xrightarrow{b} r, \quad \xrightarrow{a} q \}$$

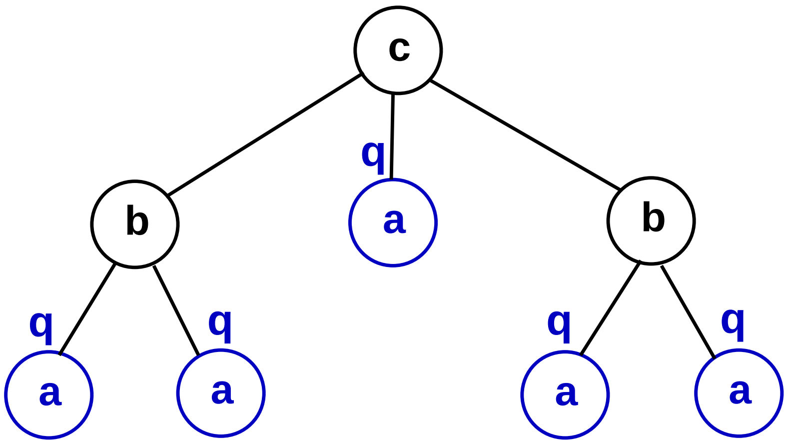


Tree Automata

❖ A bottom-up tree automaton: $A = (Q, \Sigma, F, \Delta)$ where

- Q is a finite set of states,
- $F \subseteq Q$ is a set of final states,
- Σ a ranked alphabet with a rank function $\# : \Sigma \to \mathbb{N}$,
- Δ is a set of tree transition rules of the form as in the following example:

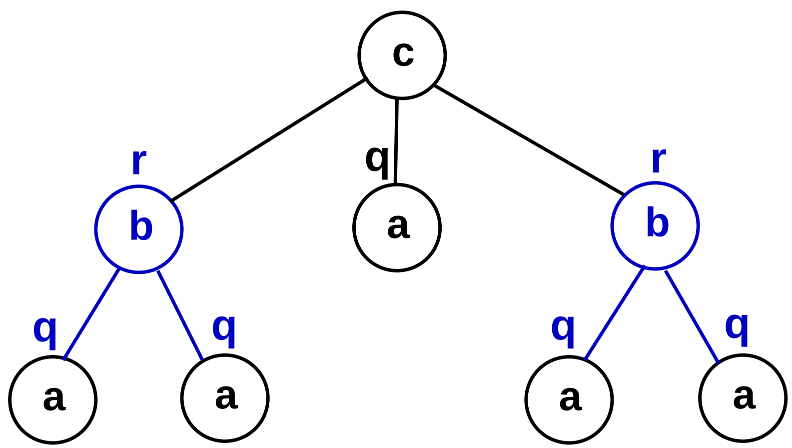
$$\Delta = \{ (r, q, r) \xrightarrow{c} s, \quad (q, q) \xrightarrow{b} r, \quad \xrightarrow{a} q \}$$



Tree Automata

- A bottom-up tree automaton: $A = (Q, \Sigma, F, \Delta)$ where
  - Q is a finite set of states,
  - $F \subseteq Q$ is a set of final states,
  - Σ a ranked alphabet with a rank function $\# : \Sigma \to \mathbb{N}$,
  - Δ is a set of tree transition rules of the form as in the following example:

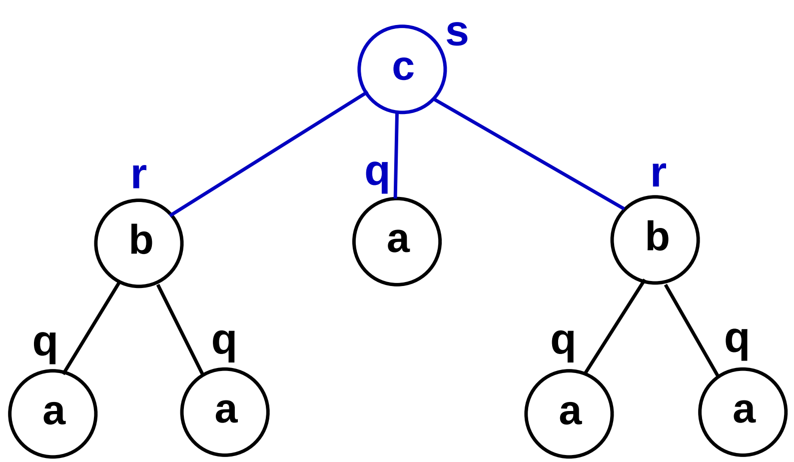
$$\Delta = \{ (r,q,r) \xrightarrow{c} s, \quad (q,q) \xrightarrow{b} r, \quad \xrightarrow{a} q \}$$



Tree Automata

- A bottom-up tree automaton: $A = (Q, \Sigma, F, \Delta)$ where
  - Q is a finite set of states,
  - $F \subseteq Q$ is a set of final states,
  - Σ a ranked alphabet with a rank function $\# : \Sigma \rightarrow \mathbb{N}$,
  - Δ is a set of tree transition rules of the form as in the following example:

$$\Delta = \{ \quad (r, q, r) \xrightarrow{c} s \quad (q, q) \xrightarrow{b} r \quad \xrightarrow{a} q \quad \}$$



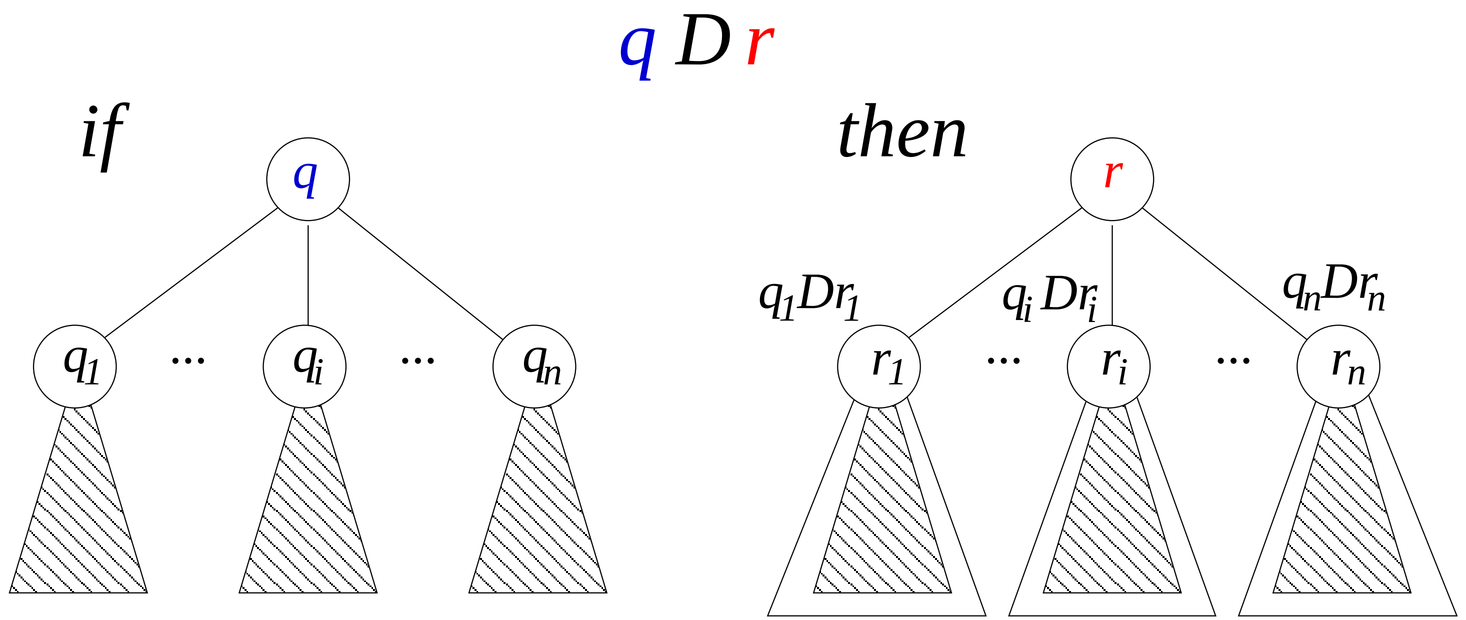
Downward Simulation

- $D \subseteq Q \times Q$ is a downward simulation

  if $q \; D \; r$ implies that

  whenever $(q_1, \ldots, q_n) \xrightarrow{f} q$,

  then also $(r_1, \ldots, r_n) \xrightarrow{f} r$ with $q_i \; D \; r_i$ for all $1 \leq i \leq n$.



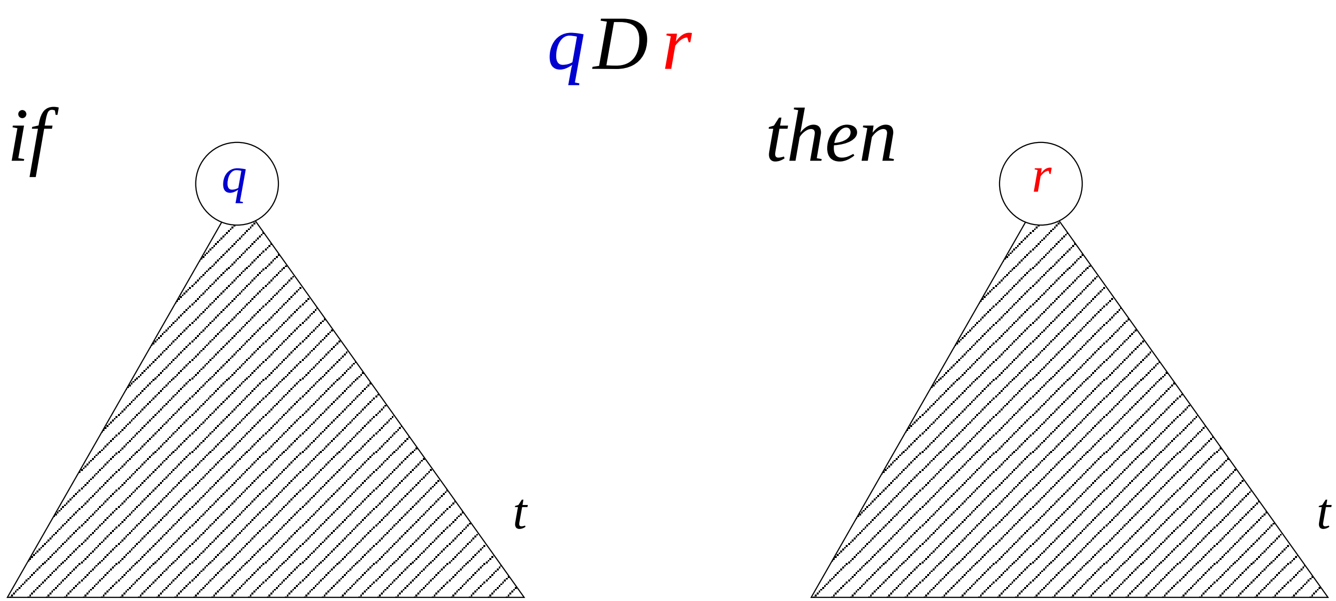
Downward Simulation

- $D \subseteq Q \times Q$ is a downward simulation

  if $q \; D \; r$ implies that

  whenever $(q_1, \ldots, q_n) \xrightarrow{f} q$,

  then also $(r_1, \ldots, r_n) \xrightarrow{f} r$ with $q_i \; D \; r_i$ for all $1 \leq i \leq n$.



Upward Simulation

- Let D be a downward simulation.

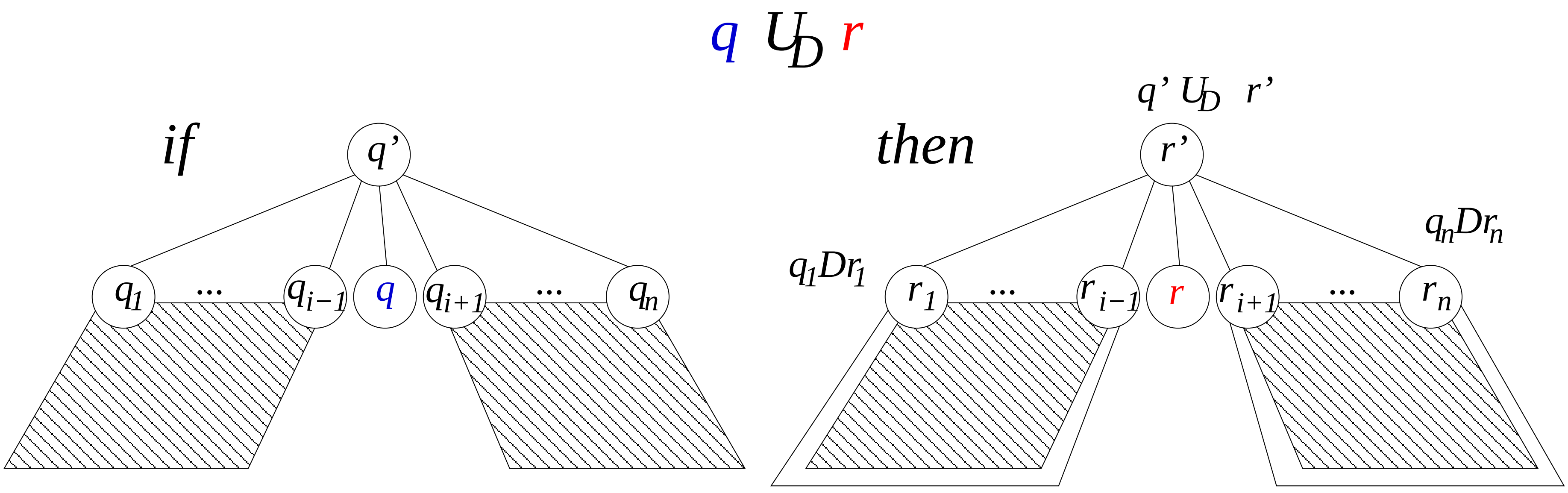
- $U_D \subseteq Q \times Q$ is an upward simulation induced by D

  if $q\ U_D\ r$ implies that

  whenever $(q_1, \ldots, q_n) \xrightarrow{f} q'$ where $q_i = q$,

  then also $(r_1, \ldots, r_n) \xrightarrow{f} r'$ where $r_i = r$, $q'\ U_D\ r'$, and $q_j D r_j$ for all $1 \leq i \neq j \leq n$,

  moreover, $q \in F \implies r \in F$.



Upward Simulation

- Let D be a downward simulation.

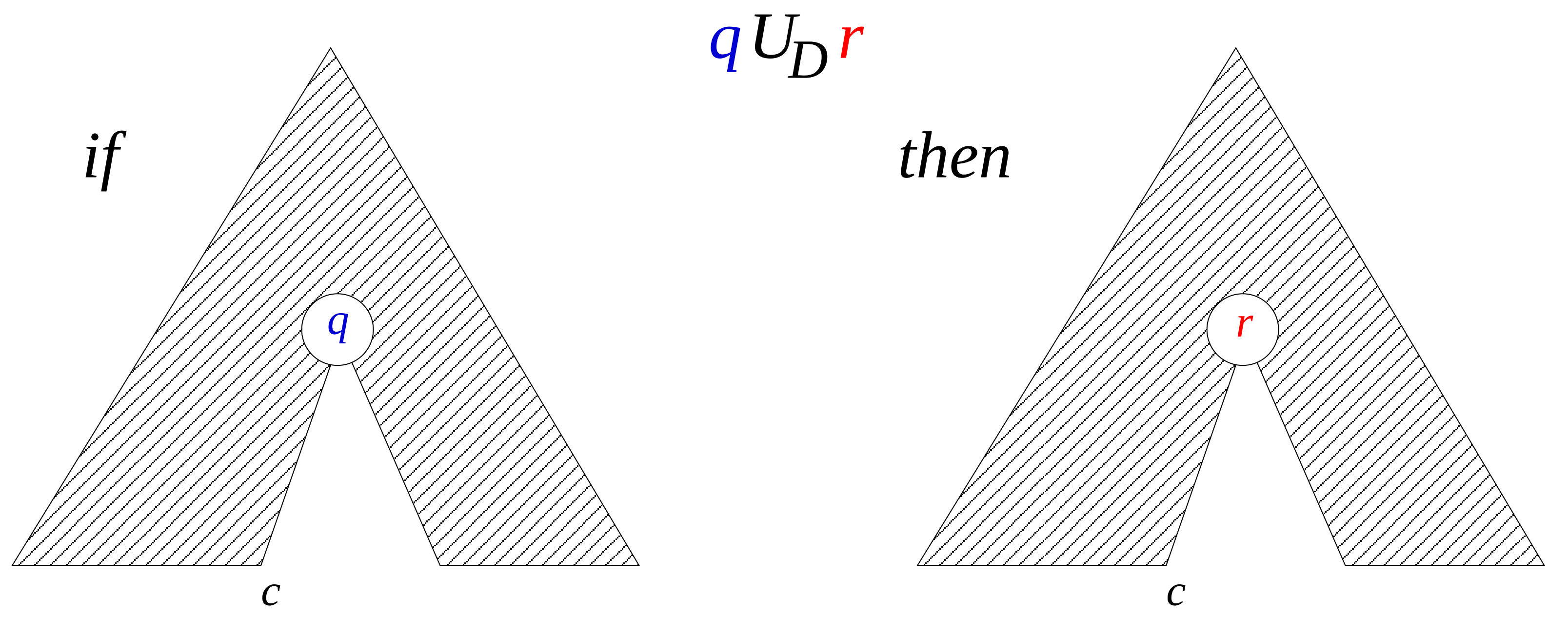
- $U_D \subseteq Q \times Q$ is an upward simulation induced by D

  if $q \; U_D \; r$ implies that

  whenever $(q_1, \ldots, q_n) \xrightarrow{f} q'$ where $q_i = q$,

  then also $(r_1, \ldots, r_n) \xrightarrow{f} r'$ where $r_i = r$, $q' \; U_D \; r'$, and $q_j D r_j$ for all $1 \leq i \neq j \leq n$,

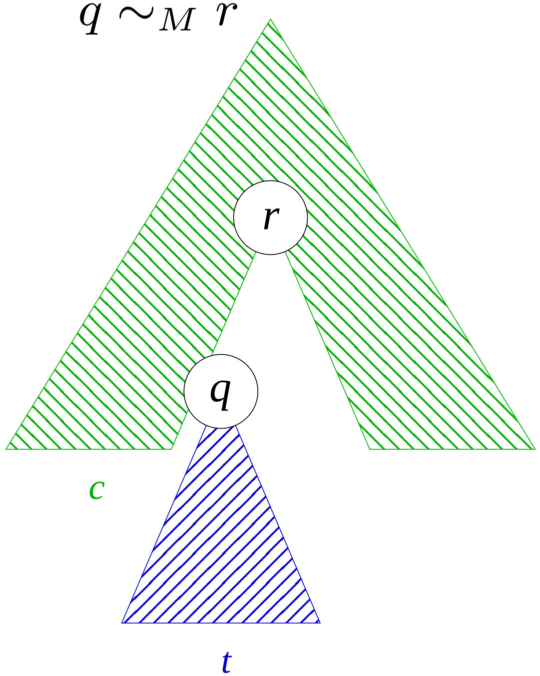
  moreover, $q \in F \implies r \in F$.



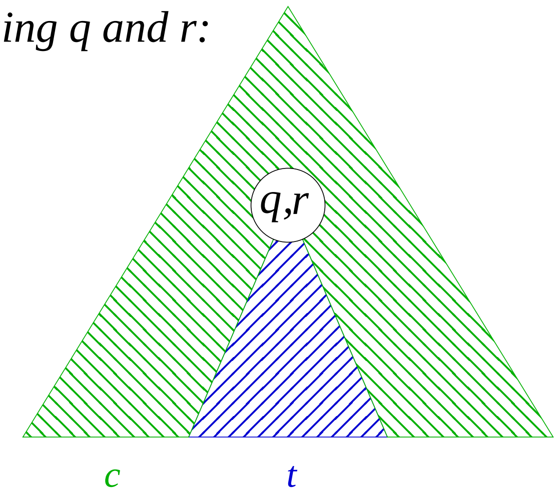
Mediation and Tree Automata

- A mediated preorder $D \oplus U$ is the maximal transitive fragment of $D \circ U_D^{-1}$ containing D.

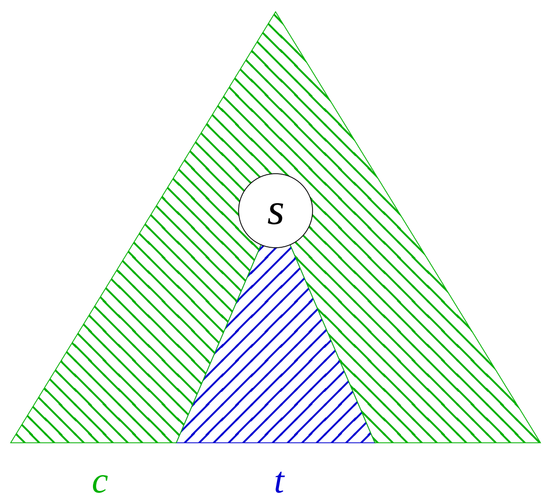
$q \sim_M r$



After collapsing q and r:



$$qMr \implies \exists s : \quad qDs \wedge rUs$$

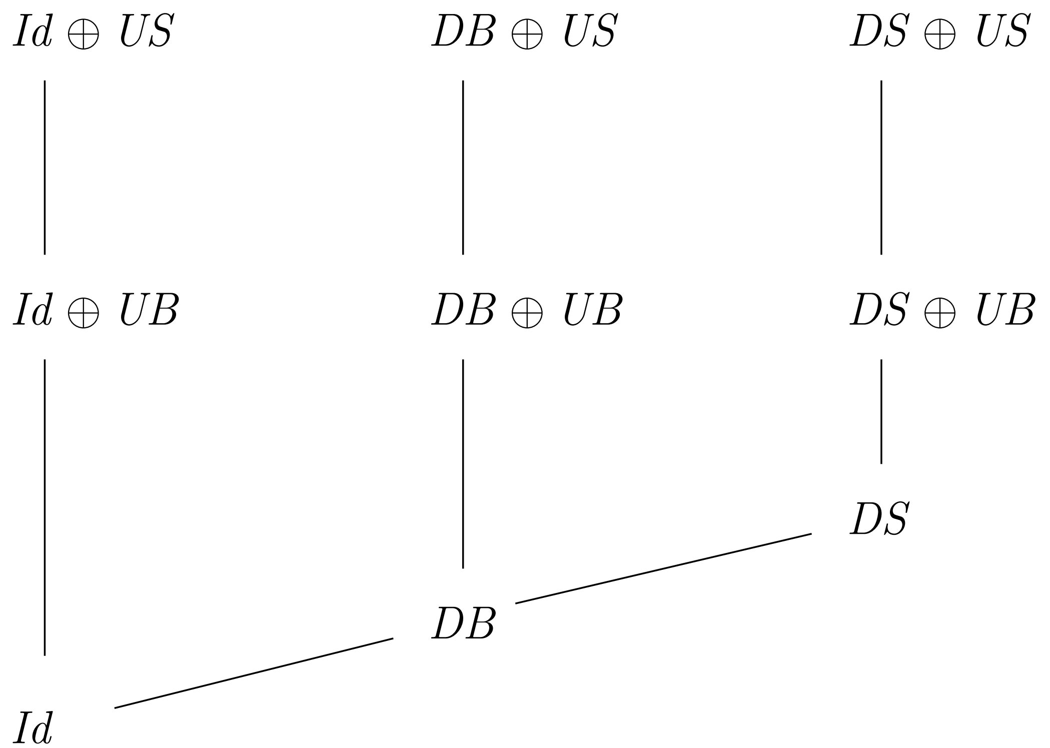


Mediation and Tree Automata

- A mediated preorder $D \oplus U$ is the maximal transitive fragment of $D \circ U_D^{-1}$ containing D.
- In fact, one can combine:
  - an inducing downward relation: simulation (DS), bisimulation (DB), identity (Id).
  - an induced upward relation: simulation (US), bisimulation (UB), identity.

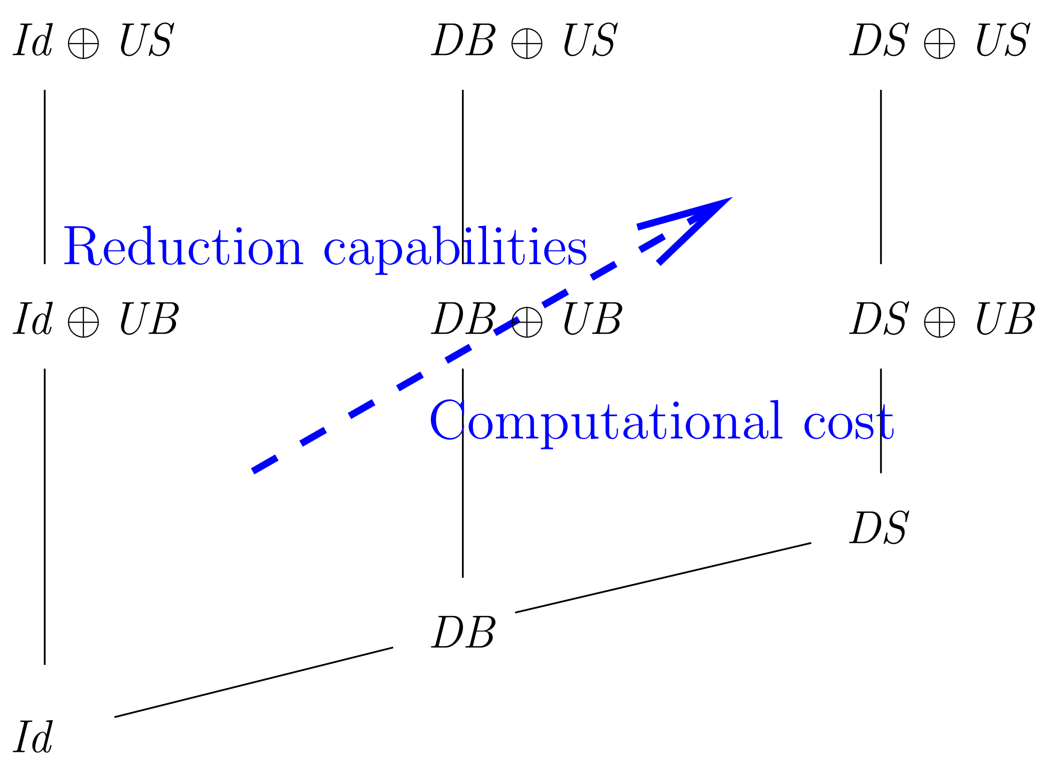
Mediation and Tree Automata

- A mediated preorder $D \oplus U$ is the maximal transitive fragment of $D \circ U_D^{-1}$ containing D.
- In fact, one can combine:
  - an inducing downward relation: simulation (DS), bisimulation (DB), identity (Id).
  - an induced upward relation: simulation (US), bisimulation (UB), identity.



Mediation and Tree Automata

- A mediated preorder $D \oplus U$ is the maximal transitive fragment of $D \circ U_D^{-1}$ containing D.
- In fact, one can combine:
  - an inducing downward relation: simulation (DS), bisimulation (DB), identity (Id).
  - an induced upward relation: simulation (US), bisimulation (UB), identity.



Experiments with Mediated Reduction on TA

TA		DS		$Id \oplus US$		$DB \oplus US$		$DS \oplus US$	
origin	size	reduction	time	reduction	time	reduction	time	reduction	time
RTMC	909	52%	3.6 s	72%	3.1 s	82%	3.4 s	89%	35.1 s
ARTMC	2029	10%	27.0 s	37%	26.0 s	33%	29.0 s	93%	39.0 s
RTMC	2403	26%	31.0 s	0%	25.0 s	0%	34.0 s	82%	37.1 s

TA		DB		$Id \oplus UB$		$DB \oplus UB$		$DS \oplus UB$	
origin	size	reduction	time	reduction	time	reduction	time	reduction	time
RTMC	909	14%	0.6 s	72%	0.4 s	82%	0.8 s	83%	4.1 s
ARTMC	2029	10%	1.7 s	14%	1.4 s	19%	3.1 s	44%	29.0 s
RTMC	2403	0%	0.3 s	0%	0.6 s	0%	0.7 s	27%	31.0 s

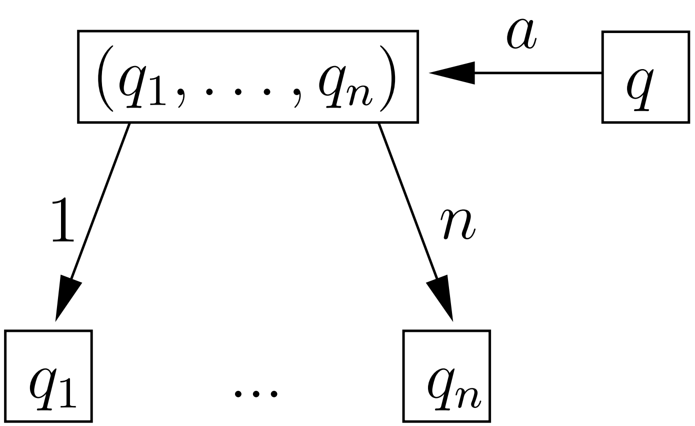
Computing Simulations on Tree Automata and Labelled Transition Systems

Computing Downward Simulations

- Via a translation from NTA to LTS:

TA: $(q_1, \ldots, q_n) \xrightarrow{a} q$

LTS:



- Theorem: $q \ D \ r$ iff $\boxed{q} \preccurlyeq \boxed{r}$.

Computing Upward Simulations

- Via a translation from NTA to LTS:

TA: $$(q_1, \ldots, q_n) \xrightarrow{a} q$$

LTS: $$\forall i \quad \boxed{q_i} \xrightarrow{\lambda} \boxed{(q_1, \ldots, \square_i, \ldots, q_n) \xrightarrow{a} q} \xrightarrow{a} \boxed{q}$$

- Theorem: $q \; U_D \; r$ iff $\boxed{q} \preccurlyeq^I \boxed{r}$.
  - $\preccurlyeq^I$ is the maximal upward simulation included in the relation I defined as follows:
    - $(\boxed{q}, \boxed{r}) \in I$ for all $q, r \in Q$ and
    - $\left(\boxed{(q_1, \ldots, \square_i, \ldots, q_n) \xrightarrow{a} q}, \boxed{(r_1, \ldots, \square_i, \ldots, r_n) \xrightarrow{a} r}\right) \in I$ iff $q_j \; D \; r_j$ for all $1 \leq j \neq i \leq n$.

Complexity

- There exist many algorithms for computing simulations on Kripke structures/LTSs.

- Fix a TA $A = (Q, \Sigma, \Delta, F)$ and let $n = |Q|$, $m = |\Delta|$, $\ell = |\Sigma|$, and r be the rank of Σ.

- We use a modification of the fast algorithm for computing simulations on Kripke structures by Ranzato and Tapparo (2007) for LTS: $\mathcal{O}(|Lab| \cdot |P_{sim}| \cdot |S| + |P_{sim}| \cdot |{\rightarrow}|)$.
  - Maximal downward simulations: $\mathcal{O}((r + \ell) \cdot m^2)$.
  - Maximal downward simulations: $\mathcal{O}(\ell \cdot r^2 \cdot m^2 + T(D))$.

- For bisimulations, one can use an LTS modification of the Paige and Tarjan (1987) partition refinement algorithm that runs in time $\mathcal{O}(|Lab| \cdot |{\rightarrow}| \cdot \log |S|)$.
  - Maximal downward bisimulations: $\mathcal{O}(r^3 \cdot m \cdot \log n)$.
  - Maximal upward bisimulations: $\mathcal{O}(m \cdot \log(n + \ell) + T(D))$.

- Specialised algorithms for downward bisimulation and upward simulation induced by identity by Högberg, Maletti, and May (2007): $\mathcal{O}(r^2 \cdot m \cdot \log n)$ and $\mathcal{O}(r \cdot m \cdot \log n)$.

Computing Simulations on LTS

Input: an LTS $T = (S, \Sigma, \{\delta_a \mid a \in \Sigma\})$, partition-relation pair $\langle P_I, Rel_I \rangle$
Output: partition-relation pair $\langle P, Rel \rangle$

`/* initialization */`
$\langle P, Rel \rangle \leftarrow \langle P_I, Rel_I \rangle$ `/*` $\leftarrow \langle P_{I\cap Out}, Rel_{I\cap Out} \rangle$ `*/`
foreach $B \in P$ **and** $a \in \Sigma$ **do** `/*` $a \in \mathrm{in}(B)$ `*/`
    foreach $v \in S$ **do** `/*` $v \in \delta_a^{-1}(S)$ `*/`
        $Count_a(v, B) = |\delta_a(v) \cap \bigcup Rel(B)|$;
    $Remove_a(B) \leftarrow S \setminus \delta_a^{-1}(\bigcup Rel(B))$ `/*` $\leftarrow \delta_a^{-1}(S) \setminus \delta_a^{-1}(\bigcup Rel(B))$ `*/`

`/* computation */`
while exists $B \in P$ **and** $a \in \Sigma$ **such that** $Remove_a(B) \neq \emptyset$ **do**
    $Remove \leftarrow Remove_a(B)$;
    $Remove_a(B) \leftarrow \emptyset$;
    $\langle P_{\mathsf{prev}}, Rel_{\mathsf{prev}} \rangle \leftarrow \langle P, Rel \rangle$;
    $P \leftarrow Split(P, Remove)$;
    $Rel \leftarrow \{(C, D) \in P \times P \mid (C_{\mathsf{prev}}, D_{\mathsf{prev}}) \in Rel_{\mathsf{prev}}\}$;
    foreach $C \in P$ **and** $b \in \Sigma$ **do** `/*` $b \in \mathrm{in}(C)$ `*/`
        $Remove_b(C) \leftarrow Remove_b(C_{\mathsf{prev}})$;
        foreach $v \in S$ **do** `/*` $v \in \delta_b^{-1}(S)$ `*/`
            $Count_b(v, C) \leftarrow Count_b(v, C_{\mathsf{prev}})$;
    foreach $C \in P$ **such that** $C \cap \delta_a^{-1}(B) \neq \emptyset$ **do**
        foreach $D \in P$ **such that** $D \subseteq Remove$ **do**
            if $(C, D) \in Rel$ **then**
                $Rel \leftarrow Rel \setminus \{(C, D)\}$;
                foreach $b \in \Sigma$ **and** $v \in \delta_b^{-1}(D)$ **do** `/*` $b \in \mathrm{in}(D) \cap \mathrm{in}(C)$ `*/`
                    $Count_b(v, C) \leftarrow Count_b(v, C) - 1$;
                    if $Count_b(v, C) = 0$ **then**
                        $Remove_b(C) \leftarrow Remove_b(C) \cup \{v\}$;

[Holík, Šimáček MOSIS'09]