

# 1 List Of Models

## 1.1 Infinite Words

- DBA
- NBA
- GBA
- Rabin automaton
- Muller automaton
- Parity automaton
- det. E automaton
- det. A automaton
- det. coBA
- weak BA
- Staiger-Wagner automaton
- ABA
- LTL
- S1S
- $\exists$ S1S
- S1S<sub>0</sub>

## 1.2 Finite Trees

- DTA
- NTA
- $\downarrow$ DTA
- $\downarrow$ NTA
- DUTA
- NUTA
- deterministic DTD
- DTD
- deterministic EDTD

- single-type EDTD
- EDTD
- Relax NG
- FO
- MSO
- Regular expressions
- DTWA
- TWA

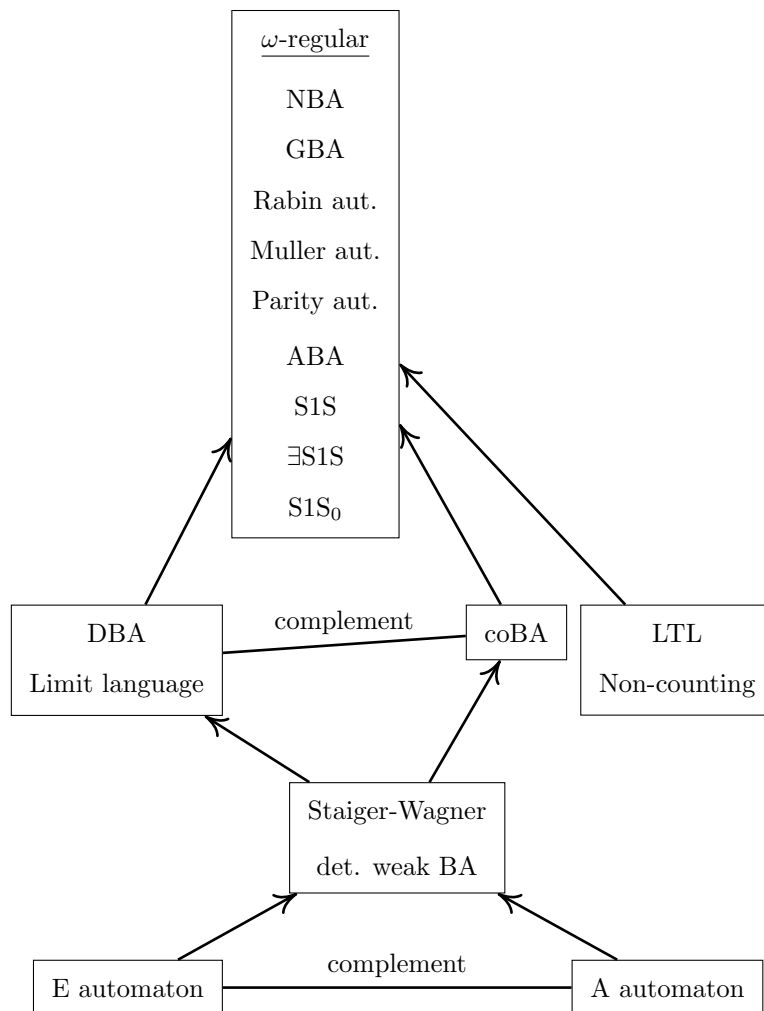
### 1.3 Infinite Trees

- BTA
- Muller TA
- Parity TA
- DMTA
- S2S (MSO / WMSO)
- S2S<sub>0</sub> (MSO / WMSO)

## 2 List Of Games

- Büchi
- Staiger-Wagner
- weak Parity
- Reachability
- Safety
- Muller
- Parity
- Rabin
- Streett
- Gale-Stewart
- Wadge

### 3 Infinite Word Models



#### 3.1 Class Inclusions

- $E \text{ aut.} \subseteq \text{Staiger-Wagner}$   
**Proof:** SWA with  $\mathcal{F} = \{Q' \subseteq Q \mid F \cap Q' \neq \emptyset\}$ .
- $A \text{ aut.} \subseteq \text{Staiger-Wagner}$   
**Proof:** SW closed under complement,
- $\text{Staiger-Wagner} \subseteq \text{DBA} / \text{coBA}$   
**Proof:**  $\mathcal{A} \text{ SWA} \Rightarrow \mathcal{A}' = (Q \times 2^Q, \Sigma, (q_0, \emptyset), \delta', F')$   
 Collect all visited states and accept if that set stays in  $\mathcal{F}$ .

- DBA  $\subseteq$  NBA  
trivial
- coBA  $\subseteq$  NBA  
**Proof:** NBA closed under complement.
- LTL  $\subseteq$  GBA (exponential size transformation)  
**Proof:** Theorem ??
- LTL  $\subseteq$  ABA (linear size transformation)  
**Proof:** Similar to theorem 2.  $\mathcal{A}_\varphi = (\text{cl}(\varphi), \mathbb{B}^n, \varphi, \delta, F)$  with  $F = \{\psi \in \text{cl}(\varphi) \mid \psi = G\vartheta\}$  and  $\delta$  is defined as follows:
  - $\delta(p_i, a)$  and  $\delta(\neg p_i, a)$  are tt or ff depending on the  $i$ -th component of  $a$ .
  - $\delta(\psi_1 \oplus \psi_2, a) = \delta(\psi_1, a) \oplus \delta(\psi_2, a)$  for  $\oplus \in \{\wedge, \vee\}$ .
  - $\delta(X\psi, a) = \psi$ .
  - $\delta(G\psi, a) = \delta(\psi, a) \wedge G\psi$ .
  - $\delta(\psi_1 U \psi_2, a) = (\delta(\psi_1, a) \wedge \psi_1 U \psi_2) \vee \delta(\psi_2, a)$ .

### 3.2 Class Exclusions

- E aut.  $\not\subseteq$  A aut.  
**Example:**  $b^*a(a+b)^\omega$   
**Proof:** Assume the A automaton  $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$  recognizes  $L$ , so for every  $b^nab^\omega$ , the run  $\rho_n$  is accepting. Let  $\rho^*(n) = \rho_{n+1}(n)$ . This is an accepting run on  $b^\omega$ , which is a contradiction.
- A aut.  $\not\subseteq$  E aut.  
**Example:**  $\{a^\omega\}$   
**Proof:** Assume the E automaton  $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$  recognizes  $L$ , so the run  $\rho$  of  $\mathcal{A}$  on  $a^\omega$  is accepting. That means there is an  $n$  s.t.  $\rho(n) \in F$ . Therefore,  $\mathcal{A}$  accepts the word  $a^n b^\omega$ , which is a contradiction.
- DBA  $\not\subseteq$  coBA  
**Example:**  $(a^*b)^\omega$   
**Proof:** Assume  $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$  is a coBA recognizing  $L$ . Let  $n = |Q|$ ,  $w = (a^n b)^\omega$ , and  $\rho$  the run of  $\mathcal{A}$  on  $w$ . From some  $m$  on,  $\rho$  only visits states in  $F$ . Consider  $(a^n b)^m a^n$ . By choice of  $n$ , there must be  $i < j$  such that after  $(a^n b)^m a^i$  and  $(a^n b)^m a^j$ , the automaton is in the same state  $q \in F$ . Therefore, the run on  $(a^n b)^m a^\omega$  is accepting, which is a contradiction.
- coBA  $\not\subseteq$  DBA  
**Example:**  $(a+b)^*a^\omega$   
**Proof:** Assume  $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$  is a DBA recognizing  $L$ . We inductively define words  $w_n$  with runs  $\rho_n$  s.t.  $w_n \sqsubseteq w_{n+1}$  and  $\rho_n$  visits  $F$  at least  $n$  times and  $|w_n|_b = n$ . Then the “limit” of those words contains infinitely many  $b$  but is accepted by  $\mathcal{A}$ , which is a contradiction.  
Let  $w_0 = \varepsilon$  and  $\rho_0 = q_0$ . For  $n+1$ , consider  $w_n a^\omega$  with the run  $\rho_n \pi$ . Since  $w_n a^\omega \in L$ , there is a  $k$  s.t.  $\pi(k) \in F$ . Let  $w_{n+1} = w_n a^k b$  and  $\rho_{n+1}$  accordingly.

- LTL  $\not\subseteq$  NBA

**Example:**  $((a + b)a)^\omega$

**Proof:** Show that  $L$  is counting. Then it follows that it is not LTL-definable. Assume that  $L$  is non-counting, so there is an  $n_0$  according to the definition. Let  $n = n_0 + 1$ ,  $u = \varepsilon$ ,  $v = a$ , and  $\beta = ba^\omega$ . Due to symmetry, we can assume that  $n_0$  is even, so  $uv^n\beta \notin L$  but  $uv^{n+1}\beta \in L$ .

### 3.3 Class Equalities

#### 3.3.1 NBA

- NBA  $\Leftrightarrow \omega$ -regular

**Proof:**  $\Leftarrow$  All three operations used in regular expressions ( $\cup, \cdot, ^\omega$ ) correspond to easy NBA constructions.

$\Rightarrow$  Let  $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$  be an NBA. For every final state  $q \in F$ , we consider the finite words  $U_q$  which lead from  $q_0$  to  $q$ , and the finite words  $V_q$  which lead from  $q$  back to  $q$  itself. Then  $L(\mathcal{A})$  is the union of  $U_q \cdot V_q^\omega$  for all final states  $q$ .

- GBA  $\Rightarrow$  NBA

**Proof:** Let  $\mathcal{A} = (Q, \Sigma, q_0, \Delta, (F_1, \dots, F_k))$  be a GBA. An equivalent NBA is  $\mathcal{A}' = (Q \times \{1, \dots, k\}, \Sigma, (q_0, 1), \Delta', F_k \times \{k\})$ . The transitions  $\Delta'$  contain all those from  $\Delta$  and allow to switch from  $(q, i)$  to  $(q, i + 1)$  if  $q \in F_i$ .

- NBA  $\Leftrightarrow$  S1S

**Proof:** Theorem 4.

- Det. Muller  $\Leftrightarrow$  NBA

**Proof:** Theorem 5.

- (det.) Muller  $\Rightarrow$  (det.) Parity

**Proof:** Theorem 6.

- ABA  $\Rightarrow$  NBA

**Proof:** Theorem 7.

#### 3.3.2 LTL

LTL  $\Leftrightarrow$  Non-counting

No proof. Remarks in F8.

A language  $L$  is called **non-counting** if there is an  $n_0$  such that for all  $n > n_0$ , all  $u, v \in \Sigma^*$ , and all  $\beta \in \Sigma^\omega$ :  $uv^n\beta \in L \Leftrightarrow uv^{n+1}\beta \in L$ .

#### 3.3.3 SW

Staiger-Wagner  $\Leftrightarrow$  det. weak BA

**Proof:**  $\Rightarrow$  Every SWA-recognizable language is also DBA- and coBA-recognizable (see above) and therefore DWBA-recognizable (see below).

$\Leftarrow$  Let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$  be a DWBA. We define  $\preceq \subseteq Q \times Q$  with  $p \preceq q$  iff  $q$  is reachable from  $p$ . Let  $\mathcal{F} = \{P \subseteq Q \mid \max_{\preceq} P \subseteq F\}$ . Then the SWA  $\mathcal{A}' = (Q, \Sigma, q_0, \delta, \mathcal{F})$  recognizes  $L(\mathcal{A})$ .

### 3.4 Closures

#### 3.4.1 NBA

- Closed under union  
**Proof:** Product automaton with  $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$ .
- Closed under intersection  
**Proof:** GBA  $\mathcal{A} = (Q_1 \times Q_2, \Sigma, (q_0^1, q_0^2), \Delta, (F_1 \times Q_2, Q_1 \times F_2))$  where  $\Delta = \{((p_1, p_2), a, (q_1, q_2)) \mid (p_1, a, q_1) \in \Delta_1, (p_2, a, q_2) \in \Delta_2\}$ .
- Closed under complement  
**Proof:** Theorem 1.

#### 3.4.2 DBA

- Closed under union and intersection  
**Proof:** Same construction as for NBA.
- Not closed under complement (inf. many  $a \leftrightarrow$  fin. many  $a$ )

#### 3.4.3 SW

- Closed under union and intersection  
**Proof:** Product automaton with  $\mathcal{F}_\cap = \{F \subseteq Q_1 \times Q_2 \mid \pi_1(F) \in \mathcal{F}_1, \pi_2(F) \in \mathcal{F}_2\}$  where  $\pi_i((x_1, x_2)) = x_i$ .
- Closed under complement  
**Proof:**  $\overline{\mathcal{F}} = 2^Q \setminus \mathcal{F}$ .

### 3.5 Characterizations

- Parity conditions are directly convertible to Rabin chain conditions and vice-versa.  
**Proof:** Assign priorities in ascending order;  $E_k \rightarrow 0, F_k \setminus E_k \rightarrow 1, E_{k-1} \setminus F_k \rightarrow 2 \dots$
- $U$  is  $\omega$ -regular iff  $U$  is a Boolean combination of DBA-languages  
**Proof:**  $\Leftarrow$  NBAs are closed under Boolean operations.  
 $\Rightarrow$  Let  $\mathcal{A}$  be a DRA for  $U$ . It suffices to consider  $\Omega = \{(E, F)\}$  as any other language is a finite union of these conditions. Let  $\mathcal{A}_E$  and  $\mathcal{A}_F$  be the modifications of  $\mathcal{A}$  with conditions  $\{(E, Q)\}$  and  $\{\emptyset, F\}$  respectively. Then  $L(\mathcal{A}_E)^G$  and  $L(\mathcal{A}_F)$  are DBA-recognizable.
- $U$  is DBA-recog. iff  $U = \lim(L)$  for some regular  $L \subseteq \Sigma^*$ .  
**Proof:** Use the DBA as a DFA or vice-versa.
- $U$  is E-recog. iff  $U = L \cdot \Sigma^*$  for some regular  $L \subseteq \Sigma^*$ .  
**Proof:**  $\Rightarrow$  Let  $\mathcal{A}$  be an E-automaton for  $U$ . For every  $q \in F, a \in \Sigma$ , add a transition  $(q, a, q)$ . The resulting automaton as an NFA accepts  $L$  with  $U = L \cdot \Sigma^*$ .  
 $\Leftarrow$  Let  $\mathcal{A}$  be a DFA for  $L$ . The same automaton as an E-automaton recognizes  $L \cdot \Sigma^*$ .

- Landweber's theorem

**Proof:** Theorem 3

- $\text{DBA} \cap \text{coBA} \Rightarrow \text{DBWA}$

**Proof:** Let  $U$  be DBA-recognizable and coBA-recognizable and let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$  be a DMA for  $U$ . Then  $\mathcal{F}$  must be closed under super loops and under sub loops by Landweber's theorem. Therefore,  $\mathcal{A}$  can also be considered as a DBWA with  $F = \bigcup \mathcal{F}$ .

- $\text{SW} \Leftrightarrow$  Boolean combination of E-recognizable languages.

**Proof:**  $\Leftarrow$  SWAs are closed under Boolean operations.

$\Rightarrow$  Let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$  be the SWA and let  $\mathcal{A}_q = (Q, \Sigma, q_0, \delta, \{q\})$ . For every  $q \in Q$ , the language  $L(\mathcal{A}_q)$  is E-recognizable. For every  $F \subseteq Q$ , let  $L_F = \bigcap_{q \in F} L(\mathcal{A}_q) \cap \bigcap_{q \notin F} L(\mathcal{A}_q)^c$ , which is a Boolean combination of E-recognizable languages. Then  $L(\mathcal{A}) = \bigcup \{L_F \mid F \in \mathcal{F}\}$ .

### 3.6 Duality

- $U$  is A-recog. iff  $\Sigma^\omega \setminus U$  is E-recog.

**Proof:** same automaton,  $\bar{F} = Q \setminus F$

- $U$  is coBA-recog. iff  $\Sigma^\omega \setminus U$  is DBA-recog.

**Proof:** same automaton,  $\bar{F} = Q \setminus F$

### 3.7 Problems / Complexity

- Emptiness problem for NBAs is decidable in poly. time.

**Algorithm:**

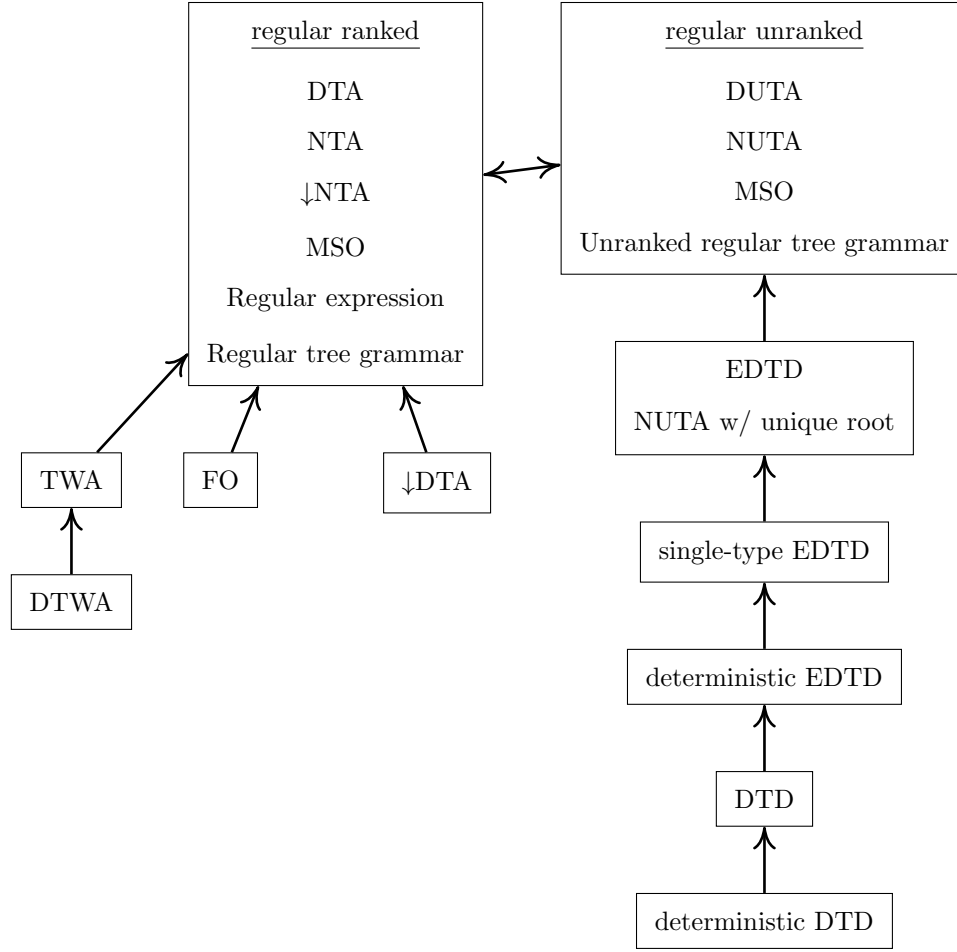
1. Compute all states that are reachable from  $q_0$ . ( $\mathcal{O}(|Q| + |\Delta|)$ ).
2. For each of these reachable states that is also accepting, check if there is a path to itself. ( $\mathcal{O}(|F| \cdot (|Q| + |\Delta|))$ ).
3. If any state like that was found, the language is not empty.

- Emptiness problem for ABAs is PSPACE-complete.

- Membership problem for ABAs is decidable in poly. time.

**Proof:** Model the problem as a Büchi game. In each turn, starting at a node  $v \in Q$ , player 1 chooses an  $a \in \Sigma$ . Then, according to the Boolean operators of  $\delta(v, a)$ , the players choose a next state. If player 0 wins, the choices of player 1 model an element of the ABA language.

## 4 Finite Tree Models



### 4.1 Class Inclusions

- Regular Ranked  $\subseteq$  Regular Unranked

**FCNS:** Let  $\Sigma$  be an unranked alphabet. We define  $\Gamma_0 = \{\#\}$  and  $\Gamma_2 = \Sigma$ . Let  $\bar{t} = t_1 \dots t_n \in T_\Sigma^*$  with  $t_1 = a(t'_1 \dots t'_m)$ . We define

$$\text{fcns}(\bar{t}) = \begin{cases} \# & \text{if } n = 0 \\ a(\text{fcns}(t'_1, \dots, t'_m), \text{fcns}(t_2 \dots t_n)) & \text{else} \end{cases}$$

See Theorem 12.

- det. DTD  $\subseteq$  DTD  $\subseteq$  det. EDTD  $\subseteq$  single-type EDTD  $\subseteq$  EDTD  
trivial



- EDTD  $\subseteq$  Regular tree grammar  
**Proof:**  $N = \Sigma', P_{\text{gram}} = P \cup \{a^{(n)} \rightarrow a \mid a, n\}$
- TWA  $\subseteq$  NTA  
**Proof:** Theorem 8

## 4.2 Class Exclusions

- $\downarrow\text{DTA} \not\subseteq \text{NTA}$   
**Example:**  $T = \{f(a, b), f(b, a)\}$   
**Proof:** Assume the  $\downarrow\text{DTA } \mathcal{A} = (Q, \Sigma, q_0, \Delta)$  accepts  $T$ . Let  $(q_0, f, (q_1, q_2)) \in \Delta$ , so also  $(q_1, a), (q_1, b), (q_2, a), (q_2, b) \in \Delta$ . However, that means the tree  $f(a, a)$  is accepted by the run  $q_0(q_1, q_2)$  which is a contradiction. On the other hand, NTAs can clearly recognize this property.
- DTD  $\not\subseteq$  single-type EDTD  
**Example:**  $T = \{t \in T_{\{a, b\}} \mid \text{there is a path in } t \text{ on which } a \text{ occurs exactly twice}\}$
- NUTA w/ unique root  $\not\subseteq$  NUTA  
**Example:**  $T = \{a, b\}$
- FO  $\not\subseteq$  MSO  
**Example:**  $T =$  positive boolean terms that evaluate to true
- DTWA  $\not\subseteq$  TWA  
**Example:**  $T_{N \setminus D}$   
 $\Sigma_0 = \{a, b\}, \Sigma_2 = \{f\}$   
 $t \in T_{N \setminus D}$  iff  $|t|_a = 3 \wedge \text{lca}(u, v) \sqsubseteq \text{lca}(v, w)$  **Proof:** DTWAs cannot recognize this tree language (no proof). TWAs can:  
  - Check whether there are exactly three  $a$  and move to the right-most one (DFS).
  - While going up, guess a node and go to the left-most ancestor.
  - The tree is in  $T_{N \setminus D}$  iff there are exactly two leafs labeled  $a$  right of that node. (DFS)
- TWA  $\not\subseteq$  NTA  
**Example:** all paths in the skeleton have even length  
**Proof:**  $\Sigma_0 = \{a, c\}, \Sigma_2 = \{b\}$   
Skeleton of a tree  $t$ : replace all subtrees that contain exactly one  $a$ .  
TWAs cannot recognize this tree language (no proof). NTAs can (find "merge" points and count the parity)

## 4.3 Class Equalities

### 4.3.1 Regular Ranked

- $\text{NTA} \Rightarrow \text{DTA}$   
**Proof:** Subset construction.
- $\text{NTA} \Leftrightarrow \downarrow\text{NTA}$   
**Proof:** Reverse the transitions and initial states  $\leftrightarrow$  final states.

- $\downarrow\text{NTA} \Leftrightarrow \text{Regular Tree Grammar}$   
**Proof:** Non-terminals correspond to states and transitions  $(q, a, q_1, \dots, q_n)$  correspond to rules  $A_q \rightarrow a(A_{q_1}, \dots, A_{q_n})$
- $\text{MSO} \Leftrightarrow \text{NTA}$   
**Proof:** same as S2S
- $\text{Reg. exp.} \Leftrightarrow \text{NTA}$   
**Proof:** Theorem 9

#### 4.3.2 Regular Unranked

- $\text{NUTA} \Rightarrow \text{DUTA}$   
**Proof:** Specialized subset construction.  
 $\mathcal{A} = (Q, \Sigma, \Delta, F) \Rightarrow \mathcal{A}' = (2^Q, \Sigma, \Delta', \{P \subseteq Q \mid P \cap F \neq \emptyset\})$   
 $\Delta' = \{(L_{a,P}, a, P) \mid a \in \Sigma, P \subseteq Q\}$  with  $L_{a,P} = \bigcap_{p \in P} K_{a,p} \cap \bigcap_{p \notin P} K_{a,p}^c$   
 $K_{a,p} = \{P_1 \dots P_n \in (2^Q)^* \mid \exists p_1 \in P_1, \dots, p_n \in P_n : p_1 \dots p_n \in L_{a,p}\}.$
- $\text{MSO} \Leftrightarrow \text{NUTA}$

#### 4.3.3 EDTD

- $\text{NUTA with unique root} \Leftrightarrow \text{EDTD (in poly. time)}$   
**Proof:**  $\Rightarrow$  Let  $q_0, \dots, q_n$  be an enumeration of  $Q$ . We then set  $D = \{\{a^{(i)} \mid a \in \Sigma, 0 \leq i \leq n\}, P, a^{q_0}\}$  where  $a$  is the unique root symbol of the automaton. Rules in  $P$  guess arbitrary symbols for fitting states.  
 $\Leftarrow$  Analogously, we use  $Q = \Sigma \times \{1, \dots, n\}$  where  $n$  is the maximal type in  $\Sigma'$  and rules mimic the transitions.

### 4.4 Closures

#### 4.4.1 Regular Ranked

- Regular (ranked) trees are closed under complement.  
**Proof:** Make  $\Delta$  total (add sink state) and set  $F' := Q \setminus F$ .
- Regular (ranked) trees are closed under union and intersection.  
**Proof:** Product construction ( $F_\cap = F_1 \times F_2$ ,  $F_\cup = (Q_1 \times F_2) \cup (F_1 \times Q_2)$ )

#### 4.4.2 Regular Unranked

- Regular unranked trees are closed under complement, union, and intersection.  
**Proof:** via FCNS, because ranked trees are closed under these operations

#### 4.4.3 TWA

- TWAs are closed under union.  
**Proof:** Non-deterministically choose at the start whether to execute  $\mathcal{A}_1$  or  $\mathcal{A}_2$ .
- TWAs and DTWAs are closed under intersection.  
**Proof:** Execute  $\mathcal{A}_1$ . If it accepts, move to the root and execute  $\mathcal{A}_2$ .
- Open Problem: Are TWAs closed under complement?
- DTWAs are closed under complement.  
**Proof:** Theorem 11

### 4.5 Problems / Complexity

#### 4.5.1 Regular Ranked

- Membership problem for NTAs is decidable.
- Reachable states of NTAs can be computed in linear time in  $|\mathcal{A}|$ . **Algorithm:**
  1. Initial reachable states  $R = \{q \in Q \mid \exists a \in \Sigma_0 : (a, q) \in \Delta\}$
  2. For each transition  $\tau = \{\bar{q}, a, p\} \in \Delta$ , count  $\text{in}(\tau) = |\bar{q}|$  and remember  $\forall q \in \bar{q} : \tau \in \text{tr}(q)$ .
  3. Grow  $R$  by processing each reachable state once: Decrement  $\text{in}(\tau)$  by 1; if that value reaches 0, every ingoing state of  $\tau$  is reachable and therefore, the outgoing state is.
- Emptiness of an NTA can be decided in linear time in  $|\mathcal{A}|$ .  
**Algorithm:**  $T(\mathcal{A}) = \emptyset$  iff  $\text{Reachable}(\mathcal{A}) \cap F = \emptyset$
- Given a DTA  $\mathcal{A}$ ,  $\sim_{T(\mathcal{A})}$  can be computed in time  $\text{poly}(|Q^m \times \Sigma \times Q|)$  where  $m$  is the maximal arity in  $\Sigma$ .  
**Algorithm:**
  1. Mark all  $(q, q')$  with  $(q, q') \in (F \times Q \setminus F) \cup (Q \setminus F \times F)$ .
  2. As long as there is still change in the marks, execute step 3:
  3. If  $(p, p')$  is already marked and there are  $q_1, \dots, q_{i-1}, q, q', q_{i+1}, \dots, q_n \in Q$  and  $a \in \Sigma_n$  such that  $p = \delta(q_1, \dots, q, \dots, q_n, a)$  and  $p' = \delta(q_1, \dots, q', \dots, q_n, a)$ , then mark the pair  $(q, q')$ .
  4.  $p \sim_{T(\mathcal{A})} q$  iff  $(p, q)$  is not marked.

#### 4.5.2 Regular Unranked

- Emptiness / membership / inclusion for NUTAs is decidable in polynomial / polynomial / exponential time.  
**Algorithm** (Emptiness): Use FCNS transformation.
- Algorithm** (Membership): Construct  $\mathcal{A}_t$  with  $T(\mathcal{A}_t) = \{t\}$  and check  $\mathcal{A} \cap \mathcal{A}_t \stackrel{?}{=} \emptyset$ .
- Algorithm** (Inclusion): Check  $T(\mathcal{A}_1) \cap T(\mathcal{A}_2)^c \stackrel{?}{=} \emptyset$ . Complementation is exponential.
- Inclusion for complete DUTAs is decidable in polynomial time.  
**Algorithm:** Same algorithm as for general NUTAs but complementation can be done in polynomial time.

### 4.5.3 Grammars

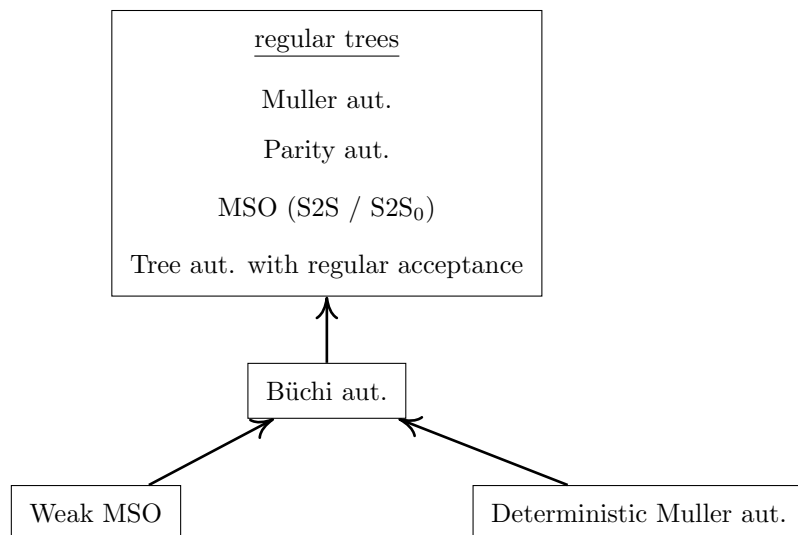
- Emptiness / membership for EDTDs is decidable in polynomial time.

**Proof:** EDTD can be converted to NUTA in polynomial time.

- Inclusion for deterministic EDTDs is decidable in polynomial time.

**Proof:** Let  $D_1, D_2$  be deterministic EDTDs. Using the previous results and theorem 10, we can construct NUTAs  $\mathcal{A}, \mathcal{B}$  with  $T(\mathcal{A}) = T(D_1)$  and  $T(\mathcal{B}) = T(D_2)^{\mathfrak{G}}$  in polynomial time. Then  $T(D_1) \subseteq T(D_2)$  iff  $T(\mathcal{A}) \cap T(\mathcal{B}) = \emptyset$ .

## 5 Infinite Tree Models



### 5.1 Class Exclusions

- $\text{BTA} \not\subseteq \text{Regular tree}$   
**Example:**  $T_{\text{fin}} = \{t \in T_{\{a,b\}} \mid \text{every infinite path has only fin. many } b\}$   
**Proof:**
- $\text{DMTA} \not\subseteq \text{BTA}$   
**Example:**  $T_{\text{fin}}^{\text{c}} = \{t \in T_{\{a,b\}} \mid \text{there is a path with inf. many } b\}$   
**Proof:** BTA-recognizable (see below) but not a path tree language.
- $\text{WMSO} \not\subseteq \text{BTA}$   
**Example:**  $\{t \in T_{\{a,b\}} \mid \text{every infinite path has infinitely many } b\}$   
**Proof:**

### 5.2 Class Equalities

#### 5.2.1 Regular Trees

- $\text{PTA}, \text{MTA} \Rightarrow \text{TA with regular acceptance}$   
**Proof:** By definition and PA/MA regularity.
- $\text{TA with reg. acc.} \Rightarrow \text{PTA}$   
**Proof:** TA  $\mathcal{A}$ , DPA  $\mathcal{A}'$  over alphabet  $Q$  that defines Acc. Define PTA with state space  $Q \times P$ .  
 $(q, a, q_1, q_2) \in \Delta \Rightarrow ((q, p), a, (q_1, \delta(p, q)), (q_2, \delta(p, q))) \in \Delta'$
- $\text{PTA} \Leftrightarrow \text{MSO (S2S)}$  **Proof:** ??

## 5.3 Closures

### 5.3.1 Regular Trees

- The class of regular tree languages is closed under union and intersection.  
**Proof:** ??
- The class of regular tree languages is closed under projection.  
**Proof:** ??
- PTAs are closed under complement ( $2^{\mathcal{O}(kn \cdot \log(kn))}$  states for  $|Q| = n$ ,  $|\text{img}(c)| = k$ ) **Proof:** ??

### 5.3.2 BTA

- BTAs are not closed under complement.  
**Proof:**  $T_{\text{fin}}^{\mathbb{C}} = \{t \in T_{\{a,b\}} \mid \text{there is a path with inf. many } b\}$   
 $T_{\text{fin}}^{\mathbb{C}}$  is BTA-recognizable but its complement is not.

### 5.3.3 DMTA

- DMTAs are not closed under union or complement.
- DMTAs are closed under intersection.  
**Proof:** product automaton

## 5.4 Characterizations

- $T \subseteq T_{\Sigma}$  is DMTA-recognizable iff  $T$  is a path tree language.  
 $T : 2^{(\{0,1\} \times \Sigma)^{\omega}} \rightarrow 2^{T_{\Sigma}}, L \mapsto \{t \in T_{\Sigma} \mid \forall \pi \in \{0,1\}^{\omega} : \pi^{\frown}(t|_{\pi}) \in L\}$   
**Proof:**
- $T \subseteq T_{\Sigma}$  is WMSO-definable iff  $T$  and  $T^{\mathbb{C}}$  are BTA-recognizable.

## 5.5 Problems

### 5.5.1 Regular Trees

- The membership problem for regular tree languages is decidable.  
**Proof:** via membership game as in complementation
- The emptiness problem for regular tree languages is decidable.  
**Proof:** ??