On the inference of finite automata

Damián López and Pedro García

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia (Spain)

September 13th, 2010
Tutorial index

- Notation
Tutorial index

- Notation
- Inference of deterministic automata
  - Trakhtenbrot-Barzdin and Gold
  - RPNI and Lang
  - EDSM and Blue-Fringe
Note: Notation

Inference of deterministic automata
- Trakhtenbrot-Barzdin and Gold
- RPNI and Lang
- EDSM and Blue-Fringe

Inference of non-deterministic automata
- Inference of canonical NFAs (DeLeTe2)
- Inference by juxtaposition of automata (WASRI)
- Order-independent merging inference (OIL)
Tutorial index

- Notation
- Inference of deterministic automata
  - Trakhtenbrot-Barzdin and Gold
  - \textit{RPNI} and Lang
  - \textit{EDSM} and \textit{Blue-Fringe}
- Inference of non-deterministic automata
  - Inference of canonical NFAs (\textit{DeLeTe2})
  - Inference by juxtaposition of automata (\textit{WASRI})
  - Order-independent merging inference (\textit{OIL})
- (Experimental) comparison of different approaches
Notation
Notation

- A finite automaton (NFA) is a 5-tuple $A = (Q, \Sigma, \delta, I, F)$
- Deterministic Finite Automaton (DFA)
Notation

- A finite automaton (NFA) is a 5-tuple \( A = (Q, \Sigma, \delta, I, F) \)
- Deterministic Finite Automaton (DFA)
- A Moore machine is a 6-tuple \( M = (Q, \Sigma, \Delta, \delta, q_0, \Phi) \)
- prefix Moore machine (PTMM\((D_+, D_-))\)
Notation

- A finite automaton (NFA) is a 5-tuple $A = (Q, \Sigma, \delta, I, F)$
- Deterministic Finite Automaton (DFA)
- A Moore machine is a 6-tuple $M = (Q, \Sigma, \Delta, \delta, q_0, \Phi)$
- Prefix Moore machine ($PTMM(D_+, D_-)$)
- $MA(D_+)$ denotes the maximal (non-deterministic) automaton for $D_+$
A finite automaton (NFA) is a 5-tuple $A = (Q, \Sigma, \delta, I, F)$

Deterministic Finite Automaton (DFA)

A Moore machine is a 6-tuple $M = (Q, \Sigma, \Delta, \delta, q_0, \Phi)$

Prefix Moore machine ($PTMM(D_+, D_-)$)

$MA(D_+)$ denotes the maximal (non-deterministic) automaton for $D_+$

Note that, $L(MA(D_+)) = PTMM(D_+, D_-) = D_+$
Inference of deterministic automata

Inference of non-deterministic automata

(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

Trakhtenbrot-Barzdin and Gold

D. López and P. García

On the inference of finite automata
Trakhtenbrot-Barzdin and Gold

- Works by Trakhtenbrot-Barzdin and Gold (1970 decade) can be considered as the first results on the identification of Finite Automata from given data.
Trakhtenbrot-Barzdin and Gold

- Works by Trakhtenbrot-Barzdin and Gold (1970 decade) can be considered as the first results on the identification of Finite Automata from given data.
- Trakhtenbrot-Barzdin proposes an algorithm for automata synthesis using an uniform and complete set of strings of the language.
- In the context of the GI field, Gold proposes an algorithm that converges to the minimal DFA of a target regular language from complete presentation.
Trakhtenbrot-Barzdin and Gold

- Works by Trakhtenbrot-Barzdin and Gold (1970 decade) can be considered as the first results on the identification of Finite Automata from given data.
- Trakhtenbrot-Barzdin proposes an algorithm for automata synthesis using an uniform and complete set of strings of the language.
- In the context of the GI field, Gold proposes an algorithm that converges to the minimal DFA of a target regular language from complete presentation.
- Both algorithms are the same (TBG in the sequel).
Trakhtenbrot-Barzdin and Gold

- Works by Trakhtenbrot-Barzdin and Gold (1970 decade) can be considered as the first results on the identification of Finite Automata from given data.
- Trakhtenbrot-Barzdin proposes an algorithm for automata synthesis using an uniform and complete set of strings of the language.
- In the context of the GI field, Gold proposes an algorithm that converges to the minimal DFA of a target regular language from complete presentation.
- Both algorithms are the same (TBG in the sequel).
- Non-merging algorithm.
- Based on the notion of obviously distinguishable states.
Trakhtenbrot-Barzdin and Gold: \textit{TBG} algorithm

1: Input: Two disjoint finite sets \((D_+, D_-)\)
2: Output: A consistent Moore Machine
3: Method
4: \(M_0 = PTMM(D_+, D_-) = (Q_0, \Sigma, \{0, 1, ?\}, \delta, q_0, \Phi_0)\);
5: \(R = \{\lambda\}\);
6: while \(\exists s' \in R\Sigma - R : \forall s \in R, \text{od}(s, s', M_0) = \text{True}\) do
7: \hspace{1em} choose \(s'\);
8: \hspace{1em} \(R = R \cup \{s'\}\);
9: end while
10: \(Q = R;\)
11: \(q_0 = \lambda;\)
12: for \(s \in R\) do
13: \hspace{1em} \(\Phi(s) = \Phi_0(s);\)
14: \hspace{1em} for \(a \in \Sigma\) do
15: \hspace{2em} if \(sa \in R\) then
16: \hspace{3em} \(\delta(s, a) = sa\)
17: \hspace{2em} else
18: \hspace{3em} \(\delta(s, a) = \text{any} s' \in R \text{ such that } \text{od}(sa, s', M_0) = \text{False}\)
19: \hspace{2em} end if
20: \hspace{em} end for
21: end for
22: \(M = (Q, \Sigma, \{0, 1, ?\}, \delta, q_0, \Phi);\)
23: if \(M\) is consistent with \((D_+, D_-)\) then Return\(M);\)
24: else Return\(M_0);\)
25: end if
26: End Method.
Trakhtenbrot-Barzdin and Gold: Example

$$D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \}$$
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010\} \quad D_- = \{0, 1, 001\} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]

\[ D + = \{ \lambda, 00, 10, 11, 010 \} \quad D - = \{ 0, 1, 001 \} \]

D. López and P. García
On the inference of finite automata
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{\lambda, 00, 10, 11, 010\} \quad D_- = \{0, 1, 001\} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 001 \} \]
Trakhtenbrot-Barzdin and Gold: Convergence

- **TBG** algorithm identifies in the limit the class of regular languages.
- Proof is based on the existence of a characteristic set.
- The size of the characteristic set is polynomially bounded with respect to the size of the minimal DFA.
- Using supersets of this set, the algorithm converges to the minimal DFA of the target regular language.
Trakhtenbrot-Barzdin and Gold: Convergence

- TBG algorithm identifies in the limit the class of regular languages
- Proof is based on the existence of a characteristic set
- The size of the characteristic set is polynomially bounded with respect to the size of the minimal DFA
- Using supersets of this set, the algorithm converges to the minimal DFA of the target regular language
- Consistency is not assured when the training set does not contain the characteristic sample
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: \textit{RPNI} and Lang
Algorithms guided by heuristics: \textit{EDSM} and \textit{Blue-Fringe}

\textit{RPNI} and Lang
Two algorithms were proposed in the early ’90s of the last century to solve the inconsistency of TBG.
Two algorithms were proposed in the early ’90s of the last century to solve the inconsistency of TBG.

Both are (in essence) identical.
RPNI and Lang

- Two algorithms were proposed in the early ’90s of the last century to solve the inconsistency of TBG.
- Both are (in essence) identical.
- RPNI algorithm performs a traversal of the PTMM($D_+, D_-$) in canonic order.
- Equivalent states (TBG and RPNI sense) are those for which there is no sample that contradicts the equivalence.
- Whenever two states are found to be equivalent, they are deterministically merged.
**RPNI : Algorithm**

1. **Input:** Two disjoint finite sets \((D_+, D_-)\)
2. **Output:** A consistent Moore Machine
3. **Method**
4. \(M = PTMM(D_+, D_-)\);
5. //\(\{u_0, u_1, ..., u_r\}\) states of \(M\) in lexicographic order, \(u_0 = \lambda\)//;
6. \(R = \{u_0\}\);
7. \(B = R\Sigma - R\);
8. **while** \(B\) not empty **do**
9. \(q = First(B)\); //in lexicographic order//
10. \(B = B - \{q\}\);
11. \(merged = False\);
12. **for** \(p \in list\) //in lexicographic order// **do**
13. **if** \(detmerge(M, p, q)\) is possible **then**
14. \(merged = True;\)
15. \(M = detmerge(M, p, q);\)
16. BreakFor
17. **end if**
18. **end for**
19. **if** not \(merged\) **then**
20. \(R = Append(R, q)\);
21. **end if**
22. \(B = R\Sigma - R;\)
23. **end while**
24. Return\((M)\);
25. End Method.

D. López and P. García  
On the inference of finite automata
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

**RPNI : Example**

\[
D_+ = \{a, aba, abba, abbb\}
\]

\[
D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\}
\]

![Automaton Diagram](image-url)
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

**RPNI : Example**

\[ D_+ = \{a, aba, abba, abbb\} \]
\[ D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\} \]

![Automaton Diagram]

D. López and P. García
On the inference of finite automata
$D_+ = \{a, aba, abba, abbb\}$

$D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\}$

### RPNI: Example

![Diagram of an automaton](image)
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

**RPNI: Example**

\[ D_+ = \{a, aba, abba, abbb \} \]

\[ D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba \} \]

![RPNI Example Diagram](image-url)
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

RPNI : Example

\[ D_+ = \{a, aba, abba, abbb\a\} \]
\[ D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\} \]

![Automaton Diagram]
RPNI: Example

\[ D_+ = \{a, aba, abba, abbb\} \]
\[ D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\} \]
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

**RPNI: Example**

\[ D_+ = \{a, aba, abba, abbb\} \]
\[ D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\} \]
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

**RPNI: Example**

\[ D_+ = \{a, aba, abba, abbb\} \]
\[ D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\} \]

![Diagram of a deterministic automaton](image-url)
RPNI: Example

\[ D_+ = \{ a, aba, abba, abbbba \} \]

\[ D_- = \{ \lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba \} \]
\( D_+ = \{a, aba, abba, abbba\} \)

\( D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\} \)
**RPNI** : Example

\[
D_+ = \{a, aba, abba, abbb\}
\]
\[
D_- = \{\lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba\}
\]
**RPNI: Example**

\[ D_+ = \{ a, aba, abba, abbbba \} \]

\[ D_- = \{ \lambda, b, aa, ab, ba, bb, aaa, abb, baa, bba \} \]

![Automaton Diagram](image-url)
**RPNI**: Convergence

- Convergence of RPNI algorithm is assured when the TBG characteristic set is contained into the training one.
Convergence of *RPNI* algorithm is assured when the *TBG* characteristic set is contained into the training one.

Consistency is also assured because assumptions on *undefined* states are not changed.
EDSM and Blue-Fringe
In a run, \textit{RPNI} merges each pair of equivalent states it finds.
EDSM and Blue-Fringe

- In a run, *RPNI* merges each pair of equivalent states it finds.
- What happens when the training set is not representing enough?
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

EDSM and Blue-Fringe

- In a run, RPNI merges each pair of equivalent states it finds.
- What happens when the training set is not representing enough?
  Some inconvenient merges can be done.
In a run, \textit{RPNI} merges each pair of equivalent states it finds.

What happens when the training set is not representing enough? Some \textit{inconvenient} merges can be done.

de la Higuera \textit{et al.} propose to modify the lexicographic order used by \textit{RPNI} to consider equivalence evidence.

Approach revisited by Price (\textit{EDSM} algorithm) to win the \textit{Abbadingo} contest.

Lang proposes \textit{EDSM-Blue-Fringe} as a time improvement with respect to the original \textit{EDSM}. 
**EDSM and Blue-Fringe: Blue-Fringe Algorithm**

1: Input: Two disjoint finite sets \((D_+, D_-)\)
2: Output: A consistent Moore Machine
3: Method:
4: \(M = PTMM(D_+, D_-) = (Q, \Sigma, \{0, 1, ?\}, \delta, q_0, \Phi)\);
5: \(red = \{\lambda\}; \quad \text{score} = \emptyset; \quad blue = \{q \in Q : q = \delta(p, a), p \in red \land a \in \Sigma\} - red;\)
6: while blue \(\neq \emptyset\) do
7:     for \(q \in blue\) do
8:         merged = False;
9:         for \(p \in red\) do
10:             if \((p, q)\) have an score in score then merged = True;
11:             else
12:                 if \(p\) and \(q\) are mergible \(\land p \neq q\) then
13:                     score = score \(\cup\) \{\((p, q), 100 \ast \text{FindScore}(M, p, q) + 99 - \text{depth}(p)\)\};
14:                     merged = True;
15:             end if
16:         end for
17:     if not merged then red = red \(\cup\) \{q\}; BreakFor;
18: end if
19: end for
20: if merged then
21:     \((p, q) = \text{MaximumScorePair}(score); M = \text{detmerge}(M, p, q); \text{score} = \emptyset;\)
22: end if
23: blue = \{\(q \in Q | q = \delta(p, a)\) for \(p \in red \land a \in \Sigma\)\} - red;
24: end while
25: Return(M);
26: End Method.
Blue-Fringe: Example
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

Blue-Fringe: Example
Blue-Fringe: Example

\[ \text{score}(1,2) = 99 \]
\[ \text{score}(1,3) = 299 \]
Blue-Fringe: Example

Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

D. López and P. García
On the inference of finite automata
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

Blue-Fringe: Example

D. López and P. García
On the inference of finite automata
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Non-merging algorithms: Trakhtenbrot-Barzdin and Gold
Merging algorithms: RPNI and Lang
Algorithms guided by heuristics: EDSM and Blue-Fringe

**Blue-Fringe: Example**

```
1 \rightarrow a \rightarrow 2 \rightarrow b \rightarrow 5 \rightarrow a \rightarrow 8
```

```
\begin{align*}
score_{(1,4)} &= 199 \\
score_{(2,4)} &= 98 \\
score_{(2,5)} &= 98
\end{align*}
```
**Blue-Fringe: Example**

\[
\begin{align*}
\text{score}_{(1,4)} &= 199 \\
\text{score}_{(2,4)} &= 98 \\
\text{score}_{(2,5)} &= 98 \\
\end{align*}
\]

States 1 and 5 are not mergible
Blue-Fringe: Example

```
D. López and P. García
On the inference of finite automata
```
Blue-Fringe: Example

![Diagram of a finite automaton](image-url)
Blue-Fringe: Example

Blue-Fringe output

RPNI output
polynomial convergence, and the existence of a characteristic polynomial set, are guaranteed for RPNI-type (namely red-blue-type) algorithms
EDSM and Blue-Fringe: Convergence

- Polynomial convergence, and the existence of a characteristic polynomial set, are guaranteed for RPNI-type (namely red-blue-type) algorithms.
- Not generally true when the order of merging states is guided by the available training data.
EDSM and Blue-Fringe: Convergence

- polynomial convergence, and the existence of a characteristic polynomial set, are guaranteed for RPNI-type (namely red-blue-type) algorithms
- not generally true when the order of merging states is guided by the available training data: convergence is assured using, in the worst case, an exponential, w.r.t. the size of the minimum automaton, training set
EDSM and Blue-Fringe: Convergence

- polynomial convergence, and the existence of a characteristic polynomial set, are guaranteed for RPNI-type (namely red-blue-type) algorithms
- not generally true when the order of merging states is guided by the available training data: convergence is assured using, in the worst case, an exponential, w.r.t. the size of the minimum automaton, training set
- data-driven algorithms may behave well (confirmed later by the Abbadingo results)
<table>
<thead>
<tr>
<th>DeLeTe2</th>
<th>On the inference of finite automata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inference of deterministic automata</td>
<td>D. López and P. García</td>
</tr>
<tr>
<td>Inference of non-deterministic automata</td>
<td></td>
</tr>
<tr>
<td>(Experimental) comparison of different approaches</td>
<td></td>
</tr>
<tr>
<td>Inference of canonical NFAs</td>
<td></td>
</tr>
<tr>
<td>Inference by juxtaposition of automata</td>
<td></td>
</tr>
<tr>
<td>Order-independent merging inference</td>
<td></td>
</tr>
</tbody>
</table>
A residual finite state automaton (RFSA) for a regular language $L$ is a FA $A$ that accepts the language $L$ and such that, for any state $q$, $R_q^A$ is a derivative of $L$.

Canonical RFSA is an interesting representation for regular languages.
DeLeTe2

- A residual finite state automaton (RFSA) for a regular language $L$ is a FA $A$ that accepts the language $L$ and such that, for any state $q$, $R_q^A$ is a derivative of $L$
- Canonical RFSA is an interesting representation for regular languages
- Relation $\prec$ is of special interest
A residual finite state automaton (RFSA) for a regular language \( L \) is a FA \( A \) that accepts the language \( L \) and such that, for any state \( q \), \( R_q^A \) is a derivative of \( L \).

 Canonical RFSA is an interesting representation for regular languages.

 Relation \( \prec \) is of special interest: \( u \prec v \) if there is no string \( w \) such that \( uw \in D_+ \) and \( vw \in D_- \).
A residual finite state automaton (RFSA) for a regular language $L$ is a FA $A$ that accepts the language $L$ and such that, for any state $q$, $R_q^A$ is a derivative of $L$.

Canonical RFSA is an interesting representation for regular languages.

Relation $\prec$ is of special interest: $u \prec v$ if there is no string $w$ such that $uw \in D_+$ and $vw \in D_-$.

$od(u_1, u_2, PTMM(D_+, D_-)) = False \iff u \prec v \land v \prec u; (u \simeq v$ in the sequel)
DeLeTe2: Algorithm

1: Input: Two finite sets of data $D_+ \cup D_-$
2: Output: A finite automaton
3: Method:
4: Let $Pref$ be the set of prefixes of $D_+$ in lexicographic order
5: $Q = \emptyset$; $I = \emptyset$; $F = \emptyset$; $\delta = \emptyset$; $u = \lambda$
6: while True do
7:   if $\exists v \in Q : u \simeq v$ then
8:     Delete $u\Sigma^*$ from $Pref$
9:   else
10:      $Q = Q \cup \{u\}$
11:      if $u \prec \lambda$ then
12:         $I = I \cup \{u\}$
13:      end if
14:      if $u \in D_+$ then
15:         $F = F \cup \{u\}$
16:      end if
17:      $\delta = \delta \cup \{(v, x, u) | v \in Q, vx \in Pref, u \prec vx\} \cup \{(u, x, v) | v \in Q, ux \in Pref, v \prec ux\}$
18:   end if
19: if $u$ is the last string in $Pref$ or $A = (Q, \Sigma, I, F)$ is consistent with respect to $D_+, D_-$ then
20:   Return $A = (Q, \Sigma, \delta, I, F)$
21: else
22:   $u = $ next string in $Pref$
23: end if
24: end while
25: End Method:
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

DeLeTe2: Example

\[
\text{mDFA: } \lambda + 00^+ + 0^*1(0 + 1)^+ 
\]
**DeLeTe2: Example**

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 01, 001 \} \]

\[
mDFA: \lambda + 00^+ + 0^*1(0 + 1)^+ \]
**DeLeTe2: Example**

\[
D_+ = \{ \lambda, 00, 10, 11, 010 \} \quad D_- = \{ 0, 1, 01, 001 \}
\]

\[
\text{mDFA: } \lambda + 00^+ + 0^*1(0 + 1)^+
\]

![Diagram of mDFA and RPNI output](image)
\textbf{DeLeTe2: Example}

\begin{align*}
D_+ &= \{\lambda, 00, 10, 11, 010\} \\
D_- &= \{0, 1, 01, 001\}
\end{align*}

\textbf{mDFA:} \quad \lambda + 00^+ + 0^*1(0 + 1)^+

\textbf{RPNI output}

\textbf{DeLeTe2 second hypothesis}
DeLeTe2: Example

\[ D_+ = \{ \lambda, 00, 10, 11, 010 \} \]

\[ D_- = \{ 0, 1, 01, 001 \} \]

mDFA: \( \lambda + 00^+ + 0^*1(0+1)^+ \)

RPNI output

DeLeTe2 output
DeLeTe2

- DeLeTe2 algorithm is closely related with TBG method
- The authors claim that DeLeTe2 converges to the saturated automata of the minimal DFA
- DeLeTe2 algorithm can output smaller automata than the minimal DFA
- The convergence of DeLeTe2 it can be proved using a similar argument to the one used to prove the RPNI convergence
DeLeTe2

- DeLeTe2 algorithm is closely related with TBG method
- The authors claim that DeLeTe2 converges to the saturated automata of the minimal DFA
- DeLeTe2 algorithm can output smaller automata than the minimal DFA
- The convergence of DeLeTe2 it can be proved using a similar argument to the one used to prove the RPNI convergence
- DeLeTe2 outputs, in some cases, non-consistent automata
DeLeTe2

- *DeLeTe2* algorithm is closely related with *TBG* method
- The authors claim that *DeLeTe2* converges to the saturated automata of the minimal DFA
- *DeLeTe2* algorithm can output smaller automata than the minimal DFA
- The convergence of *DeLeTe2* it can be proved using a similar argument to the one used to prove the *RPNI* convergence
- *DeLeTe2* outputs, in some cases, non-consistent automata: the authors propose an improved algorithm to solve this drawback
DeLeTe2 procedure: An improved algorithm

1: Input: $M = (Q, \Sigma, \{0, 1, \?\}, \delta, q_0, \Phi) = PTMM(D_+, D_-)$
2: Output: A finite automaton consistent with the input sets
3: Method:
4: Initialize inclusion relationships
5: $S = \emptyset; \; M' = (\emptyset, \Sigma, \{0, 1, \?\}, \emptyset, \emptyset)$ // Empty Moore machine //
6: for all $q_1 \in Q$ // breadth-first search do
7: \hspace{1em} if $q_1$ is active then
8: \hspace{2em} for all $q_2 \in states$ such that $q_2 \ll q_1$ //breadth-first traversal// do
9: \hspace{3em} if $q_2$ is active then
10: \hspace{4em} if $q_1 \neq q_2$ then
11: \hspace{5em} if find\_relationship($R, q_1, q_2$) = Unknown then find\_inclusion($M, R, q_1, q_2$)
12: \hspace{5em} end if
13: \hspace{5em} if find\_relationship($R, q_2, q_1$) = Unknown then find\_inclusion($M, R, q_2, q_1$)
14: \hspace{4em} end if
15: \hspace{3em} end if
16: \hspace{2em} end for
17: \hspace{1em} if $q_1$ is active then $S = S \cup \{q_1\}$
18: \hspace{1em} end if
19: $M' = $Automaton induced by $S$ in $M$
20: if $M'$ is consistent with respect to $D_+$ then Return $M'$
21: end if
22: end if
23: end for
24: Return $M'$
25: End Method
DeLeTe2 procedure

- Merging-states algorithm
- Massive use of transitivity property on language inclusion relation
- Consistent with respect to the input data
- Experiments show that DeLeTe2 procedure output hypothesis whose size converge to the size of the mDFA
DeLeTe2 procedure: Example

\[ D_+ = \{ \lambda, 00, 010, 10, 11 \} \]
\[ D_- = \{ 0, 001, 1 \} \]
DeLeTe2 procedure: Example

\[ D_+ = \{ \lambda, 00, 010, 10, 11 \} \quad D_- = \{ 0, 001, 1 \} \]
DeLeTe2 procedure: Example

\[ D_+ = \{ \lambda, 00, 010, 10, 11 \} \quad D_- = \{ 0, 001, 1 \} \]

\[ \begin{array}{c}
\text{DeLeTe2 proc. output} \\
\end{array} \]

\[ \begin{array}{c}
\text{RPNI output} \\
\end{array} \]
<table>
<thead>
<tr>
<th>WASRI</th>
<th>Inference of deterministic automata</th>
<th>Inference of non-deterministic automata</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Experimental) comparison of different approaches</td>
<td>Inference by juxtaposition of automata</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Order-independent merging inference</td>
</tr>
</tbody>
</table>

D. López and P. García

On the inference of finite automata
Vazquez de Parga *et al.* propose a general scheme that considers one positive string in each iteration

- Quite flexible general scheme
- Time complexity is function of the number of samples and the length of the longest string in $D_+$
WASRI: A general algorithm

1: **Input:** Two finite sets \( D_+ = \{x_1, x_2, \ldots, x_n\} \) and \( D_- \)
2: **Output:** A finite automaton consistent with the input sets
3: **Method:**
4: \( A = (Q, \Sigma, \delta, I, F) \) where \( Q = \delta = I = F = \emptyset \);
5: for \( i = 1 \) to \( n \) do
6: if \( x_i \notin L(A) \) then
7: for at least \( j = 1 \) do
8: Infer an automaton \( A^j_i \) consistent with \( D_+ = \{x_i\} \) and \( D_- \);
9: if \( A^j_i \) is good enough then
10: \( A = A \uplus A^j_i \);
11: //where \( \uplus \) stands for disjoint union of the automata//
12: end if
13: end for
14: end if
15: end for
16: Return \( A \);
17: End Method.
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches
Inference of canonical NFAs
Inference by juxtaposition of automata
Order-independent merging inference

WASRI: Example. Instance details
WASRI : Example. Instance details

- Inference of one automaton for each string in $D_+$
- Canonic traverse of the states
- Non-deterministic merging
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches
Inference of canonical NFAs
Inference by juxtaposition of automata
Order-independent merging inference

**WASRI : Example**

\[ D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{01000010\} \]
WASRI: Example

\[
D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{01000010\}
\]

\[
1 \xrightarrow{0} 2 \xrightarrow{1} 3 \xrightarrow{0} 4 \xrightarrow{1} 5 \xrightarrow{0} 6 \xrightarrow{1} 7 \xrightarrow{0} 8
\]
\( D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{01000010\} \)
\[ D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{01000010\} \]

![Diagram of automata with states 1, 5, 6, and 8, transitions labeled by 0 and 1, and an initial and final state marked with circles.]
\[ D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{01000010\} \]
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

**WASRI**: Example

\[ D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{0100010\} \]
\( D_+ = \{0, 000011, 001, 0101010\} \) \hspace{1cm} \( D_- = \{01000010\} \)
$D_+ = \{0, 000011, 001, 0101010\}$ \quad $D_- = \{01000010\}$

D. López and P. García

On the inference of finite automata
\( D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{01000010\\)

\[ \begin{align*}
1 & \xrightarrow{0} 0
\end{align*} \]
WASRI: Example

\[ D_+ = \{0, 000011, 001, 0101010\} \quad D_- = \{0100010\} \]
Taking into account the universal automaton for the language $U_L$, it is possible to compute an *universal sample* $D_+$.

From $MA(D_+)$, any irreducible automaton in $\overline{L}$ accepts the target language.
Taking into account the universal automaton for the language $U_L$, it is possible to compute an *universal sample* $D_+$.

From $MA(D_+)$, any irreducible automaton in $\overline{L}$ accepts the target language.

There is a (finite) set of negative samples $D_-$ which allows to implement the algorithm.

Taking into account $D_-$, for any string in $L$, it is possible to obtain an irreducible automaton which accepts a sublanguage of the target language.
Algorithm based on a result by García et al. who propose a theoretical algorithm based on the notion of automata irreducibility.
Algorithm based on a result by García et al. who propose a theoretical algorithm based on the notion of automata irreducibility

OIL algorithm does not consider any fixed order over the states of the hypothesis and outputs an NFA consistent with the training data
Algorithm based on a result by García et al. who propose a theoretical algorithm based on the notion of automata irreducibility.

OIL algorithm does not consider any fixed order over the states of the hypothesis and outputs an NFA consistent with the training data.

Non-deterministic algorithm
OIL

- Algorithm based on a result by García et al. who propose a theoretical algorithm based on the notion of automata irreducibility
- OIL algorithm does not consider any fixed order over the states of the hypothesis and outputs an NFA consistent with the training data non-deterministic algorithm
- Convergence can be proved using a similar argument to the one used by WASRI algorithm
**OIL : Algorithm**

1: **Input:** A sequence of finite sets \(\langle (D^1_+, D^-_1), (D^2_+, D^-_2), \ldots, (D^n_+, D^-_n)\rangle\)

2: **Output:** A finite automaton consistent with the input sets

3: **Method:**

4: \(A = MA(D^1_+); D_- = D^1_-;\)

5: Find a partition \(\pi\) of the states of \(A\) irreducible in \(D_-;\)

6: \(A = A/\pi;\)

7: for \(i = 2\) to \(n\) do

8: \(D_- = D_- \cup D^i_-;\)

9: if \(A\) is consistent with \((D^i_+, D^i_-)\) then

10: Continue;

11: end if

12: if \(A\) is consistent with \(D^i_-\) then

13: \(S_+ = D^i_+ - L(A);\)

14: \(A' = (Q', \Sigma, \delta', I', F') = MA(S_+);\)

15: \(A = (Q \cup Q', \Sigma, \delta \cup \delta', I \cup I', F \cup F');\)

16: //where \(\cup\) stands for disjoint union //

17: Find a partition \(\pi\) of the states of \(A\) irreducible in \(D_-;\)

18: \(A = A/\pi;\)

19: else

20: \(A = OIL(D^1_+, D_-), (D^2_+, D_-), \ldots, (D^i_+, D_-));\)

21: end if

22: end for

23: Return \(A;\)

24: End Method.
**OIL : Example**

\[
\begin{align*}
D_1^+ &= \{a, bb, aa\} & D_1^- &= \{ab, bba\} \\
D_2^+ &= \{b, aaa\} & D_2^- &= \{aaab, aab\} \\
D_3^+ &= \{\lambda, bbb, aaaa\} & D_3^- &= \{abb, ba\}.
\end{align*}
\]
\[ D_+^1 = \{a, bb, aa\} \quad D_- = D_-^1 = \{ab, bba\} \]

![Diagram](image)
OIL : Example

\[ D_+^1 = \{ a, bb, aa \} \quad D_- = \{ ab, bba \} \]
OIL: Example

\[ D_+^1 = \{a, bb, aa\} \quad D_- = \{ab, bba\} \]

\[ D_+^2 = \{b, aaa\} \quad D_-^2 = \{aaab, aab\} \]
OIL: Example

\[ D_+^1 = \{a, bb, aa\} \quad D_- = \{ab, bba\} \]

\[
\begin{array}{c}
D_+^2 = \{b, aaa\} \\
D_-^2 = \{aaab, aab\}
\end{array}
\]

\[ S_+ = D_+^2 - L(A) = \{b\} \]
OIL : Example

\[ D_+^2 = \{aaa, b\} \quad D_+ = D_+ \cup D_+^2 = \{ab, bba, aaab, aab\} \]
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches
Inference of canonical NFAs
Inference by juxtaposition of automata
Order-independent merging inference

OIL : Example

$D_+ = \{ a, bb, aa, aaa, b \}$ $D_- = \{ ab, bba, aaab, aab \}$

D. López and P. García
On the inference of finite automata
$D_+ = \{a, bb, aa, aaa, b\} \quad D_- = \{ab, bba, aaab, aab\}$

$$D^3_+ = \{\lambda, bbb, aaaa\} \quad D^3_- = \{abb, ba\}$$
OIL: Example

\[ D^1_+ = \{a, bb, aa\} \quad D_- = \{ab, bba, aaab, aab, abb, ba\} \]
OIL : Example

\[ D_1^+ = \{a, bb, aa\} \quad D_- = \{ab, bba, aaab, aab, abb, ba\} \]

\[ D_2^+ = \{aaa, b\} \quad D_- = \{ab, bba, aaab, aab, abb, ba\} \]
Inference of deterministic automata
Inference of non-deterministic automata
(Experimental) comparison of different approaches

Order-independent merging inference

OIL : Example

\[ D_1^+ = \{ a, bb, aa \} \]
\[ D^- = \{ ab, bba, aaab, aab, abb, ba \} \]

\[ D_2^+ = \{ aaa, b \} \]
\[ D^- = \{ ab, bba, aaab, aab, abb, ba \} \]

\[ D_3^+ = \{ \lambda, bbb, aaaa \} \]
\[ D^- = \{ ab, bba, aaab, aab, abb, ba \} \]
OIL: Example

\[ D^1_+ = \{a, bb, aa\} \quad D_- = \{ab, bba, aaab, aab, abb, ba\} \]

\[ D^2_+ = \{aaa, b\} \quad D_- = \{ab, bba, aaab, aab, abb, ba\} \]

\[ D^3_+ = \{\lambda, bbb, aaaa\} \quad D_- = \{ab, bba, aaab, aab, abb, ba\} \]
Experimental behavior
Denis et al. built in a corpus of data to empirically test their results.
Experimental behavior: Database

- Denis et al. built in a corpus of data to empirically test their results
- Coste and Fredouille extend this dataset to consider languages represented by DFAs and Unambiguous Finite Automata (UFAs)
Denis et al. built in a corpus of data to empirically test their results.

Coste and Fredouille extend this dataset to consider languages represented by DFAs and Unambiguous Finite Automata (UFAs).

Alvarez processes both datasets to obtain an incremental corpus which considers all the distinct languages in the original corpora.
Denis et al. built in a corpus of data to empirically test their results.

Coste and Fredouille extend this dataset to consider languages represented by DFAs and Unambiguous Finite Automata (UFAs).

Alvarez processes both datasets to obtain an incremental corpus which considers all the distinct languages in the original corpora.

A naive baseline algorithm (majority algorithm) obtains (approx.) 66.5%, 67% and 72% for the regular expressions, NFAs and DFAs subcorpora respectively.
Experimental behavior: *Blue-Fringe, RPNI and DeLeTe2*

<table>
<thead>
<tr>
<th></th>
<th>Blue-Fringe</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rec. rate</td>
<td>av. (</td>
<td>A</td>
<td>)</td>
</tr>
<tr>
<td>er_100</td>
<td>87.94%</td>
<td>10</td>
<td>83.35%</td>
<td>12.39</td>
</tr>
<tr>
<td>er_200</td>
<td>94.81%</td>
<td>9.97</td>
<td>93.91%</td>
<td>11.52</td>
</tr>
<tr>
<td>er_300</td>
<td>96.46%</td>
<td>11.05</td>
<td>96.32%</td>
<td>11.17</td>
</tr>
<tr>
<td>er_400</td>
<td>97.74%</td>
<td>10.43</td>
<td>97.45%</td>
<td>10.98</td>
</tr>
<tr>
<td>er_500</td>
<td>98.54%</td>
<td>10.47</td>
<td>98.11%</td>
<td>10.95</td>
</tr>
<tr>
<td>nfa_100</td>
<td>68.15%</td>
<td>18.83</td>
<td>66.50%</td>
<td>20.31</td>
</tr>
<tr>
<td>nfa_200</td>
<td>72.08%</td>
<td>28.80</td>
<td>69.27%</td>
<td>32.35</td>
</tr>
<tr>
<td>nfa_300</td>
<td>74.55%</td>
<td>36.45</td>
<td>72.90%</td>
<td>40.86</td>
</tr>
<tr>
<td>nfa_400</td>
<td>77.53%</td>
<td>42.58</td>
<td>74.59%</td>
<td>49.75</td>
</tr>
<tr>
<td>nfa_500</td>
<td>80.88%</td>
<td>47.54</td>
<td>76.75%</td>
<td>55.91</td>
</tr>
<tr>
<td>dfa_100</td>
<td>69.12%</td>
<td>18.59</td>
<td>66.29%</td>
<td>20.27</td>
</tr>
<tr>
<td>dfa_200</td>
<td>77.18%</td>
<td>25.83</td>
<td>71.01%</td>
<td>31.11</td>
</tr>
<tr>
<td>dfa_300</td>
<td>88.53%</td>
<td>25.10</td>
<td>80.61%</td>
<td>33.33</td>
</tr>
<tr>
<td>dfa_400</td>
<td>94.42%</td>
<td>21.36</td>
<td>87.39%</td>
<td>31.90</td>
</tr>
<tr>
<td>dfa_500</td>
<td>97.88%</td>
<td>18.75</td>
<td>91.67%</td>
<td>29.61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RPNI</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rec. rate</td>
<td>av. (</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>83.35%</td>
<td>12.39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>93.91%</td>
<td>11.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>96.32%</td>
<td>11.17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>97.45%</td>
<td>10.98</td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.11%</td>
<td>10.95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66.50%</td>
<td>20.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>69.27%</td>
<td>32.35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>72.90%</td>
<td>40.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>74.59%</td>
<td>49.75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>76.75%</td>
<td>55.91</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66.29%</td>
<td>20.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>71.01%</td>
<td>31.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80.61%</td>
<td>33.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>87.39%</td>
<td>31.90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>91.67%</td>
<td>29.61</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DeLeTe2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rec. rate</td>
<td>av. (</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>91.65%</td>
<td>30.23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>96.96%</td>
<td>24.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>97.80%</td>
<td>31.41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.49%</td>
<td>27.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.75%</td>
<td>29.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td>73.95%</td>
<td>98.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>77.79%</td>
<td>220.93</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80.86%</td>
<td>322.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>82.66%</td>
<td>421.30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>84.29%</td>
<td>512.55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>62.94%</td>
<td>156.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64.88%</td>
<td>432.88</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66.37%</td>
<td>706.64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>69.07%</td>
<td>903.32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>72.41%</td>
<td>1027.42</td>
<td></td>
</tr>
</tbody>
</table>
### Experimental behavior: WASRI, Blue-Fringe and DeLeTe2

<table>
<thead>
<tr>
<th></th>
<th>Blue-Fringe</th>
<th>WASRI</th>
<th>DeLeTe2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rec. rate</td>
<td>av.</td>
<td>Rec. rate</td>
</tr>
<tr>
<td>er_100</td>
<td>87.94%</td>
<td>25.3</td>
<td>91.65%</td>
</tr>
<tr>
<td>er_200</td>
<td>94.81%</td>
<td>33.4</td>
<td>96.96%</td>
</tr>
<tr>
<td>er_300</td>
<td>96.46%</td>
<td>47.0</td>
<td>97.80%</td>
</tr>
<tr>
<td>er_400</td>
<td>97.74%</td>
<td>51.3</td>
<td>98.49%</td>
</tr>
<tr>
<td>er_500</td>
<td>98.54%</td>
<td>48.6</td>
<td>98.75%</td>
</tr>
<tr>
<td>nfa_100</td>
<td>68.15%</td>
<td>82.9</td>
<td>73.95%</td>
</tr>
<tr>
<td>nfa_200</td>
<td>72.08%</td>
<td>170.4</td>
<td>77.79%</td>
</tr>
<tr>
<td>nfa_300</td>
<td>74.55%</td>
<td>267.2</td>
<td>80.86%</td>
</tr>
<tr>
<td>nfa_400</td>
<td>77.53%</td>
<td>350.1</td>
<td>82.66%</td>
</tr>
<tr>
<td>nfa_500</td>
<td>80.88%</td>
<td>431.2</td>
<td>84.29%</td>
</tr>
</tbody>
</table>
### Experimental behavior: OIL, Blue-Fringe and DeLeTe2

<table>
<thead>
<tr>
<th></th>
<th>Blue-Fringe</th>
<th>OIL</th>
<th>DeLeTe2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>smallest</td>
<td>av.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rec. rate</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>er_100</td>
<td>87.94%</td>
<td>93.79%</td>
<td>8.27</td>
</tr>
<tr>
<td>er_200</td>
<td>94.81%</td>
<td>97.83%</td>
<td>7.80</td>
</tr>
<tr>
<td>er_300</td>
<td>96.46%</td>
<td>98.77%</td>
<td>7.68</td>
</tr>
<tr>
<td>er_400</td>
<td>97.74%</td>
<td>99.20%</td>
<td>7.55</td>
</tr>
<tr>
<td>er_500</td>
<td>98.54%</td>
<td>99.66%</td>
<td>6.82</td>
</tr>
<tr>
<td>nfa_100</td>
<td>68.15%</td>
<td>75.00%</td>
<td>21.46</td>
</tr>
<tr>
<td>nfa_200</td>
<td>72.08%</td>
<td>78.05%</td>
<td>35.23</td>
</tr>
<tr>
<td>nfa_300</td>
<td>74.55%</td>
<td>81.27%</td>
<td>45.81</td>
</tr>
<tr>
<td>nfa_400</td>
<td>77.53%</td>
<td>83.87%</td>
<td>52.40</td>
</tr>
<tr>
<td>nfa_500</td>
<td>80.88%</td>
<td>85.64%</td>
<td>58.81</td>
</tr>
<tr>
<td>dfa_100</td>
<td>69.12%</td>
<td>60.17%</td>
<td>28.01</td>
</tr>
<tr>
<td>dfa_200</td>
<td>77.18%</td>
<td>63.05%</td>
<td>49.63</td>
</tr>
<tr>
<td>dfa_300</td>
<td>88.53%</td>
<td>66.01%</td>
<td>65.17</td>
</tr>
<tr>
<td>dfa_400</td>
<td>94.42%</td>
<td>69.12%</td>
<td>78.66</td>
</tr>
<tr>
<td>dfa_500</td>
<td>97.88%</td>
<td>72.29%</td>
<td>88.30</td>
</tr>
</tbody>
</table>
Future work
Future work

iii !!! ???
Future work

- Boosting techniques...
Future work

- Boosting techniques...
- Inference of teams of automata...
Future work

- Boosting techniques...
- Inference of teams of automata...
- ...

D. López and P. García
On the inference of finite automata
Thank you!