

Approximate Schemas and Data Exchange

Michel de Rougemont
University Paris II & LRI

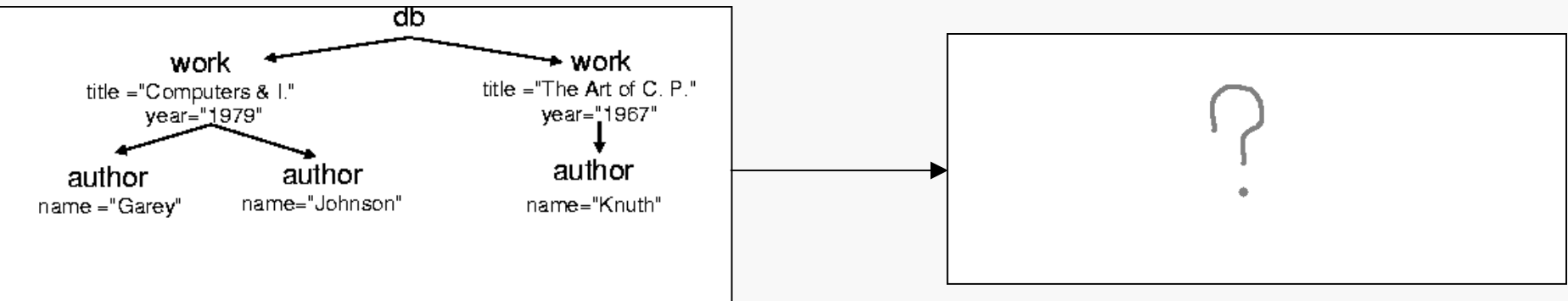


Joint work with Adrien Viellerivière,
University Paris-South

Plan

1. Classical Data Exchange on words and trees
2. Approximation based on Property Testing
3. Tester for regular words and regular trees with the Edit Distance with Moves
4. Approximate Data Exchange
5. Composition of Data Exchange setting

1. Data Exchange on Words and Trees

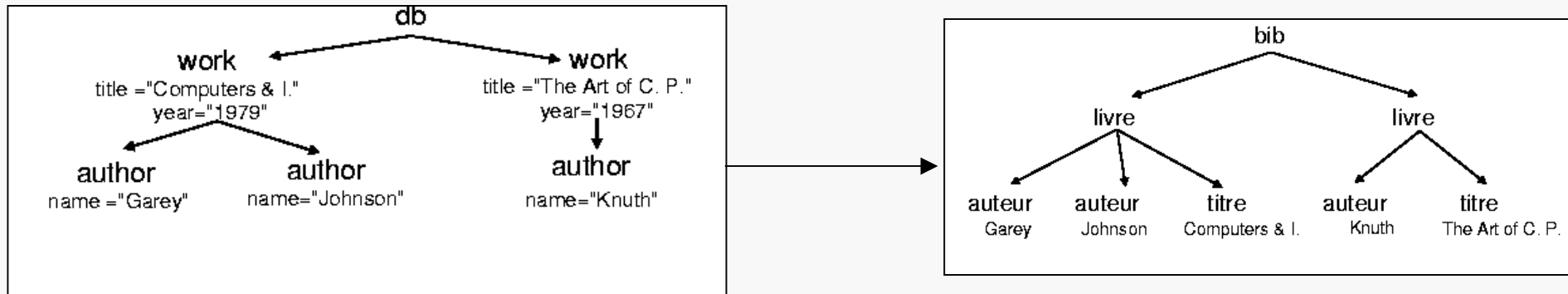
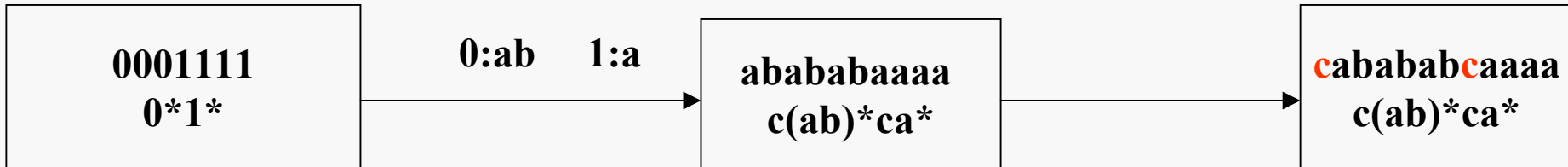


```
<!ELEMENT db (work*)>
<!ELEMENT work (author*)>
<!ATTLIST work title CDATA #REQUIRED year CDATA>
<!ELEMENT author (EMPTY)>
<!ATTLIST author name CDATA #REQUIRED>
```

```
<!ELEMENT bib (livre*)>
<!ELEMENT livre (auteur+, titre , annee)>
<!ELEMENT auteur #PCDATA>
<!ELEMENT titre #PCDATA>
<!ELEMENT annee #PCDATA>
```

Transducers transform the data

Transducer is an XSLT program:



.....
`<xsl:template match="work"> <livre><xsl:apply-templates/> <titre>`
`<xsl:value-of select="@title" /></titre></livre></xsl:template>`

Main Problems

Data Exchange setting: (K_S, τ, K_T) :

- Fagin et al. 2002: τ defined by Source-Target-Dependencies on relations
- Libkin et al. 2005: τ defined by Tree-Pattern-Formulas on trees

1. **Source-Consistency:** Given a source structure I in K_S , is there a target J in K_T s.t. (I, J) in τ ?
2. **Typechecking:** decide if for all I in K_S , there is a target J in K_T s.t. (I, J) in τ .
3. **Composition** of settings ?

Approximate Data Exchange

Data Exchange setting: (K_S, τ, K_T) , where τ is a transducer :

1. **ε -Source-Consistency:** Given a source structure I , is there a source I' ε -close to K_S s.t. $\tau(I)$ ε -close to K_T ?
2. **ε -Typechecking:** decide if for all I in K_S , $\tau(I)$ is ε -close to K_T .
3. **ε -Composition** of settings.

2. Property Testing

Let F be a property on a class K of structures U

An ϵ -**tester** for F is a probabilistic algorithm A such that:

- If $U \models F$, A accepts
- If U is ϵ far from F , A rejects with high probability

A property F is **testable** if there exists a probabilistic algorithm A s.t.

- For all ϵ it is an ϵ -**tester** for F
- $\text{Time}(A)$ independent of n .

Robust characterizations of polynomials, R. Rubinfeld, M. Sudan, 1994

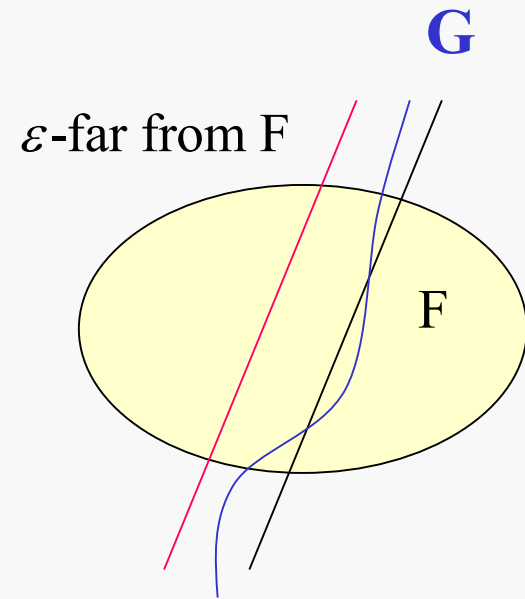
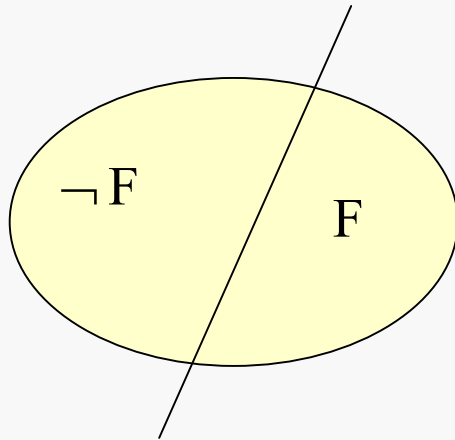
O. Goldreich, S. Goldwasser and D. Ron, [Property Testing and its connection to Learning and Approximation](#), 1996.

Tester usually implies a linear time corrector. (ϵ_1, ϵ_2) -Tolerant Tester.

Approximate Satisfiability and Equivalence

1. Satisfiability : $T \models F$
2. Approximate Satisfiability $T \models_{\varepsilon} F$
3. Approximate Equivalence $F \equiv_{\varepsilon} G$

Image on a class K of trees



History of Testers

Self-testers and correctors for Linear Algebra ,Blum & Kanan 1989

Robust characterizations of polynomials, R. Rubinfeld, M. Sudan, 1994

Testers for graph properties : k-colorability, Goldreich and al. 1996

Regular languages have testers, Alon et al. 2000s

Testers for Regular tree languages , Mdr and Magniez, 2004

Charaterization of testable properties on graphs, Alon et al. 2005

New areas: Sublinear algorithms, Approximation of decision problems

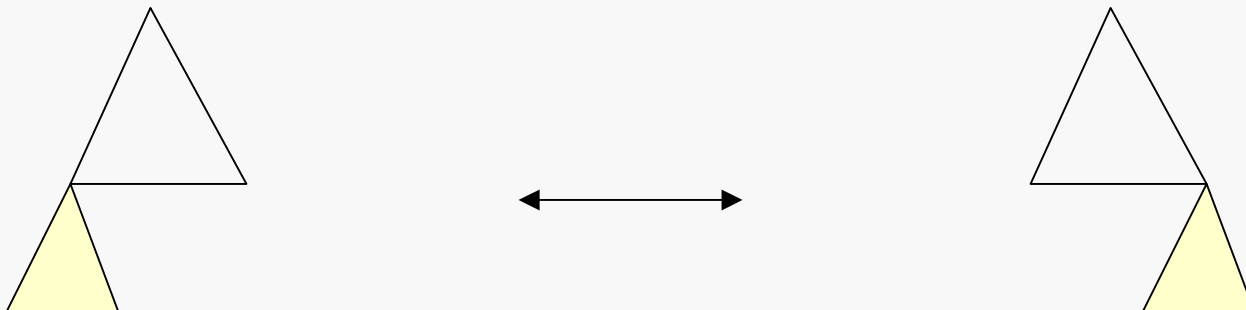
Edit Distances with Moves

1. Classical Edit Distance: *Insertions, Deletions, Modifications*
2. Edit Distance with moves

$$dist(W, W') ; dist(W, L) = \text{Min}_{W' \in L} \{dist(W, W')\}$$

01110000**111100**11001
01110**1111000000**11001

3. Edit Distance with Moves generalizes to Ordered Trees



Uniform Statistics

W=001010101110 length n , $n-k+1$ blocks of length $k=1/\epsilon$

$$u.stat(W) = \begin{pmatrix} \#n_1 \\ \dots \\ \#n_{2^k} \end{pmatrix} \cdot \frac{1}{n-k+1}$$

$\#n_1$ number of "00...0"
 $\#n_2$ number of "00...1"
 \dots
 \dots
 $\#n_{2^k}$ number of "11...1"

For $k=2$, $n-k+1=11$

$$u.stat(W) = \begin{pmatrix} 1 \\ 4 \\ 4 \\ 2 \end{pmatrix} \cdot \frac{1}{11} \approx Y(W) + \epsilon$$

$dist(W, W') \approx |u.stat(W) - u.stat(W')|$, for words of similar length,

$Y(W)$, statistics on N samples: $|u.stat(W) - Y(W)| \leq \epsilon$,

Distance between words:

- NP-complete
- Testable, $O(1)$: Sample N subwords of length k : $Y(W)$ and $Y(W')$
 If $|Y(w) - Y(w')| < \epsilon$. accept, else reject

3. Tester for a regular language

Automaton A defines L, and a polytope H for u.stats

$$u.stat(W) \approx \begin{pmatrix} 0.5 - \epsilon/2 \\ \epsilon/2 \\ \epsilon/2 \\ 0.5 - \epsilon/2 \end{pmatrix} \approx u.stat(Z) \approx u.stat(Y)$$

$$\begin{pmatrix} 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \end{pmatrix} \approx u.stat(T)$$

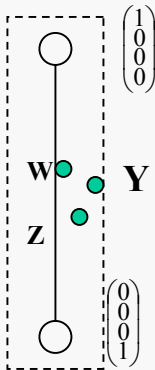
W: 0000000000111111111111

Y: 000001000011111101111

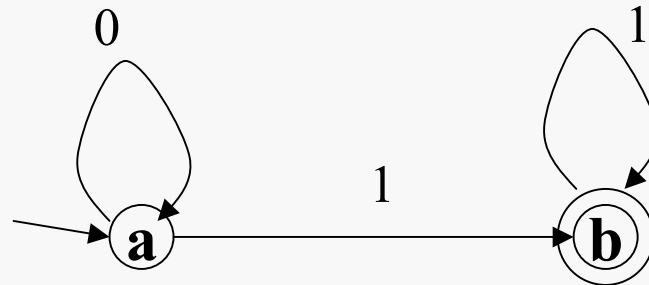
Z: 11111111111110000000000

T: 01001010001011000111010101

H



A



Tester W in L:

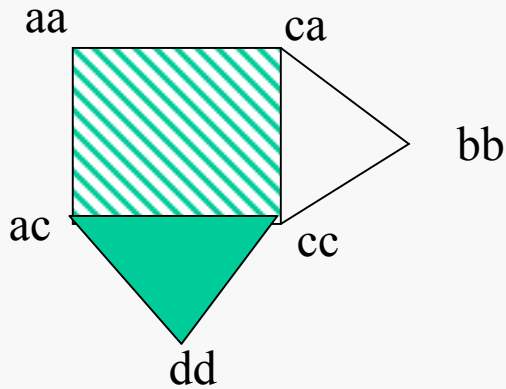
- Testable, $O(1)$: compute $Y(W)$,
- If $\text{dist}(Y(w), H) < \epsilon$. accept, else reject

Remark: robust to noise.

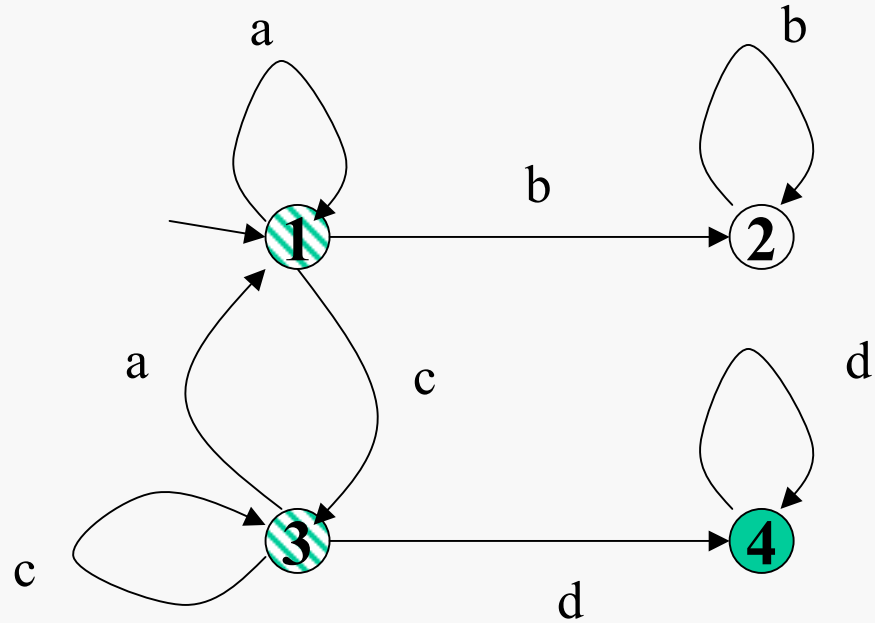
Pair (A,H)

Blocks, $k=2$, $m=4$, $|\Sigma|=4$, $|\Sigma|^k+1=17$:

Boucles de taille 1 bloc: $\{(aa,ca:1),(bb,2),(cc,ac:3),(dd:4)\}$



H



A

Corrector of a regular language

W: 000001000011111101111 is ε -close to $L(A)$

Deterministic Correction:

1. Decomposition in admissible subwords:

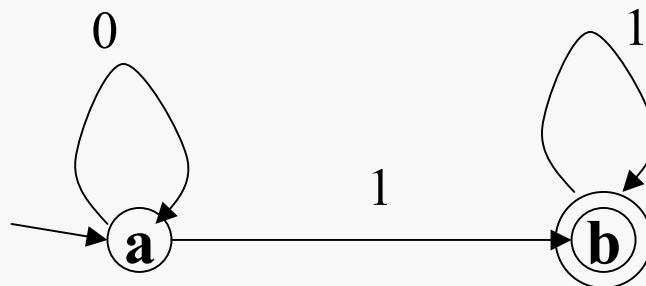
000001000011111101111
000001 000111111 1111

2. Decomposition in connected components

000001 000 111111 1111

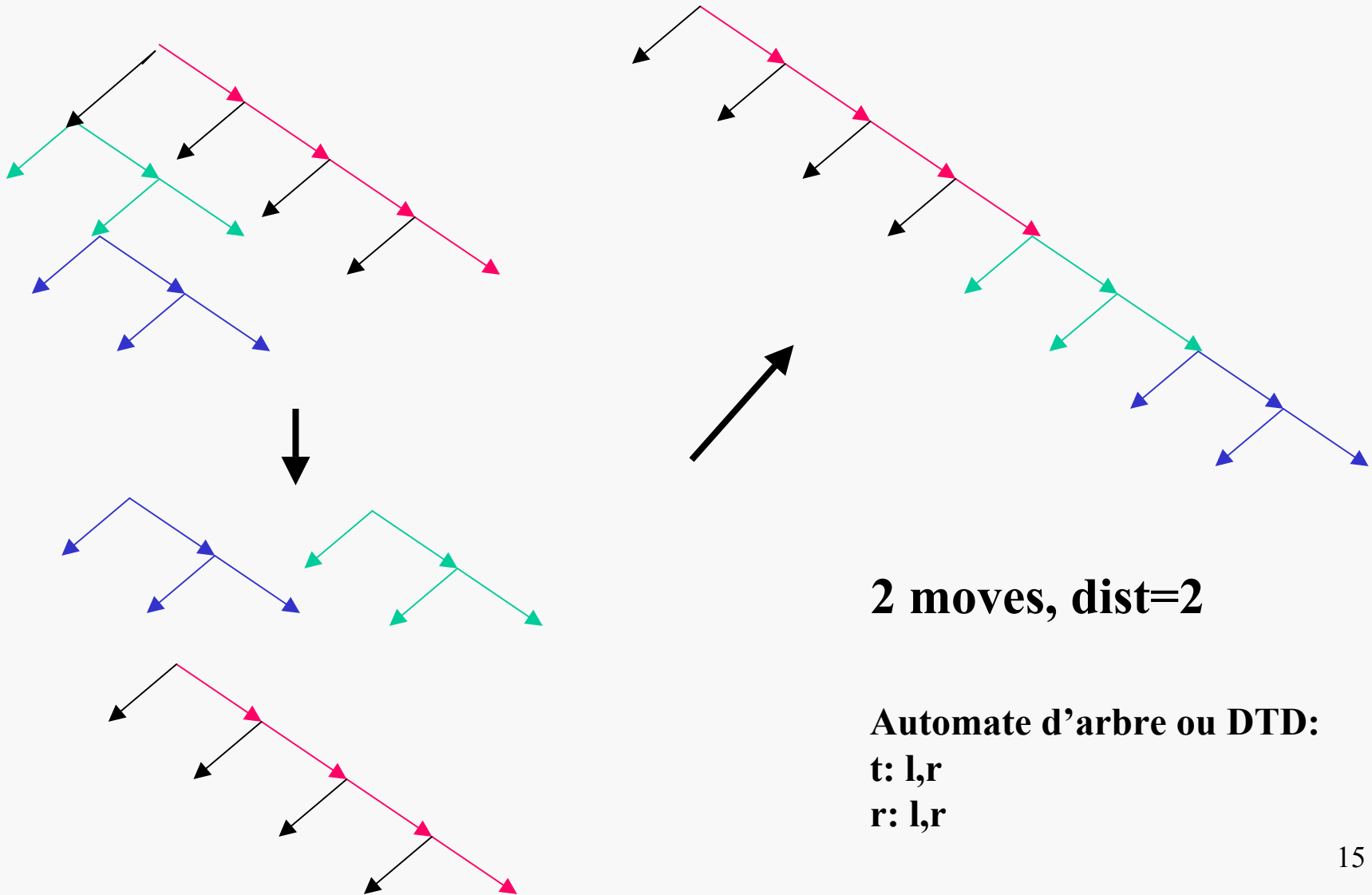
3. Recomposition (Moves)

000 000001 111111 1111 distance 3 from W



A

Corrector of an ordered tree



2 moves, dist=2

Automate d'arbre ou DTD:

t: l,r

r: l,r

XML Corrector: <http://www.lri.fr/~mdr/xml/>

XML Corrector - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Recherche Favoris

Adresse <http://www.lri.fr/~mdr/xml/?res=1&larg=1920&haut=1200>

Google Recherche Nouveau 1 bloquée(s) Orthographe Options

Info.

D e m o

Select an XML file or enter your own:

Predefined files: Ordering error in bibliographic data (bibxml.txt) ▼

Your File (Taille Max. = 10 Mo) Parcourir...

Input file: bib.xml

```
<?xml version="1.0"?>
1 <!DOCTYPE bib SYSTEM "bib.dtd">
2
3 <bib>
4 <vendor id="id1_2">
5 <name>Barnes and Nobel</name>
6 <publisher>McGraw-Hill</publisher>
7 <year>1990</year>
8 <author>
9 <lastname>Hollister</lastname>
10 <firstname>Warren</firstname>
11 </author>
12 <price>73.77</price>
13 <book>
14 <title>Crafting a Compiler with C</title>
```

Browse for DTDs:

Parcourir...

bib.dtd

Clear Remove selected DTDs

Predefineds Sets of DTDs: Select Predefined set ▼

Correct

Applications

Testers:

- Estimate the distance between two XML files,
- Décide if an XML F is ε -valid,
- Décide if two DTDs are close.

Correctors: If an XML file F is ε -close from a DTD,

- Find a valid F' ε -close to F ;
- Rank XML files for a set of DTD's (supervised learning)

Program Verification:

- Decide if two automata are ε -close in polynomial time.
- Approximate Model-Checking: <http://www.lri.fr/~mdr/vera/>
 - Specification language
 - Model
 - Distance

4. Approximate Data Exchange: typechecking

Data Exchange setting: (K_S, τ, K_T) , where τ is a transducer :

ε -Typechecking: decide if for I in K_S , $\tau(I)$ ε -close to K_T .

Words: $\tau(K_S)$ ε -close to K_T ? Apply the Equivalence Tester in polynomial time, as $\tau(K_S)$ is regular.

Trees: Similar technique, exponential in $|DTD|$.

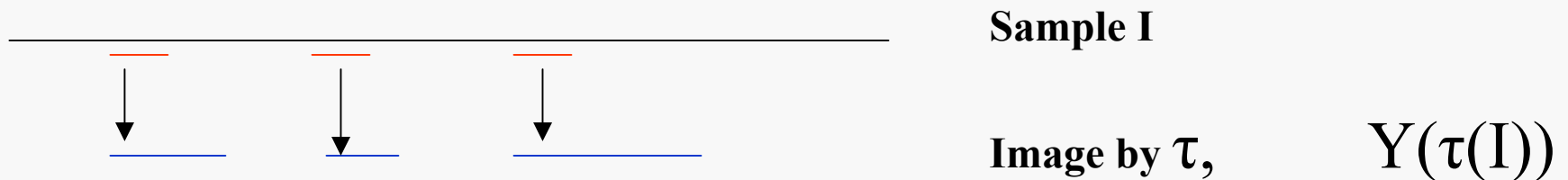
Open problem: Is DTD ε -Equivalence in P ?

Approximate Data Exchange: Source-Consistency

Data Exchange setting: (K_S, τ, K_T) , where τ is a transducer :

ε -Source-Consistency: Given a source structure I , is there a source I' ε -close to K_S s.t. $\tau(I)$ ε -close to K_T ?

Words: Case 1: Transducer with one state.



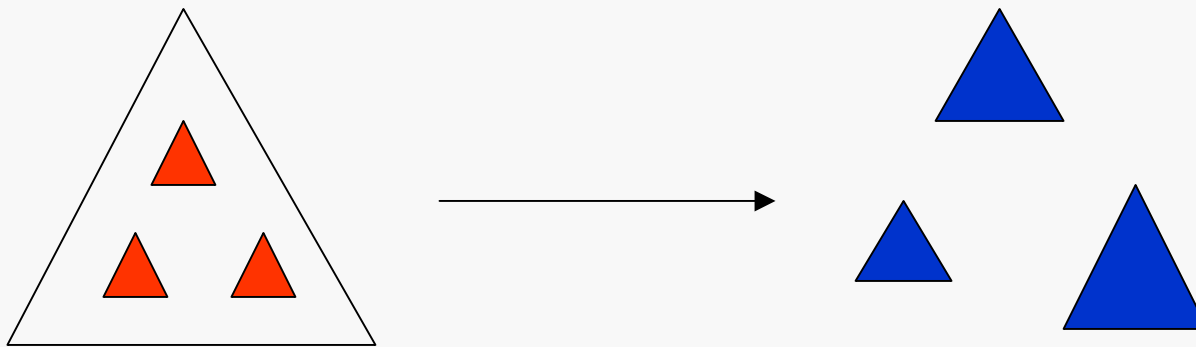
Statistics : test if $Y(\tau(I))$ is ε -close to K_T .

Case 2: Transducer with many states. Distinguish between compatible paths.

Approximate Data Exchange: Source-Consistency

Data Exchange setting: (K_S, τ, K_T) , where τ is a transducer :

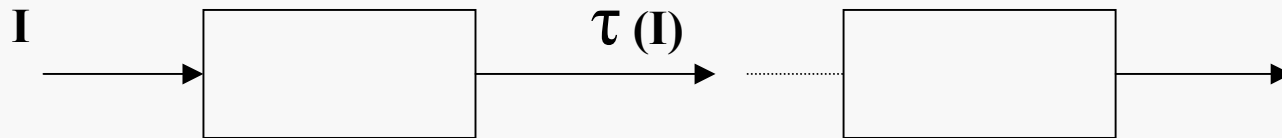
ϵ -Source-Consistency on trees:



Sampling in T provides Statistics on $\tau(T)$. Apply tester on trees.

5. Composition of close settings

Data Exchange settings: (K_{S1}, τ, K_{T1}) , (K_{S2}, τ', K_{T2}) :



Possible when the schemas are ϵ -close.

- Apply corrector at every stage to define the new τ'' for (K_{S1}, τ'', K_{T2}) : Apply corrector to $\tau(I)$ and obtain $C_1 \cdot \tau(I)$ in K_{T1} then the corrector C for K_{S2} then τ' then the corrector C_2 for K_{T2} :

$$\tau'' : C_2 \cdot \tau' \cdot C \cdot C_1 \cdot \tau(I)$$

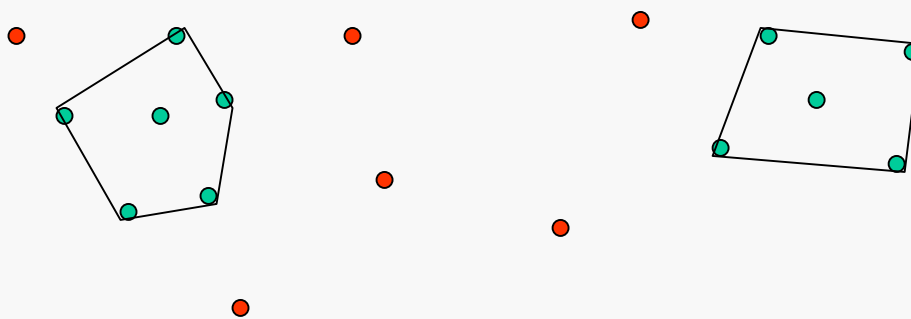
Conclusion

1. **Data Exchange:** Source-Consistency, Typechecking, Composition.
2. **Property Testing based Approximation**
3. **Tester and Corrector of regular languages**
4. **Equivalence tester for automata**
 - Polynomial time approximate algorithm (PSPACE-complete)
 - Generalization to Buchi automata : approximate Model-Checking
 - Context-Free Languages: exponential algorithm (undecidable problem)
5. **Approximate Data Exchange**
6. **Connection to PAC-Learning**

Application to learning

Model: take random words according to a distribution D :

U.stat representation:



Negative examples could include the distance.

Learning algorithm: convex hulls of positive examples.

PAC learning

The regular language is a polytope for u.stat.

Polytopes have a finite VC dimension. Hence they are PAC learnable.

Problem: the learnt concept may be ε -far from the language L.

For special distributions D, it may be ε -close.

Example: D is uniform and the polytopes are « large ».

Block and Uniform statistics

W=001010101110 length n , $k=\frac{1}{\varepsilon}$

b.stat: consecutive subwords of length k , n/k blocks

u.stat: any subwords of length k , $n-k+1$ blocks

$$b.stat(W) = \frac{1}{n/k} \begin{pmatrix} \#n_1 \\ \dots \\ \dots \\ \#n_{2^k} \end{pmatrix} \quad \begin{array}{l} \#n_1 \text{ number of "00...0"} \\ \#n_2 \text{ number of "00...1"} \\ \dots \quad \quad \quad \dots \\ \dots \quad \quad \quad \dots \\ \#n_{2^k} \text{ number of "11...1"} \end{array}$$

$$b.stat(W) = \frac{1}{6} \begin{pmatrix} 1 \\ 0 \\ 4 \\ 1 \end{pmatrix} \quad \text{For } k=2, n/k=6 \quad u.stat(W) = \frac{1}{11} \begin{pmatrix} 1 \\ 4 \\ 4 \\ 2 \end{pmatrix}$$

Main study: $|u.stat(W) - u.stat(W')|_1$

Tester for equality of strings

Edit distance with moves. NP-complete problem, but approximable in constant time with additive error.

Uniform statistics ($k=\frac{1}{\epsilon}$): $\mathbf{W}=\mathbf{001010101110}$ $u.stat(W)=\frac{1}{11}\begin{pmatrix} 1 \\ 4 \\ 4 \\ 2 \end{pmatrix}$

Theorem 1. $|u.stat(w)-u.stat(w')|$ approximates $dist(w,w')$.

Sample N subwords of length k , compute $Y(w)$ and $Y(w')$:

$$Y(w)=\frac{1}{N}\sum_{i=1\dots N}X_i \quad Y(w')=\frac{1}{N}\sum_{i=1\dots N}X_i \quad X_i=\begin{pmatrix} 0 \\ 1 \\ 0 \\ \ddots \\ 0 \end{pmatrix}$$

Lemma (Chernoff). $Y(w)$ approximates $u.stat(w)$.

Corollary. $|Y(w)-Y(w')|$ approximates $dist(w,w')$.

Tester: If $|Y(w)-Y(w')| < \epsilon$. accept, else reject.

Soundness and Robustness

Let F be a property on strings.

Soundness: ε -close strings have close statistics
 $dist(w, w') \leq \varepsilon \cdot n$

Robustness: ε -far strings have far statistics
 $dist(w, w') \geq \varepsilon \cdot n$

F is Equality on pairs of strings.

For theorem 1, we prove:

1. b.stat is robust
2. u.stat is sound
3. u.stat is robust

Robustness of b.stat

Robustness of b-stat: $dist(w, w') \leq (\frac{1}{2} \cdot |b.stat(w) - b.stat(w')| + \varepsilon) \cdot n$

If $b.stat(w) = b.stat(w')$ then $dist(w, w') \leq \varepsilon \cdot n$

If $b.stat(w) \neq b.stat(w')$ then construct w'' s. t. $b.stat(w'') = b.stat(w')$ after at most $\frac{n}{2} \cdot |b.stat(w) - b.stat(w')|$ substitutions on w . Example:

$$b.stat(W) = \frac{1}{6} \begin{pmatrix} 1 \\ 0 \\ 4 \\ 1 \end{pmatrix} \qquad b.stat(W') = \frac{1}{6} \begin{pmatrix} 2 \\ 0 \\ 3 \\ 1 \end{pmatrix}$$

#"00"=1 in W and 2 in W' but #"10"= 4 in W and 3 in W'
 W'' : take one block of "00" in W and change it into "10"

Soundness of u.stat

Soundness of u-stat: $dist(w, w') \leq \varepsilon^2 \cdot n \implies |u.stat(w) - u.stat(w')| \leq 6 \cdot \varepsilon$

Simple edit: $|u.stat(w) - u.stat(w')| \leq \frac{2k}{n-k+1} \leq \frac{2}{n \cdot \varepsilon}$

Move $w = A.B.C.D$, $w' = A.C.B.D$: $|u.stat(w) - u.stat(w')| \leq \frac{2 \cdot 3(k-1)}{n-k+1} \leq \frac{6}{n \cdot \varepsilon}$

Hence, for $\varepsilon^2 \cdot n$ operations, $|u.stat(w) - u.stat(w')| \leq 6 \cdot \varepsilon$

Remark: **b.stat** is not sound.

Problem: robustness of **u.stat** ?

Harder! We need an auxiliary distribution and two key lemmas.

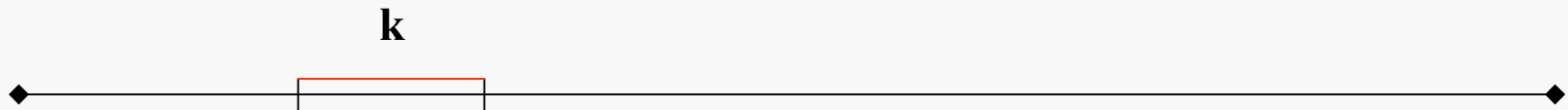
Statistics on words

Block statistics: b.stat

$$k = \frac{1}{\varepsilon}$$

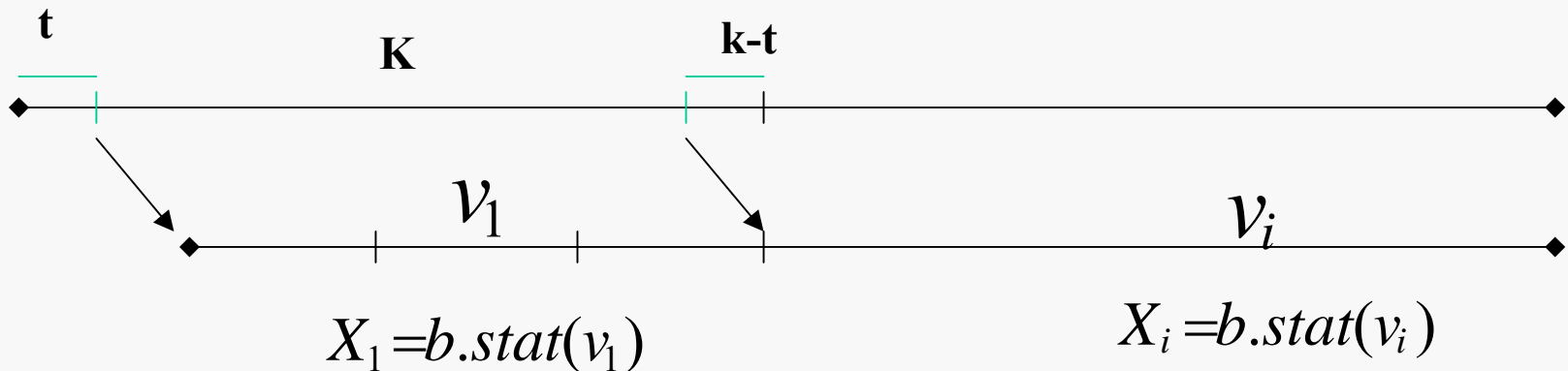


Uniform statistics: u.stat



Block Uniform statistics: bu.stat

$$K = c.k^2$$

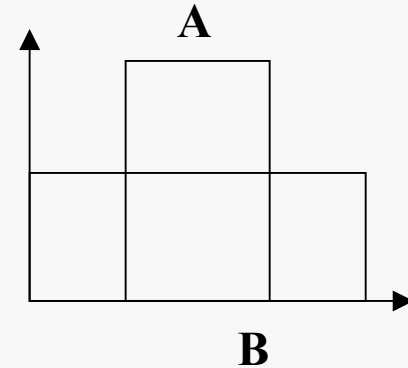


$$bu.stat(w) = \frac{K}{n} \sum_{i=1, \dots, n/K} E_{t_i}(b.stat(v_i)) = E(b.stat(v))$$

Uniform Statistics

Lemma : Let $A \subseteq B$ and two uniform distributions μ_A, μ_B .

$$\text{Then } |\mu_A - \mu_B| = 2 \cdot \frac{|B| - |A|}{|B|}$$



Lemma 2: $\forall w \left| bu.stat(w) - u.stat(w) \right| \leq \frac{|\Sigma|^{2/\varepsilon}}{\varepsilon^4 \cdot n}$

#subwords of length k missed by bu : $(k-1) \cdot \left(\frac{n}{K} - 1\right) = |B| - |A|$

Apply the previous lemma with $|B| = n - k + 1$, $K \approx \frac{\varepsilon^3 \cdot n}{|\Sigma|^{2/\varepsilon}}$

$$\left| u.stat(v) - bu.stat(w) \right| = O\left(\frac{|\Sigma|^{2/\varepsilon}}{\varepsilon^4 \cdot n}\right)$$

Block Uniform Statistics

Lemma 1: $\forall w \exists v |bu.stat(w) - b.stat(v)| \leq \frac{\varepsilon}{2}$ and $dist(v, w) \leq c_\varepsilon$

$$bu.stat(w) = \frac{K}{n} \sum_{i=1, \dots, n/K} E_{t_i}(b.stat(v_i)) = E(b.stat(v))$$

$$X_i = b.stat(v_i), \quad X_i[u] = b.stat(v_i)[u], \quad 0 \leq X_i[u] \leq 1$$

Each $X_i[u]$ is independent. Average on i is $bu.stat(w)[u]$

Chernoff Bound: $\Pr[|b.stat(v)[u] - bu.stat(w)[u]| \geq t \times bu.stat(w)[u]] \leq e^{-\frac{8n}{K}t^2}$

Union Bound: $\Pr[|b.stat(v) - bu.stat(w)| \geq t \times bu.stat(w)] \leq |\Sigma|^k \cdot e^{-\frac{8n}{K}t^2}$

For large enough n and $t = \frac{\varepsilon}{2|\Sigma|^k} \Rightarrow \Pr[|b.stat(v) - bu.stat(w)| \leq \frac{\varepsilon}{2}] > 0$

Robustness of the uniform Statistics

Robustness of u-stat: $dist(w, w') \geq 5\varepsilon .n \Rightarrow |u.stat(w) - u.stat(w')| \geq 6,5.\varepsilon$

Robustness of bstat: $(\frac{1}{2} \cdot |b.stat(w) - b.stat(w')| + \varepsilon) .n \geq dist(w, w') \geq 5.\varepsilon .n$

By Lemma 1: $\forall w \exists v |bu.stat(w) - b.stat(v)| \leq \frac{\varepsilon}{2}$ Get v, v' close from w, w'

By Lemma 2: $\forall w |bu.stat(w) - u.stat(w)| \leq \frac{|\Sigma|^{2/\varepsilon}}{\varepsilon^4 .n}$

Robustness of b-stat implies robustness of u-stat.

Tolerant tester: $N = O(c_\varepsilon)$, Accept if $|Y(w) - Y(w')| \leq 5.\varepsilon$

Theorem: for two words w and w' large enough, the tester:

1. Accepts if $w = w'$ with probability 1
2. Accepts if w, w' are ε^2 -close with probability 2/3
3. Rejects if w, w' are ε -far with probability 2/3

Membership and Equivalence tester

Membership Tester for w in L (regular):

1. Construction of the tester: Precompute H_ϵ
2. Tester: Compute $Y(w)$ (approx. $b.stat(w)$).
Accept iff $Y(w)$ is at distance less than ϵ to H_ϵ

Construction: Time is $m^{|\Sigma|^{O(k)}}$.

Tester: query complexity in $|\Sigma|^{O(k)}$
time complexity in $2^{|\Sigma|^{O(k)}}$.

Remark 1: Time complexity of previous testers was exponential in m .

Remark 2: The same method works for L context-free.

Tester of $A \equiv_\epsilon B$

1. Compute $H_{\epsilon,A}$ and $H_{\epsilon,B}$
2. Reject if $H_{\epsilon,A}$ and $H_{\epsilon,B}$ are different.

Time polynomial in $m = \text{Max}(|A|, |B|)$: $m^{|\Sigma|^{O(k)}}$.

Generalizations

Buchi Automata.

Distance on infinite words:

Two words are ε -close if $\sup \lim_{n \rightarrow \infty} \text{dist}(w(n), w'(n)) \leq \varepsilon$

A word is ε -close to a language L if there exists w' in L s. t. w and w' are ε -close.

Statistics: set of accumulation points of $b.stat(w(n))_n$

H: compatible loops of connected components of accepting states

Tester for Buchi Automata:

Compute H_A and H_B

Reject if H_A and H_B are different.

Equivalence of CF grammars is undecidable, Approximate equivalence in exponential.