

Schema-Guided Induction of Monadic Queries

Jérôme Champavère

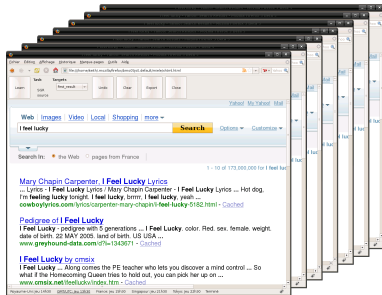
Joint work with R. Gilleron, A. Lemay and J. Niehren

Université de Lille, France
LIFL (Grappa), INRIA (Mostrare)

ICGI 2008, Saint-Malo, France
September 24, 2008

Queries for Web Information Extraction

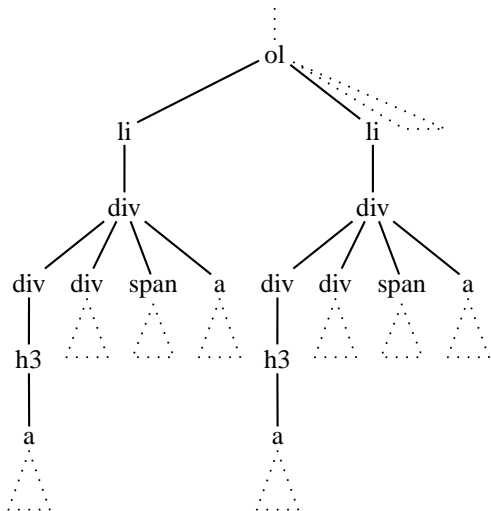
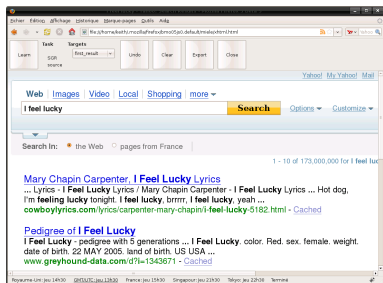
- Basic problem: find queries that select data in a set of Web sources



- Various machine learning techniques
 - Classification [Marty et al., 2006]
 - Conditional random fields [Kristjansson et al., 2004]
 - Inductive logic programming [Cohen et al., 2002]
 - Tree automata inference [Kosala, 2003]

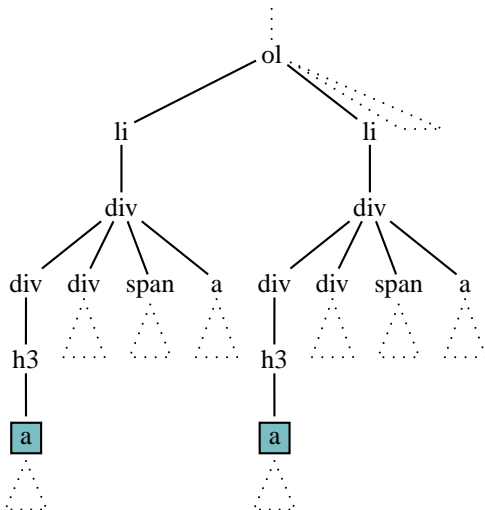
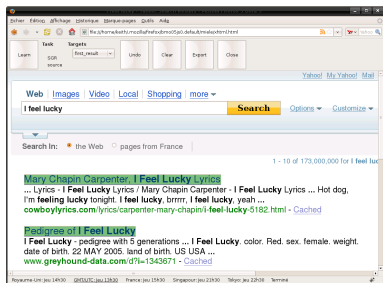
Query in XHTML Documents

Select nodes in unranked trees.



Query in XHTML Documents

Select nodes in unranked trees.



Using Schema Information

- Motivation: find better heuristics for learning
- Schemas describe valid document collections
 - Web pages: DTD of XHTML
 - Inferred schemas, e.g. [Bex et al., 2006]
- No schema information taken into account so far
- Representation of DTDs by tree automata
- Idea: prevent from wrong out-of-domain generalizations

Using Schema Information

- Motivation: find better heuristics for learning
- Schemas describe valid document collections
 - Web pages: DTD of XHTML
 - Inferred schemas, e.g. [Bex et al., 2006]
- No schema information taken into account so far
- Representation of DTDs by tree automata
- Idea: prevent from wrong out-of-domain generalizations

For strings: domain bias [Coste et al., 2004], Pierre Dupont yesterday's talk [Dupont et al., 2008]

Main Contributions

- Framework: RPNI-based algorithm for stepwise tree automata [Carme et al., 2007]
- Integration of schema information
 - Addition of schema consistency
 - Pruning with schemas
- Implementation of both aspects of schema guidance

Outline

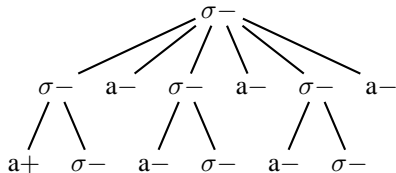
- 1 Schema-Guided Query Induction
- 2 Schema-Guided Pruning for Interactive Query Induction
- 3 Implementation and Experiments

Outline

- 1 Schema-Guided Query Induction
- 2 Schema-Guided Pruning for Interactive Query Induction
- 3 Implementation and Experiments

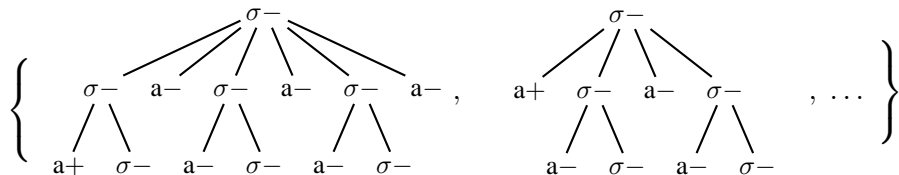
Queries as Tree Languages

- Language of annotated trees, i.e. trees over $\Sigma \times \{+, -\}$



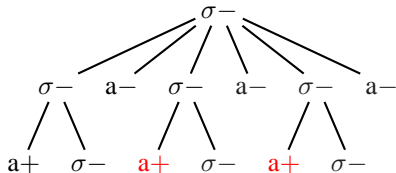
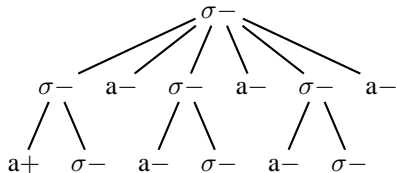
Queries as Tree Languages

- Language of annotated trees, i.e. trees over $\Sigma \times \{+, -\}$
- Sample
 - Set of correctly annotated trees
 - Only positive examples for learning



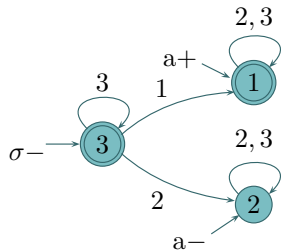
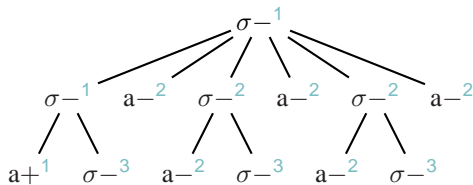
Queries as Tree Languages

- Language of annotated trees, i.e. trees over $\Sigma \times \{+, -\}$
- Sample
 - Set of correctly annotated trees
 - Only positive examples for learning
- Functionality: no contradictory annotations



Queries as Tree Languages

- Language of annotated trees, i.e. trees over $\Sigma \times \{+, -\}$
- Sample
 - Set of correctly annotated trees
 - Only positive examples for learning
- Functionality: no contradictory annotations
- Target: stepwise tree automaton over $\Sigma \times \{+, -\}$ that recognizes functional tree languages

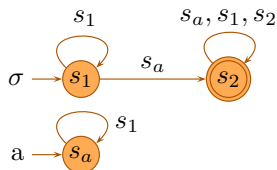


Schema-Consistent Queries

- RPNI-based learning algorithm
 - Merge states
 - Test functionality
- Schema-guided RPNI: check schema-consistency of queries by language inclusion

Schema-Consistent Queries

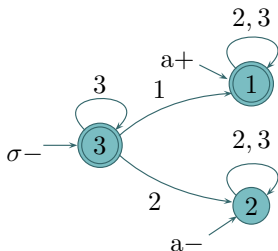
- RPNI-based learning algorithm
 - Merge states
 - Test functionality
- Schema-guided RPNI: check schema-consistency of queries by language inclusion



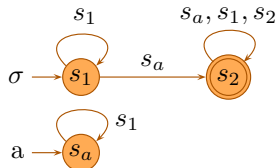
No 'a' descendent of another 'a'; at least one 'a'.

Schema-Consistent Queries

- RPNI-based learning algorithm
 - Merge states
 - Test functionality
- Schema-guided RPNI: check schema-consistency of queries by language inclusion



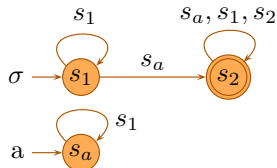
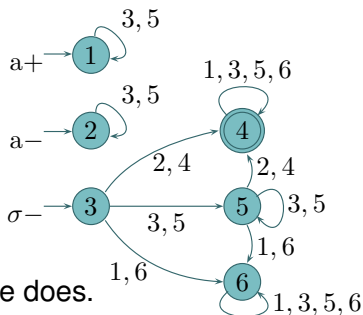
This one does not satisfy the schema.



No 'a' descendent of another 'a'; at least one 'a'.

Schema-Consistent Queries

- RPNI-based learning algorithm
 - Merge states
 - Test functionality
- Schema-guided RPNI: check schema-consistency of queries by language inclusion



No 'a' descendent of another 'a'; at least one 'a'.

Schema-Consistent Queries

- RPNI-based learning algorithm
 - Merge states
 - Test functionality
- Schema-guided RPNI: check schema-consistency of queries by language inclusion

Problem: how to test inclusion efficiently?

Efficient Inclusion Checking

- Automata for schemas have to be deterministic
- Projection of automata for queries can be non-deterministic
- Efficient inclusion test in $O(|A| * |\Sigma| * |D|)$ for stepwise tree automata A and DTDs D over Σ [Champavère et al., 2008]
 - Non trivial: naive algorithm in $O(|A| * |\Sigma| * |D|^2)$
 - Factorized automata to avoid the DTD transformation blowup

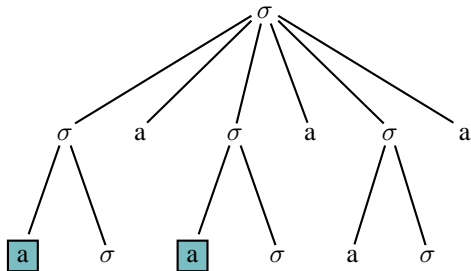
RPNI with schema-consistency checking: $O(|S|^4 * |\Sigma| * |D|)$, where S is a sample of positive examples.

Outline

- 1 Schema-Guided Query Induction
- 2 Schema-Guided Pruning for Interactive Query Induction**
- 3 Implementation and Experiments

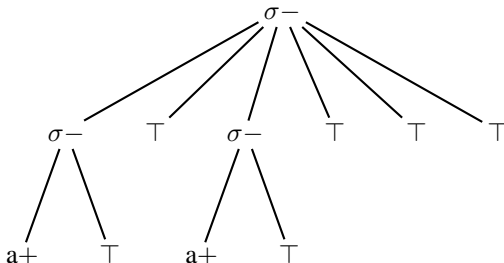
Learning with Partial Information

- Users should not have to annotate whole documents
- Pruning heuristics [Carme et al., 2007]



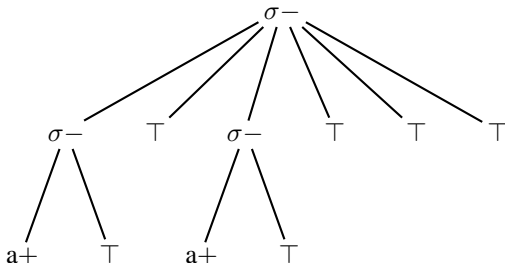
Learning with Partial Information

- Users should not have to annotate whole documents
- Pruning heuristics [Carme et al., 2007]



Learning with Partial Information

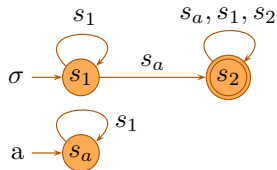
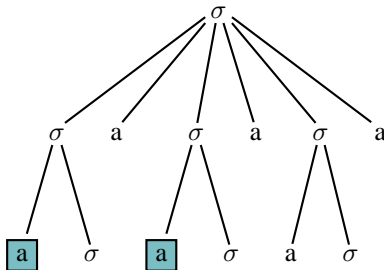
- Users should not have to annotate whole documents
- Pruning heuristics [Carme et al., 2007]



How to use schema information?

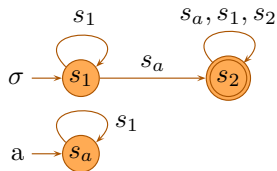
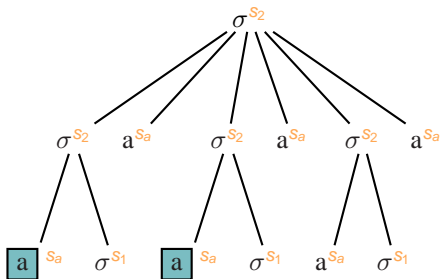
Pruning with Schemas

The trees can be pruned by using states of the schema instead of \top .



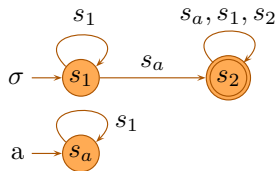
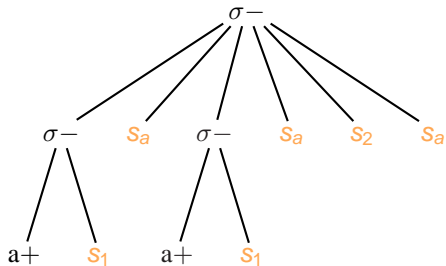
Pruning with Schemas

The trees can be pruned by using states of the schema instead of \top .



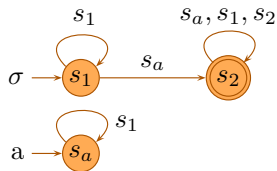
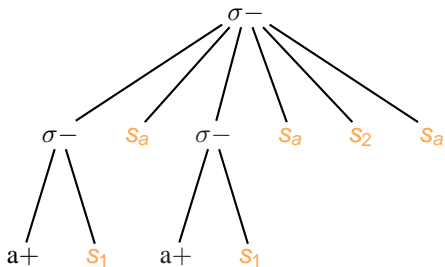
Pruning with Schemas

The trees can be pruned by using states of the schema instead of \top .



Pruning with Schemas

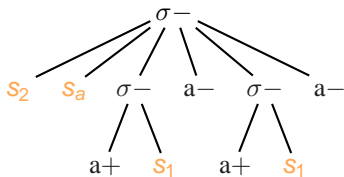
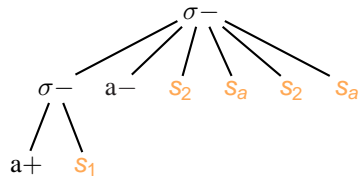
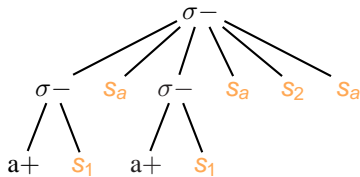
The trees can be pruned by using states of the schema instead of \top .



What about functionality?

S-cut-functionality

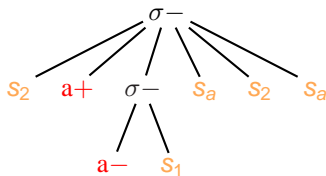
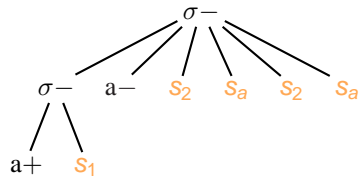
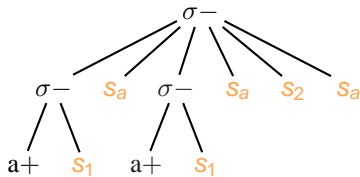
- Partially annotated trees
- Same tree, different prunings: no contradictory annotations



Those are compatible.

S-cut-functionality

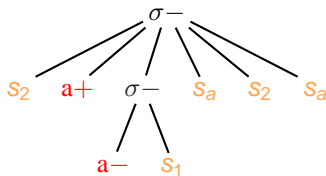
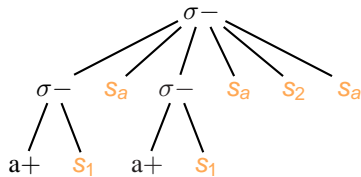
- Partially annotated trees
- Same tree, different prunings: no contradictory annotations



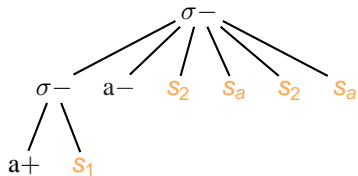
Those are not.

S-cut-functionality

- Partially annotated trees
- Same tree, different prunings: no contradictory annotations



Those are not.



S-cut-functionality can be checked in $O(|S|^2 + |S| * |D|)$.

Outline

- 1 Schema-Guided Query Induction
- 2 Schema-Guided Pruning for Interactive Query Induction
- 3 Implementation and Experiments**

Inclusion Algorithm

- DTDs to tree automata
 - One-unambiguous regular expressions e to deterministic Glushkov automata: $O(|\Sigma| * |e|)$ [Brüggemann-Klein & Wood, 1998]
 - Simple combination of G-automata: unwanted quadratic blowup
 - Factorized automata: more compact, sufficient notion of determinism, linear time transformation from G-automata

Inclusion Algorithm

- DTDs to tree automata
 - One-unambiguous regular expressions e to deterministic Glushkov automata: $O(|\Sigma| * |e|)$ [Brüggemann-Klein & Wood, 1998]
 - Simple combination of G-automata: unwanted quadratic blowup
 - Factorized automata: more compact, sufficient notion of determinism, linear time transformation from G-automata
- Incrementality
 - Initial automaton obviously schema-consistent
 - Inclusion test based on accessibility
 - Add ϵ -transitions between states of the initial automaton
 - Update accessible states and check for inclusion failure

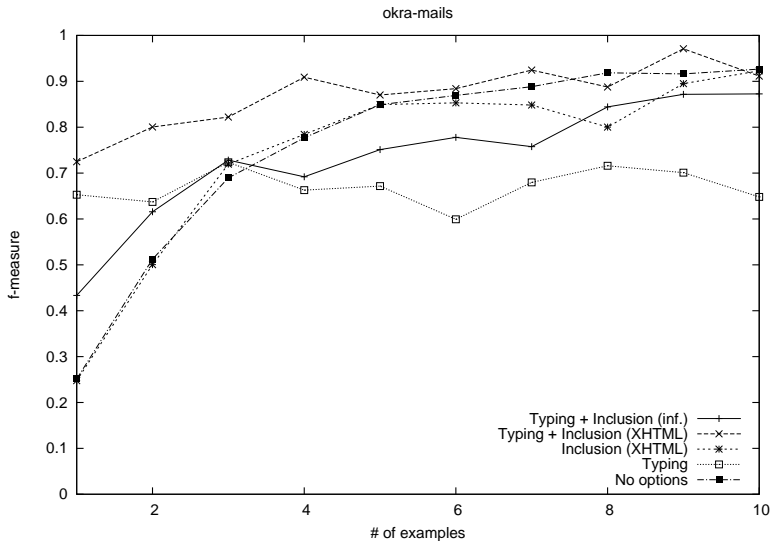
Measuring the Effect of Several Heuristics

- Parameters of the learning algorithm
 - Do verify schema consistency, or not
 - Pruning with the help of schema, or with universal language
 - Do use a simple state typing heuristics, or not
- Different combinations of previous heuristics are possible

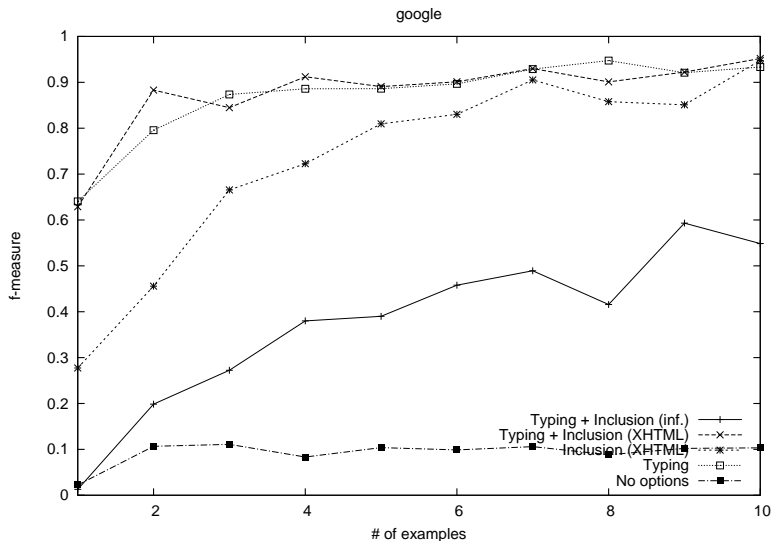
Experimental Insights

- Two scenarios of learning
 - Non-interactive, i.e. complete annotations
 - Interactive, i.e. partial annotations
- Extra-time to check schema-consistency: not so expansive
- Schema-consistency has uncertain impact on learning quality
- Schema-guided pruning is of interest in interactive settings

Experimental Insights



Experimental Insights



Experimental Insights

- Two scenarios of learning
 - Non-interactive, i.e. complete annotations
 - Interactive, i.e. partial annotations
- Extra-time to check schema-consistency: not so expansive
- Schema-consistency has uncertain impact on learning quality
- Schema-guided pruning is of interest in interactive settings

Experimental Insights

Table 1. Interactive learning. For each dataset, we present the number of necessary corrections/pages to learn the target query (T=typing heuristics; I=inclusion; P=schema-guided pruning). All experiments have been done with regular pruning, unless P is specified.

| | T | I (HTML DTD) | T + I (HTML DTD) | T + I (Inferred DTD) | T + P (HTML DTD) |
|---------|-----------|-----------------|---------------------|-------------------------|---------------------|
| Okra | failed | 17.93/3.87 | 4.00/2.03 | 4.60/2.73 | 3.73/1.87 |
| Bigbook | 3.03/1.37 | 3.20/1.57 | 2.77/1.77 | 2.33/1.33 | 3.90/1.37 |
| Google | 4.53/2.33 | 9.60/3.43 | 8.00/4.00 | 28.60/12.03 | 6.90/3.53 |

Experimental Insights

- Two scenarios of learning
 - Non-interactive, i.e. complete annotations
 - Interactive, i.e. partial annotations
- Extra-time to check schema-consistency: not so expansive
- Schema-consistency has uncertain impact on learning quality
- Schema-guided pruning is of interest in interactive settings

Clearly, we need further heuristics and better control on data.

Conclusion




Summary

- Two aspects of schema-guidance
 - Consistency checking
 - Pruning heuristics
- Preliminary experimental results

Future Work

- Further heuristics, e.g. state merging ordering
- Text content
- n -ary queries
- Tree transformations

Some References

-  F. Coste, D. Fredouille, C. Kermovant & C. de la Higuera (2004)
Introducing Domain and Typing Bias in Automata Inference
In Proceedings of the 7th International Colloquium on Grammatical Inference
-  J. Carme, R. Gilleron, A. Lemay & J. Niehren (2007)
Interactive Learning of Node Selecting Tree Transducers
Machine Learning, 66(1)
-  J. Champavère, R. Gilleron, A. Lemay & J. Niehren (2008)
Efficient Inclusion Checking for Tree Automata and DTDs
In Proceedings of the 2nd International Conference on Language and Automata Theory and Applications

Learning from Completely & Partially Annotated Trees

```

RPNIprune,consS,type ( $E, \langle t, e_+, e_- \rangle$ )
// sample of completely annotated examples  $E \subseteq T_{\Sigma \times \mathbb{B}}$ 
// partially annotated example  $\langle t, e_+, e_- \rangle \in T_{\Sigma} \times \text{nodes}(t)^2$ 
// schema defined by a deterministic factorized tree automaton  $S$  over
 $\Sigma$ 

```

```

// prune all example trees w.r.t. schema definition  $S$ //
let  $E' = \{\text{prune}_S(t' * \beta) \mid t' * \beta \in E\} \cup \{\text{prune}_S(t * p_+)\}$ 
// compute the initial automaton
let  $A$  be a deterministic  $S$ -pNSTT such that  $L(A) = E'$ 
let  $\text{states}(A) = \{q_1, \dots, q_n\}$  in some admissible order
// generalize  $A$  by state merging //
for  $i = 1$  to  $n$  do
  for  $j = 1$  to  $i - 1$  with type  $(q_j) = \text{type}(q_i)$  do
    let  $A' = \text{det-merge}(A, q_i, q_j)$ 
    if  $A'$  is  $S$  cut-functional //  $S$ -consistency of annotations on pruned trees
    and if  $\text{cons} = \text{yes}$  then  $\{t \mid t * \beta \in L(A')\} \subseteq L(S)$  // query  $S$ -consistent
    and  $A'$  consistent with sample  $E$  and example  $\langle t, e_+, e_- \rangle$ 
    then  $A \leftarrow A'$ 
    else skip

```

Output : A
