Predictable and Consistent Information Extraction

Besat Kassaie   Frank Wm. Tompa

The Cheriton School of Computer Science, University of Waterloo

Problem

Given an information extraction specification and view/document update languages how can source documents be updated to produce the modified extracted view?

Information Extraction

Information extraction identifies and isolates words and phrases within documents and presents them as relational tables in order to reveal the relationships among those text fragments in structured form [1].

JAPE

JAPE [2] is a well-established rule-based extraction language. Running a JAPE program involves executing a set of matching rules, written as regular expressions over annotations that label edges in a rooted directed acyclic graph.

Robust Extractors

A given extraction algorithm is robust if it produces expected records from a legitimately modified document, *all* possible input document collections, *all* entries in the extracted table, and *all* values in each entry's domain.

Characterization of Robust Extractors

STRICT

An extractor is *strict* if for every possible input document, the set of extracted values in the corresponding record is a subset of words and phrases appearing in the input.

COMPUTABLE

A strict extractor is *computable* if for all possible input documents and corresponding extracted attributes, we have access to lineage of the attributes are extracted.

STABLE

With a *stable* extractor, changing values in appropriate positions in a document affects only the expected attribute in the extracted record.

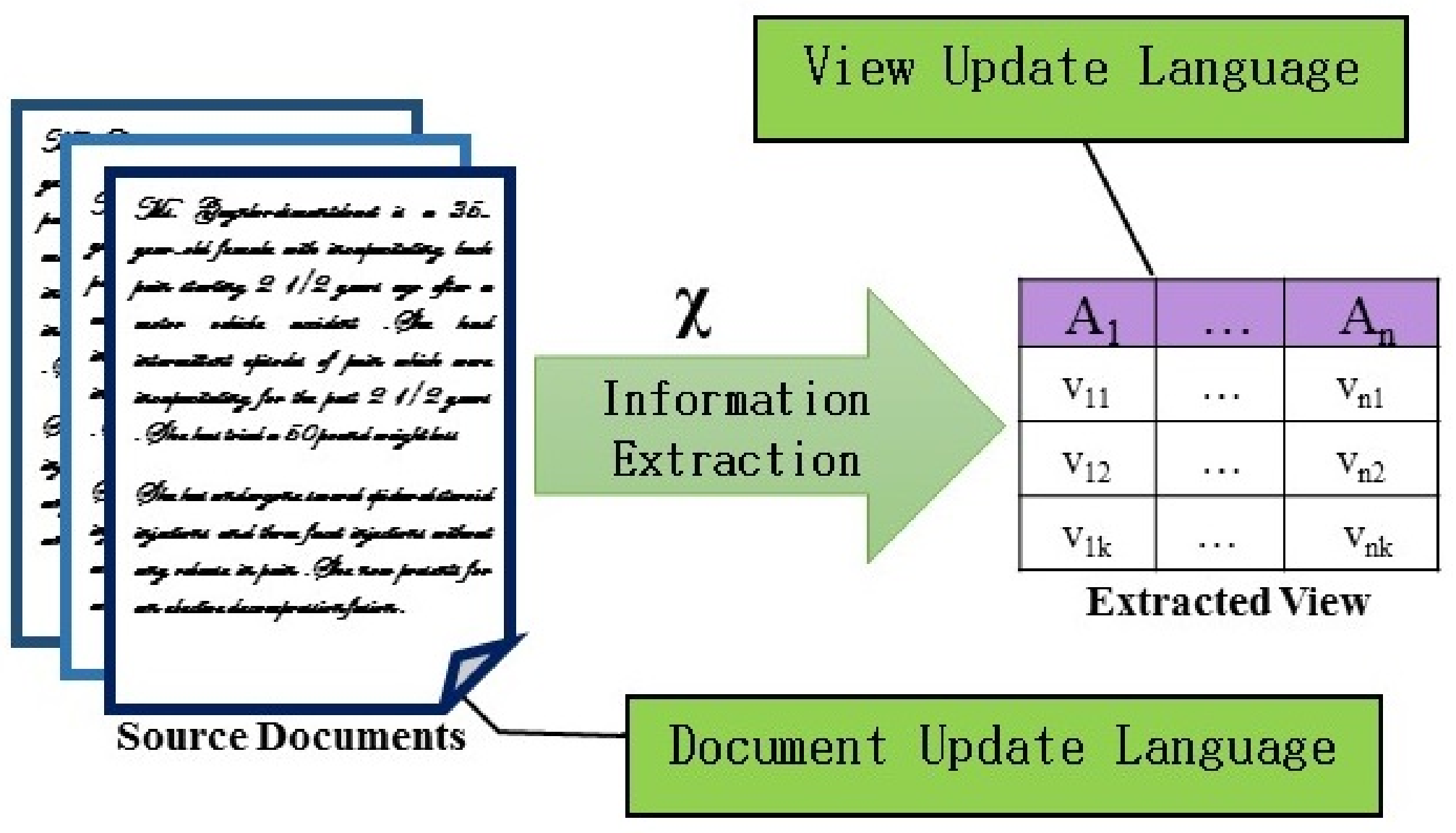

Figure 1: Two components are added to extraction process.

Proposition

For any strict, computable, and stable extractor $\mathcal{X} : \mathcal{D} \to \mathcal{R}$, there exists an algorithm $A(\mathcal{F}, D, P_{\mathcal{X}}(D, j))$ such that for all indexed sets of domain preserving functions $\mathcal{F} = \{f_i | f_i : W_i \to W_i$, where $i \in [1 \ldots \mathcal{T}]\}$ and any document $D \in \mathcal{D}$, $A(\mathcal{F}, D, P_{\mathcal{X}}(D, j))$ produces $D^{\mathcal{P}}_{\mathcal{F}}$ in such way that $F(\mathcal{X}(D)) = \mathcal{X}(D^{\mathcal{P}}_{\mathcal{F}})$.

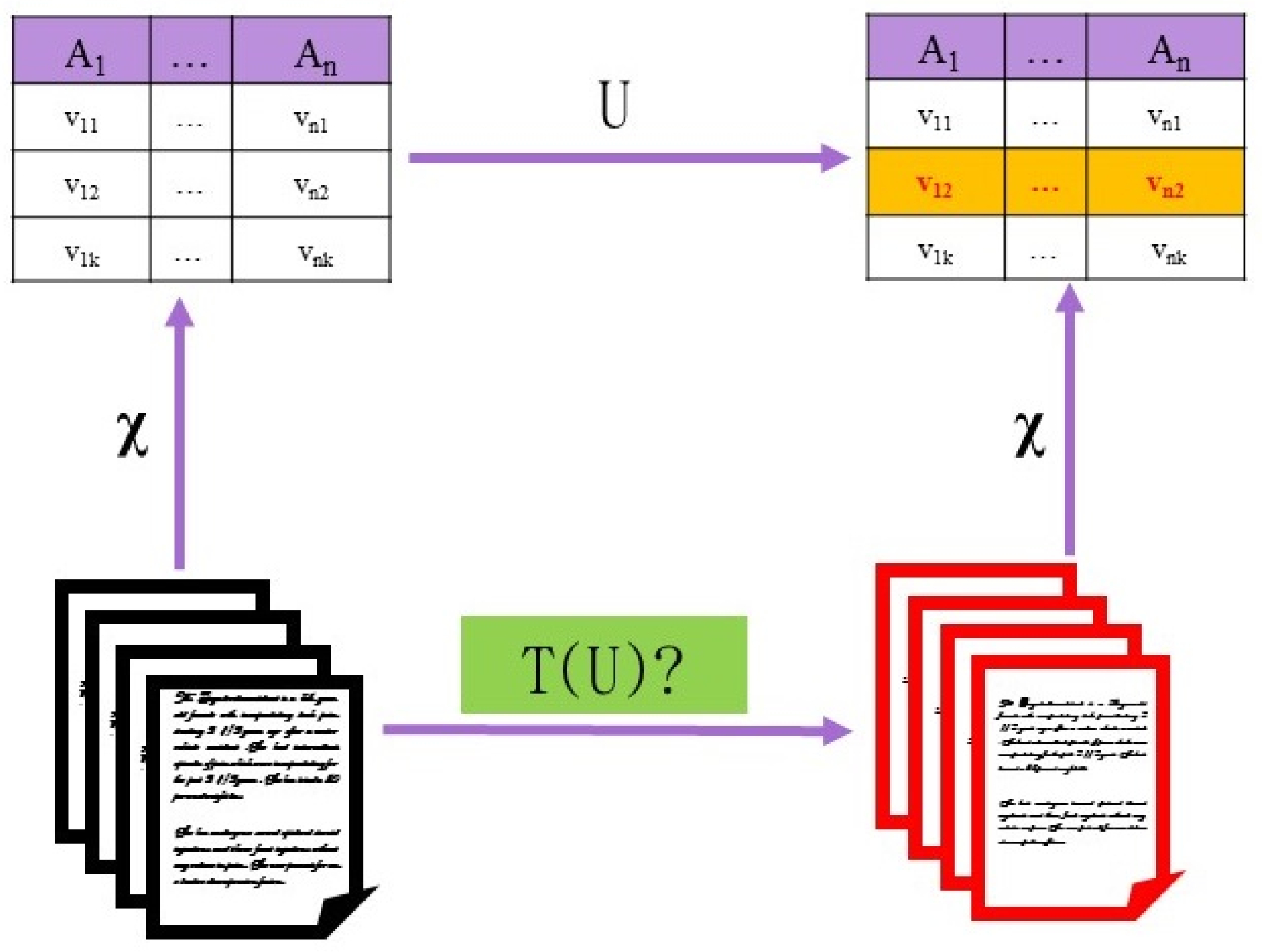

Figure 2: The goal is to find a translation of updates over extracted view to updates over documents.

Claim

For any information extraction algorithm $\mathcal{X}$ having the aforementioned properties, Algorithm 1 produces $D^{\mathcal{P}}_{\mathcal{F}}$ in such a way that $F(\mathcal{X}(D))= \mathcal{X}(D^{\mathcal{P}}_{\mathcal{F}})$.

```
Input: $\mathcal{F}$, $D$, $j \to P_{\mathcal{X}}(D, j)$
Output: $D^{\mathcal{P}}_{\mathcal{F}}$
$D^{\mathcal{P}}_{\mathcal{F}} \leftarrow D$
for $j \in [1 \ldots \mathcal{T}]$ do
  for $\langle a, b \rangle \in P_{\mathcal{X}}(D, j)$ do
    replace $D[a, b] \in D^{\mathcal{P}}_{\mathcal{F}}$ by $f_j(D[a, b])$
  end
end
return $D^{\mathcal{P}}_{\mathcal{F}}$
```

Algorithm 1: Updating a document.

VERIFICATION OF JAPE PROGRAMS

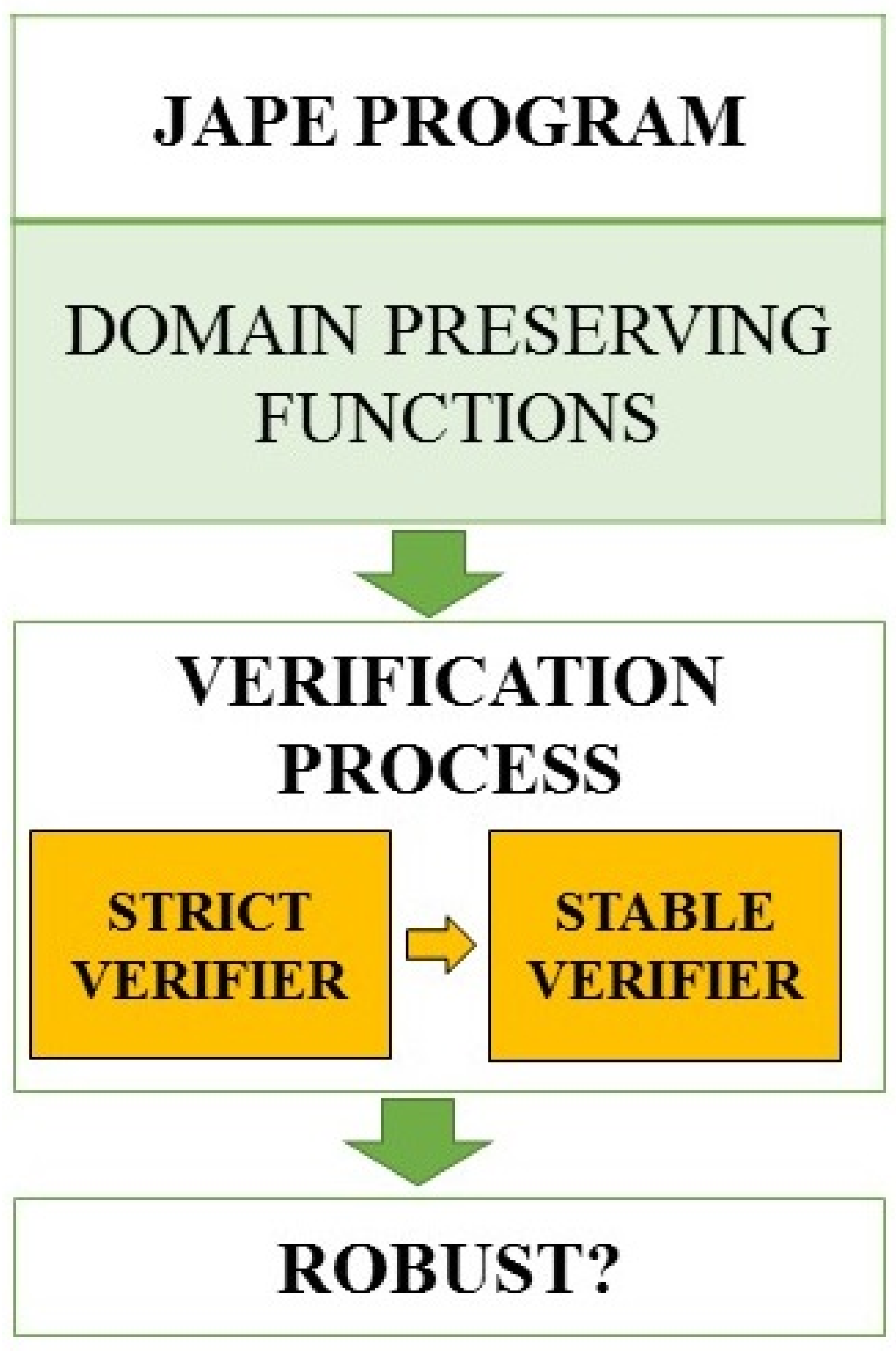

Figure 3: The verifier statically analyses regular expressions to determine whether an extractor is stable.

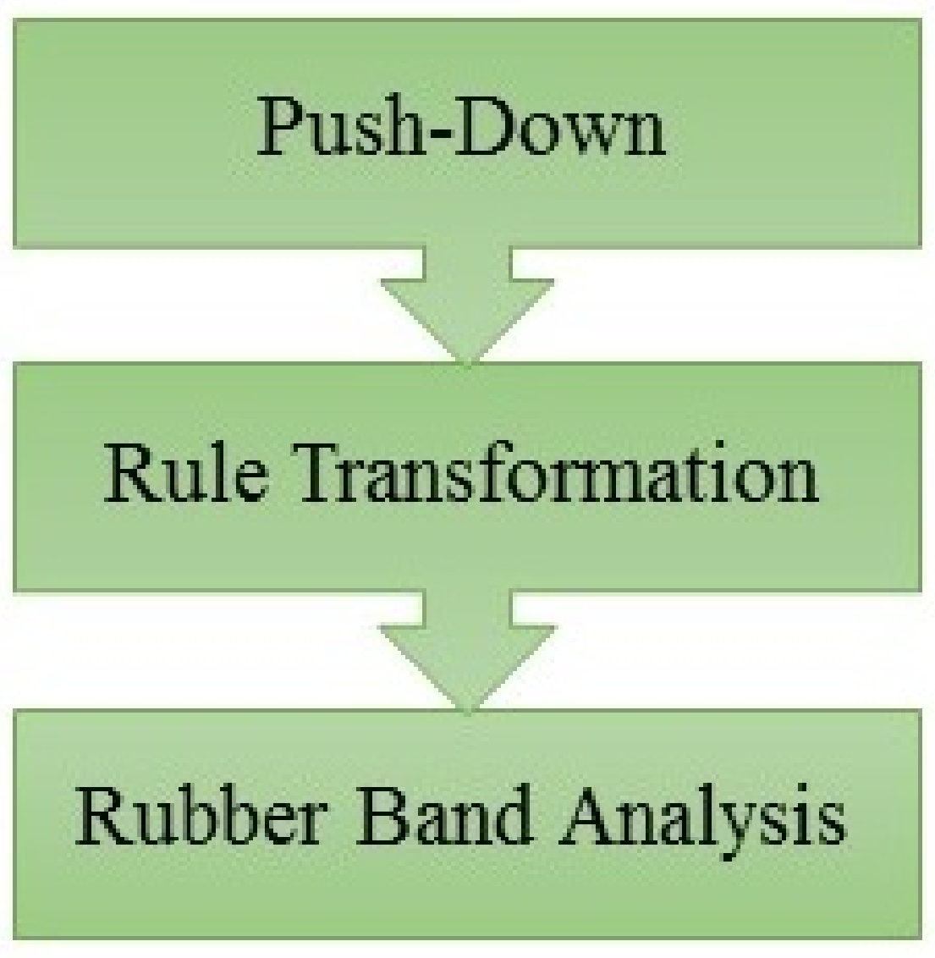

Figure 4: Main steps of the JAPE verifier.

Results

- Every JAPE program that contains only simple rules has the strict property.
- Every strict JAPE program is computable.
- A JAPE program that does not have any domain inconsistency or problematic overlaps is stable.

Conclusion

- We introduce the extracted view update problem.
- We formalize the notion of robust extraction algorithms.
- We propose three properties for robust extractors.
- We design an algorithm that modifies the input document. The modified document can be fed into the extractor to produce the updated extracted view.
- We present the essentials for designing a verification process for robustness of programs written in a significant subset of JAPE language.

References

[1] Sunita Sarawagi.
Information extraction.
Foundations and Trends in DB, 1(3):261–377, 2008.

[2] Hamish Cunningham, Diana Maynard, and Valentin Tablan.
JAPE: a Java annotation patterns engine.
Technical Report CS-00-10, Univ. Sheffield, 2000.

Contact Information

bkassaie@uwaterloo.ca
https://www.linkedin.com/in/besatkassaie/

fwtompa@uwaterloo.ca
https://cs.uwaterloo.ca/ fwtompa/

This work has been accepted to be presented at the 19th ACM Symposium on Document Engineering, September 2019, Berlin, Germany