**INTRODUCTION**

The following transducer $T$:

```
... 2  3
 1  a : A/2  4  b : B/1
 2  c : D/4
```

- is ambiguous because it contains repeated inputs (i.e. “a b”).
- is non-functional because these input paths emit different outputs (i.e. “A B” and “A V”).

**CONTRIBUTION**

A new, more efficient, disambiguation algorithm for non-functional WFSTs.

**NON-FUNCTIONAL FST DISAMBIGUATION**

(a) Map the WFST into an equivalent weighted finite-state automata (WFSA) using weights that contain both the WFST weight and output symbols (using a special semiring)

(b) Apply WFSA determination under this semiring to ensure that only one unique path per input string survives

(c) Expand the result back to an WFST that preserves arc-level alignments

**TROPICAL SPARSE VECTOR SEMIRING**

Assume cost $c$ is a linear combination of features:

```
Tropical Semiring  Tropical Sparse Vector Semiring
```

- Allows to keep track of individual features. Note the sparse vector notation: (feature-id, value) pairs.
- Useful for lattice-based translation (HiFST), optimization (Lattice MERT) and other algorithms.

**WFSA WITH SPARSE TOPOLOGICAL FEATURES**

- With this semiring we can now use sparse Topological Features $f_j$ to identify arcs in a topologically sorted transducer. Arc $i$ fires $f_i = 1$, $f_i = 0$ otherwise.
- Assume $\omega = (1, 0, ..., 0)$, $c = f_0$.
- Consider all arcs $j$ in a path, it holds that $\sum f_j = 1$.

Map $T$ to acceptor $A$ with topological features:

- $a/((0, 2), (1, 1)]$
- $b/((0, 1), (3, 1)]$
- $c/((0, 5), (1, 1) - (2, 1), (5, 1)]$

- The topology is "over-determined"
- 1-1 topological-feature to output label mappings is no longer possible!

**EXPANSION ALGORITHM**

```plaintext
Take $A' = Reverse\left( A^0 \right)$:
```

```
1 a/((0, 2), (1, 1)]
2 b/((0, 1), (3, 1)]
3 c/((0, 5), (1, 1) - (2, 1), (5, 1)]
```

Run Expansion algorithm on $A'$:

**DISAMBIGUATION EFFICIENCY**

Number of successfully disambiguated transducers over time for PoS tagged (left) and HiFST(right) lattices, versus categorial disambiguation (Shafran et al., 2011; Sproat et al., 2014), versus OpenFst 1.4.1.

**USE CASE IN MT: RESCORING WITH BILINGUAL MODELS**

- HiFST in alignment mode can generate WFSTs with edges of the form $t_3/w$,
- Standard Hiero Grammar augmented with affiliations (Devlin et al., 2014):

$$X \rightarrow < s_1 X s_2 s_3 t_1 X t_2 > / 2.1$$

MT Rescoring Pipeline:
- Generate WFSTs with affiliation
- Disambiguate WFSTs.
- Apply bilingual composition
- Run Lattice MERT