

The *reversal* of a word $w = a_0a_1\cdots a_n$, written w^R , is $a_n\cdots a_1a_0$. The reversal of a language $L \subseteq \Sigma^*$ is $L^R = \{w^R \mid w \in L\}$.

Brzowski's automata minimization algorithm [Brz63] is based on two successive reverse and determinization operations and the entire algorithm is presented (in one single line!) on Algorithm 1.2. This algorithm is the only one able to process both DFAs and NFAs, always yielding a minimal DFA that accepts the same language as the input automaton. Given an automaton \mathcal{A} , let **det** be the subset construction method and **rev** to the procedure that computes an NFA that accepts $L(\mathcal{A})^R$. Brzowski's algorithm can be stated as a simple function composition of the previous procedures: $\text{det} \cdot \text{rev} \cdot \text{det} \cdot \text{rev}$. Since the *rev* method can be applied to any finite automaton, perhaps with non-determinism, this algorithm is able to minimize both DFAs and NFAs.

```
def brzowski(fa):
    return det(rev(det(rev(fa))))
```

Algorithm 1.2. Brzowski's algorithm.