Kleene Theorem Generalized to Regular Word Transformations or a Notion of Regular Expression for String Transformations

Regular languages bridge many different fundamental models: Kleene established the equiexpressiveness of finite state automata and regular expressions, Bchi showed that MSO logic was also equiexpressive, and Schtzenberger did it for recognizability by finite monoids.

There exists a model of regular transformations that is also definable both in logics (through Courcelle's MSO graph transformations restricted to words), and through computationalautomata-like models (deterministic two-way transducers) and streaming string transducers. There is however no natural extension of regular expressions to define string transformations, that would capture exactly that class of regular transformations. Some directions are worth investigating:

- Perl is a programming language which places regular expressions at its core. Perl "regular expressions" have this particularity that they are more expressive than regular expressions. Perl can also define some regular string transformations. One approach would be to determine what misses from these Perl definable transformations so as to capture all regular string transformations. Some implementation of an equivalence tester for these transforming regular expressions could be appealing.

- A more theoretic approach could also be considered by extending the classical automata-regular expression translation (by successive state removing) to a translation for streaming string transducers.