Motivation

- We are interested in studying the behaviour of systems (potentially infinite)
- Coalgebraic framework is suitable
- Our most tool related research: regular expressions
- Regular expressions are a powerful tool in specification: they are the basis of PSL (Intel and IBM).
- Generalization to other types of systems together with algebraic counterpart.
- Deciding equivalence of regular expressions is essential: algebraically or coalgebraically.

Motivation

- We are interested in studying the behaviour of systems (potentially infinite)
- Coalgebraic framework is suitable
- Our most tool related research: regular expressions
- Regular expressions are a powerful tool in specification: they are the basis of PSL (Intel and IBM).
- Generalization to other types of systems together with algebraic counterpart.
- Deciding equivalence of regular expressions is essential: algebraically or coalgebraically.

Motivation

- We are interested in studying the behaviour of systems (potentially infinite)
- Coalgebraic framework is suitable
- Our most tool related research: regular expressions
- Regular expressions are a powerful tool in specification: they are the basis of PSL (Intel and IBM).
- Generalization to other types of systems together with algebraic counterpart.
- Deciding equivalence of regular expressions is essential: algebraically or coalgebraically.

Motivation

- We are interested in studying the behaviour of systems (potentially infinite)
- Coalgebraic framework is suitable
- Our most tool related research: regular expressions
- Regular expressions are a powerful tool in specification: they are the basis of PSL (Intel and IBM).
- Generalization to other types of systems together with algebraic counterpart.
- Deciding equivalence of regular expressions is essential: algebraically or coalgebraically.

Motivation

- We are interested in studying the behaviour of systems (potentially infinite)
- Coalgebraic framework is suitable
- Our most tool related research: regular expressions
- Regular expressions are a powerful tool in specification: they are the basis of PSL (Intel and IBM).
- Generalization to other types of systems together with algebraic counterpart.
- Deciding equivalence of regular expressions is essential: algebraically or coalgebraically.

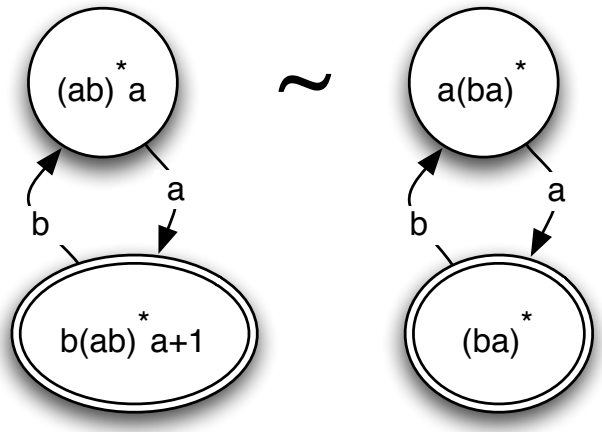
Coalgebraic vs algebraic

Theorem

Sliding Rule

$$(ab)^* a = a(ba)^*$$

Coalgebraic



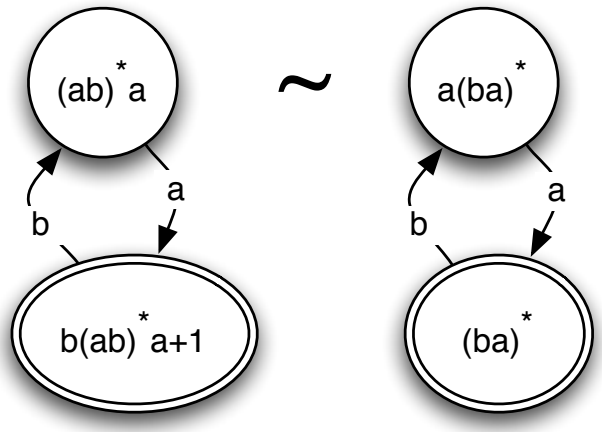
Coalgebraic vs algebraic

Theorem

Sliding Rule

$$(ab)^* a = a(ba)^*$$

Coalgebraic



Algebraic

$$\begin{aligned}
& (ab)^* a \leq a(ba)^* \\
\iff \quad & a + (ab)a(ba)^* \leq a(ba)* \\
\iff \quad & a + a(ba)(ba)^* \leq a(ba)* \\
\iff \quad & a(1 + ba(ba))^* \leq a(ba)* \\
\iff \quad & a(1 + (ba))^* \leq a(ba)* \\
\iff \quad & a(ba)^* \leq a(ba)*
\end{aligned}$$

and similarly
$a(ba)^* \leq (ab)^* a$.

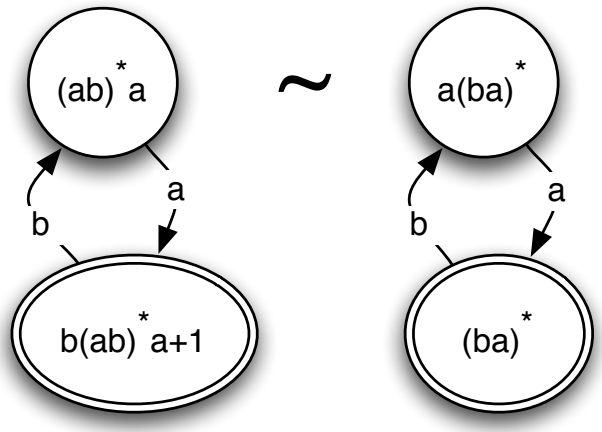
Coalgebraic vs algebraic

Theorem

Sliding Rule

$$(ab)^* a = a(ba)^*$$

Coalgebraic semantic



Algebraic syntactic

$$\begin{aligned}
& (ab)^* a \leq a(ba)^* \\
\iff \quad & a + (ab)a(ba)^* \leq a(ba)* \\
\iff \quad & a + a(ba)(ba)^* \leq a(ba)* \\
\iff \quad & a(1 + ba(ba))^* \leq a(ba)* \\
\iff \quad & a(1 + (ba))^* \leq a(ba)* \\
\iff \quad & a(ba)^* \leq a(ba)*
\end{aligned}$$

and similarly
$a(ba)^* \leq (ab)^* a$.

Coalgebraic or algebraic

- Both the coalgebraic and the algebraic approach have (dis)advantages
- Combining both results in powerful proof techniques
- Collaboration with Dorel Lucanu (`Romania`) and Grigore Rosu (`USA`) :

`CIRC: a (co)inductive prover`

Example

```
while b
  p
  while c
    q
  endwhile
endwhile
```

$b(p(cq)^*\overline{c})^*\overline{b}$

```
if b
  then p;
    while (c or b)
      if c
        then q
        else p
      endif
    endwhile
endif
```

$bp((cq) + b\overline{c}p)^* + \overline{b}$