

Propagating Regular Counting with Lazy Clause Generation

DFA vs counter-DFA for finite domain satisfaction problems

Daniël Ravensbergen¹
 Supervisors: Emir Demirović¹, Imko Marijnissen¹
¹EEMCS, Delft University of Technology, The Netherlands
 E-mail: d.g.s.ravensbergen@student.tudelft.nl

1) Introduction

Regular counting¹ is a feature of **regular** languages useful for modeling problems like:

- Sequencing
- Time-tabling

It can be modeled for **constraint solvers** with constraints based on:

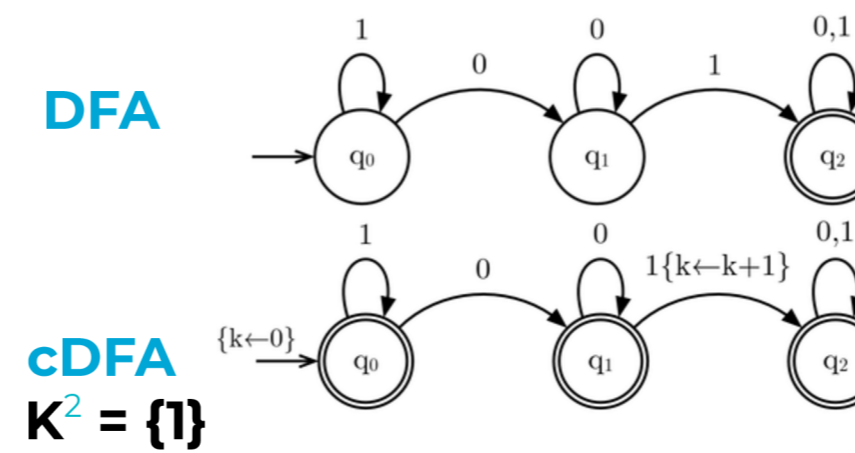
- Deterministic Finite Automata (**DFA**)
- Counter Deterministic Finite Automata (**cDFA**)

cDFA-based has been shown to find solutions more efficiently for finite domain satisfaction problems, but it was unknown if this is still the case when using Lazy Clause Generation (**LCG**).

We created a **cDFA**-based propagator for **LCG** and compared it to **DFA**-based approaches.

2) Counter-DFA

Uses a **counter** to decide when to accept an input instead of accepting states.



3) Lazy Clause Generation

Uses **explanations** to learn from conflicts.

Consider $A = B$, where $A \in \{1, 2, 3\}$, $B \in \{2, 3\}$

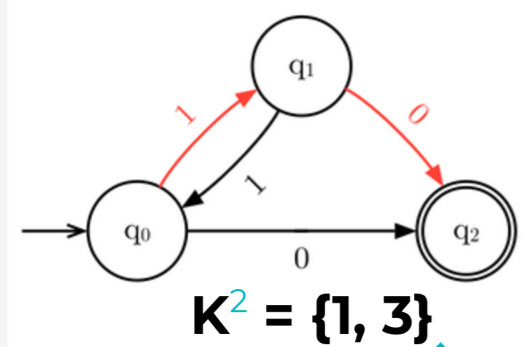
Updating the domain (**propagation**):

[A ≥ 2]

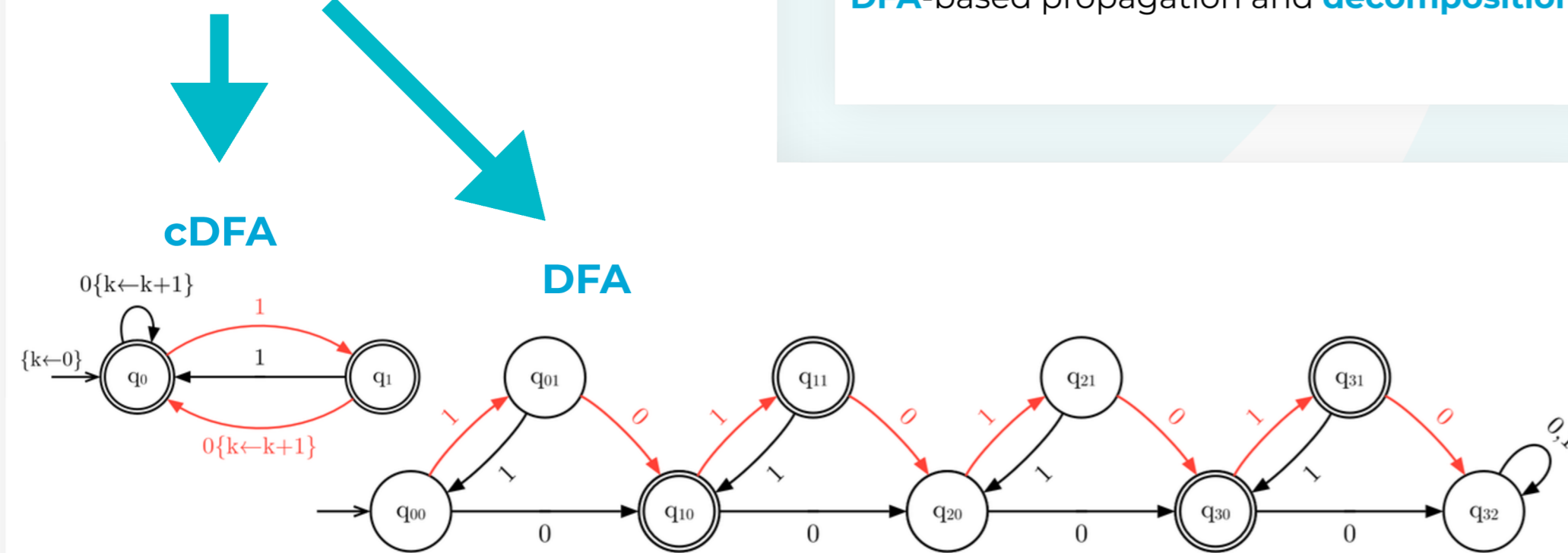
And giving the **reason**:

[B ≥ 2]

4) Generating Equivalent Automata



Intermediate Automaton

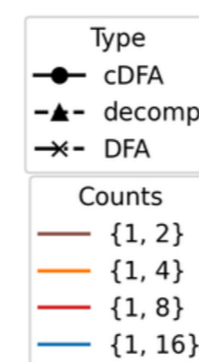


cDFA is more compact

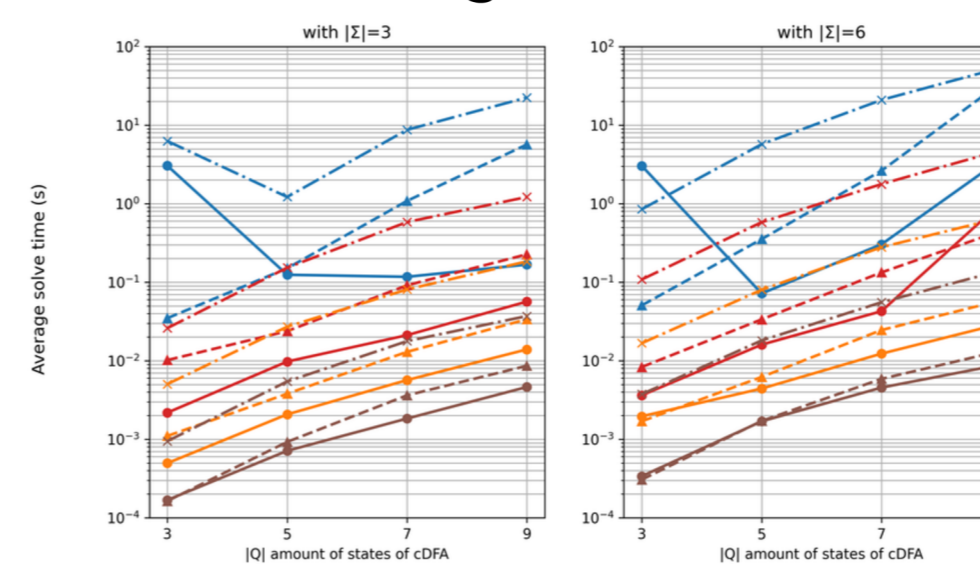
5) Results

We created equivalent random finite domain satisfaction problems using our **Generator**.

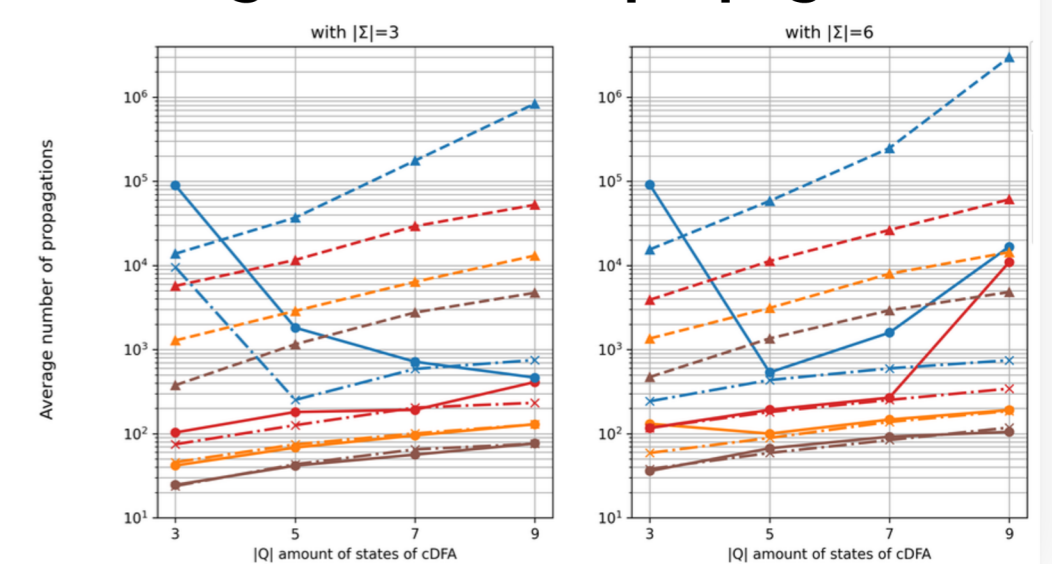
We compared our **cDFA**-based propagator to **DFA**-based propagation and **decomposition**³.



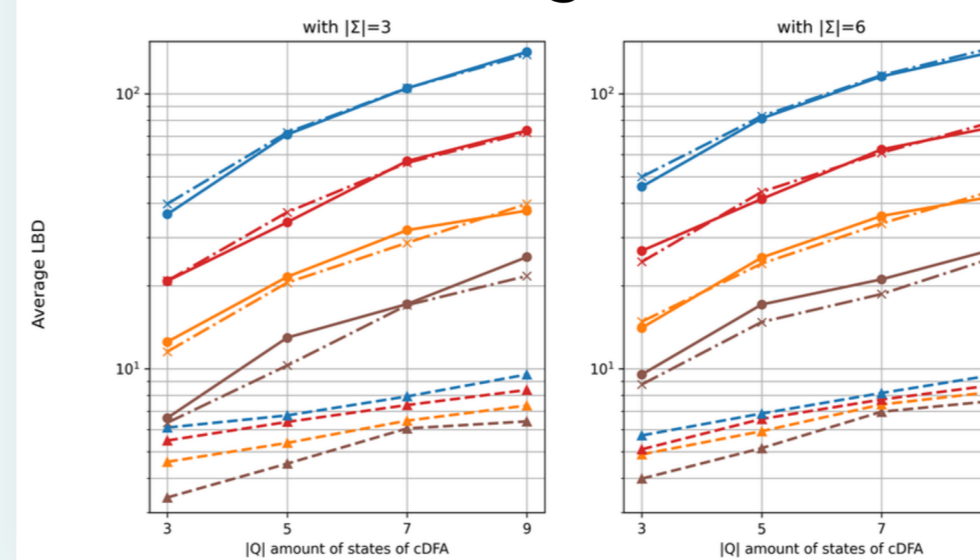
Average Solve Time



Average number of propagations



Average LBD⁴



6) Conclusion

Generally **solves faster**, but has **worse explanations**.

We set a **baseline** for future research to improve.

Footnotes:

1. Regular counting: How many times a given regular pattern occurs. You may know it from RegEx, where counting n times is represented by "{n}".
2. K: the set of accepted counts.
3. Decomposition: Breaking up a constraint into many simpler ones.
4. LBD: Literal Block Distance, a measure of quality for clauses learned from conflict learning. The lower the better.