Progress on Developing a Uni-Prover for Program Verification: Theorem Application

RESOLVE Workshop

18th May 2023

Denison University, Granville, Ohio

Nicodemus Msafiri Mbwambo

ARUSHA TECHNICAL COLLEGE

'Skills make the Difference'

Uni-Prover: A Universal Automated Prover for Specificationally Rich Languages

- Specificationally rich languages allow development and use of arbitrary mathematical theories.

- They provide suitable mathematical models and notations for describing components in optimally insightful ways.

Verification Support for Rich Languages

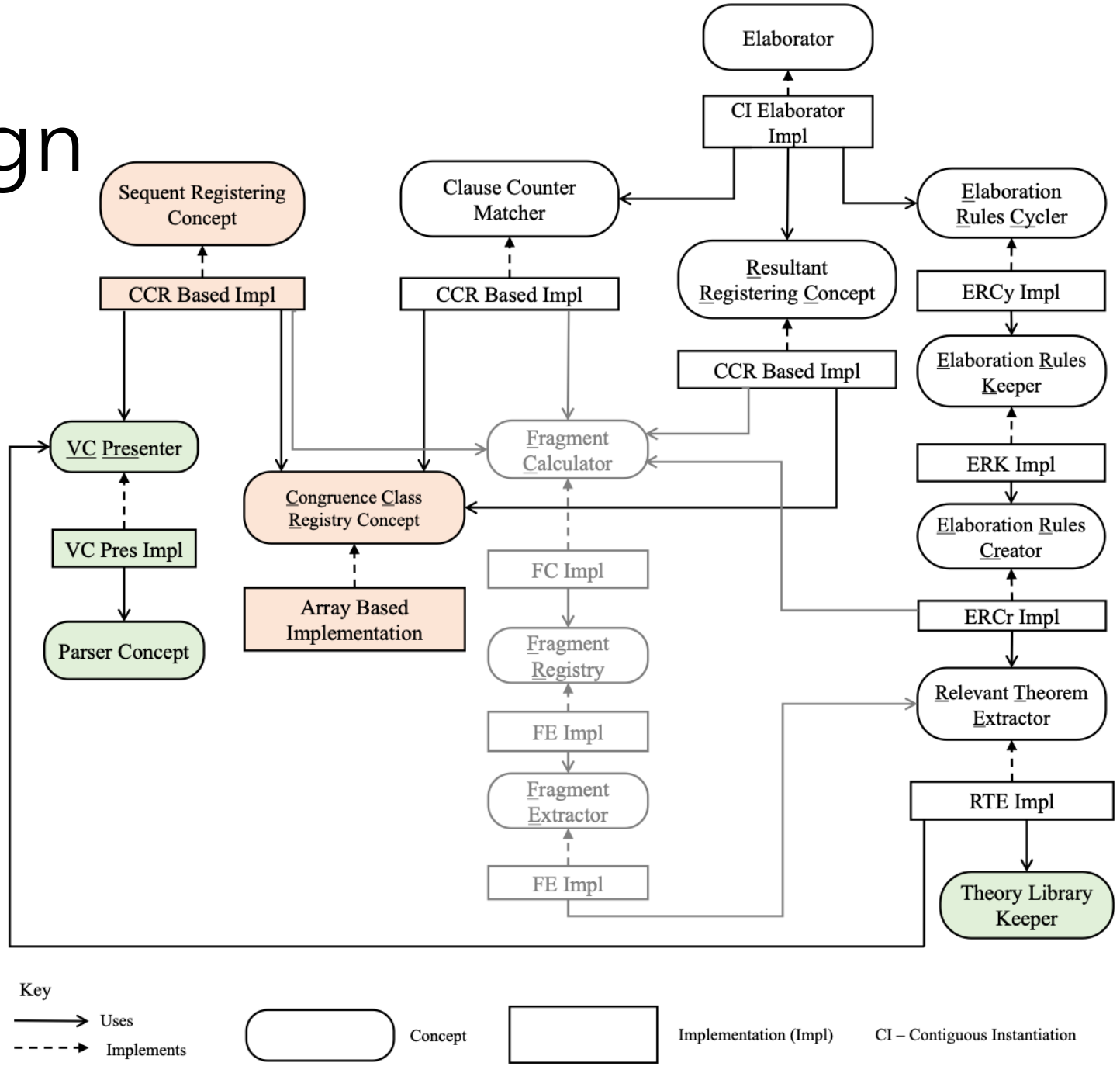
- A fully automated prover must be able to employ results (e.g., theorems) from arbitrary mathematical domains.

- The prover should employ strategies that attempt to minimize unproductive verification steps in order to prove as many VCs as possible.

Uni-Prover Design

- Adequate handling of equality.
- Minimizing the number of steps taken in the verification process.
  - Effective theorem instantiation process.
  - Effective searching strategies.
- Maintaining a uniform performance when dealing with existing and new theorems from the theory library.

Uni-Prover Design



Experimentation Results

```
Realization Do_Nothing_Realiz for
Do_Nothing_Capability of Stack_Template;
        Procedure Do_Nothing(restores S: Stack);
                Var Next_Entry: Entry;
                Pop(Next_Entry, S);
                Push(Next_Entry, S);
        end Do_Nothing;
end Do_Nothing_Realiz;
```

[VC Results]

VC 0_1	.....	proved in 5 ms.
VC 0_2	.....	not proved in 2 ms.
VC 0_3	.....	proved in 1 ms.

Proved VCs

VC 0_1:
Requires Clause for Pop at Do_Nothing_Realiz.rb(5:2)

Goal:
(1 <= |S|)

Given(s):
1. (1 <= |S|)

[Prover Result]: Proved

VC 0_3:
Ensures Clause of Do_Nothing (Condition from "RESTORES" parameter mode) at Do_Nothing_Realiz.rb(2:31)

Goal:
((<Next_Entry''> o S'') = S)

Given(s):
1. (S = (<Next_Entry''> o S''))
2. (1 <= |S|)

[Prover Result]: Proved

Unproved VC

VC 0_2:
Requires Clause for Push at Do_Nothing_Realiz.rb(6:2)

Goal:
((1 + |S''|) <= Max_Depth)

Given(s):
1. (S = (<Next_Entry''> o S''))
2. (1 <= Max_Depth)
3. (1 <= |S|)

[Prover Result]: Not Proved

Other Results

Recursive realization for Position_Depth_Capability of Exploration_Tree_Template	40 VCs	[Aggregate Results] Total number of proved VCs: 15 Total number of unproved and skipped VCs: 25 Total elapsed time in generating proofs: 381 ms
Grid positioning realization for Grid_Positioning_Template	76 VCs	[Aggregate Results] Total number of proved VCs: 56 Total number of unproved and skipped VCs: 20 Total elapsed time in generating proofs: 418 ms

Potential Theorems