Guiding Constraint Horn Clauses Solving using Graph Neural Networks

Parosh Aziz Abdulla[1], Chencheng Liang[1], Philipp Rümmer[1,2]

[1] Uppsala University, Sweden
[2] University of Regensburg, Germany

May 07 2024

UPPSALA UNIVERSITET

Universität Regensburg

Verification and Deep Learning

- Verify neural networks
- Apply deep learning to improve verification

Verification and Deep Learning

- Verify neural networks
- Apply deep learning to improve verification (formal methods)

Verification and Deep Learning

- Verify neural networks.
- Apply deep learning to improve verification (examples)
  - Premise selection for Automatic Theorem Provers (ATPs)
  - Variable branching decision for SAT solvers
  - Instance selection in SMT solving
  - Algorithm selection for software verification

Verification and Deep Learning

- Verify neural networks.
- Apply deep learning to improve verification.
    - Decision making problems (classification task)

What to learn

- Apply deep learning to improve verification (formal methods)
    - Learn ?

What to learn

- Apply deep learning to improve recommendation system
  - Learn relations

What to learn

- Apply deep learning to improve $\boxed{\text{language translation}}$
  - Learn $\boxed{\text{semantics}}$

What to learn

- Apply deep learning to improve $\boxed{\text{protein structure prediction}}$
    - Learn $\boxed{\text{isomorphism}}$

What to learn

- Apply deep learning to improve verification (formal methods)
  - Learn ?

What to learn

- Apply deep learning to improve verification (formal methods)
    - Learn reasoning

Background

- Program verification

Program verification (example)

```
int x = _ ;
while (x > 0){
        x--;
}
if (x != 0){ error(); }
```

- Whether exists a path that leads to the error state

Background

- Program verification
- Encode program verification to Constraint Horn Clauses (CHCs)
  - Solving the CHCs is solving the program verification problem

Constraint Horn Clauses (CHCs)

A CHC is a formula in the format

$$\forall V.\, L[X] \leftarrow L_1[X_1] \wedge \ldots \wedge L_n[X_n] \wedge \varphi$$

Where

V are variables,

X_i are terms over V,

$L, L_1, \ldots, L_n$ are n-ary relation symbols,

$L_i[X_i]$ is an atom of relation symbol to the terms,

φ is a constraint in the background theory T.

A set of CHCs (example)

A CHC is a formula in the format

$$\forall V.\, L[X] \leftarrow L_1[X_1] \wedge \ldots \wedge L_n[X_n] \wedge \varphi$$

$$
\begin{aligned}
L_1(x) &\leftarrow true \\
L_2(x) &\leftarrow L_1(x) \wedge x > 0 \\
L_1(x') &\leftarrow L_2(x) \wedge x' = x - 1 \\
L_3(x) &\leftarrow L_1(x) \wedge x <= 0 \\
false &\leftarrow L_3(x) \wedge x \neq 0
\end{aligned}
$$

Background

- Program verification
- Constraint Horn Clauses (CHCs)
- Encode program verification to CHCs

A program and its CHCs (example)

```
int x = _ ;
while (x > 0){
      x--;
}
if (x != 0){ error(); }
```

$$
\begin{aligned}
L_1(x) &\leftarrow \mathit{true}\\
L_2(x) &\leftarrow L_1(x) \land x > 0\\
L_1(x') &\leftarrow L_2(x) \land x' = x - 1\\
L_3(x) &\leftarrow L_1(x) \land x <= 0\\
\mathit{false} &\leftarrow L_3(x) \land x \neq 0
\end{aligned}
$$

Background

- Program verification
- Constraint Horn Clauses (CHCs)
- Encode program verification to CHCs
- Solving CHCs

A program and its CHCs (example)

```
int x = _ ;
while (x > 0){
      x--;
}
if (x != 0){ error(); }
```

$$L_1(x) \leftarrow true$$
$$L_2(x) \leftarrow L_1(x) \wedge x > 0$$
$$L_1(x') \leftarrow L_2(x) \wedge x' = x - 1$$
$$L_3(x) \leftarrow L_1(x) \wedge x <= 0$$
$$false \leftarrow L_3(x) \wedge x \neq 0$$

A path to error

Cannot find interpretations to atoms to make the set of CHCs true

Background

- Techniques for solving CHCs
  - Counterexample-guided abstraction refinement (CEGAR)
  - Symbolic execution based technique

Background

- Which CHC is processed first in a set of CHCs is important
  - General ranking problem

Background

- CHC selection
  - Premise selection for Automatic Theorem Provers (ATPs)
  - Variable branching decision for SAT solvers
  - Instance selection in SMT solving
- Classification task with various input and output

Background

- Examples of prioritizing (ranking) CHCs
  - The fewer dependencies the higher priority
  - Solving simpler CHCs outside of cycles may reduce complexity within the cycles or overall problem space
  - Domain specific heuristics: in program verification, clauses representing base cases in recursive functions might be simpler to solve.

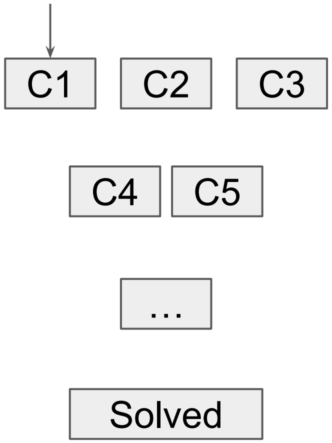
Motivation

- Target: data-driving method to prioritize CHCs (deep learning)
- Challenge: hard to form training data

C1	C2	C3

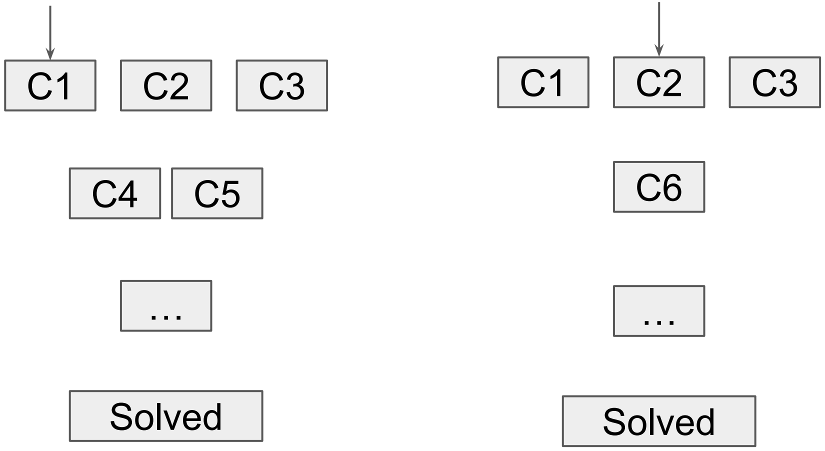
Motivation

- Target: data-driving method to prioritize CHCs (deep learning)
- Challenge: hard to form training data



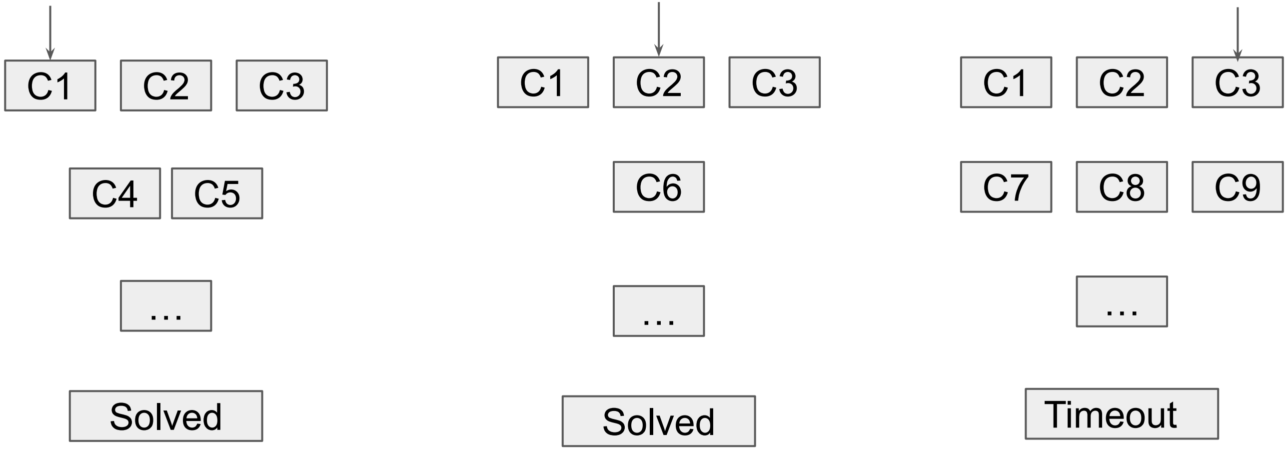
Motivation

- Target: data-driving method to prioritize CHCs (deep learning)
- Challenge: hard to form training data



Motivation

- Target: data-driving method to prioritize CHCs (deep learning)
- Challenge: hard to form training data



Motivation

- Target: data-driving method to prioritize CHCs
- Challenge: hard to form training data
- Idea: focus on learning a particular concept
  - Minimal Unsatisfiable Subsets (MUSes)

Minimal Unsatisfiable Subsets (MUSes) of CHCs

$$
\begin{aligned}
&[1]\ L_1(x) && \leftarrow \mathit{true} \\
&[2]\ L_2(x) && \leftarrow L_1(x) \wedge x > 0 \\
&[3]\ L_1(x') && \leftarrow L_2(x) \wedge x' = x - 1 \\
&[4]\ L_3(x) && \leftarrow L_1(x) \wedge x \leq 0 \\
&[5]\ \mathit{false} && \leftarrow L_3(x) \wedge x \neq 0
\end{aligned}
$$

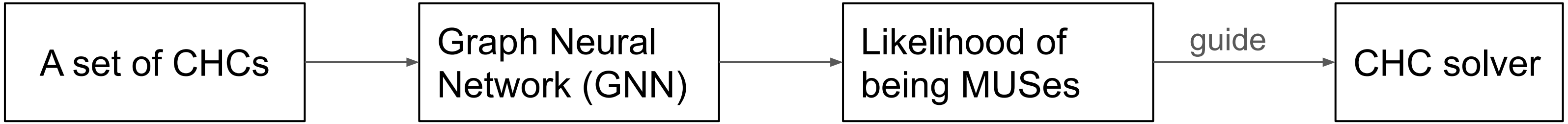
Minimal Unsatisfiable Subsets (MUSes) of CHCs

$$
\begin{aligned}
&[1]\ L_1(x) &&\leftarrow true \\
&[2]\ L_2(x) &&\leftarrow L_1(x) \wedge x > 0 \\
&[3]\ L_1(x') &&\leftarrow L_2(x) \wedge x' = x - 1 \\
&[4]\ L_3(x) &&\leftarrow L_1(x) \wedge x \leq 0 \\
&[5]\ false &&\leftarrow L_3(x) \wedge x \neq 0
\end{aligned}
$$

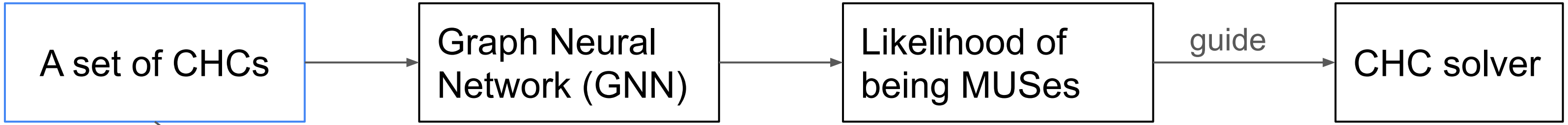
{[1], [4], [5]} is the only MUS

Property: If any subset of the set of CHCs is UNSAT, then the entire set of CHCs is also UNSAT.

Deep Learning-Based Framework (prediction phase)

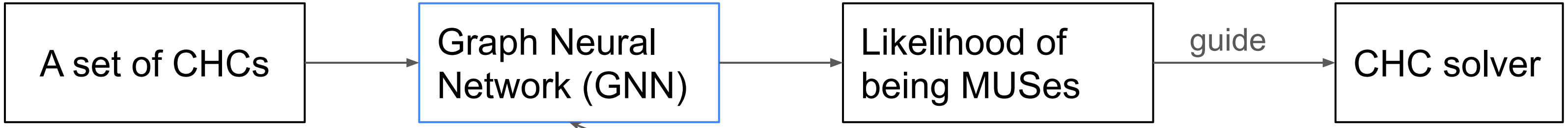


Prediction phase



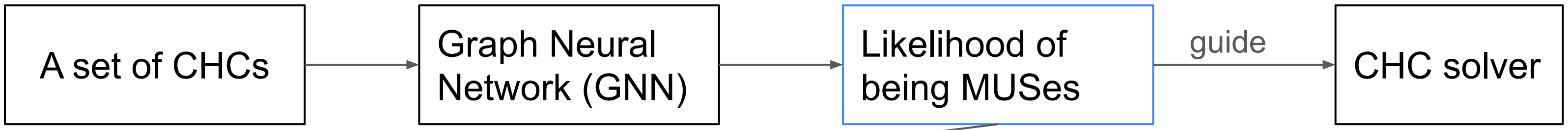
$$
\begin{aligned}
&[1]\ L_1(x) && \leftarrow \mathit{true}\\
&[2]\ L_2(x) && \leftarrow L_1(x) \wedge x > 0\\
&[3]\ L_1(x') && \leftarrow L_2(x) \wedge x' = x - 1\\
&[4]\ L_3(x) && \leftarrow L_1(x) \wedge x \leq 0\\
&[5]\ \mathit{false} && \leftarrow L_3(x) \wedge x \neq 0
\end{aligned}
$$

Prediction phase



$$
\begin{array}{lll}
[1] & L_1(x) & \leftarrow true \\
[2] & L_2(x) & \leftarrow L_1(x) \land x > 0 \\
[3] & L_1(x') & \leftarrow L_2(x) \land x' = x - 1 \\
[4] & L_3(x) & \leftarrow L_1(x) \land x \leq 0 \\
[5] & false & \leftarrow L_3(x) \land x \neq 0
\end{array}
$$

Prediction phase



Score		
0.8	$[1]\ L_1(x)$	$\leftarrow true$
0.2	$[2]\ L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0.1	$[3]\ L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
0.75	$[4]\ L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
0.6	$[5]\ false$	$\leftarrow L_3(x) \wedge x \neq 0$

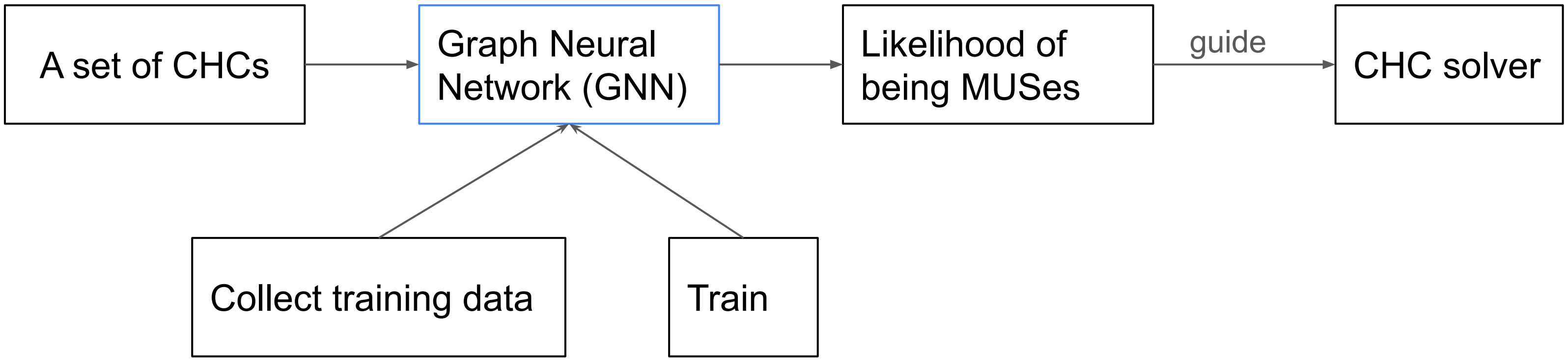
Prediction phase

A set of CHCs → Graph Neural Network (GNN) → Likelihood of being MUSes $\xrightarrow{\text{guide}}$ CHC solver

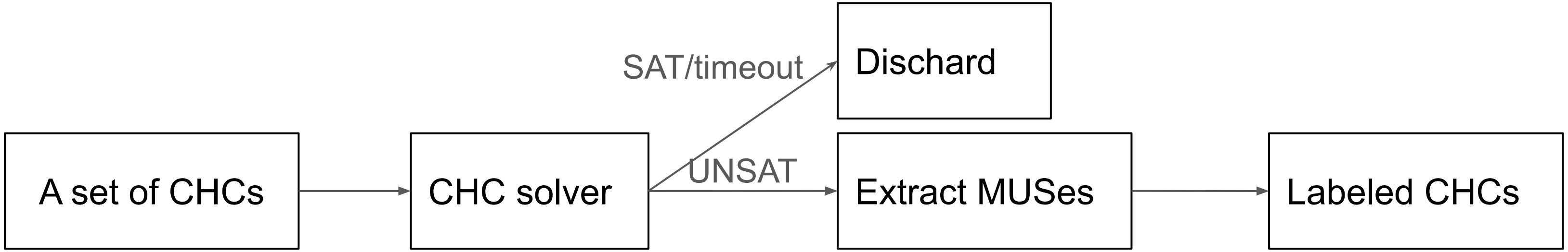
Score		Head		Body
0.8	[1]	$L_1(x)$	$\leftarrow$	$true$
0.2	[2]	$L_2(x)$	$\leftarrow$	$L_1(x) \land x > 0$
0.1	[3]	$L_1(x')$	$\leftarrow$	$L_2(x) \land x' = x - 1$
0.75	[4]	$L_3(x)$	$\leftarrow$	$L_1(x) \land x \leq 0$
0.6	[5]	$false$	$\leftarrow$	$L_3(x) \land x \neq 0$

Prioritized CHCs:
[1], [4], [5], [2], [3]

Deep Learning-Based Framework



Training phase (collect training data)

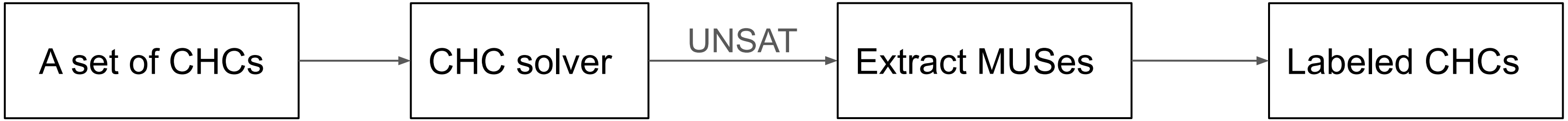


Training phase (collect training data)

A set of CHCs → CHC solver $\xrightarrow{\text{UNSAT}}$ Extract MUSes → Labeled CHCs

$$
\begin{array}{lll}
 & \text{Clauses} & \\
[1] & L_1(x) & \leftarrow \mathit{true} \\
[2] & L_2(x) & \leftarrow L_1(x) \land x > 0 \\
[3] & L_1(x') & \leftarrow L_2(x) \land x' = x - 1 \\
[4] & L_3(x) & \leftarrow L_1(x) \land x \leq 0 \\
[5] & \mathit{false} & \leftarrow L_3(x) \land x \neq 0
\end{array}
$$

Training phase (collect training data)



Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow \mathit{true}$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] false	$\leftarrow L_3(x) \wedge x \neq 0$

Training phase (collect training data)

A set of CHCs → CHC solver $\xrightarrow{\text{UNSAT}}$ Extract MUSes → Labeled CHCs

Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \wedge x \neq 0$

When there are multiple MUSes

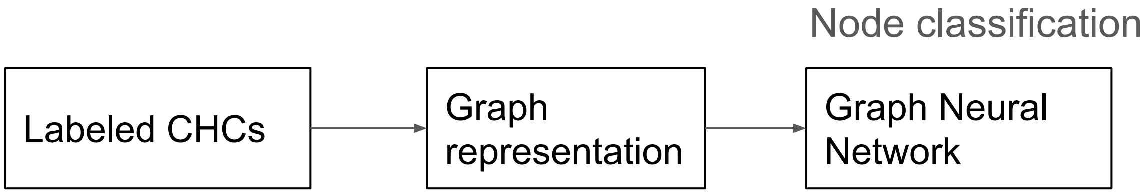
- Union
- Intersection
- Single

Training phase (train a model)

Labeled CHCs → Graph representation → Graph Neural Network

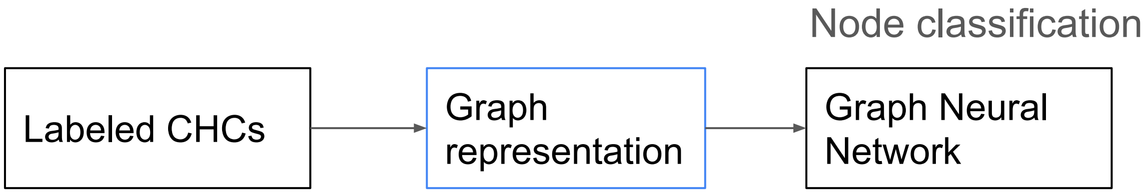
Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \wedge x \neq 0$

Training phase (train a model)



Label		Clauses	
1	[1]	$L_1(x)$	$\leftarrow true$
0	[2]	$L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3]	$L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4]	$L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5]	$false$	$\leftarrow L_3(x) \wedge x \neq 0$

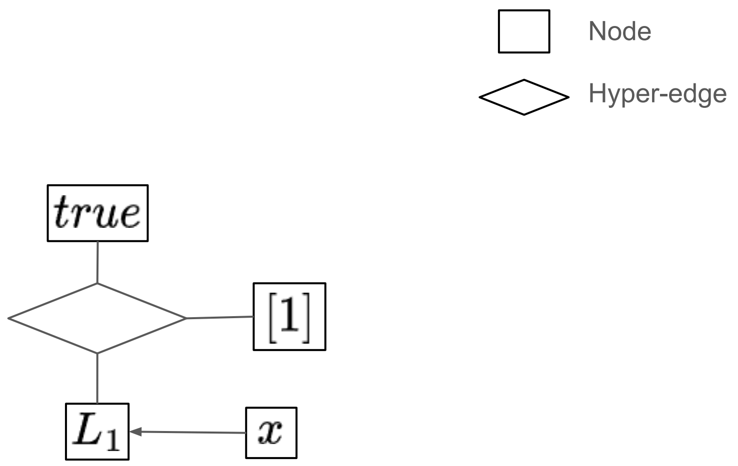
Training phase (train a model)



Label		Clauses
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \wedge x \neq 0$

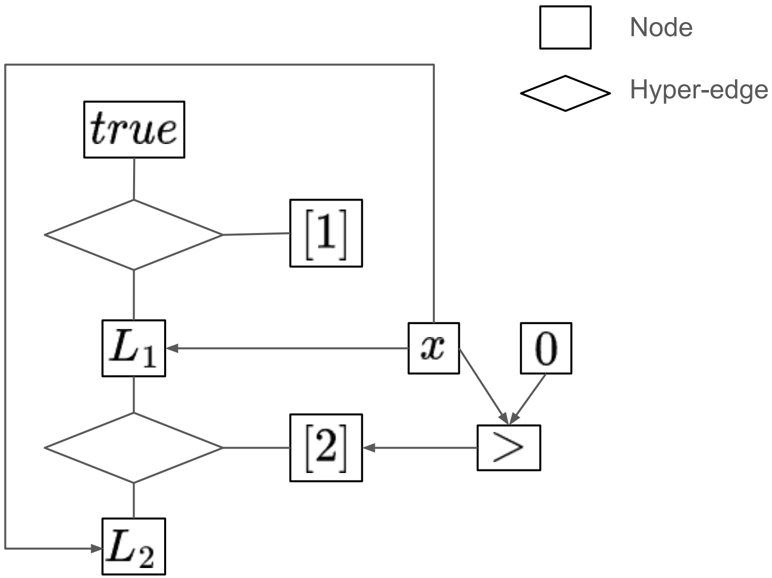
Represent CHCs by graph (example)

Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \wedge x \neq 0$



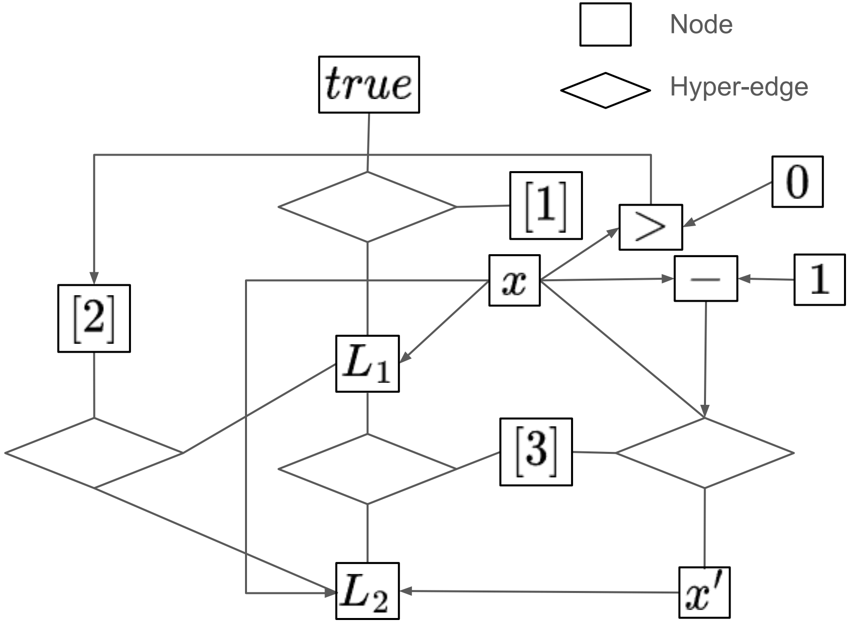
Represent CHCs by graph (example)

Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \land x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \land x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \land x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \land x \neq 0$



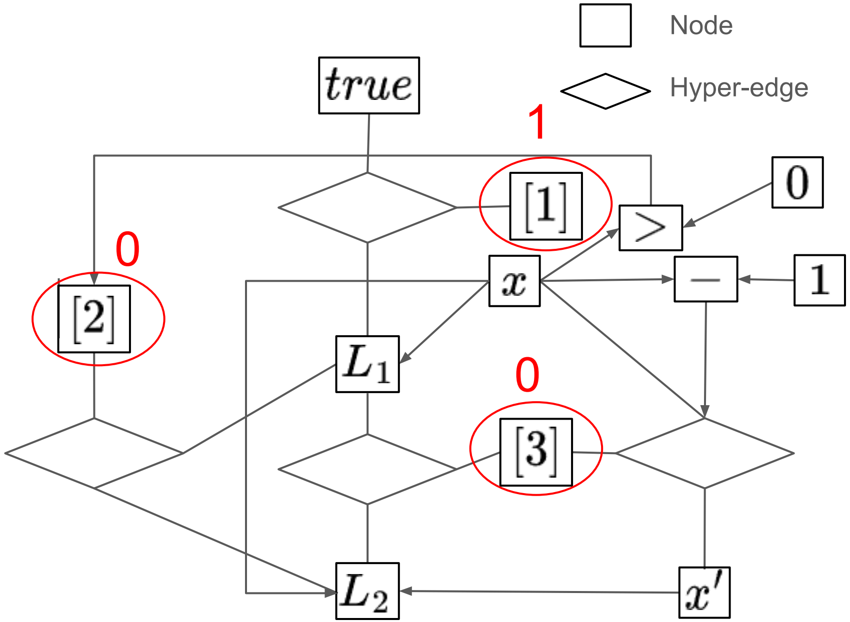
Represent CHCs by graph (example)

Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \wedge x \neq 0$

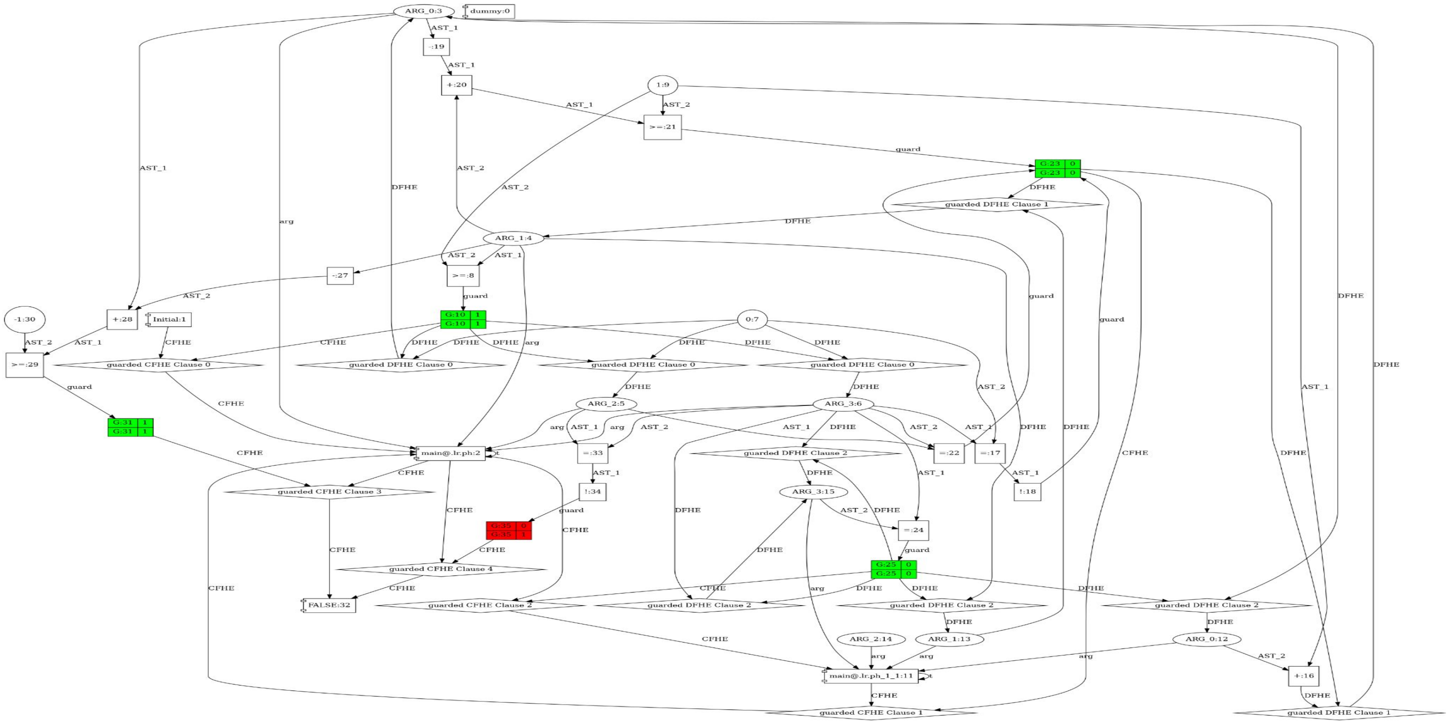


Represent CHCs by graph (example)

Label		Clauses
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \wedge x \neq 0$



Represent CHCs by graph (example from benchmark)



Training phase (train a model)

Node classification

Labeled CHCs → Graph representation → Graph Neural Network

Training phase (train a model)

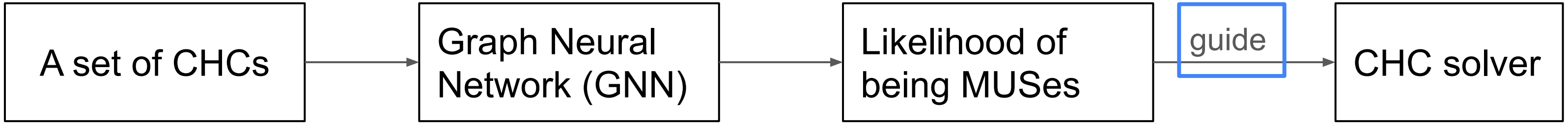
Node classification

Labeled CHCs → Graph representation → Graph Neural Network

- Relational Hypergraph Neural Network [1]
    - Can handle different types of hyperedges

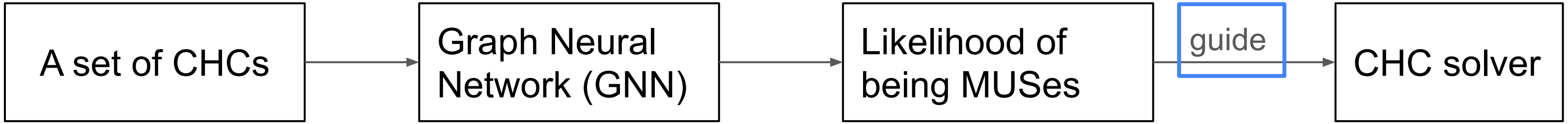
[1] Chencheng Liang, Philipp Rümmer, and Marc Brockschmidt. Exploring Representation of Horn Clauses using GNNs

Prediction phase



Score				
0.8	[1]	$L_1(x)$	$\leftarrow$	$true$
0.2	[2]	$L_2(x)$	$\leftarrow$	$L_1(x) \wedge x > 0$
0.1	[3]	$L_1(x')$	$\leftarrow$	$L_2(x) \wedge x' = x - 1$
0.75	[4]	$L_3(x)$	$\leftarrow$	$L_1(x) \wedge x \leq 0$
0.6	[5]	$false$	$\leftarrow$	$L_3(x) \wedge x \neq 0$

Prediction phase



- Use scores alone
- Combine with original prioritizing scores
    - Add/subtract normalized or ranked scores with coefficient
    - Randomly shifting between MUS and original score

Experimental results

- Benchmarks from CHC-COMP

<table>
<tr><td colspan="7">Linear LIA problems</td></tr>
<tr><td colspan="7">8705</td></tr>
<tr><td colspan="5">Benchmarks for training</td><td colspan="2">Holdout set</td></tr>
<tr><td colspan="5">7834 (90%)</td><td colspan="2">871 (10%)</td></tr>
<tr><td colspan="3">UNSAT</td><td>SAT</td><td>T/O</td><td>Eval.</td><td>N/A</td></tr>
<tr><td colspan="3">1585</td><td>4004</td><td>2245</td><td>383</td><td>488</td></tr>
<tr><td>Train</td><td>Valid</td><td>N/A</td><td colspan="4"></td></tr>
<tr><td>782</td><td>87</td><td>716</td><td colspan="4"></td></tr>
</table>

Experimental results

- Benchmarks from CHC-COMP

<table>
<tr><td colspan="7">Linear LIA problems</td><td colspan="7">Non-linear LIA problems</td></tr>
<tr><td colspan="7">8705</td><td colspan="7">8425</td></tr>
<tr><td colspan="5">Benchmarks for training</td><td colspan="2">Holdout set</td><td colspan="5">Benchmarks for training</td><td colspan="2">Holdout set</td></tr>
<tr><td colspan="5">7834 (90%)</td><td colspan="2">871 (10%)</td><td colspan="5">7579 (90%)</td><td colspan="2">846 (10%)</td></tr>
<tr><td colspan="3">UNSAT</td><td>SAT</td><td>T/O</td><td>Eval.</td><td>N/A</td><td colspan="3">UNSAT</td><td>SAT</td><td>T/O</td><td>Eval.</td><td>N/A</td></tr>
<tr><td colspan="3">1585</td><td>4004</td><td>2245</td><td>383</td><td>488</td><td colspan="3">3315</td><td>4010</td><td>254</td><td>488</td><td>358</td></tr>
<tr><td>Train</td><td>Valid</td><td>N/A</td><td colspan="4"></td><td>Train</td><td>Valid</td><td>N/A</td><td colspan="4"></td></tr>
<tr><td>782</td><td>87</td><td>716</td><td colspan="4"></td><td>1617</td><td>180</td><td>1518</td><td colspan="4"></td></tr>
</table>

Experimental results

- Algorithms of CHC solver (Eldarica)
  - Counterexample-guided abstraction refinement (CEGAR)
  - Symbolic execution (SymEx)

Experimental results (Improved percentage)

Benchmark Algorithm	MUS data set (best count)	Best ranking function (improvement in %)						
		Number of Solved Problems			Average Time			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Linear CEGAR	Union (0)	R-Plus (1.4%)	R-Plus (2.4%)	R-Minus (1.0%)	R-Plus (1.3%)	S-Plus (19.1%)	S-Minus (46.5%)	Rank (31.1%)
	Single (3)	Rank (3.6%)	**R-Plus (4.0%)**	Rank (8.2%)	R-Plus (1.9%)	S-Plus (26.6%)	**R-Minus (57.9%)**	**Rank (36.3%)**
	Intersection (4)	**R-Plus (4.1%)**	S-Plus (0.8%)	**R-Plus (9.3%)**	**R-Plus (3.1%)**	**S-Plus (27.6%)**	R-Minus (45.0%)	S-Plus (0.0%)
Linear SymEx	Union (4)	**Two-Q (1.0%)**	**S-Plus* (0.0%)**	**Random (2.0%)**	Two-Q (0.9%)	R-Minus (12.7%)	**R-Minus (30.2%)**	S-Plus (26.5%)
	Single (3)	S-Minus* (0.5%)	**S-Plus* (0.0%)**	**Random (2.0%)**	Random (0.8%)	**S-Plus (12.9%)**	Random (28.4%)	S-Plus (17.6%)
	Intersection (5)	**S-Plus* (1.0%)**	**S-Plus* (0.0%)**	**S-Plus* (2.0%)**	**S-Plus (1.3%)**	Score (9.5%)	Random (28.4%)	**R-Plus (35.8%)**

Experimental results (Improved percentage)

Benchmark Algorithm	MUS data set (best count)	Best ranking function (improvement in %)						
		Number of Solved Problems			Average Time			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Linear CEGAR	Union (0)	R-Plus (1.4%)	R-Plus (2.4%)	R-Minus (1.0%)	R-Plus (1.3%)	S-Plus (19.1%)	S-Minus (46.5%)	Rank (31.1%)
	Single (3)	Rank (3.6%)	**R-Plus (4.0%)**	Rank (8.2%)	R-Plus (1.9%)	S-Plus (26.6%)	**R-Minus (57.9%)**	**Rank (36.3%)**
	Intersection (4)	**R-Plus (4.1%)**	S-Plus (0.8%)	**R-Plus (9.3%)**	**R-Plus (3.1%)**	**S-Plus (27.6%)**	R-Minus (45.0%)	S-Plus (0.0%)
Linear SymEx	Union (4)	**Two-Q (1.0%)**	**S-Plus* (0.0%)**	**Random (2.0%)**	Two-Q (0.9%)	R-Minus (12.7%)	**R-Minus (30.2%)**	S-Plus (26.5%)
	Single (3)	S-Minus* (0.5%)	**S-Plus* (0.0%)**	**Random (2.0%)**	Random (0.8%)	**S-Plus (12.9%)**	Random (28.4%)	S-Plus (17.6%)
	Intersection (5)	**S-Plus* (1.0%)**	**S-Plus* (0.0%)**	**S-Plus* (2.0%)**	**S-Plus (1.3%)**	Score (9.5%)	Random (28.4%)	**R-Plus (35.8%)**

Experimental results (Improved percentage)

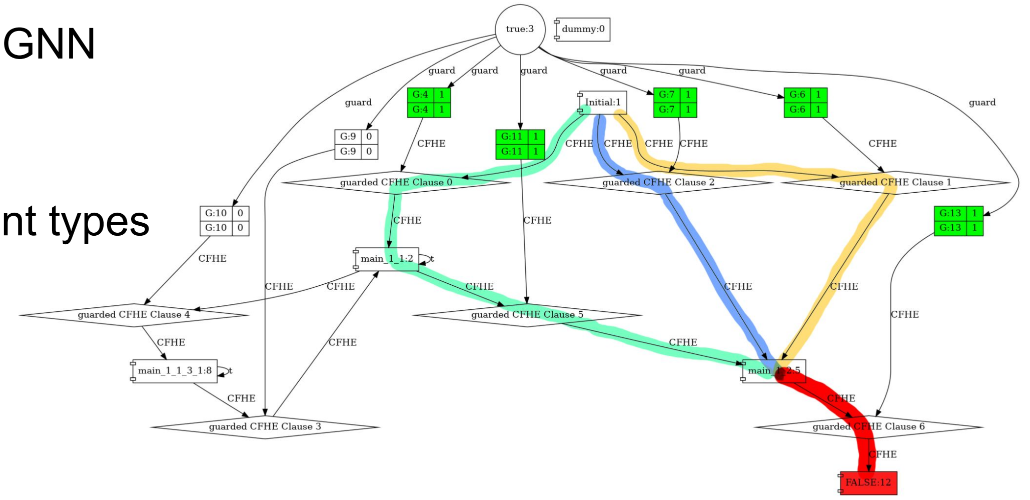
Benchmark Algorithm	MUS data set (best count)	Best ranking function (improvement in %)						
		Number of Solved Problems			Average Time			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Non-Linear CEGAR	Union (7)	**S-Plus (0.5%)**	**S-Plus (0.8%)**	**S-Plus* (0.0%)**	**S-Plus (7.1%)**	**R-Minus (20.8%)**	**Rank (53.5%)**	**S-Plus (19.4%)**
	Single (1)	R-Plus (0.2%)	R-Plus (0.4%)	**R-Plus* (0.0%)**	R-Plus (6.6%)	S-Plus (18.4%)	R-Minus (52.8%)	R-Minus (14.2%)
	Intersection (1)	R-Plus* (0.0%)	S-Plus (0.5%)	**S-Plus* (0.0%)**	R-Plus (5.9%)	R-Plus (20.3%)	Rank (45.8%)	S-Plus (16.8%)
Non-Linear SymEx	Union (6)	**Two-Q (6.1%)**	**S-Minus* (1.6%)**	Random (12.3%)	**Two-Q (13.3%)**	**R-Minus (7.3%)**	**Score (5.1%)**	**R-Plus (19.9%)**
	Single (3)	**Two-Q (6.1%)**	**Score (1.6%)**	**Two-Q (12.9%)**	Two-Q (12.4%)	Rank (-2.2%)	R-Minus (0.2%)	Two-Q (11.2%)
	Intersection (3)	**Two-Q (6.1%)**	**S-Plus (1.6%)**	**Two-Q (12.9%)**	Two-Q (12.7%)	S-Minus (0.6%)	Two-Q (1.7%)	S-Plus (5.4%)

Experimental results (Improved percentage)

Benchmark Algorithm	MUS data set (best count)	Best ranking function (improvement in %)						
		Number of Solved Problems			Average Time			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Non-Linear CEGAR	Union (7)	**S-Plus (0.5%)**	**S-Plus (0.8%)**	**S-Plus* (0.0%)**	**S-Plus (7.1%)**	**R-Minus (20.8%)**	**Rank (53.5%)**	**S-Plus (19.4%)**
	Single (1)	R-Plus (0.2%)	R-Plus (0.4%)	**R-Plus* (0.0%)**	R-Plus (6.6%)	S-Plus (18.4%)	R-Minus (52.8%)	R-Minus (14.2%)
	Intersection (1)	R-Plus* (0.0%)	S-Plus (0.5%)	**S-Plus* (0.0%)**	R-Plus (5.9%)	R-Plus (20.3%)	Rank (45.8%)	S-Plus (16.8%)
Non-Linear SymEx	Union (6)	**Two-Q (6.1%)**	**S-Minus* (1.6%)**	Random (12.3%)	**Two-Q (13.3%)**	**R-Minus (7.3%)**	**Score (5.1%)**	**R-Plus (19.9%)**
	Single (3)	**Two-Q (6.1%)**	**Score (1.6%)**	**Two-Q (12.9%)**	Two-Q (12.4%)	Rank (-2.2%)	R-Minus (0.2%)	Two-Q (11.2%)
	Intersection (3)	**Two-Q (6.1%)**	**S-Plus (1.6%)**	**Two-Q (12.9%)**	Two-Q (12.7%)	S-Minus (0.6%)	Two-Q (1.7%)	S-Plus (5.4%)

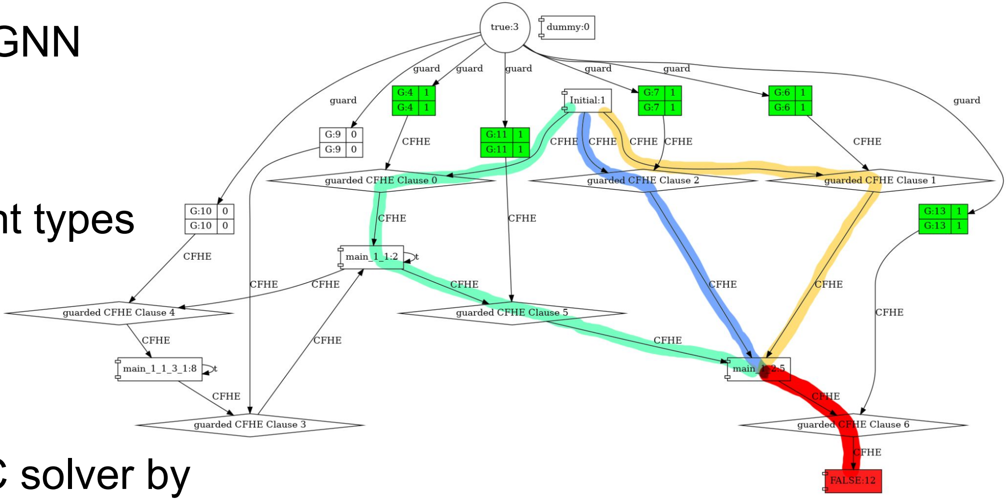
Summary

- General framework that integrates GNN guidance into a CHC solver
- Graph representation of CHCs
- A new GNN that can handle different types of hyperedges



Summary

- General framework that integrates GNN guidance into a CHC solver
- Graph representation of CHCs
- A new GNN that can handle different types of hyperedges



Conclusion

- GNN can be used to speed up CHC solver by predicting MUSes
- GNN learns simple patterns
- It is difficult to learn intricate patterns

What to learn

- Apply deep learning to improve $\boxed{\text{verification (formal methods)}}$
    - Learn reasoning

Reasoning

- Inductive reasoning
- Deductive reasoning

Reasoning

- Inductive reasoning (learn from examples)
- Deductive reasoning

Reasoning

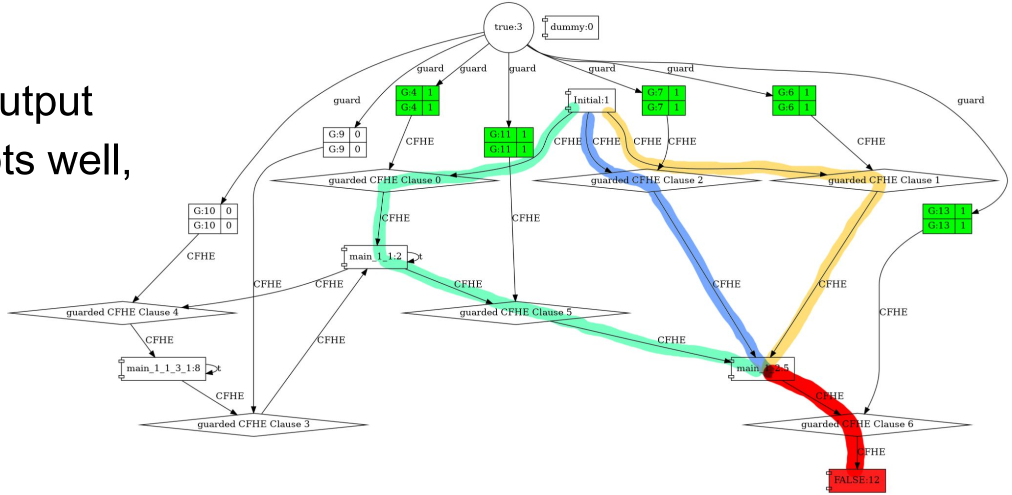
- Inductive reasoning (learn from examples)
- Deductive reasoning (learn from instructions)

Thank you!

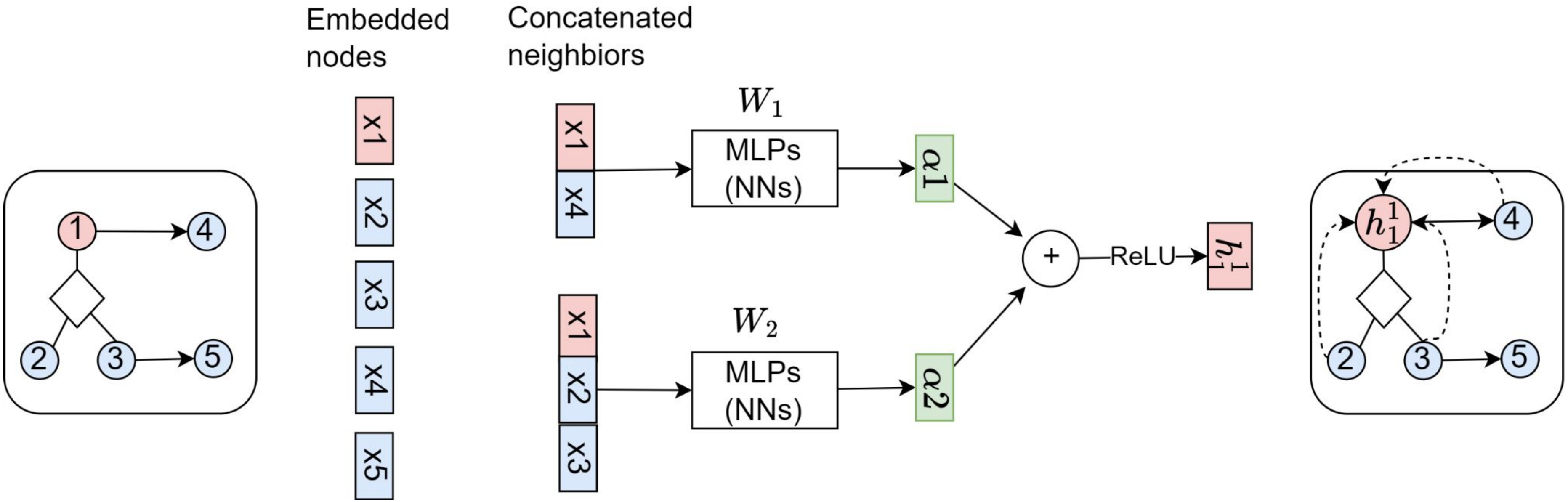
Q & A

Discussion

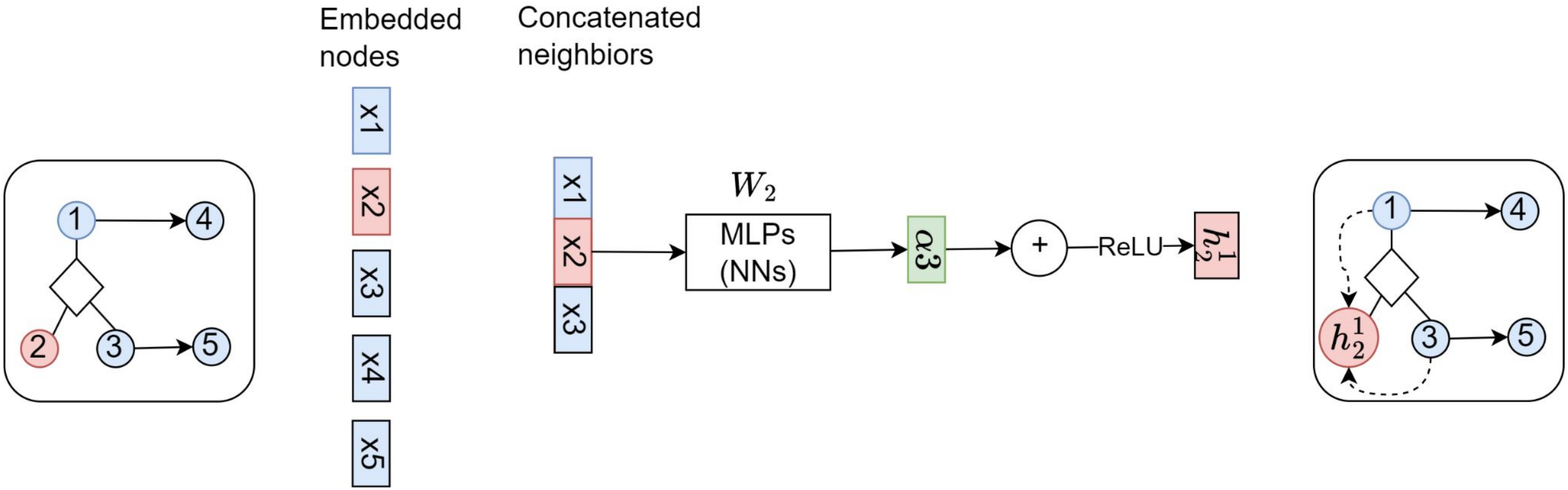
- Deep learning for ranking problem
  - Various number of input and output
- Each model learn particular concepts well, but how to combine them
  - Transfer learning



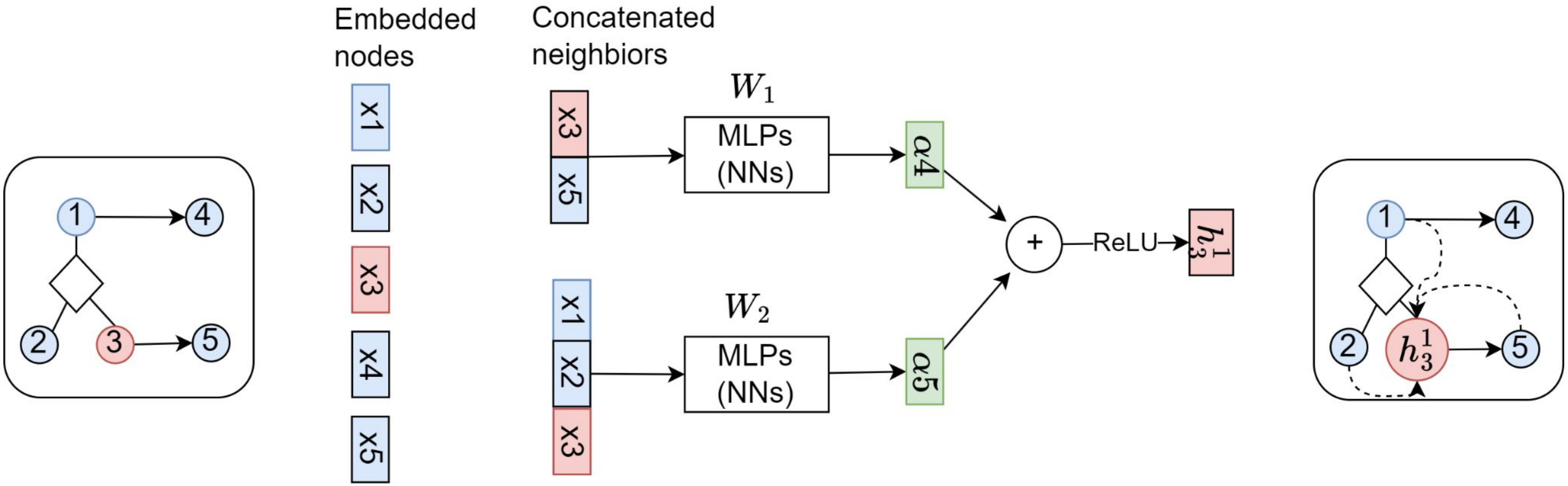
MUSHyperNet Framework (GNN model):



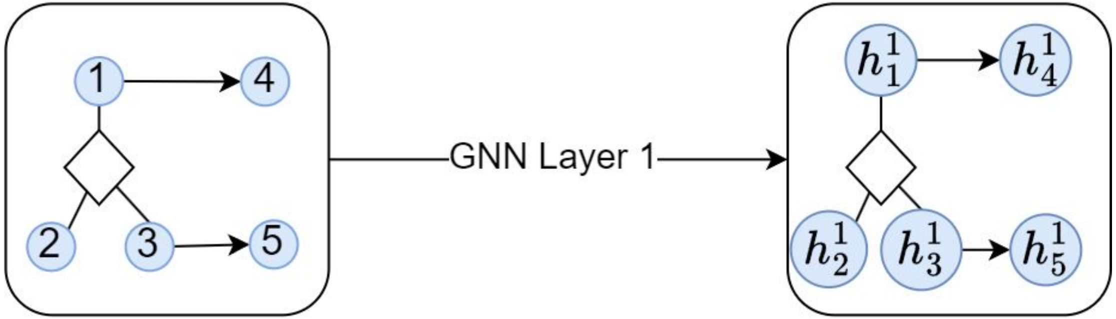
MUSHyperNet Framework (GNN model):



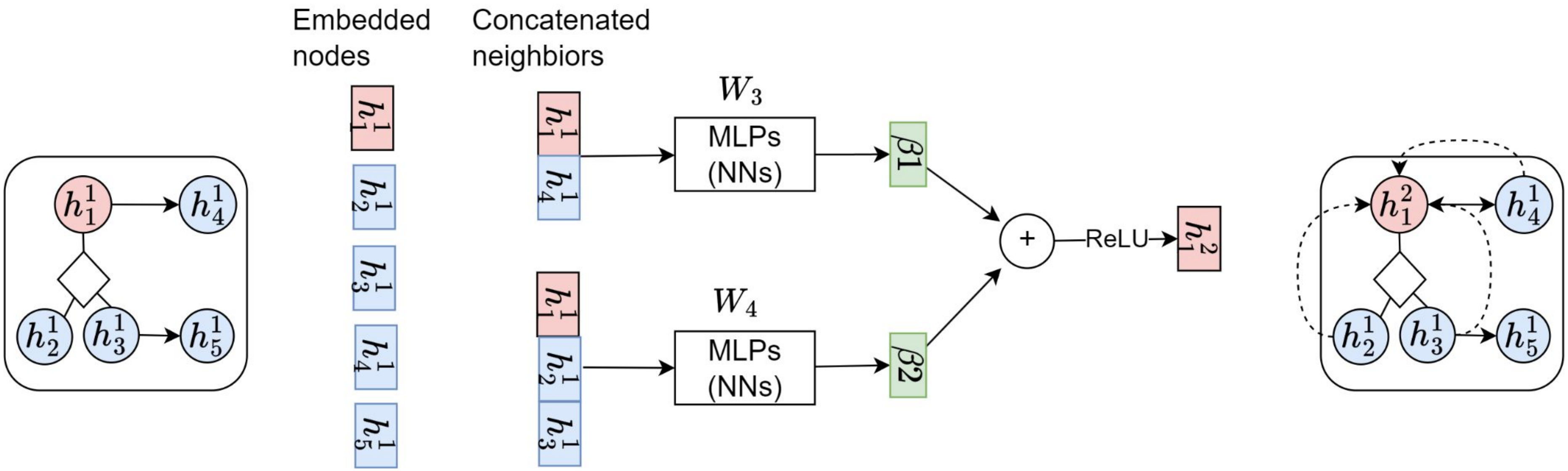
MUSHyperNet Framework (GNN model):



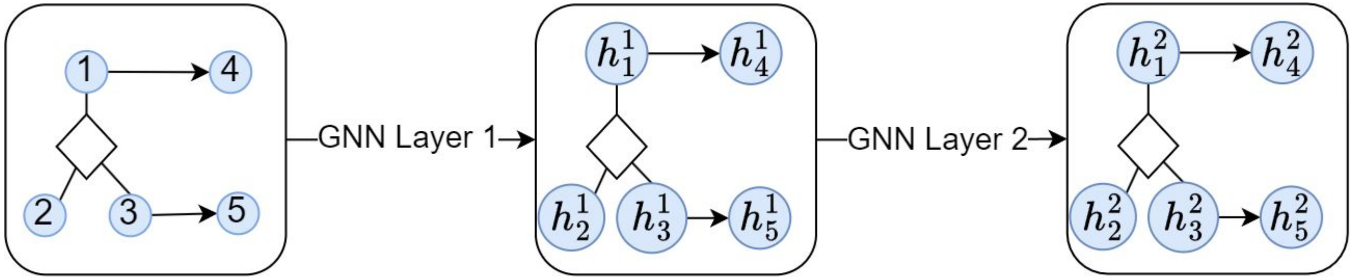
MUSHyperNet Framework (GNN model):



MUSHyperNet Framework (GNN model):



MUSHyperNet Framework (GNN model):

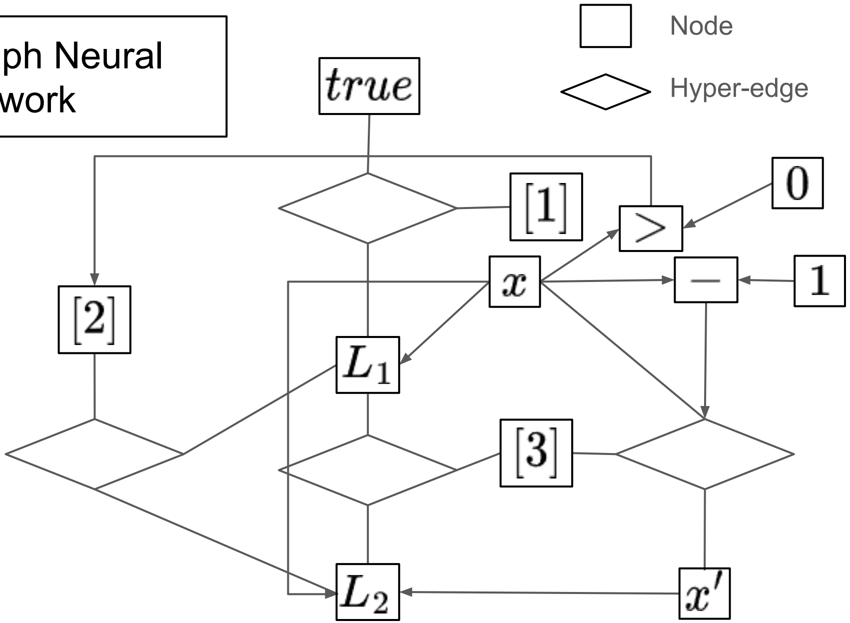


Training phase (train a model)

Node classification

Labeled CHCs → Graph representation → Graph Neural Network

Label		Clauses
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \land x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \land x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \land x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \land x \neq 0$

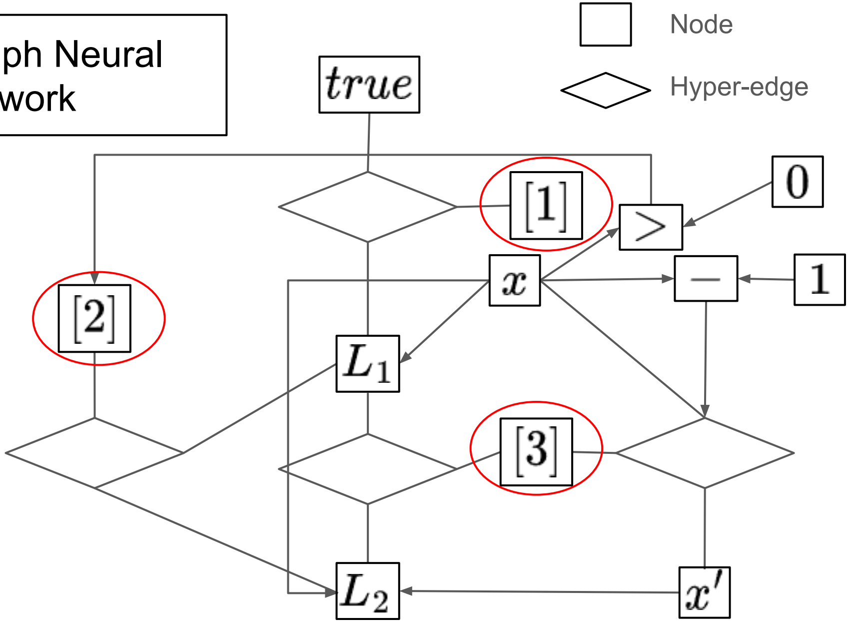


Training phase (train a model)

Node classification

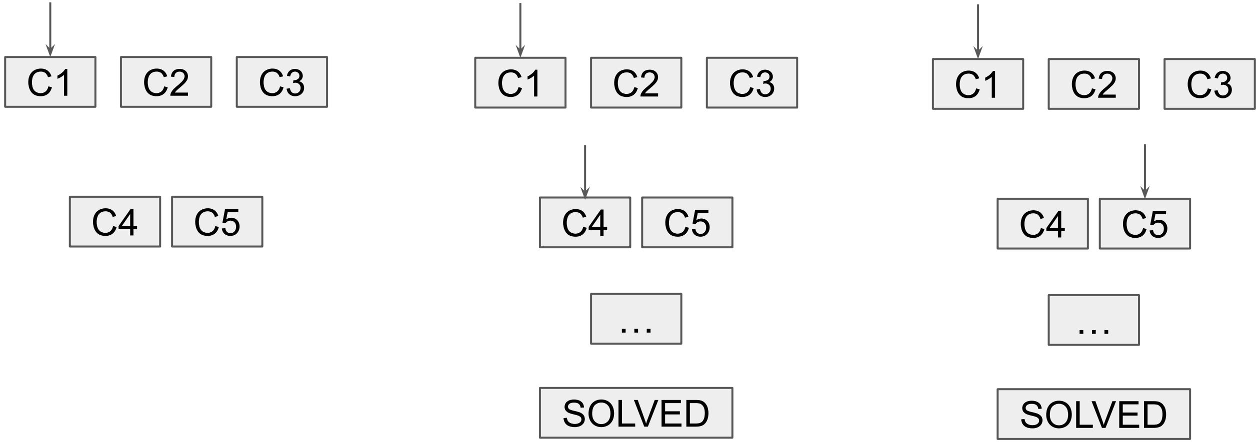
Labeled CHCs → Graph representation → Graph Neural Network

Label		Clauses	
1	[1] $L_1(x)$	$\leftarrow$	$true$
0	[2] $L_2(x)$	$\leftarrow$	$L_1(x) \land x > 0$
0	[3] $L_1(x')$	$\leftarrow$	$L_2(x) \land x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow$	$L_1(x) \land x \leq 0$
1	[5] $false$	$\leftarrow$	$L_3(x) \land x \neq 0$



Motivation

- Target: data-driving method to prioritizing CHCs
- Challenge: search space for collecting training data is too big



MUSes of CHCs

$$
\begin{array}{lll}
[1] & L_1(x) & \leftarrow \mathit{true} \\
[2] & L_2(x) & \leftarrow L_1(x) \wedge x > 0 \\
[3] & L_1(x') & \leftarrow L_2(x) \wedge x' = x - 1 \\
[4] & L_3(x) & \leftarrow L_1(x) \wedge x \leq 0 \\
[5] & \mathit{false} & \leftarrow L_3(x) \wedge x \neq 0
\end{array}
$$

{[1], [4], [5]} is the only MUSes

- Algorithms
  - Counterexample-guided abstraction refinement (CEGAR)
  - Symbolic execution (Symex)

MUSes of CHCs

Score			
0.8	[1]	$L_1(x)$	$\leftarrow \mathit{true}$
0.2	[2]	$L_2(x)$	$\leftarrow L_1(x) \wedge x > 0$
0.1	[3]	$L_1(x')$	$\leftarrow L_2(x) \wedge x' = x - 1$
0.75	[4]	$L_3(x)$	$\leftarrow L_1(x) \wedge x \leq 0$
0.6	[5]	false	$\leftarrow L_3(x) \wedge x \neq 0$

{[1], [4], [5]} is the only MUSes

- Algorithms
  - Counterexample-guided abstraction refinement (CEGAR)
  - Symbolic execution (Symex)

Use predicted MUSes to guide the algorithms

- Prioritize CHCs by using predicted scores of CHCs
  - Use scores alone
  - Combine with original prioritizing scores
    - Add/subtract normalized or ranked scores with coefficient
    - Randomly shift to MUS and original score

Algorithm	Name
CEGAR	Fixed
	Random
	Score
	Rank
	R-Plus
	S-Plus
	R-Minus
	S-Minus
SymEx	Fixed
	Random
	Score
	Rank
	R-Plus
	S-Plus
	R-Minus
	S-Minus
	Two-queue

Experimental results (Improved percentage)

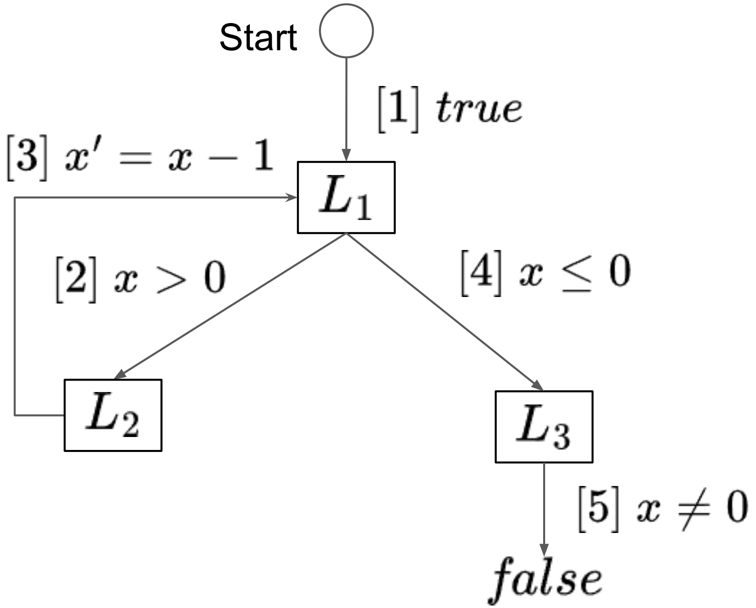
Benchmark Algorithm	MUS data set (best count)	Best ranking function (improvement in %)						
		Number of Solved Problems			Average Time			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Linear CEGAR	Union (0)	R-Plus (1.4%)	R-Plus (2.4%)	R-Minus (1.0%)	R-Plus (1.3%)	S-Plus (19.1%)	S-Minus (46.5%)	Rank (31.1%)
	Single (3)	Rank (3.6%)	**R-Plus (4.0%)**	Rank (8.2%)	R-Plus (1.9%)	S-Plus (26.6%)	**R-Minus (57.9%)**	**Rank (36.3%)**
	Intersection (4)	**R-Plus (4.1%)**	S-Plus (0.8%)	**R-Plus (9.3%)**	**R-Plus (3.1%)**	**S-Plus (27.6%)**	R-Minus (45.0%)	S-Plus (0.0%)
Linear SymEx	Union (4)	**Two-Q (1.0%)**	**S-Plus* (0.0%)**	**Random (2.0%)**	Two-Q (0.9%)	R-Minus (12.7%)	**R-Minus (30.2%)**	S-Plus (26.5%)
	Single (3)	S-Minus* (0.5%)	**S-Plus* (0.0%)**	**Random (2.0%)**	Random (0.8%)	**S-Plus (12.9%)**	Random (28.4%)	S-Plus (17.6%)
	Intersection (5)	**S-Plus* (1.0%)**	**S-Plus* (0.0%)**	**S-Plus* (2.0%)**	**S-Plus (1.3%)**	Score (9.5%)	Random (28.4%)	**R-Plus (35.8%)**

Experimental results (Improved percentage)

Benchmark Algorithm	MUS data set (best count)	Best ranking function (improvement in %)						
		Number of Solved Problems			Average Time			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Non-Linear CEGAR	Union (7)	**S-Plus (0.5%)**	**S-Plus (0.8%)**	**S-Plus* (0.0%)**	**S-Plus (7.1%)**	**R-Minus (20.8%)**	**Rank (53.5%)**	**S-Plus (19.4%)**
	Single (1)	R-Plus (0.2%)	R-Plus (0.4%)	**R-Plus* (0.0%)**	R-Plus (6.6%)	S-Plus (18.4%)	R-Minus (52.8%)	R-Minus (14.2%)
	Intersection (1)	R-Plus* (0.0%)	S-Plus (0.5%)	**S-Plus* (0.0%)**	R-Plus (5.9%)	R-Plus (20.3%)	Rank (45.8%)	S-Plus (16.8%)
Non-Linear SymEx	Union (6)	**Two-Q (6.1%)**	**S-Minus* (1.6%)**	Random (12.3%)	**Two-Q (13.3%)**	**R-Minus (7.3%)**	**Score (5.1%)**	**R-Plus (19.9%)**
	Single (3)	**Two-Q (6.1%)**	**Score (1.6%)**	**Two-Q (12.9%)**	Two-Q (12.4%)	Rank (-2.2%)	R-Minus (0.2%)	Two-Q (11.2%)
	Intersection (3)	**Two-Q (6.1%)**	**S-Plus (1.6%)**	**Two-Q (12.9%)**	Two-Q (12.7%)	S-Minus (0.6%)	Two-Q (1.7%)	S-Plus (5.4%)

Visualize CHCs with dependency graph

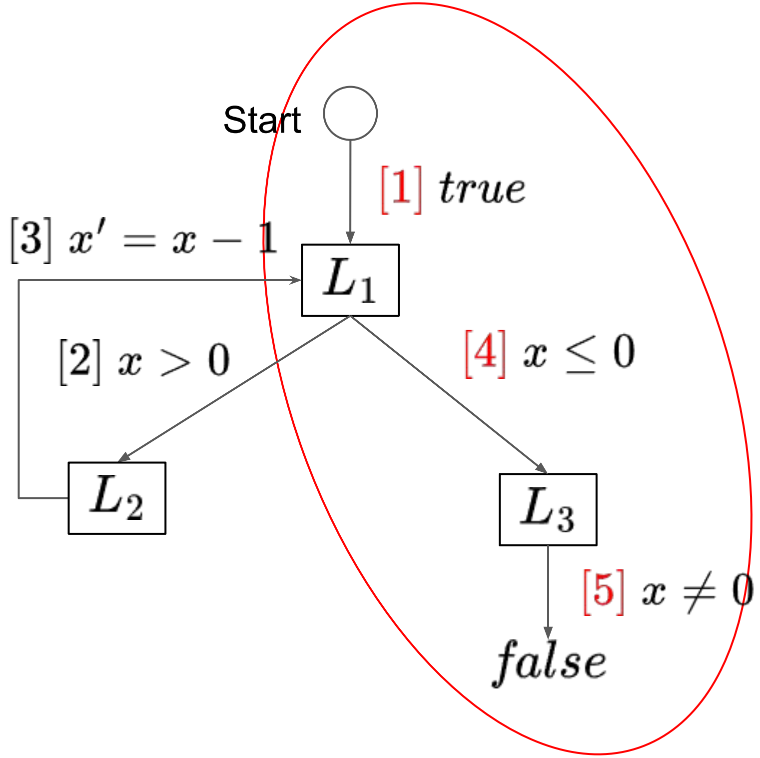
$$
\begin{aligned}
&[1]\ L_1(x) && \leftarrow true \\
&[2]\ L_2(x) && \leftarrow L_1(x) \wedge x > 0 \\
&[3]\ L_1(x') && \leftarrow L_2(x) \wedge x' = x - 1 \\
&[4]\ L_3(x) && \leftarrow L_1(x) \wedge x \leq 0 \\
&[5]\ false && \leftarrow L_3(x) \wedge x \neq 0
\end{aligned}
$$



MUSes of CHCs

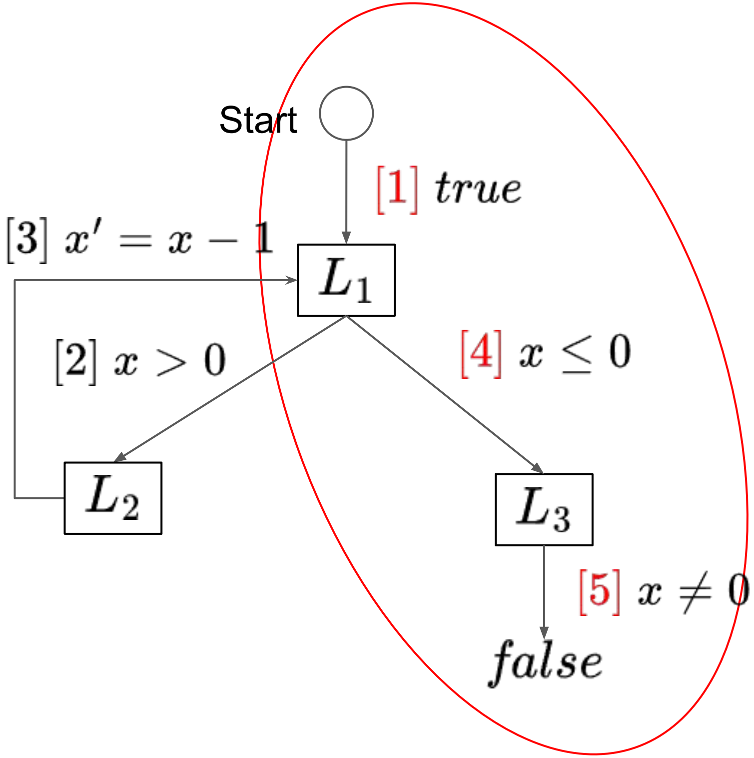
$$
\begin{aligned}
&[1]\ L_1(x) && \leftarrow \mathit{true}\\
&[2]\ L_2(x) && \leftarrow L_1(x) \land x > 0\\
&[3]\ L_1(x') && \leftarrow L_2(x) \land x' = x - 1\\
&[4]\ L_3(x) && \leftarrow L_1(x) \land x \leq 0\\
&[5]\ \mathit{false} && \leftarrow L_3(x) \land x \neq 0
\end{aligned}
$$

{[1], [4], [5]} is the only MUSes



MUSes of CHCs

$$
\begin{array}{lll}
[1] & L_1(x) & \leftarrow true \\
[2] & L_2(x) & \leftarrow L_1(x) \wedge x > 0 \\
[3] & L_1(x') & \leftarrow L_2(x) \wedge x' = x - 1 \\
[4] & L_3(x) & \leftarrow L_1(x) \wedge x \leq 0 \\
[5] & false & \leftarrow L_3(x) \wedge x \neq 0
\end{array}
$$

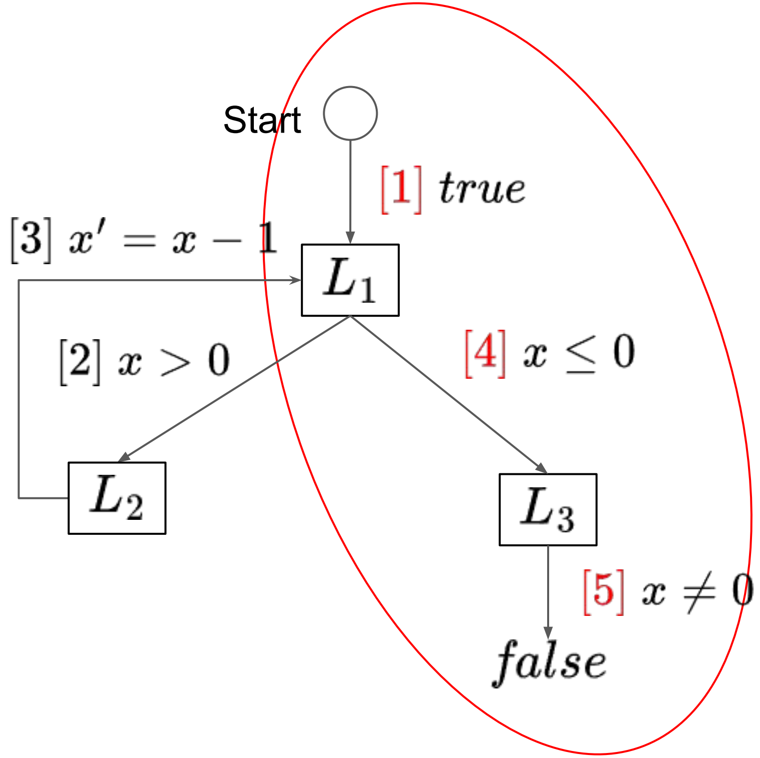


- Algorithms
  - Counterexample-guided abstraction refinement (CEGAR)
  - Symbolic execution (Symex)

MUSes of CHCs

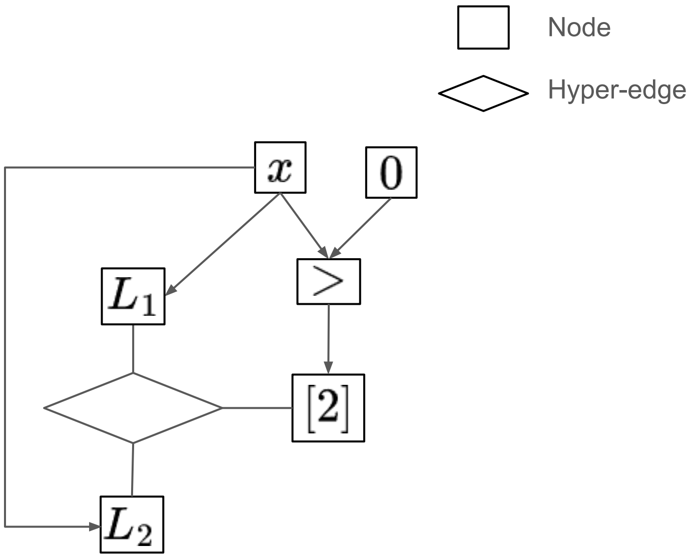
Score			
0.8	[1] $L_1(x)$	$\leftarrow$	$true$
0.2	[2] $L_2(x)$	$\leftarrow$	$L_1(x) \wedge x > 0$
0.1	[3] $L_1(x')$	$\leftarrow$	$L_2(x) \wedge x' = x - 1$
0.75	[4] $L_3(x)$	$\leftarrow$	$L_1(x) \wedge x \leq 0$
0.6	[5] $false$	$\leftarrow$	$L_3(x) \wedge x \neq 0$

- Algorithms
  - Counterexample-guided abstraction refinement (CEGAR)
  - Symbolic execution (Symex)



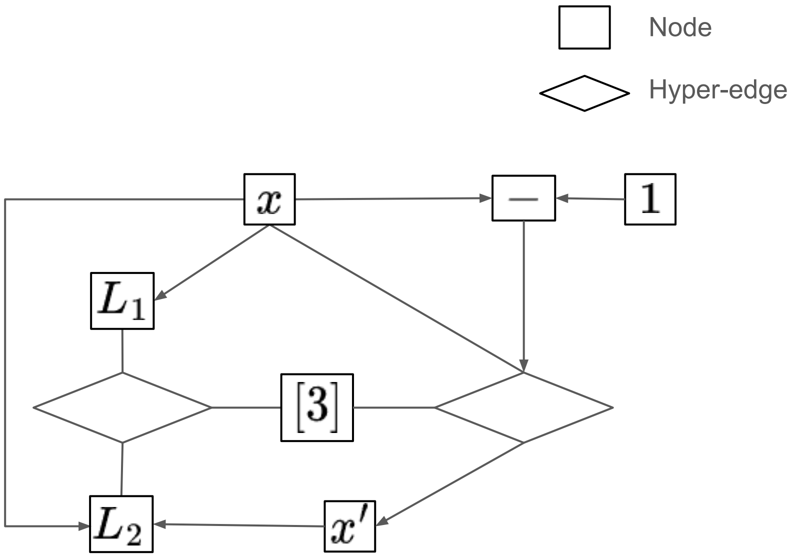
Represent CHCs by graphs

Label	Clauses	
1	[1] $L_1(x)$	$\leftarrow true$
0	[2] $L_2(x)$	$\leftarrow L_1(x) \land x > 0$
0	[3] $L_1(x')$	$\leftarrow L_2(x) \land x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow L_1(x) \land x \leq 0$
1	[5] $false$	$\leftarrow L_3(x) \land x \neq 0$



Represent CHCs by graphs

Label		Clauses	
1	[1] $L_1(x)$	$\leftarrow$	$true$
0	[2] $L_2(x)$	$\leftarrow$	$L_1(x) \wedge x > 0$
0	[3] $L_1(x')$	$\leftarrow$	$L_2(x) \wedge x' = x - 1$
1	[4] $L_3(x)$	$\leftarrow$	$L_1(x) \wedge x \leq 0$
1	[5] $false$	$\leftarrow$	$L_3(x) \wedge x \neq 0$



Experimental results

Benchmark Algorithm	Ranking Function	Number of Solved Problems (improvement %) Total	SAT	UNSAT	Average Time (improvement %) All	Common	SAT	UNSAT
Non Linear CEGAR	Default	432	250	182	131.12	42.05	43.34	40.28
	Random	425 (-1.6%)	243 (-2.8%)	**182 (0.0%)**	143.42 (-9.4%)	34.27 (-11.1%)	34.84 (19.6%)	38.75 (3.8%)
	R-Plus	432 (0.0%)	250 (0.0%)	**182 (0.0%)**	122.29 (6.7%)	31.74 (17.8%)	28.59 (34.0%)	37.82 (6.1%)
	R-Minus	417 (-3.5%)	240 (-4.0%)	177 (-2.7%)	154.07 (-17.5%)	**26.20 (20.8%)**	**21.46 (50.5%)**	**32.51 (19.3%)**
	S-Plus	**434 (0.5%)**	**252 (0.8%)**	**182 (0.0%)**	**121.75 (7.1%)**	34.64 (13.1%)	35.97 (17.0%)	39.10 (2.9%)
	S-Minus	421 (-2.5%)	242 (-3.2%)	179 (-1.6%)	149.02 (-13.7%)	31.76 (-2.0%)	26.33 (39.2%)	38.95 (3.3%)
	Portfolio	435 (0.7%)	253 (1.2%)	182 (0.0%)	113.49 (13.4%)	28.24 (29.1%)	30.57 (29.5%)	31.75 (21.2%)
Non Linear SymEx	Default	342	187	155	343.82	28.39	29.05	27.59
	Random	362 (5.8%)	188 (0.5%)	**174 (12.3%)**	301.90 (12.2%)	32.67 (-15.1%)	**36.24 (-24.8%)**	41.83 (-51.6%)
	R-Plus	339 (-0.9%)	**190 (1.6%)**	149 (-3.9%)	357.18 (-3.9%)	27.88 (0.3%)	47.71 (-64.2%)	**22.10 (19.9%)**
	R-Minus	361 (5.6%)	189 (1.1%)	172 (11.0%)	299.86 (12.8%)	**26.35 (7.3%)**	37.68 (-29.7%)	27.98 (-1.4%)
	S-Plus	340 (-0.6%)	189 (1.1%)	151 (-2.6%)	352.84 (-2.6%)	29.04 (-0.3%)	41.41 (-42.5%)	24.54 (11.1%)
	S-Minus	362 (5.8%)	**190 (1.6%)**	172 (11.0%)	303.65 (11.7%)	28.62 (-0.4%)	44.11 (-51.8%)	37.95 (-37.5%)
	Two-queue	**363 (6.1%)**	189 (1.1%)	**174 (12.3%)**	**297.93 (13.3%)**	30.15 (-6.2%)	41.14 (-41.6%)	32.51 (-17.8%)
	Portfolio	366 (7.0%)	191 (2.1%)	175 (12.9%)	288.85 (16.0%)	22.29 (21.4%)	42.42 (-46.0%)	26.75 (3.0%)

Experimental results

- At least one setting has improvement

Benchmark Algorithm	Ranking Function	Number of Solved Problems (improvement %)			Average Time (improvement %)			
		Total	SAT	UNSAT	All	Common	SAT	UNSAT
Linear CEGAR	Default	222	125	97	519.38	25.77	38.97	8.77
	Random	221 (-0.5%)	124 (-0.8%)	97 (0.0%)	523.58 (-0.8%)	27.49 (-29.5%)	37.05 (4.9%)	15.85 (-80.7%)
	R-Plus	**225 (1.4%)**	**128 (2.4%)**	97 (0.0%)	**512.41 (1.3%)**	21.65 (16.0%)	42.89 (-10.1%)	11.99 (-36.7%)
	R-Minus	220 (-0.9%)	122 (-2.4%)	**98 (1.0%)**	526.08 (-1.3%)	18.02 (-24.4%)	30.93 (20.6%)	21.60 (-146.3%)
	S-Plus	222 (0.0%)	125 (0.0%)	97 (0.0%)	517.43 (0.4%)	**20.92 (19.1%)**	34.13 (12.4%)	**7.32 (16.5%)**
	S-Minus	219 (-1.4%)	122 (-2.4%)	97 (0.0%)	522.97 (-0.7%)	12.56 (2.4%)	**20.86 (46.5%)**	9.81 (-11.9%)
	Portfolio	229 (3.2%)	130 (4.0%)	99 (2.1%)	503.16 (3.1%)	18.28 (29.1%)	45.67 (-17.2%)	19.94 (-127.4%)
Linear SymEx	Default	200	101	99	590.68	33.16	55.42	10.44
	Random	201 (0.5%)	100 (-1.0%)	101 (2.0%)	586.12 (0.8%)	30.08 (-8.5%)	39.69 (28.4%)	20.95 (-100.7%)
	R-Plus	192 (-4.0%)	**101 (0.0%)**	91 (-8.1%)	617.60 (-4.6%)	38.59 (-10.9%)	52.87 (4.6%)	21.99 (-110.6%)
	R-Minus	200 (0.0%)	100 (-1.0%)	100 (1.0%)	586.24 (0.8%)	**24.67 (12.7%)**	**38.69 (30.2%)**	10.60 (-1.5%)
	S-Plus	198 (-1.0%)	**101 (0.0%)**	97 (-2.0%)	595.02 (-0.7%)	30.22 (11.6%)	50.97 (8.0%)	**7.67 (26.5%)**
	S-Minus	201 (0.5%)	**101 (0.0%)**	100 (1.0%)	586.35 (0.7%)	30.64 (7.8%)	50.57 (8.8%)	10.65 (-2.0%)
	Two-queue	**202 (1.0%)**	**101 (0.0%)**	**101 (2.0%)**	**585.58 (0.9%)**	35.11 (-5.9%)	49.94 (9.9%)	20.14 (-92.9%)
	Portfolio	206 (3%)	101 (0.0%)	105 (6.1%)	569.1 (3.7%)	25.79 (22.2%)	44.58 (19.6%)	10.16 (2.6%)

Minimal Unsatisfiable Subsets (MUSes) of CHCs

$$
\begin{aligned}
&[1]\ L_1(x) && \leftarrow true \\
&[2]\ L_2(x) && \leftarrow L_1(x) \land x > 0 \\
&[3]\ L_1(x') && \leftarrow L_2(x) \land x' = x - 1 \\
&[4]\ L_3(x) && \leftarrow L_1(x) \land x \leq 0 \\
&[5]\ false && \leftarrow L_3(x) \land x \neq 0
\end{aligned}
$$

$$L_3(x) \leftarrow true \land x \leq 0$$

$$false \leftarrow x \leq 0 \land x \neq 0$$