CSIsat: Interpolation for LA+EUF

Dirk Beyer[1] **Damien Zufferey**[2] Rupak Majumdar[3]

[1] Simon Fraser University, BC, Canada

[2] EPFL, Switzerland

[3] UCLA, CA, USA

CAV'08, Princeton, July 11, 2008

Outline

Definition [Craig 57]

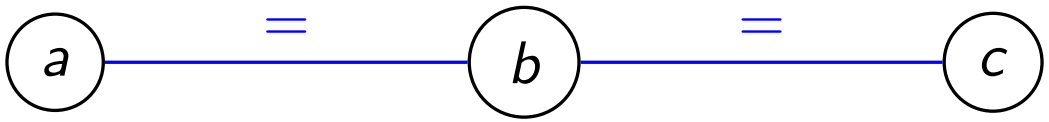
Let A and B be two formulas such that $A \wedge B$ unsat.
An interpolant I has the following properties:

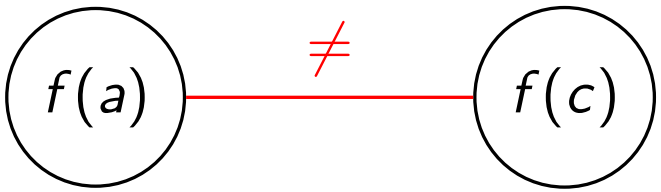
- I contains only AB-common symbols.
- A implies I
- $I \wedge B$ unsat.

Interpolation exists for $\mathrm{LA}+\mathrm{EUF}$.

Interpolant for EUF

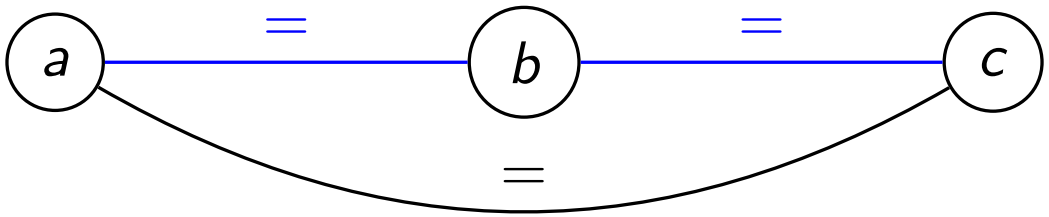
A: $a = b \wedge b = c$ B: $f(a) \neq f(c)$

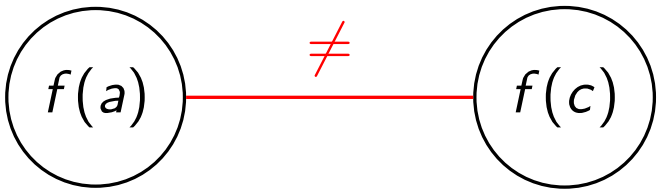




Interpolant for EUF

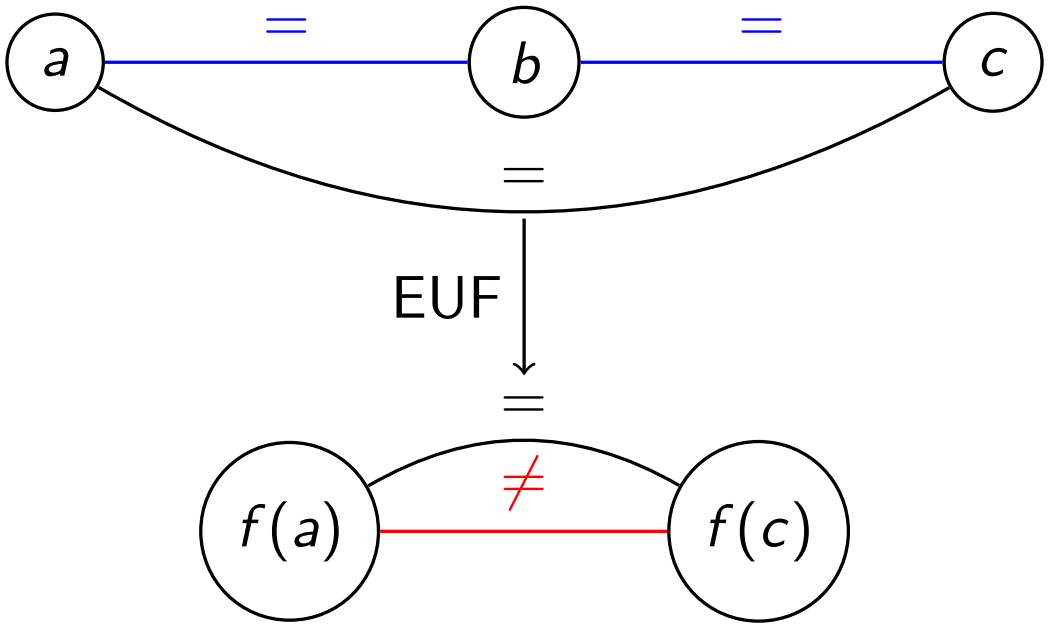
A: $a = b \wedge b = c$ B: $f(a) \neq f(c)$





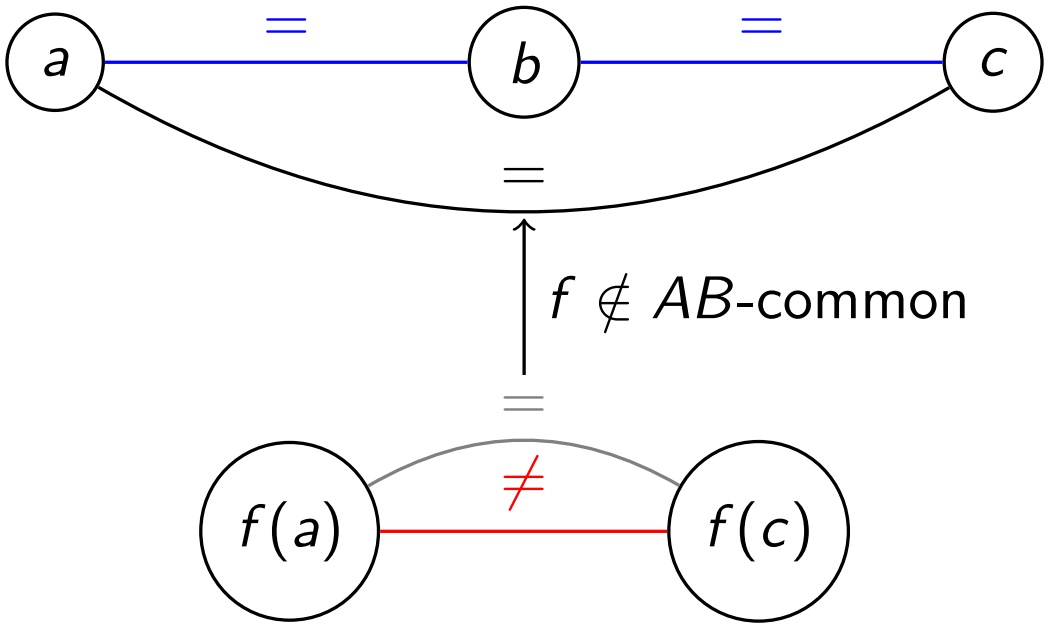
Interpolant for EUF

A: $a = b \wedge b = c$ $\qquad$ B: $f(a) \neq f(c)$



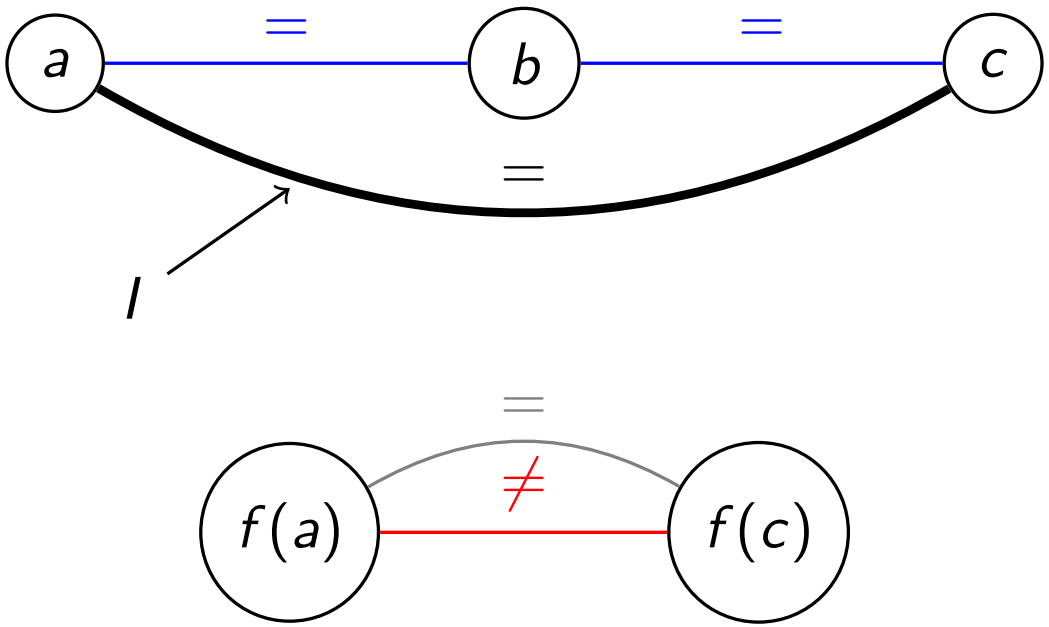
Interpolant for EUF

A: $a = b \wedge b = c$ B: $f(a) \neq f(c)$



Interpolant for EUF

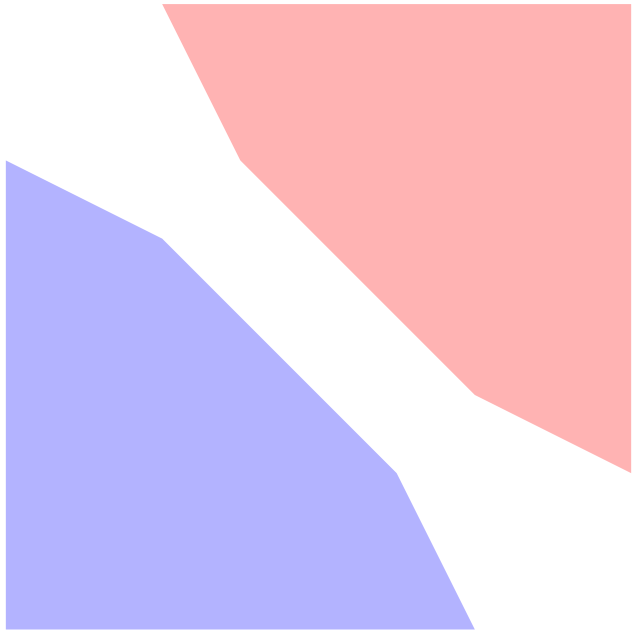
A: $a = b \wedge b = c$ B: $f(a) \neq f(c)$



Interpolant for LA

$A\vec{x} \leq \vec{a}$

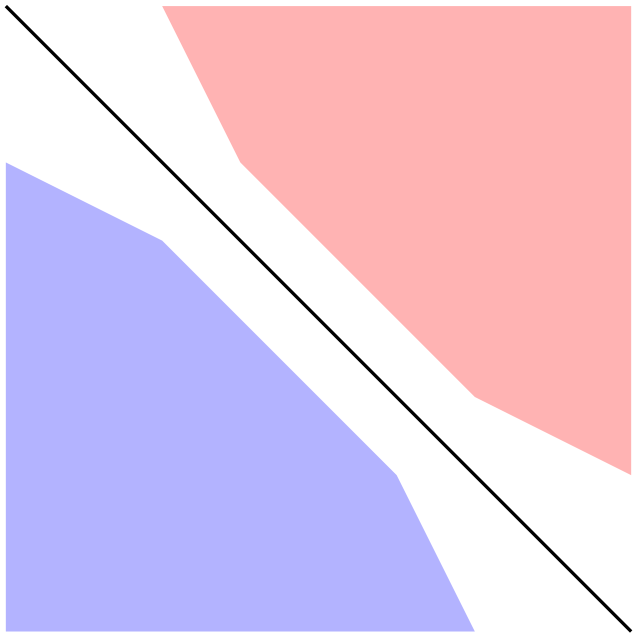
$B\vec{x} \leq \vec{b}$



Interpolant for LA

$A\vec{x} \leq \vec{a}$

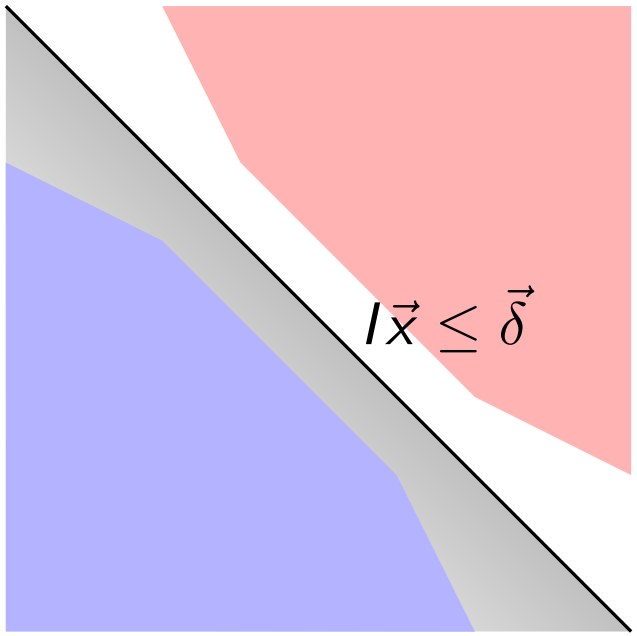
$B\vec{x} \leq \vec{b}$



Interpolant for LA

$A\vec{x} \leq \vec{a}$

$B\vec{x} \leq \vec{b}$



Applications

- Predicate discovery for CEGAR-based model checkers for refinement of abstract states.
- Example: Blast[1] 2.5 is based on CSIsat:

- Open-source software and freely extendable by others.
  - Total of 7500 lines of code written in Ocaml.
  - Includes interpolation code and SMT solver.

[1] http://mtc.epfl.ch/blast/
[2] http://www.kenmcmil.com/foci.html
[3] http://www.mpi-sws.mpg.de/~rybal/clp-prover/

Applications

- Predicate discovery for CEGAR-based model checkers for refinement of abstract states.
- Example: Blast[1] 2.5 is based on CSIsat:
  - Foci[2] for DL + EUF.

- Open-source software and freely extendable by others.
  - Total of 7500 lines of code written in Ocaml.
  - Includes interpolation code and SMT solver.

[1] http://mtc.epfl.ch/blast/
[2] http://www.kenmcmil.com/foci.html
[3] http://www.mpi-sws.mpg.de/~rybal/clp-prover/

Applications

- Predicate discovery for CEGAR-based model checkers for refinement of abstract states.
- Example: Blast[1] 2.5 is based on CSIsat:
  - Foci[2] for DL + EUF.
  - CLProver[3] for LA + EUF, but only conjunctions.
- Open-source software and freely extendable by others.
  - Total of 7500 lines of code written in Ocaml.
  - Includes interpolation code and SMT solver.

[1] http://mtc.epfl.ch/blast/
[2] http://www.kenmcmil.com/foci.html
[3] http://www.mpi-sws.mpg.de/~rybal/clp-prover/

Applications

- Predicate discovery for CEGAR-based model checkers for refinement of abstract states.
- Example: Blast[1] 2.5 is based on CSIsat:
  - Foci[2] for DL + EUF.
  - CLProver[3] for LA + EUF, but only conjunctions.
  - CSIsat is a new implementation for LA + EUF.
- Open-source software and freely extendable by others.
  - Total of 7500 lines of code written in Ocaml.
  - Includes interpolation code and SMT solver.

[1] http://mtc.epfl.ch/blast/
[2] http://www.kenmcmil.com/foci.html
[3] http://www.mpi-sws.mpg.de/~rybal/clp-prover/

Outline

What to give, what to expect

- Input: n formulae $X_1, \ldots, X_n$ such that

$$\bigwedge_{i=1}^{n} X_i \models \bot$$

- Output: $n-1$ interpolants such that

$$\bigwedge_{j=1}^{i} X_j \models I_i$$

$$I_i \wedge \bigwedge_{j=i+1}^{n} X_j \models \bot$$

Syntax

- Formula syntax is very simple and easy to integrate.
- CSIsat supports also Foci syntax.

Example: A: $a = b \wedge b = c$ B: $f(a) \neq f(c)$

```
a = b & b = c    ;    not  f(a) = f(c)
```

Example

Input:

```
a = b;

b = c;

not f(a) = f(c)
```

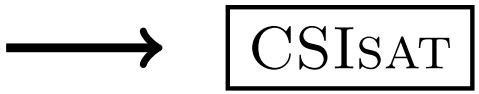
Example

Input:

```
a = b;

b = c;

not f(a) = f(c)
```



Example

Input:

```
a = b;

b = c;

not f(a) = f(c)
```

Output:

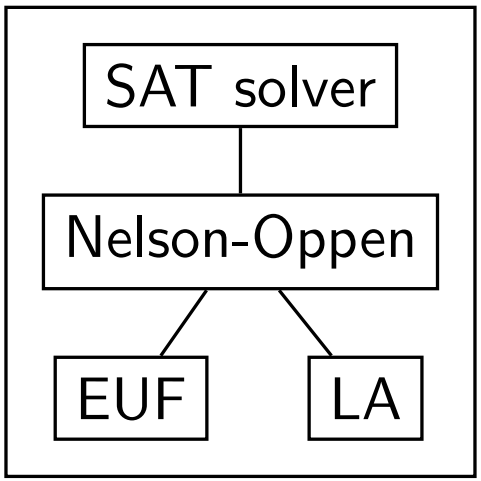
```
a = b

a = c
```

Outline

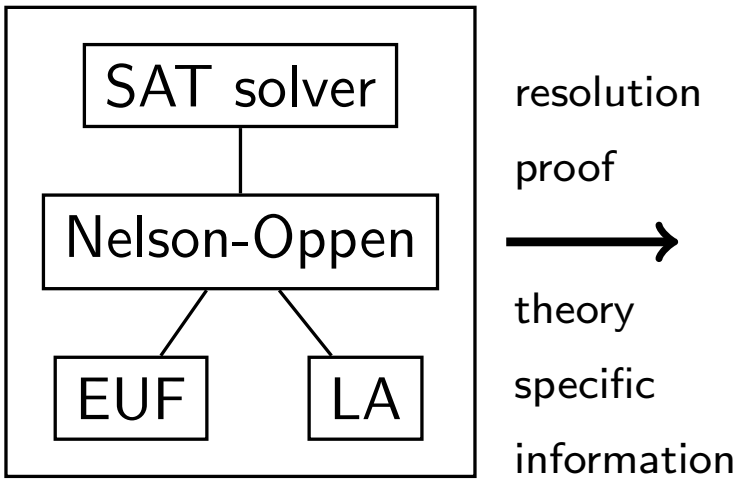
Architecture

1. Generating a resolution proof of unsatisfiability.



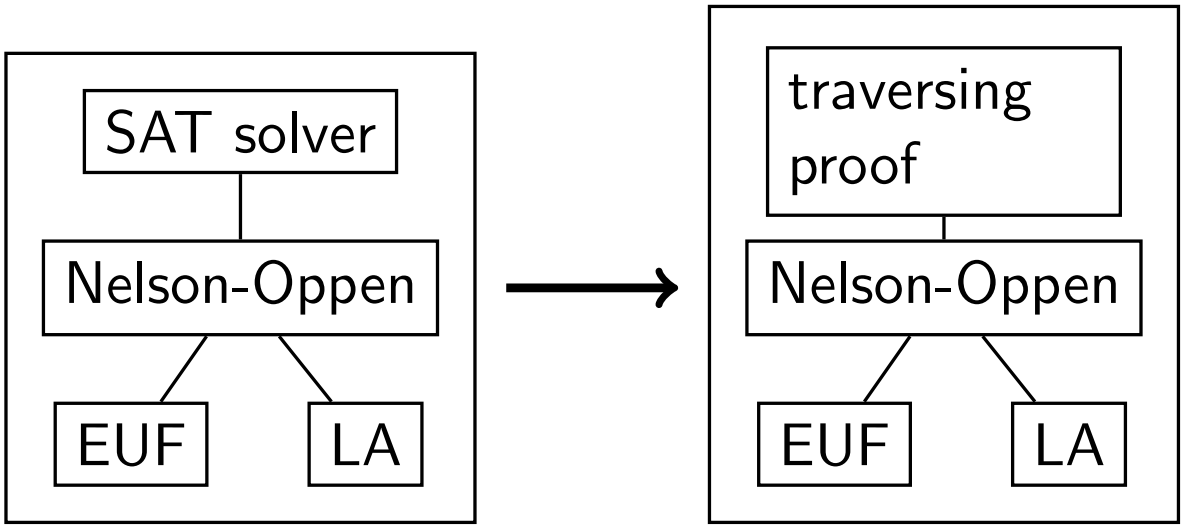
Architecture

1. Generating a resolution proof of unsatisfiability.



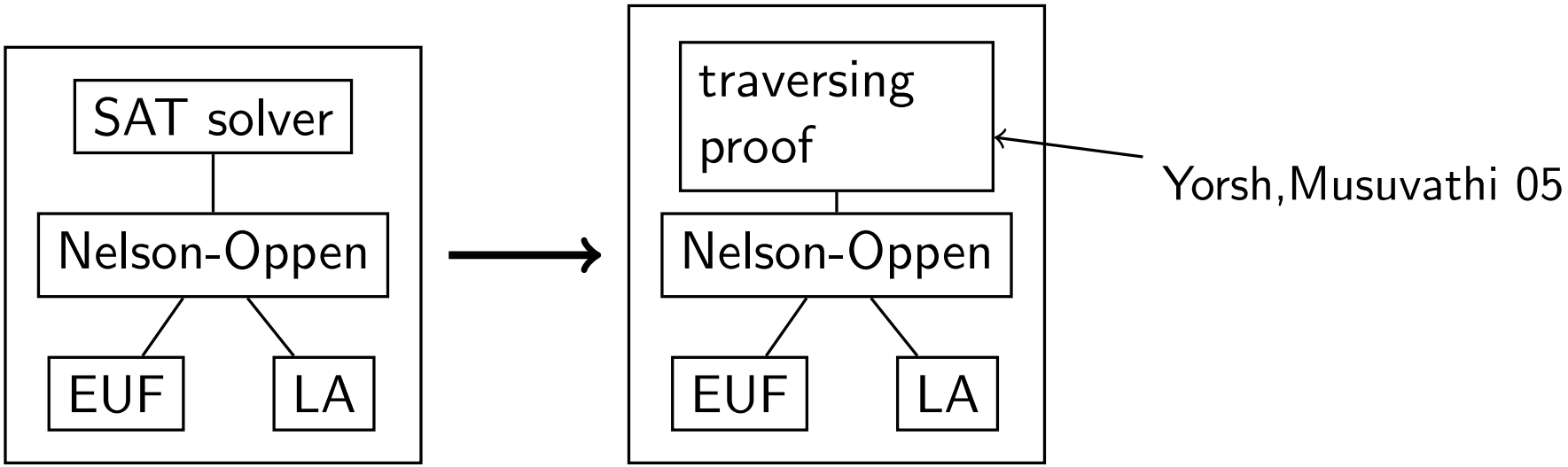
Architecture

1. Generating a resolution proof of unsatisfiability.
2. Constructing the interpolant.



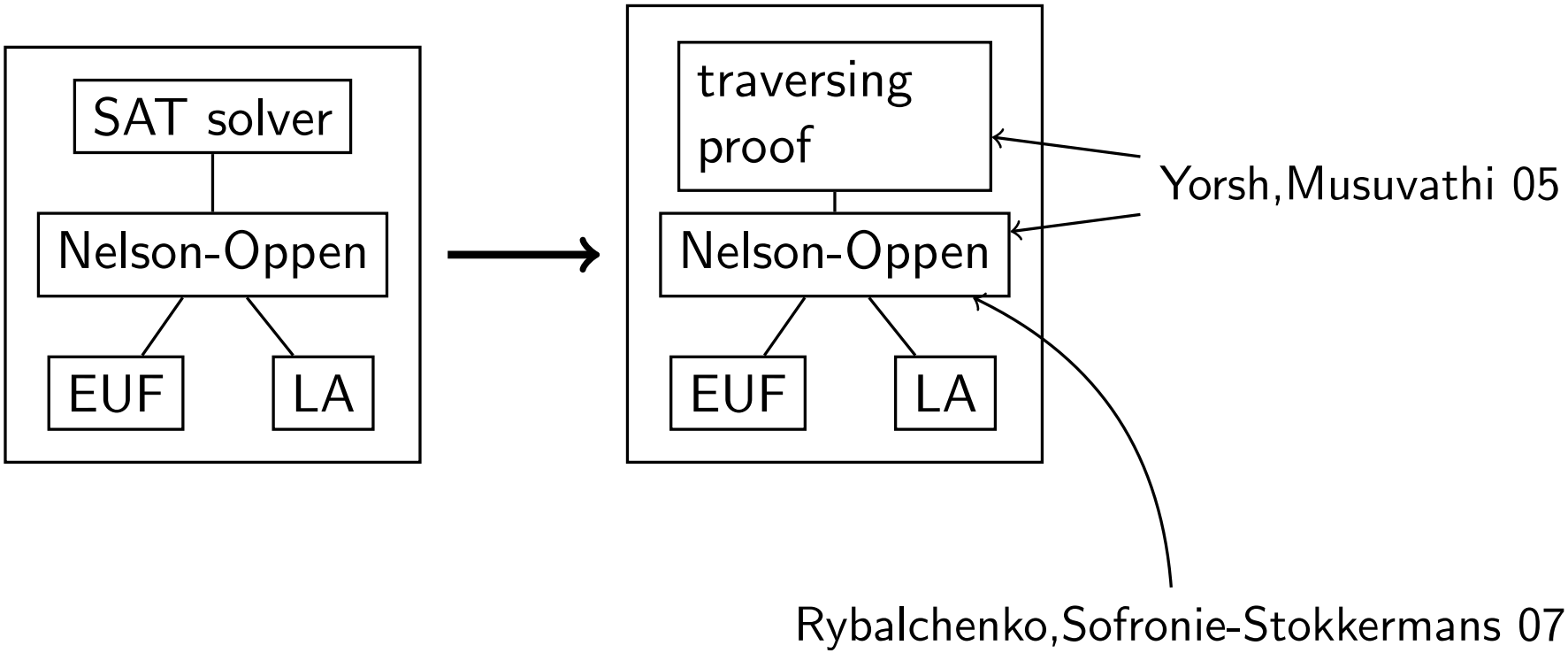
Architecture

1. Generating a resolution proof of unsatisfiability.
2. Constructing the interpolant.



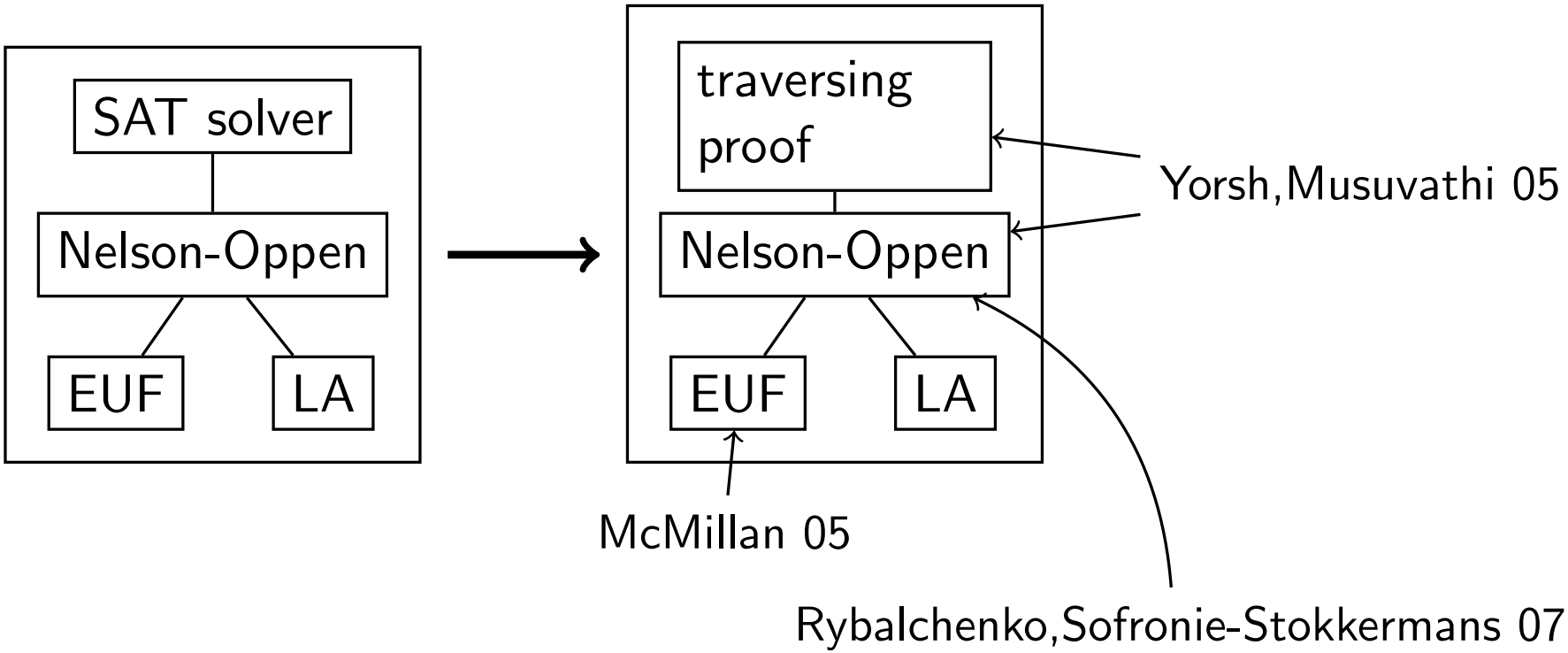
Architecture

1. Generating a resolution proof of unsatisfiability.
2. Constructing the interpolant.



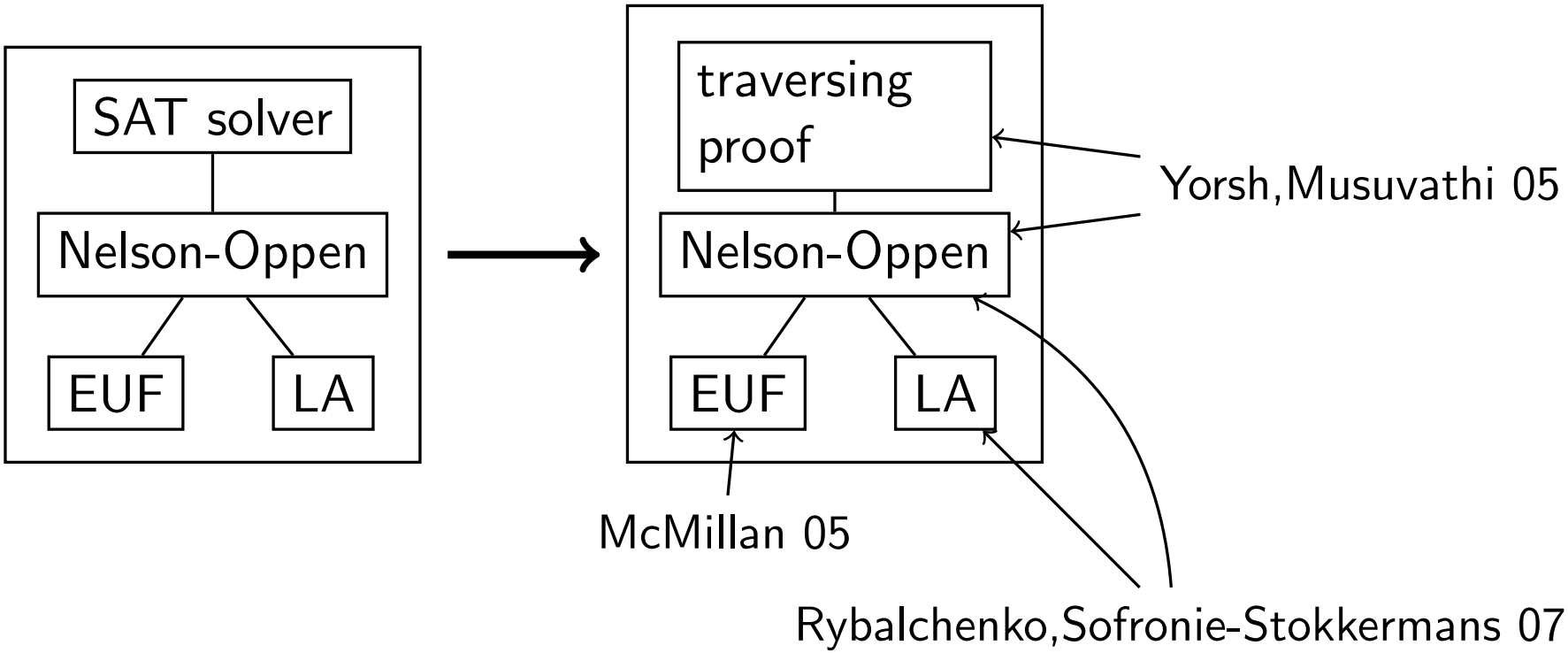
Architecture

1. Generating a resolution proof of unsatisfiability.
2. Constructing the interpolant.



Architecture

1. Generating a resolution proof of unsatisfiability.
2. Constructing the interpolant.



Performance

Program	#queries	Foci	CLPprover	CSIsat
Blast[4]				
floppy	235	1.17 s	1.55 s	**0.55 s**
cdaudio	130	0.60 s	0.70 s	**0.26 s**
ssh	6881	29 s	—	**17 s**
ARMC[5]				
magill	9860	—	30 s	**21 s**

Related work: the new version of MathSAT [CAV 08] can generate interpolants.

[4] http://mtc.epfl.ch/blast/
[5] http://www.mpi-sws.mpg.de/~rybal/armc/

Try it out!

CSIsat is freely available online:

- Project web page:
  `http://www.cs.sfu.ca/~dbeyer/CSIsat`
- Sources and bug reports:
  `http://csisat.googlecode.com`
- Feedback very welcome!

- Questions?