Quantifier Elimination Via Clause Redundancy

Eugene Goldberg, Pete Manolios
Northeastern University, USA

FMCAD-2013, October 20-23,
Portland, OR, USA

Outline

- **Introduction**
- Clause D-sequents
- Example
- Experimental results
- Conclusions

Quantifier Elimination (QE)

Let F be a Boolean CNF formula and $X \subseteq Vars(F)$.

QE problem:

Given $\exists X\,[F\,]$, find a quantifier free CNF formula G such that $G \equiv \exists X\,[F\,]$

$G \equiv \exists X\,[F\,]$ means that $G_{\mathbf{s}} = \exists X\,[F_{\mathbf{s}}]$ for every complete assignment $\mathbf{s}$ to $Vars(F) \setminus X$

QE is important in many areas e.g model checking

SAT-based QE Methods

- **Enumeration of satisfying assignments:**

  McMillan 2002, Ganai,Gupta, Ashar 2004,
  Jin, Somenzi 2005, Brauer, King, Kriener 2011

- **Variable elimination:**

  Davis, Putnam 1960, Jiang 2009, Goldberg, Manolios 2010

- **Computing redundancy of variables**

  Goldberg, Manolios 2012

Three Ideas of Our Method

1) **Add resolvent-clauses** to F until clauses with variables of X (X-clauses) are **redundant** in $\exists X\,[H]$, $H \supseteq F$

Redundancy of X-clause C means $\exists X\,[H] \equiv \exists X\,[H \setminus \{C\}]$

2) **Use branching** to prove redundancy of X-clauses in subspaces and merge results of different branches

3) **Compute termination condition:** (all X-clauses are redundant in $\exists X\,[H]$. This is done by machinery of dependency sequents (D-sequents)

Clause And Variable Redundancy

D-sequents based on **redundancy of variables** (FMCAD-12)

A variable $v \in X$ is redundant in $\exists X\,[F\,]$

if the clauses of F with v are redundant in $\exists X\,[F\,]$

D-sequents based on **clause redundancy** (FMCAD-13)

- Clause D-sequents can express redundancy of any subset of X-clauses
- Derivation of termination condition in terms of clause D-sequents cannot be simulated by variable D-sequents

Outline

- Introduction
- **Clause D-sequents**
- Example
- Experimental results
- Conclusions

Clause Dependency Sequents (D-sequents)

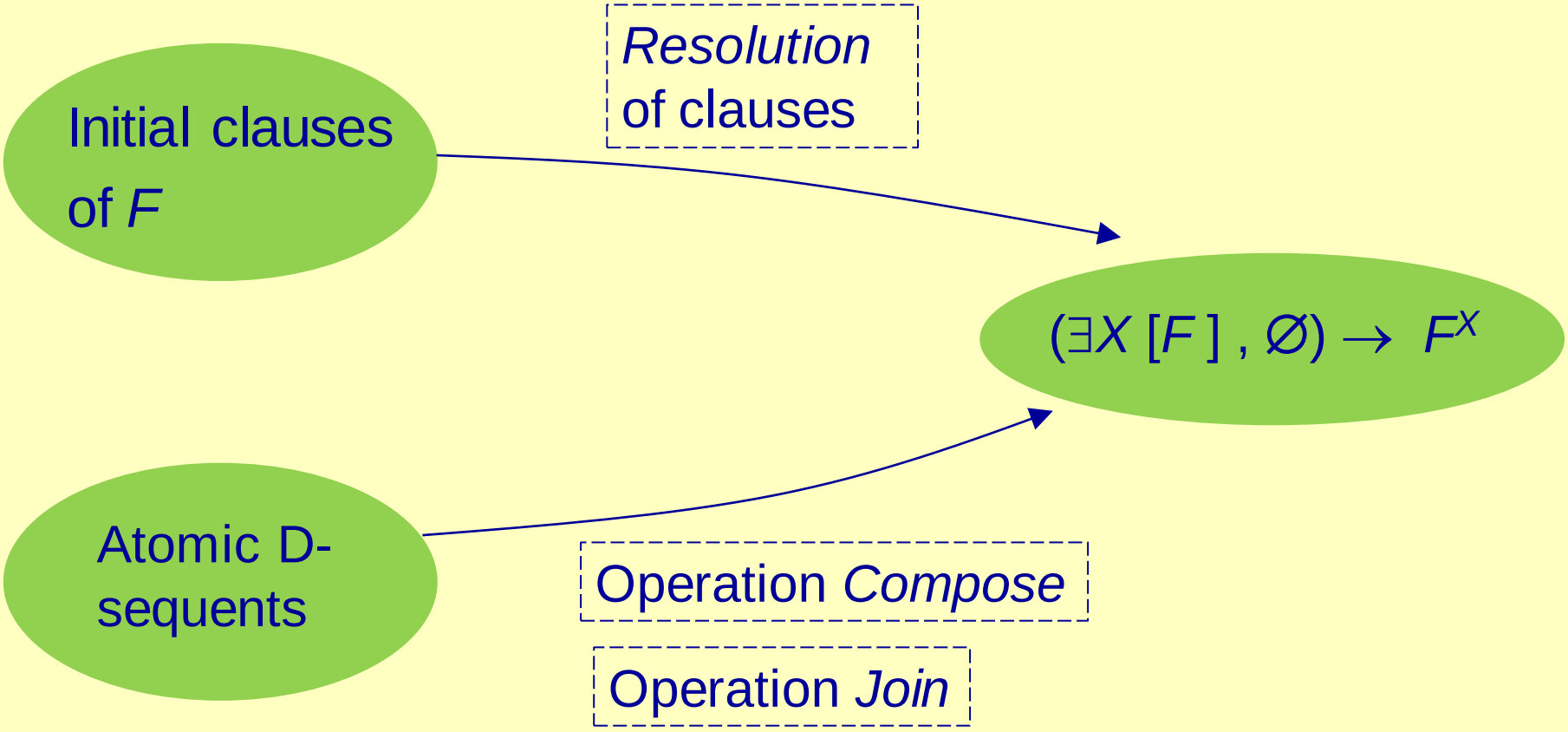
Let **s** be a partial assignment to *Vars*(*F*).

Let F^X denote the *X*-clauses of $\exists X\,[F\,]$

A clause D-sequent $(\exists X\,[F\,]\,,\,\mathbf{s}) \rightarrow R$, *where* $R \subseteq F^X$ states that *R* is redundant in $\exists X\,[F_{\mathbf{s}}]$.

We will call **s** the conditional part of the D-sequent

D-Sequent Calculus



Atomic D-sequents

Let C be an X-clause of $\exists X\,[F\,]$,

Atomic D-sequent $(\exists X\,[F\,], \boldsymbol{s}) \rightarrow \{C\,\}$

is derived when C is

satisfied

by $\boldsymbol{s}$

subsumed

by $C' \in F_{\boldsymbol{s}}$

blocked

(cannot be resolved with clauses of $F_{\boldsymbol{s}}$)

Outline

- Introduction
- Clause D-sequents
- **Example**
- Experimental results
- Conclusions

A Run of DCDS on a Simple Formula

Derivation of **C**lause **D**-**S**equents (**DCDS**)

Consider $\exists x\,[F\,]$ where

$F = C_1 \wedge C_2$,

$C_1 = \sim y_1 \vee x$,

$C_2 = y_2 \vee \sim x$,

$\sim y_1 \vee y_2$ is a **solution** to the QE problem i.e.

$\sim y_1 \vee y_2 \equiv \exists x\,(\sim y_1 \vee x\,) \wedge (y_2 \vee \sim x\,)$

Decision Tree Built by DCDS

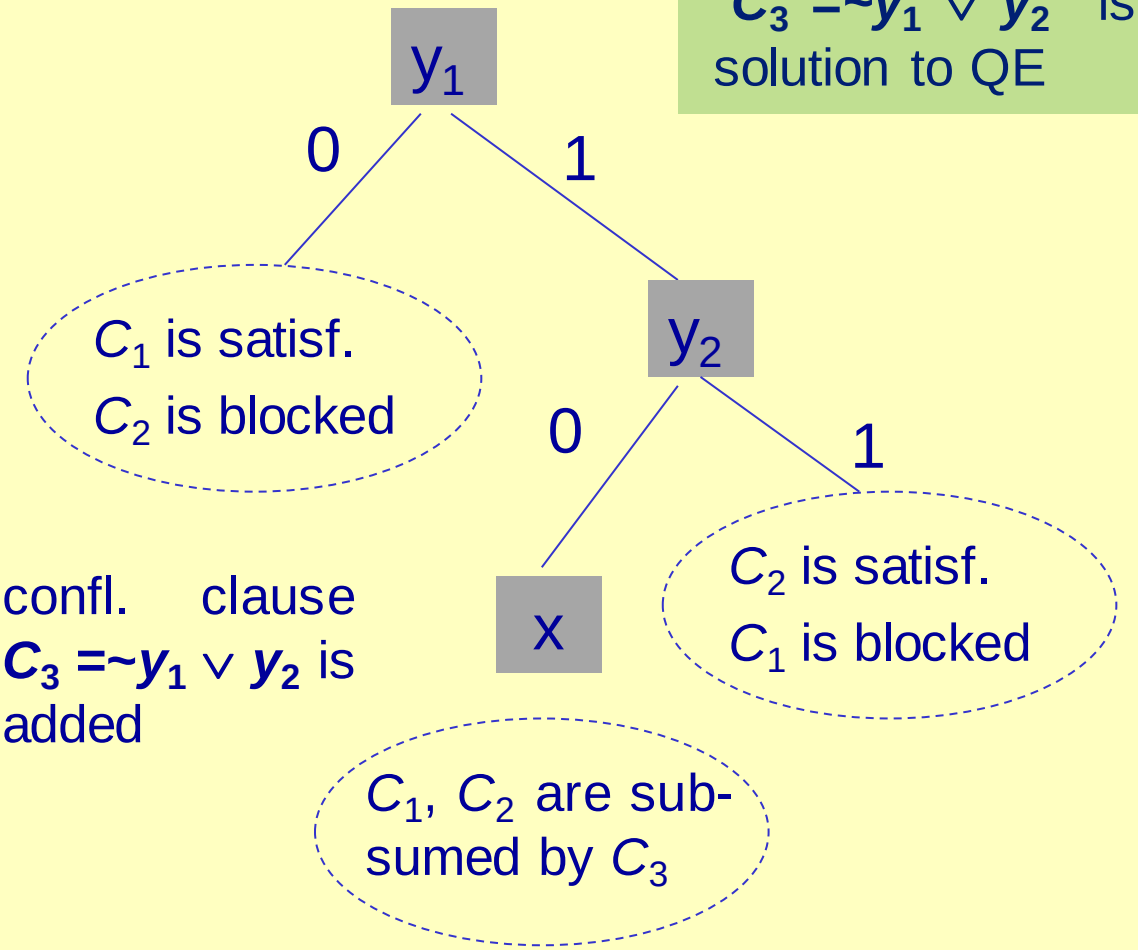
$\exists x\,[F\,]$,

$F = C_1 \wedge C_2$,

$C_1 = \sim y_1 \vee x$,

$C_2 = y_2 \vee \sim x$,

$\boldsymbol{C_3 = \sim y_1 \vee y_2}$ is a solution to QE



confl. clause $\boldsymbol{C_3 = \sim y_1 \vee y_2}$ is added

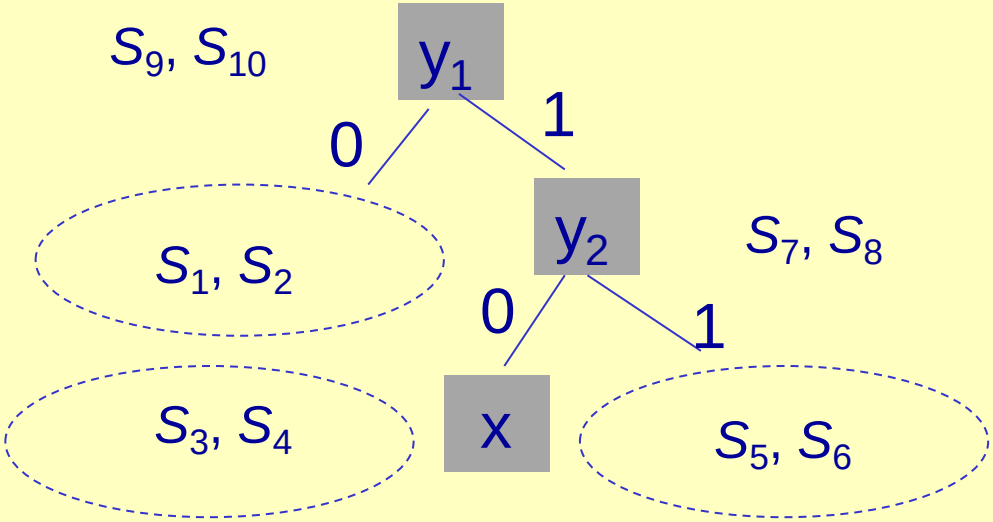
D-sequents Built by DCDS

$\exists x\,[F\,]$,

$F = C_1 \wedge C_2$,

$C_1 = \sim y_1 \vee x$,

$C_2 = y_2 \vee \sim x$,



Join operation:

S_3: $(y_1 = 1, y_2=0) \rightarrow \{C_1\}$,

S_5: $(y_2 = 1) \rightarrow \{C_1\}$,

S_7: $(y_1 = 1) \rightarrow \{C_1\}$,

S_1: $(y_1 = 0) \rightarrow \{C_1\}$, S_2: $(y_1 = 0) \rightarrow \{C_2\}$

S_3: $(y_1 = 1, y_2=0) \rightarrow \{C_1\}$, S_4: $(y_1 = 1, y_2=0) \rightarrow \{C_2\}$

S_5: $(y_2 = 1) \rightarrow \{C_1\}$, S_6: $(y_2 = 1) \rightarrow \{C_2\}$

S_7: $(y_1 = 1) \rightarrow \{C_1\}$, S_8: $(y_1 = 1) \rightarrow \{C_2\}$

S_9: $\varnothing \rightarrow \{C_1\}$, S_{10}: $\varnothing \rightarrow \{C_2\}$

Outline

- Introduction
- Clause D-sequents
- Example
- **Experimental results**
- Conclusions

Re-using D-sequents

- Current implementation of DCDS lacks a few optimizations
- Most importantly, D-sequents are not re-used
- Parent D-sequents are discarded after a join operation
- Re-using D-sequents should drastically boost performance

Backward Model Checking

758 benchmarks of HWMCC-10. Time limit is 2,000 s.

We compared three algorithms:

- **MC-DDS** is based on our QE algorithm of FMCAD-12
- **MC-DCDS** is based on our QE algorithm of FMCAD-13
- **MC-BDD** is based on PdTrav

Model checker	MC-DDS	**MC-DCDS**	MC-BDD
Solved	247	**258**	374

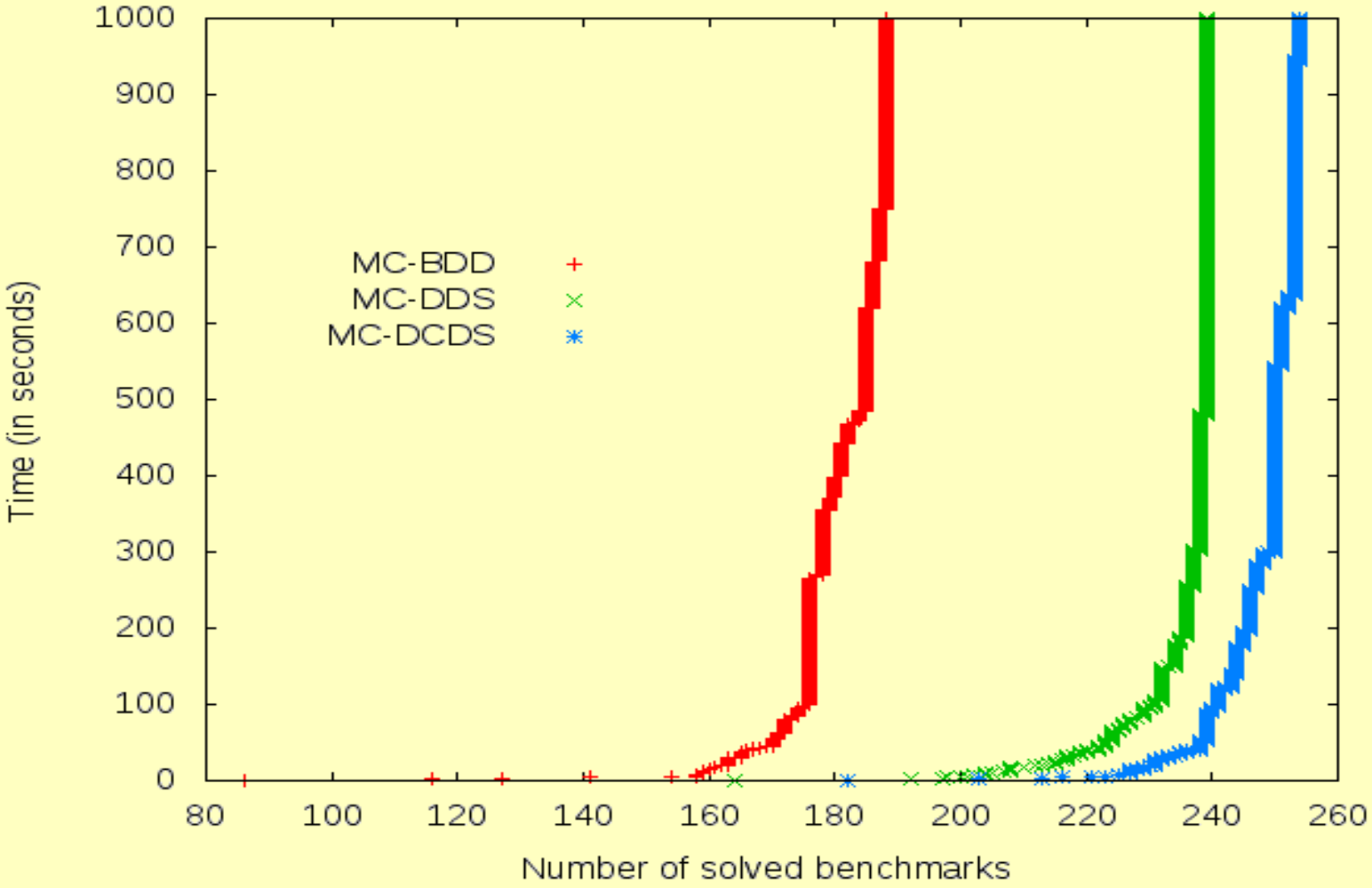
Comparison On Benchmarks Solved by MC-DDS or MC-DCDS

Number of benchmarks is 259

Time limit is 2,000 s.

Model checker	MC-DDS	**MC-DCDS**	MC-BDD
Solved	247	**258**	193
#timeouts	12	**1**	66
Time for solved by all (s.)	11,293	**1,698**	9,080

Cactus Plots For Benchmarks Solved by MC-DDS or MC-DCDS



Conclusions

- We introduced the machinery of clause D-sequents that can be used in many applications
- We showed how it works for quantifier elimination
- A model checker based on clause D-sequents can solve examples that are hard for BDDs
- We are still at the stage where adding a new technique (e.g. re-using D-sequents) can lead to drastic improvements