LECTURE 21

Last time: Covered result that if a formula has a narrow res ref, then CDCL is likely to solve the formula fast.

Final two lectures: Look at nondeterministic algorithms = proof systems. Prove that natural model of CDCL polynomially simulates resolution.

Intuitively: CDCL with right heuristics has potential to be as good as general resolution

Caveat: Nondeterminism probably needed.

Otherwise would get efficient search algorithm

$\Rightarrow$ resolution automatizable

By [Alekhnovich - Razborov '08] this is unlikely

OVERVIEW OF PREVIOUS WORK

[Beame, Kautz, Sabharwal '04]

First result showing CDCL p-simulates resolution

But requires solver to make decisions negating variables set by unit prop.

[Hertel, Bacchus, Pitassi, Van Gelder '08]

CDCL "efficiently p-simulates" resolution

Do preprocessing step $F \rightsquigarrow F'$

Have CDCL refute F'

[Buss, Hoffman, Johannsen '08]
Modify input CNF; introduce new variables
Use CDCL allowing decisions past conflicts
$\Rightarrow$ effective p-simulation

THE FOCUS OF TODAY'S LECTURE

[Pipatsrisawat, Darwiche '11]
Building on [PD '09] but not just
conference - journal relation — actually strengthening
the results

Result: CDCL can p-simulate general resolution, period.
No preprocessing
No decisions on implied variables
No decisions past conflicts.

Contrary to [BKS '04, HBPV '08, BHJ '08]
do not force solver to learn every clause
in proof
Instead somewhat similar to [AFT '11] learn
"good enough" clauses.

PRELIMINARIES

DEF 2 (POLYNOMIAL SIMULATION) adapted from [CR '79]

P_1 <u>p-simulates</u> P_2 if $\exists$ poly-time computable f s.t. if π is a P_2-refutation of F, then $\pi' = f(\pi, F)$ is a P_1-refut of F.

P_1 <u>weakly p-simulates</u> P_2 if for any P_2-refut π can find π' of size polynomial in $S(\pi)$ and $S(F)$ s.t. π' P_1-refut of F.

Dependence on formula technicality — allow proof system to look at all of input.

DEF 2 (UNIT RESOLUTION DERIVATION)

A <u>unit resolution derivation</u> is a res deriv in which at every resolution step one of the clauses resolved is a unit clause (size 1)

$\boxed{\mathbb{D} \vdash_1 a}$ Literal a can be derived from clauses in $\mathbb{D}$ by unit resolution

CDCL ~~MODEL~~ — CDCL AS PROOF SYSTEM

<u>C</u>lause <u>L</u>earning with <u>R</u>estarts

See Figure on next page (21 : IV)

21:IV

Input : CNF formula Δ
Output: A solution of Δ or unsat if Δ is not satisfiable

```
D ← ⟨⟩ // decision literals
Γ ← true // learned clauses
while true do
    if S = (Δ, Γ, D) is 1-inconsistent then
        // there is a conflict.
        if D = ⟨⟩ then
            return unsat
        end
        α ← a conflict clause of S
        m ← the assertion level of α
        D ← D_m // the first m decisions
        Γ ← Γ ∧ α
    else
        // there is no conflict.
        if time to restart then
            D ← ⟨⟩
            S ← (Δ, Γ, D)
        end
        Choose a literal ℓ such that S ⊬_1 ℓ and S ⊬_1 ¬ℓ
        if ℓ = null then
            return D // satisfiable
        end
        D ← D, ℓ
    end
end
```

Algorithm 1: CLR: Clause-learning SAT solver with restarts.

Δ – we will write F
α – -"- C
ℓ – -"- a
Γ – we will write G

NB! No restart here...

But here (!)

DECISION SEQUENCE $D = (a_1, \dots, a_k)$

ordered set of literals

a_i decision at level i

$D_m = (a_1, \dots, a_m)$

CLR STATE $S = (F, G, D)$

F & G CNF formulas; $F \models G$

(F original formula; G learned clauses)

D decision sequence

$S_m = (F, G, D_m)$

H is <u>1-inconsistent</u> if $H \vdash_1 \bot$ and

<u>1-consistent</u> otherwise

$S = (F, G, D)$ is 1-(in)consistent if

$F \wedge G \wedge D$ is 1-(in)consistent.

Write $S \vdash_1 a$ if $F \wedge G \wedge D \vdash_1 a$

$S \nvdash_1 a$ if $F \wedge G \wedge D \nvdash_1 a$

Literal a is <u>implied by state S at level m</u>

if m minimal s.t. $S_m \vdash_1 a$. If so, the

<u>implication level</u> of a (and $\bar{a}$) is m

S is <u>normal</u> if $\forall\ 1 \leq i \leq k$

S_{i-1} is 1-consistent

$S_{i-1} \nvdash_1 a_i$

$S_{i-1} \nvdash_1 \bar{a}_i$

Normality means: no decisions past conflicts

no decisions on implied variables

<u>All states in CCR model are normal</u>

DEF 3 (CONFLICT CLAUSE)

$C = a_1 \vee \dots \vee a_m$ is CONFLICT CLAUSE of 1-inconsistent state $S = (F, G, D)$ if

1) $F \wedge G \wedge \neg C \vdash_1 \bot$
2) $\forall a_i \quad S \vdash_1 \overline{a}_i$

$\neg C = \{ \overline{a} \mid a \in lit(C) \}$

i.e. can show $F \wedge G \models C$ by unit res

all literals $\overline{a}_i$ decision or implication discovered by unit prop

<u>ASSERTION LEVEL</u> 2nd largest implication level of any literal in clause wrt S

Conflict clauses usually defined in terms of cuts in implication graphs [ZMMM '01]

Every such conflict clause can be derived by <u>trivial resolution</u> from $F \wedge G$. [BKS '04, Prop 4].

DEF 4 (TRIVIAL RESOLUTION)

A <u>trivial resolution derivation</u> from F

- Every resolution step between last resolvent and axiom
- Resolved variables are all distinct

PROP 5

$S = (F, G, D)$ with $|D| > 0$ is 1 inconsistent iff has conflict clause

<u>Proof</u> By def.

So description of CDCL makes sense (can always learn clause at conflict)

DEF 4 (ASSERTING CLAUSE)

Conflict clause C of $S = (F, G, D)$ is <u>asserting</u> if has exactly one literal a with implication level $|D|$ (the asserted literal).

Why are asserting clauses nice?

When backtracking to assertion level, $C \setminus \{a\}$ is falsified, so guaranteed to get unit propagation on a

$\boxed{CLR_{LS}}$ SAT solver in Alg 1 with learning scheme LS.

Also need to specify

- branching heuristic ~~fixed bran~~
- restart policy

EXTENDED BRANCHING SEQUENCE (as introduced in [BKS'04])

sequence of literals and special symbol R = restart

$\sigma = \langle x, \bar{y}, R, \bar{x} \rangle$

Set x = true, unit propagate, resolve any conflicts, propagate, etc

then set y = false, unit propagate, — " —

Then restart (but only in 1-consistent state!)

$\boxed{CLR(F, \sigma)}$ Endstate after branching sequence σ (1-consistent or containing $\bot$)

$\boxed{KB(CLR(F, \sigma))}$ Knowledge base of this state $S = (F, G, D)$, i.e., $F \wedge G$

CLR as a proof system

Containing all proof obtainable by particular combination of
- learning scheme
- decision heuristic
- restart policy

Given CLR execution learning conflicts
$C_1, C_2, \ldots, C_k$

Can do resolution derivation
$$\pi_j : \left(F \wedge \bigwedge_{i=1}^{j-1} C_i\right) \vdash C_j$$

String all $\pi_1, \pi_2, \ldots, \pi_k$ together

Finally $F \wedge \bigwedge_{i=1}^{k} C_i \vdash_1 \bot \quad =: \pi_\bot$

In fact, all π_i can be made trivial

Consider $\pi = \pi_1, \pi_2, \ldots, \pi_k, \pi_\bot$
to be the proof generated by this execution of CLR

With a bit of overloading notation...

DEF 5 Given learning scheme LS, CLR_{LS} is the proof system consisting of all refutations that can be generated by the algorithm CRR with learning scheme LS.

With some more formal notation...

DEF 6

Suppose CLR_{LS} run on F with branching sequence σ generating conflict clauses $C_1, \ldots, C_k$

Let S^i ~~state~~ 1-inconsistent for which C_i conflict clause.

Let π_i trivial res deriv of C_i from $KB(S^i)$.

Then σ <u>induces</u> resolution derivation

$DER_{LS}(F, \sigma) = \pi_1, \pi_2, \ldots, \pi_k$

<u>KEY CONCEPTS</u>

If $C = \overline{a}_1 \vee \ldots \vee \overline{a}_{m-1} \vee a_m$ (or $\in$) is a clause we can write this $\bigwedge_{i=1}^{m-1} a_i \to a_m$

(just to follow [PD11] notation

Let $T = \bigwedge_{i=1}^{m+1} a_i$ denote such terms.

DEF 7 (1-EMPOWERMENT)

The clause $C = T \to a$ is <u>1-empowering</u> wrt $\mathbb{D}$ if

1) $\mathbb{D} \models T \to a$
2) $\mathbb{D} \wedge T$ is 1-consistent
3) $\mathbb{D} \wedge T \nvdash_1 a$

a is <u>empowering literal</u> of C. If $\mathbb{D} \models C$ and C contains no empowering literals, then C is <u>absorbed</u> by F

cf [AFT11]

Clause C implied by $\mathbb{D}$ is 1-empowering if it allows unit resolution to derive new implication that would be impossible otherwise

__Ex 8__ $\mathbb{D} =$

$$
\begin{array}{l}
x \vee y \vee z \\
x \vee y \vee \overline{z} \\
x \vee \overline{y} \vee z \\
x \vee \overline{y} \vee \overline{z} \\
\overline{z} \vee u \\
z \vee w
\end{array}
$$

1) $x \vee y$ is 1-empowering, since
$\mathbb{D} \wedge \overline{y} \nvdash_1 x$

2) $u \vee w$ is __not__ 1-empowering but absorbed

$\mathbb{D} \wedge \overline{u} \vdash_1 w$
$\mathbb{D} \wedge \overline{w} \vdash_1 u$

OBSERVATION 9 (cf monotonicity lemma in [AFT 11])

1) If $C \subseteq C'$ then C' is absorbed by C

2) Adding more clauses to knowledge base can make 1-empowering clause absorbed, but can never make absorbed clause 1-empowering.

3) Every asserting clause is 1-empowering wrt knowledge base at the time of its derivation with ~~the~~ asserted literal as empowering

__Proof__ Exercise (juggle with the definitions).

DEF 10 (1-provability)
Clause C is <u>1-provable</u> wrt clause set $\mathbb{D}$ if $\mathbb{D} \wedge \neg C \vdash_1 \bot$.

<u>Ex 11</u> Go back to Ex 8 and clause set $\mathbb{D}$ there.

$$\mathbb{D} \vDash x \vee y$$
$$\mathbb{D} \vDash x$$

$x \vee y$ is 1-provable, but x is <u>not</u>.

By Def 3 every conflict clause is 1-provable wrt the knowledgebase $F \wedge G$ at the time of its derivation from $S = (F, G, D)$.

<u>LEMMA 12</u> (1ST KEY LEMMA)
Let F be a 1-consistent CNF formula and suppose $\pi : F \vdash \bot$ is a res ref.
Then $\exists\, C \in \pi$ which is both 1-empowering and 1-provable wrt F.

Follows from two propositions

<u>PROP 13</u>
Suppose $F \vDash \overset{C_1 =}{C_1' \vee x}, \overset{C_2 =}{C_2' \vee \overline{x}}$ but that neither clause is 1-empowering wrt F.
Then $C = C_1' \vee C_2'$ is 1-provable wrt F

<u>Proof</u> Suppose C_1 not 1-empowering, ie. some condition in Def 7 fails.

$F \nvDash C_1$ by assumption $\Rightarrow$ not cond 1.

Suppose $F \wedge \neg C'_1$ not 1-consistent, i.e.

$$F \wedge \neg C'_1 \vdash_1 \bot$$

Then $F \wedge C \vdash_1 \bot$ so C 1-provable

Takes care of cond 2.

Otherwise cond 3 fails and

$$F \wedge \neg C'_1 \vdash_1 x \qquad (*)$$

Same analysis for $C_2 = C'_2 \vee \bar{x}$ yields that either we're already done or

$$F \wedge \neg C'_2 \vdash_1 \bar{x} \qquad (**)$$

Combining $(*)$ & $(**)$ get $F \wedge \neg C \vdash_1 \bot$, so $C = C'_1 \vee C'_2$ is 1-provable. QED $\boxed{\text{E}}$

PROP 14 If $\pi : F \vdash C$ but C is not 1-provable wrt F, then π contains clause that is both 1-provable & 1-empowering

Proof Let $\pi = \{C_1, C_2, \ldots, C_L\}$ $\quad C_L = C$

Let C_i first clause in π not 1-provable

Can have $C_i = C_L = C$.

C_i resolvent of C_j, C_k 1-provable.

By Prop 13 one of C_j & C_k has to be 1-empowering $\boxed{\text{E}}$

Proof of Lem 12 $F \nvdash_1 \bot$ by assumption, since F is 1-consistent. Prop 14 now provides 1-empowering & 1-provable clause $\boxed{\text{E}}$

LEMMA 15 (2ND KEY LEMMA)
Let $S = (F, G, D)$ be 1-inconsistent CLR state.
Let C conflict clause of S that is 1-empowering wrt $F \wedge G$. Then $\exists$ trivial res derv of $C' \subseteq C$ from $F \wedge G$

Proof sketch
Let $C = C' \vee a$ with empow lit a.
Let σ branching sequence on $\neg C'$ (in any order) followed by $\bar{a}$. Let DECISION be the decision learning scheme as defined in [ZMMM '01] (and as discussed in [AFT 11])
This learning scheme $\Rightarrow$ conflict clauses only over decision variables.
Following σ leads to conflict
DECISION will learn $C' \subseteq C$
Prop 4 of [BKS '04] Now implies that $\exists$ trivial res derv of C'. $\boxed{}$

Given any res ref $\pi : F \vdash \bot$, want to p-simulate it in CLR.
Focus on clauses $\in \pi$ which are both 1-empowering and 1-provable. Show that for any such C, can force CLR to derive clauses so that unit resolution sees any implications C would allow us to derive (so although we may not have learnt C, get the same effect for all practical purposes) (cf [AFT 11])