DPLL algorithm

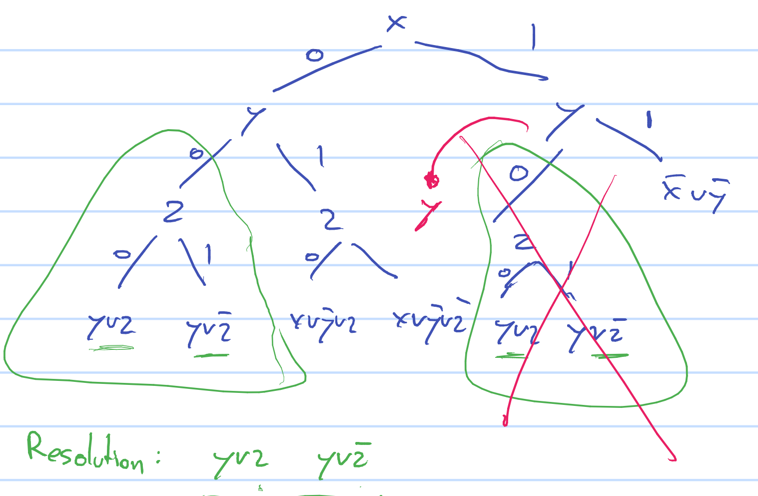
state: α.

if conflict backtrack()

if unit propagate()

o/w make decision.

$G = \{ y \vee z,\ y \vee \bar z,\ x \vee \bar y \vee z,\ x \vee \bar y \vee \bar z,\ \bar x \vee \bar y \}$.



Resolution: $\dfrac{y \vee z \quad y \vee \bar z}{y}$

CDCL: state: α (assignment), D (database)

while not solved:

if conflict: 1. learn()

2. backtrack()

if unit clause:

set forced value

"unit propagation"

else: make decision.

(add non-forced value to α).

in DPLL $\boxed{F(\alpha) = 0}$ $\exists C \in F$ st $C(\alpha) = 0$.

in CDCL $\exists C \in F \cup D$ st $C(\alpha) = 0$.

learn: add to D some clause that is a logical consequence of F

$F: x \vee y \quad x \vee \bar y \quad \bar x \vee y \quad \bar x \vee \bar y$

$\cancel{F(\phi) = 0}$.

$\exists C \in F \cup D$ st $|C(\alpha)| = 1$

How to do decisions / unit propagations.

$$\alpha = \Big(x_1 \overset{d\ (\text{decision})}{=} b_1,\ x_2 \overset{C_2 (\text{unit})}{=} b_2,\ x_3 \overset{C_3 (\text{unit})}{=} b_3,\ x_4 \overset{d}{=} b_4 \ \cdots \Big)$$

$x_1 \overset{d}{=} 0 \quad \cancel{x_1} \vee x_2$

$x_2 = 0$ ($x \vee x_2$)

trail: sequence of assignments to variables annotated with either d or C clause.

operations: → insert one var.

→ remove a set of vars.

→ find a free var.

trail (value + reason)] state

+ queue of free variables.] implementation.

state = trail + DB. learned.

def decision level of a variable.

$$\langle \underbrace{d, u, u, u}_{\text{d. level } 1},\ \underbrace{d, u, u, u}_{\text{d. level } 2},\ \underbrace{d}_{\text{dl.3}},\ \underbrace{d, u, u}_{\text{dl.4}} \rangle.$$

decision level of variable x_i is # of decisions in the trail before (or incl.) x_i.

if F contains unit clause it is possible for decision level to be 0.

for $C \in F \cup D$.

apply α to C. check if unit.] (can do unit prop like this, but very inefficient. we'll see how to do it better).

How to learn.

want to add some consequence of F to D

α, reached a conflict add negation of α to D.

$\alpha: x_1 \overset{d}{=} 0,\ x_2 \overset{u}{=} 1,\ x_3 \overset{u}{=} 0,\ x_4 \overset{d}{=} 1,\ x_5 \overset{u}{=} 0$ $\qquad \bar\alpha = (x_1 \vee \bar x_2 \vee x_3 \vee \bar x_4 \vee x_5)$.

α falsifies some $C \in F \cup D$. $\quad C \subseteq \bar\alpha \quad$ C is stronger than $\bar\alpha$.

learn that $x_1 \neq 0$ or $x_4 \neq 1$ $\quad x_1 \vee \bar x_4$. → $C \in D$

claim: we will never move into same branch of search tree again.

(iow: never reach state α again) (or any $\beta \supseteq \alpha$).

$x_1 = 0 \quad \cancel{x_1} \vee \bar x_4$ becomes unit. $x_4 = 0$.

$\beta \supseteq \alpha$ consider first time when $k-1$ decisions are made. $\gamma \subseteq \beta$.

γ agrees with α on $k-1$ decs. $C(\gamma)$ is unit, propagates var in α with value opposite to α.

$\gamma' = \gamma \cup \{x_i = b_i\} \qquad \alpha: \{x_i = 1 - b_i\}$. next state $\gamma' \not\subseteq \alpha \subseteq \beta$.

learning $\overline{\text{decisions}}$ is complete (algorithm will terminate)

is it sound? (does $\overline{\text{decisions}}$ follow from $F \wedge D$)

$\text{dec} = x_1 \overset{d}{=} 0 \wedge x_4 \overset{d}{=} 1$

$\overline{\text{dec}} = x_1 \vee \bar x_4$.

prop: there is a resolution derivation of $\overline{\text{dec}}$ from $F \wedge D$

res. proof is $(C_1, C_2, \ldots, C_L)$ $\quad C_L = !!$

C_i either in F, or follows from $\dfrac{C_j \quad C_k}{C_i}$ Res $\quad j, k < i$

res. derivation.

Res $\dfrac{C \vee x \quad C' \vee \bar x}{C \vee C'}$ (no opposite literals in C or C').

Weakening. $\dfrac{C}{C \vee x}$ x any variable.

$\dfrac{x \vee y \quad \bar x \vee \bar y}{y \vee \bar y}$ ⇝ satisfied? always.

$F = \{x \vee \bar z\}$. $\qquad F \wedge (y \vee \bar y) \equiv F$.

apply weakening $\dfrac{x \vee \bar z}{x \vee y \vee \bar z}$ ✓ $\quad \dfrac{x \vee \bar z}{x \vee \bar y \vee \bar z}$ ✓

want to derive C $\qquad x \vee y \vee \bar z$

only have $C' \subseteq C$. $\qquad x \vee \bar z$.

$C_{e,k_e} \quad E = E_{e,k_e}$

$E_{e,k_{e-1}}$

proof. (can derive $\overline{\text{dec}}$ in resolution).

trail $\quad x_{1,1} \overset{d}{=} b_{1,1} \quad x_{1,2} \overset{C_{1,2}}{=} b_{1,2} \quad \cdots \quad x_{1,k_1} \overset{C_{1,k_1}}{=} b_{1,k_1}$

$x_{2,1} \overset{d}{=} b_{2,1} \quad \cdots$

$\vdots$

$x_{e,1} \overset{d}{=} b_{e,1} \quad \cdots \quad x_{e,k_e} \overset{C_{e,k_e}}{=} b_{e,k_e}$.

$\exists$ clause $E \in F \cup D$ st $\alpha(E) = 0$. $\qquad E(\alpha) = \{x_{e,k_e} = 1 - b_{e,k_e}\}$

$\alpha \setminus \{x_{e,k_e}\}$ unit propagates because of C_{e,k_e}. → $C_{e,k_e}(\alpha) = \{x_{e,k_e} = b_{e,k_e}\}$

claim: $E_{e,k_{e-1}}$ falsified by $\alpha \setminus \{x_{e,k_e}\}$.

$\dfrac{C' \vee x^b \quad E' \vee x^{1-b}}{C' \vee E'}$

$C_{e,k_e} = C' \vee x_{e,k_e}^{b}$

$E = E' \vee x_{e,k_e}^{1-b}$

α fals. $C' \vee E'$. $= E_{e,k_{e-1}}$.

$\alpha(0 \vee 0) = 0$.

$\alpha \setminus \{x_{e,k_e}\}$ falsifies both C' and E'.

$x_1 \overset{d}{=} 0,\ x_2 \overset{x_1 \vee \bar x_2}{=} 0,\ x_3 \overset{x_1 \vee \bar x_3}{=} 0$

$x_4 \overset{d}{=} 0,\ x_5 \overset{x_2 \vee \bar x_5 \vee x_4}{=} 0,\ x_6 \overset{x_4 \vee x_2 \vee \bar x_6}{=} 0$

F.
- $x_1 \vee \bar x_2$
- $x_1 \vee \bar x_3$
- $x_2 \vee \bar x_5 \vee x_4$ → C_{e,k_e-1}
- $x_4 \vee x_2 \vee \bar x_6$ → $C_{e,k_e} = C' \vee \bar x_6$
- $x_1 \vee x_5 \vee x_6$ → $E = E' \vee x_6$.

$\alpha = (x_1 = 0, \ldots x_6 = 0)$.

$\alpha \setminus x_6$

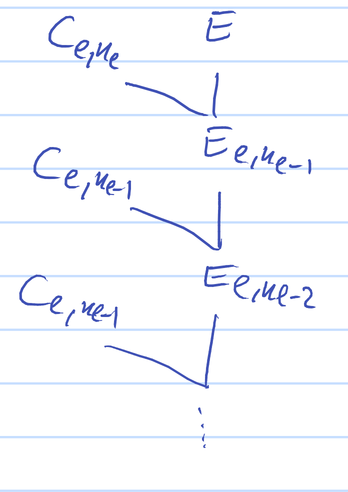
$\alpha \setminus x_6$ fals $C' = x_4 \vee x_2$.

$\alpha \setminus x_6$ fals $E' = x_1 \vee x_5$.

$\alpha \setminus x_6$ fals. $C' \vee E'$

$\dfrac{E \quad C_{e,k_e}}{C' \vee E'}$ Res

claim: $E_{e,k_{e-1}} = \text{Res}(E, C_{e,k_e})$ fals by $\alpha \setminus x_{e,k_e}$.



proof will end next day.

(after example).

PPSZ algorithm: good in worst-case.