Automatsko rezonovanje – beleške sa predavanja Iskazno rezonovanje

Milan Banković (po slajdovima Filipa Marića)

*Matematički fakultet,
Univerzitet u Beogradu

Prolećni semestar 2026/27.

Pregled

1 Metod tabloa

2 SAT problem

3 Metod iskazne rezolucije.

4 DPLL procedura

5 CDCL algoritam

Metod tabloa

O metodu tabloa

- Metod analitičkih tabloa
- Bet (Everth Beth), Hintika (Hintikka) 1955., Smaljan (Raymond Smullyan) 1971.
- Koristi se za ispitivanje (ne)zadovoljivosti iskaznih formula
  - može se koristiti za dokazivanje tautologičnosti pobijanjem
- Metod tabloa se primenjuje na formulu u proizvoljnom obliku
  - formula se ne transformiše u normalnu formu pre primene
  - zadržava se struktura originalne formule
- Tabloi se obično prikazuju u obliku stabla
  - dobijeno stablo se može smatrati dokazom nezadovoljivosti
  - metod tabloa – refutacioni deduktivni sistem
  - dokazi mogu biti čitljivi i razumljivi za čoveka

Iskazni tablo – pravila

Pravila

$$\frac{\neg\neg A}{A}$$

$$\frac{A \wedge B}{\begin{array}{c} A \\ B \end{array}}$$

$$\frac{\neg(A \vee B)}{\begin{array}{c} \neg A \\ \neg B \end{array}}$$

$$\frac{\neg(A \Rightarrow B)}{\begin{array}{c} A \\ \neg B \end{array}}$$

$$\frac{\neg(A \wedge B)}{\neg A \mid \neg B}$$

$$\frac{A \vee B}{A \mid B}$$

$$\frac{A \Rightarrow B}{\neg A \mid B}$$

Slična pravila se mogu definisati i za druge logičke veznike.

Konstrukcija tabloa

Postupak konstrukcije tabloa

1. Konstrukcija tabloa kreće od stabla koje u svom jedinom čvoru (korenu) sadrži formulu čija se zadovoljivost ispituje.
2. U svakom koraku, moguće je stablo proširiti naniže primenom nekog od datih pravila na neku formulu koja se nalaze u putanji od korena do lista koji se proširuje, a na koju to pravilo ranije nije bilo primenjeno u toj grani
3. Tablo koji se ne može proširiti, naziva se zasićenim.
4. Putanja je zatvorena ukoliko sadrži formulu i njenu negaciju (obično su u pitanju kontradiktorni literali). Tablo je zatvoren ako mu je svaka putanja zatvorena.

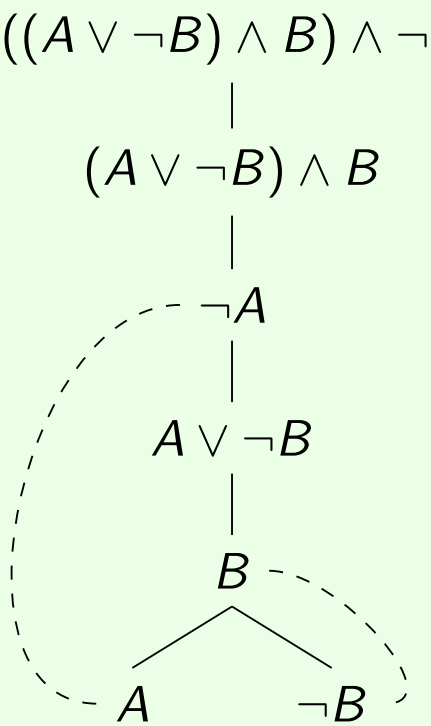
Teorema (Korektnost metoda tabloa)

Metod tabloa se zaustavlja za svaku iskaznu formulu i dobijeni tablo je zatvoren ako i samo ako je polazna iskazna formula nezadovoljiva.

Primer

Primer

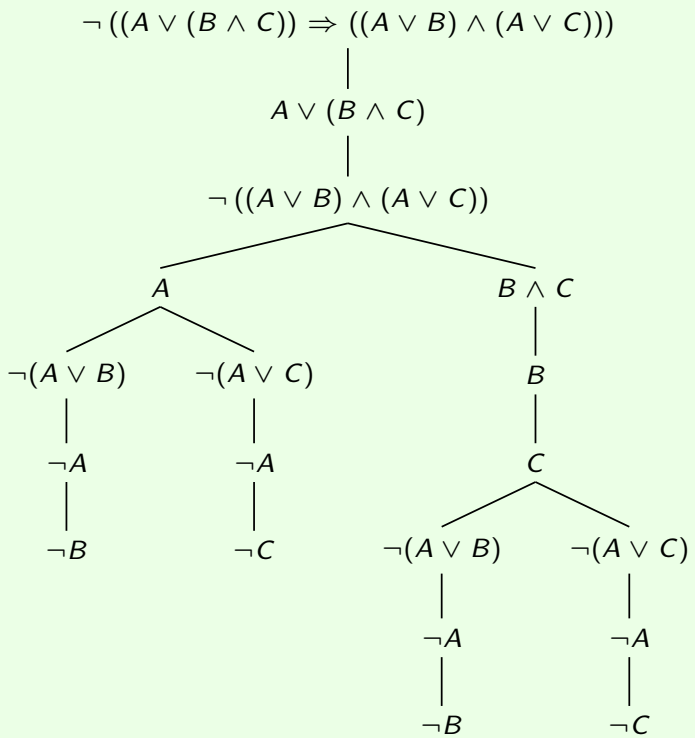
Pokažimo da je $((A \vee \neg B) \wedge B) \wedge \neg A$ *nezadovoljiva formula.*



Primer

Primer

Pokažimo da je $(A \vee (B \wedge C)) \Rightarrow ((A \vee B) \wedge (A \vee C)$ *tautologija.*



Redosled primene pravila

Heuristike

- Redosled primene pravila nije bitan za korektnost procedure.
- Bolja efikasnost se dobija ukoliko se negranajućim pravilima da prioritet.

Pregled

1 Metod tabloa

2 SAT problem

3 Metod iskazne rezolucije.

4 DPLL procedura

5 CDCL algoritam

SAT problem

Šta je SAT problem?

- Problem ispitivanja zadovoljivosti iskazne formule u KNF-u naziva se SAT problem (od engleske reči *satisfiability*).
  - ponekad se termin SAT odnosi i na opšti problem zadovoljivosti iskazne formule u proizvoljnom obliku
  - ipak, u najvećem delu literature, SAT označava specifičan slučaj zadovoljivosti KNF iskaznih formula
- Centralni problem teorijskog računarstva
  - SAT je prvi problem za koji je dokazano da je NP kompletan (Cook-Levin, 1971.)
  - Posledično, i problem zadovoljivosti iskazne formule u proizvoljnom obliku je NP-kompletan, jer je SAT njegov specijalni slučaj
- SAT ima ogromne praktične primene

SAT problem

Zašto KNF?

- Zadovoljivost formule u proizvoljnom obliku:
  - naivni metodi: metod istinitosnih tablica
  - metod tabloa: ne preterano efikasan
  - tehnički se jednostavnije rešava ako je formula u nekom jednostavnijem obliku (DNF, KNF)
- Za DNF se zadovoljivost trivijalno ispituje, ali je postupak prevođenja u DNF netrivijalan
  - u opštem slučaju dolazi do kombinatorne eksplozije
- Za KNF postoji efikasan postupak prevođenja formule u KNF koji čuva zadovoljivost (Cajtinova transformacija)
  - zbog toga je KNF prirodno ishodište transformacije formula
  - opšti problem zadovoljivosti se tako svodi na SAT, uz linearno uvećanje veličine ulaza
- Na žalost, problem ispitivanja zadovoljivosti ostaje NP kompletan i za formule u KNF-u

Pristupi rešavanju SAT problema

Pristupi

Zadovoljivost formula u KNF se ispituje nekim od tzv. klauzalnih algoritama:

- metod rezolucije
- DPLL procedura
- CDCL SAT rešavači (unapređena DPLL procedura)
- Stohastički SAT rešavači (zasnovani na randomizaciji — ne garantuju uvek definitivan odgovor)

O većini navedenih pristupa biće reči u nastavku.

SAT rešavači

SAT rešavači

Implementacije procedura odlučivanja za SAT problem nazivaju se SAT rešavači

Napomena

Moderni SAT rešavači, naročito oni zasnovani na CDCL algoritmu, su veoma efikasni!

SAT rešavači kao dokazivači valjanosti

SAT rešavači kao dokazivači valjanosti (tautologičnosti)

- Formula F je tautologija akko je $\neg F$ nezadovoljiva
- Otuda se SAT rešavači mogu koristiti i za dokazivanje tautologičnosti
  - da bismo dokazali da je F tautologija, njenu negaciju $\neg F$ svedemo na KNF i predamo SAT rešavaču
  - ako rešavač potvrdi da je $\neg F$ nezadovoljiva, tada je polazna formula F tautologija
  - ovakav način dokazivanja valjanosti se naziva dokazivanje pobijanjem (engl. refutation)
- U zavisnosti od procedure na kojoj je zasnovan, SAT rešavač može (ali ne mora) pružiti i dokaz nezadovoljivosti formule $\neg F$ (tj. valjanosti formule F) u nekom deduktivnom sistemu.

Svojstva procedura za ispitivanje (ne)zadovoljivosti

Saglasnost, potpunost, zaustavljanje

Imajući u vidu opisanu vezu između ispitivanja (ne)zadovoljivosti i dokazivanja valjanosti (pobijanjem), osobine SAT rešavača možemo formulisati na sledeći način:

Saglasnost – Ako algoritam prijavi nezadovoljivost, polazna formula je zaista nezadovoljiva

Potpunost – Ako je polazna formula nezadovoljiva, algoritam će prijaviti nezadovoljivost

Zaustavljanje – Za svaku ulaznu formulu, algoritam se zaustavlja nakon primene konačno mnogo koraka

Od SAT rešavača obično očekujemo da ima sve tri navedene osobine.

SAT rešavači kao procedure pretrage

SAT rešavači kao procedure pretrage

- U zavisnosti od algoritma na kome je zasnovan, SAT rešavač može na izlazu dati i zadovoljavajuću valuaciju u slučaju zadovoljive formule
  - takvi SAT rešavači se mogu koristiti i kao procedure za pretragu, a ne samo kao procedure odlučivanja
- Ovo omogućava primenu SAT rešavača u oblastima koje izlaze iz okvira dokazivanja tvrđenja:
  - mnogi problemi se mogu izraziti na jeziku iskazne logike
  - zadovoljavajuća valuacija kodira njihovo rešenje

DIMACS CNF

DIMACS format

- DIMACS CNF — Standardni format zapisa KNF formula u tekstualne datoteke.
- Koristi se kao ulaz SAT rešavača.

Primer

$$(x_1 \vee \neg x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge x_3$$

```
p cnf 3 4
1 -2 0
-2 3 0
-1 2 -3 0
3 0
```

Pregled

1 Metod tabloa

2 SAT problem

3 Metod iskazne rezolucije.

4 DPLL procedura

5 CDCL algoritam

Metod rezolucije

O metodu rezolucije

- Metod otkriven još 1960.-tih (Davis, Putnam, Robinson, . . .)
- Metod ispitivanja (ne)zadovoljivosti formule (ili skupa formula):
  - iskazne logike
  - logike prvog reda
- Nije metod pretrage – u slučaju zadovoljive formule ne daje zadovoljavajuću valuaciju
- Pogodan je za dokazivanje valjanosti formule pobijanjem
  - rezolucijski dokazi nezadovoljivosti se mogu izvesti i proveriti putem nezavisnog provaravača

Pravilo rezolucije

Pravilo rezolucije

$$\frac{\mathcal{C}' \cup \{l\} \quad \mathcal{C}'' \cup \{\bar{l}\}}{\mathcal{C}' \cup \mathcal{C}''}$$

Dobijena klauza $\mathcal{C}' \cup \mathcal{C}''$ se naziva rezolventa, dok se polazne klauze $\mathcal{C}' \cup \{l\}$ i $\mathcal{C}'' \cup \{\bar{l}\}$ nazivaju njenim roditeljima.
Rezolucija klauza $\mathcal{C}_1$ i $\mathcal{C}_2$ preko literala l će biti označavana i sa $\mathcal{C}_1 \oplus_l \mathcal{C}_2$.

Napomena

Suštinski, opravdanje ovog pravila je u tranzitivnosti implikacije.

$$\frac{\overline{\mathcal{C}'} \Rightarrow l \quad l \Rightarrow \mathcal{C}''}{\overline{\mathcal{C}'} \Rightarrow \mathcal{C}''}$$

Napomena

Pravilo rezolucije podrazumeva da su klauze skupovi, tj. da u njima nema ponavljanja literala. Ako u programu klauze predstavljamo listama, tada moramo obezbediti da one ne sadrže duplikate. Alternativa je da rezolviramo istovremeno sve kopije odgovarajućeg literala (tzv. opšte pravilo rezolucije).

Metod rezolucije

Algoritam

Neka je $\mathcal{S}$ skup klauza čiju (ne)zadovoljivost ispitujemo:

- Dokle god postoje dve klauze C' i C'' iz $\mathcal{S}$ i literal l takve da rezolventa $\mathcal{C}_1 \oplus_l \mathcal{C}_2$ ne pripada $\mathcal{S}$:
  - dodaj $\mathcal{C}_1 \oplus_l \mathcal{C}_2$ u skup $\mathcal{S}$
  - ako je dodata klauza $\mathcal{C}_1 \oplus_l \mathcal{C}_2$ prazna, završi algoritam i prijavi nezadovoljivost
- Kada više ne postoje nove rezolvente koje možemo izvesti i dodati u $\mathcal{S}$, prijavljujemo zadovoljivost

Metod rezolucije — primeri

Primer

$$\{\{\neg p, \neg q, r\}, \{\neg p, q\}, \{p\}, \{\neg r\}\}$$

$\mathcal{C}_1$	:	$\{\neg p, \neg q, r\}$	
$\mathcal{C}_2$	:	$\{\neg p, q\}$	
$\mathcal{C}_3$	:	$\{p\}$	
$\mathcal{C}_4$	:	$\{\neg r\}$	
$\mathcal{C}_5$	:	$\{\neg p, r\}$	$\mathcal{C}_1 \oplus_q \mathcal{C}_2$
$\mathcal{C}_6$	:	$\{\neg p\}$	$\mathcal{C}_4 \oplus_r \mathcal{C}_5$
$\mathcal{C}_7$	:	$\{\}$	$\mathcal{C}_3 \oplus_p \mathcal{C}_6$

Zaustavljanje rezolucije

Teorema

Metod rezolucije se zaustavlja.

Dokaz

Pošto je početni skup atoma konačan, postoji konačno mnogo različitih klauza sagrađenih od njegovih elemenata (za n atoma, postoji 2^{2n} različitih klauza). U svakom koraku u skup klauza se dodaje nova klauza koja sadrži samo literale iz polaznog skupa, te može da bude samo konačno mnogo koraka.

Saglasnost rezolucije

Lema (Saglasnost pravila rezolucije)

Ako je $\mathcal{C}_1 \in \Gamma$ *i* $\mathcal{C}_2 \in \Gamma$, *skupovi klauza* Γ *i* $\Gamma \cup \{\mathcal{C}_1 \oplus_l \mathcal{C}_2\}$ *su logički ekvivalentni.*

Dokaz

Ako je v *model za* $\Gamma \cup \{\mathcal{C}_1 \oplus_l \mathcal{C}_2\}$, *tada je, trivijalno,* v *model i za* Γ. *Obratno, neka je* v *model za* Γ. *Razmotrimo sledeće slučajeve:*

- $v \not\models l$. *Tada u* $\mathcal{C}_1$ *postoji literal* l' *različit od* l *takav da* $v \models l'$. *No,* $l' \in \mathcal{C}_1 \oplus_l \mathcal{C}_2$ *te* $v \models \mathcal{C}_1 \oplus_l \mathcal{C}_2$.
- $v \models l$. *Tada* $v \not\models \bar{l}$ *pa u* $\mathcal{C}_2$ *postoji literal* l' *različit od* $\bar{l}$ *takav da* $v \models l'$. *No,* $l' \in \mathcal{C}_1 \oplus_l \mathcal{C}_2$ *te* $v \models \mathcal{C}_1 \oplus_l \mathcal{C}_2$.

Saglasnost metoda rezolucije

Teorema

Ako metod rezolucije prijavi nezadovoljivost (tj. ako izvede praznu klauzu), onda je polazni skup klauza nezadovoljiv.

Dokaz

U svakom koraku metoda rezolucije u skup klauza se dodaje rezolventa neka dva njegova člana i novodobijeni skup je ekvivalentan prethodnom, pa samim tim, na osnovu tranzitivnosti logičke ekvivalentnosti, i polaznom. Kako poslednji skup sadrži praznu klauzu, on nije zadovoljiv pa ne može biti ni njemu ekvivalentan polazni skup klauza.

Potpunost rezolucije

Teorema

Ako je polazni skup klauza nezadovoljiv, onda metod rezolucije prijavljuje nezadovoljivost (tj. izvodi praznu klauzu).

Dokaz

Skica dokaza: posmatramo parcijalne valuacije oblika $\{l_1, \ldots, l_k\}$, $k \leq n$, *gde je* $l_i \in \{p_i, \neg p_i\}$. *Za ovakvu parcijalnu valuaciju kažemo da je pokrivena skupom klauza* $\mathcal{S}$, *ako postoji klauza* $C \in \mathcal{S}$ *koja je netačna za tu parcijalnu valuaciju. Pretpostavimo da imamo nezadovoljiv skup klauza* $\mathcal{S}$ *za koji smo iscrpno primenili pravilo rezolucije na sve moguće načine. Dokažimo da* $\mathcal{S}$ *mora sadržati praznu klauzu. Zaista, sve potpune valuacije su sigurno pokrivene skupom* $\mathcal{S}$, *jer je* $\mathcal{S}$ *nezadovoljiv. Dalje, ako imamo dve ,,susedne" parcijalne valuacije* $\{l_1, \ldots, l_{k-1}, l_k\}$ *i* $\{l_1, \ldots, l_{k-1}, \overline{l}_k\}$, *tada iz činjenice da su obe pokrivene (npr. klauzama* C *i* C'*), sledi da je pokrivena i parcijalna valuacija* $\{l_1, \ldots, l_{k-1}\}$: *zaista, ako je* $\overline{C}$ *(ili* $\overline{C'}$*) potskup od* $\{l_1, \ldots, l_{k-1}\}$, *tada ta klauza pokriva i* $\{l_1, \ldots, l_{k-1}\}$; *u suprotnom, klauze* C *i* C' *sadrže, respektivno, literale* $\overline{l}_k$ *i* l_k, *pa je klauza* $C \oplus_{l_k} C'$ *sigurno u* $\mathcal{S}$, *jer je rezolucija iscrpno primenjena, a ova klauza tada pokriva valuaciju* $\{l_1, \ldots, l_{k-1}\}$. *Na ovaj način, indukcijom možemo dokazati da su pokrivene sve parcijalne valuacije oblika* $\{l_1, \ldots, l_k\}$ *za svako* $k \geq 0$. *Za* $k = 0$ *imamo praznu valuaciju koja može biti pokrivena jedino praznom klauzom, pa ona mora pripadati skupu* $\mathcal{S}$.

Efikasnost metoda rezolucije

Pitanje efikasnosti i strategije pretrage

- Da bi metod rezolucije imao svojstvo potpunosti, potrebno je da se rezolucija primenjuje iscrpno
  - ovakav pristup se naziva sistematski metod rezolucije
- Ipak, nije svejedno u kom poretku će se primenjivati pravilo rezolucije nad različitim klauzama (i literalima)
  - tu na scenu stupaju različite heuristike
  - ipak, složenost je u najgorem slučaju eksponencijalna
- U nekim slučajevima potpunost je moguće obezbediti čak i bez sistematske primene rezolucije:
  - npr. ako su sve klauze takve da imaju najviše jedan pozitivan literal (tzv. Hornove klauze), tada se potpunost može obezbediti linearnom ulaznom strategijom: pravilo rezolucije se uvek primenjuje nad poslednjom izvedenom rezolventom i nekom od početnih (ulaznih) klauza
  - opisanu strategiju koristi programski jezik Prolog (doduše, u logici prvog reda)

Pregled

1 Metod tabloa

2 SAT problem

3 Metod iskazne rezolucije.

4 DPLL procedura

5 CDCL algoritam

DPLL procedura

Pregled

- DPLL (Davis-Putnam-Logemann-Loveland, 1962)
- Ispituje zadovoljivost iskaznih formula (u KNF)
  - kao i ranije, i ovde je ulaz dat kao skup klauza, pri čemu je svaka klauza skup literala
- Izmena DP procedure (Davis-Putnam, 1961) u cilju smanjivanja utroška memorije
- Osnova savremenih CDCL SAT rešavača
- Predstavlja proceduru pretrage: ako je skup klauza zadovoljiv, vraća zadovoljavajuću valuaciju
- Razlikovaćemo DPLL pretragu i DPLL proceduru (koja uz pretragu uključuje i elemente zaključivanja kojima se sužava prostor pretrage)

DPLL procedura — ideja koraka pretrage

Pretraga

- Pretraga za zadovoljavajućom valuacijom zasniva se na ispitivanju obe moguće istinitosne vrednosti promenljivih koje se javljaju u formuli
- Postavljanje vrednosti neke promenljive indukuje vrednost oba njoj odgovarajuća literala
  - postavljanje vrednosti promenljive p na *tačno* čini da je literal p tačan, a $\neg p$ netačan
  - postavljanje vrednosti promenljive p na *netačno* čini da je literal p netačan, a $\neg p$ tačan
  - otuda ćemo umesto ,,postavljanja promenljive p na tačno (netačno)" reći ,,postavljanje literala p ($\neg p$) na tačno"
- Vrednost nekog literala l se postavi na tačno
- Postavljanje literala na tačno dovodi do uprošćavanja formule:
  - klauze koje sadrže l su tačne pa se brišu
  - iz klauza koje sadrže $\overline{l}$ se taj literal briše, jer je netačan
- za uprošćeni skup klauza se rekurzivno proverava zadovoljivost
- Ako je nakon postavljanja l na tačno dobijena nezadovoljivost, pokušava se sa postavljanjem $\overline{l}$ na tačno
- Ako se i u tom slučaju dobije nezadovoljivost, formula je nezadovoljiva

DPLL procedura — $F[[l \to \top]]$

Uprošćavanje formule

- Efekat uprošćavanja formule postavljanjem nekog literala na tačno se može postići tako što se sva njegova pojavljivanja zamene logičkom konstantom $\top$, a pojavljivanja njemu suprotnog literala zamene logičkom konstantom $\bot$, a zatim primeni eliminacija logičkih konstanti
  - rezultat je upravo ono što je već rečeno: klauze koje sadrže tačan literal se brišu, a iz klauza koje sadrže suprotan literal se briše taj literal
- Neka $F[[l \to \top]]$ označava formulu $F[l \to \top][\bar{l} \to \bot]$
- Neka $F[[l \to \bot]]$ označava formulu $F[\bar{l} \to \top][l \to \bot]$ (tj. $F[[\bar{l} \to \top]]$)

Korak split

Split

Pretraga u okviru DPLL procedure je zasnovana na koraku split koji ispitivanje zadovoljivosti formule F svodi na ispitivanje zadovoljivosti formula $F[[I \to \top]]$ i $F[[I \to \bot]]$.
Logičko opravdanje za ovo je u ekvizadovoljivosti:

$$F \equiv_s F[[I \to \top]] \vee F[[I \to \bot]]$$

DPLL i tautologične klauze

Tautologične klauze

- Klauze koje sadrže međusobno suprotne literale ne utiču na zadovoljivost formule (jer su uvek tačne)
- Ove klauze je moguće ukloniti pre primene procedure (često već u fazi prevođenja u KNF)
- Primetimo da ne postoji mogućnost da se naknadno pojave nove tautologične klauze

DPLL pretraga — trivijalni slučajevi

Izlaz iz rekurzije

- Kako je DPLL pretraga rekurzivna procedura, mora imati i izlaz iz rekurzije
- Mogu nastupiti dva ,,trivijalna" slučaja:
  - skup klauza postane prazan nakon uprošćavanja: DPLL pretraga prijavljuje zadovoljivost
  - prilikom uprošćavanja u skupu klauza se pojavi prazna klauza: DPLL pretraga prijavljuje nezadovoljivost

DPLL pretraga — pseudokod

Pseudokod DPLL pretrage

```
function dpll (F : CNF Formula) : (SAT, UNSAT)
begin
   if F is empty then return SAT
   else if there is an empty clause in F then return UNSAT
   else begin
      select a literal l occurring in F
      if dpll(F[[l → ⊤]]) = SAT then return SAT
      else return dpll(F[[l → ⊥]])
   end
end
```

Korektnost koraka split

Stav (Split)

Skup klauza $\mathcal{F}$ je zadovoljiv, ako i samo ako je zadovoljiv $\mathcal{F}[[l \to \top]]$ ili je zadovoljiv $\mathcal{F}[[l \to \bot]]$, gde je l proizvoljan literal.

Dokaz

Neka je v model za $\mathcal{F}$. Ako je $v \vDash l$, onda je v model i za $\mathcal{F}[[l \to \top]]$, a ako $v \nvDash l$, pošto $v \vDash \bar{l}$, onda je v model za $\mathcal{F}[[l \to \bot]]$.
Ako je v model za $\mathcal{F}[[l \to \top]]$, onda postoji v' (dobijena od v postavljanjem vrednosti l na tačno) tako da je v' model za $\mathcal{F}$.
Ako je v model za $\mathcal{F}[[l \to \bot]]$, onda postoji v' (dobijena od v postavljanjem vrednosti l na netačno) tako da je v' model za $\mathcal{F}$.

Korektnost trivijalnih slučajeva

Stav

- *Prazan skup klauza je zadovoljen u svakoj valuaciji.*
- *Prazna klauza nema model.*
- *Ako skup formula sadrži praznu klauzu on nema model (tj. nije zadovoljiv).*

Korektnost DPLL pretrage

Teorema (Zaustavljanje)

Ukoliko se korak split uvek primenjuje na literal koji je deo tekućeg skupa klauza, DPLL pretraga se zaustavlja.

Dokaz

Broj različitih promenljivih u tekućem skupu klauza se smanjuje pri svakom rekurzivnom pozivu, pa se procedura mora zaustaviti.

Teorema (Saglasnost i potpunost)

Skup klauza $\mathcal{F}$ je zadovoljiv ako i samo je `dpll` $\mathcal{F} = True$.

Dokaz

Indukcijom, na osnovu definicije funkcije `dppl`*, uz korišćenje lema o korektnosti pojedinačnih koraka.*

Elementi zaključivanja

Elementi rezonovanja (zaključivanja)

- Prethodna procedura ne uključuje nikakve elemente zaključivanja
- U nekim slučajevima može se izbeći analiziranje obe grane u split pravilu
- DPLL procedura uvodi dva ovakva koraka koji značajno mogu da smanje prostor pretrage:
  - propagacija jediničnih klauza (eng. unit propagation)
  - eliminacija ,,čistih" literala (eng. pure literal)

Primeri

Primer

$$\mathcal{F}_0 = \{\{\neg x_1, \neg x_2\}, \{x_2\}, \{\neg x_2, x_3, \neg x_4\}, \{\neg x_3, x_4\}, \{x_4, x_5\}, \{\neg x_3, \neg x_4, x_1\}\}$$

Unit propagate: *valuacija koja x_2 postavlja na netačno ne može biti model.*

$$\mathcal{F}_1 = \mathcal{F}_0[[x_2 \to \top]] = \{\{\neg x_1\}, \{x_3, \neg x_4\}, \{\neg x_3, x_4\}, \{x_4, x_5\}, \{\neg x_3, \neg x_4, x_1\}\}$$

Unit propagate: *valuacija koja x_1 postavlja na tačno ne može biti model.*

$$\mathcal{F}_2 = \mathcal{F}_1[[x_1 \to \bot]] = \{\{x_3, \neg x_4\}, \{\neg x_3, x_4\}, \{x_4, x_5\}, \{\neg x_3, \neg x_4\}\}$$

Pure literal: *Ne smeta da se x_5 postavi na tačno.*

$$\mathcal{F}_3 = \mathcal{F}_2[[x_5 \to \top]] = \{\{x_3, \neg x_4\}, \{\neg x_3, x_4\}, \{\neg x_3, \neg x_4\}\}$$

Tek sada kreće pretraga.

Elementi zaključivanja

Pravila zaključivanja

Jedinična propagacija (engl. *unit propagation*) — Ako formula sadrži klauzu sa tačno jednim literalom $\{l\}$, onda taj literal mora biti tačan u eventualnoj zadovoljavajućoj valuaciji. Klauze sa samo jednim literalom nazivamo jedinične klauze (engl. unit clause).

Čisti literal (engl. *pure literal*) — Ako se u formuli javlja neki literal l, a ne javlja se njemu suprotan literal $\bar{l}$, onda taj literal može biti postavljen na tačno u eventualnoj zadovoljavajućoj valuaciji. Ovakav literal nazivamo čist literal.

DPLL procedura — pseudokod

Algoritam

```
function dpll (F : CNF Formula) : bool
begin
   if F is empty then return True
   else if there is an empty clause in F then return False
   else if there is a pure literal l in F then return dpll(F[[l → ⊤]])
   else if there is a unit clause [l] in F then return dpll(F[[l → ⊤]])
   else begin
      select a literal l occurring in F
      if dpll(F[[l → ⊤]]) = SAT then return SAT
      else return dpll(F[[l → ⊥]])
   end
end
```

Korektnost koraka propagacije jediničnih klauza

Stav (Unit propagate)

Ukoliko je $\{l\} \in \mathcal{F}$*, tada je*

$$\mathcal{F} \equiv_s \mathcal{F}[[l \to \top]].$$

Dokaz

Neka je v *model za* $\mathcal{F}$*. Pošto je* $\{l\} \in \mathcal{F}$*, važi* $v \vDash l$*, te* $v \vDash \mathcal{F}[[l \to \top]]$.
Ako je v *model za* $\mathcal{F}[[l \to \top]]$*, onda postoji* v' *(dobijena od* v *postavljanjem vrednosti* l *na tačno) tako da je* v' *model za* $\mathcal{F}$.

Korektnost koraka eliminacije čistih literala

Stav (Pure literal)

Ako se u skupu klauza $\mathcal{F}$ javlja literal l, a ne javlja literal $\bar{l}$, tada je

$$\mathcal{F} \equiv_s \mathcal{F}[[l \to \top]].$$

Dokaz

Ako je $\mathcal{F}$ zadovoljiv, zadovoljiv je i $\mathcal{F}[[l \to \top]]$ (kao njegov potskup). Ako je v model za $\mathcal{F}[[l \to \top]]$, onda postoji v' (dobijena od v postavljanjem vrednosti l na tačno) tako da je v' model za $\mathcal{F}$.

Efikasnost DPLL procedure

Efikasnost

- U najgorem slučaju, DPLL procedura zahteva eksponencijalni broj koraka u odnosu na veličinu ulazne formule (korak split generiše eksponencijalnost). Ovaj problem nije moguće razrešiti.
- Osnovna varijanta procedure je izrazito neefikasna:
  - Pri svakom koraku vrši se modifikacija formule (skupa klauza).
  - Rekurzivna implementacija značajno kvari efikasnost (formule koje mogu da budu veoma velike se prenose kao parametar funkcije i slažu na programski stek).

Pregled

1 Metod tabloa

2 SAT problem

3 Metod iskazne rezolucije.

4 DPLL procedura

5 CDCL algoritam

O CDCL algoritmu

CDCL – Conflict Driven Clause Learning

- CDCL algoritam je nastao početkom 21. veka, sa ciljem unapređenja efikasnosti klasičnog DPLL algoritma
- Unapređenja idu u nekoliko pravaca:
  - inkrementalna izgradnja parcijalne valuacije, umesto transformacija formule
  - iterativna implementacija umesto rekurzivne
  - efikasan obilazak skupa klauza zasnovan na pogodnim strukturama podataka
  - efikasne heuristike grananja
  - analiza konflikata zasnovana na rezoluciji i nehronološko vraćanje unazad
  - učenje klauza iz konflikata, u cilju sprečavanja sličnih konflikata u budućnosti
- Sva ova unapređenja detaljno razmatramo u nastavku

Prosleđivanje parcijalnih valuacija

Alternativni pristup – parcijalne valuacije

- Umesto zamene izabranog literala sa $\top$ u formuli (i potom pojednostavljivanja skupa klauza) moguće je da se formula ne menja, a da se literal postavlja na tačno tako što se dodaje u parcijalnu valuaciju
- Podsetnik: parcijalna valuacija može da se shvati kao skup literala koji je konzistentan tj. ne sadrži istovremeno dva suprotna literala
- Pošto je valuacija parcijalna, promenljiva, odnosno literal, mogu u njoj da budu tačni, netačni ili nedefinisani
- Sada ćemo imati samo jednu formulu $\mathcal{F}$ koja se ne menja i koja je globalno vidljiva, a rekurzivnim pozivima ćemo prosleđivati parcijalne valuacije

Pojam jedinične klauze

Jedinične klauze – uopštenje

- Ukoliko se vrši uprošćavanje formule, iz klauza se uklanjaju literali
  - otuda neka klauza može postati jedinična nakon pojednostavljivanja i ako prvobitno to nije bila
- U slučaju da se ne vrši uprošćavanje formule već izgradnja parcijalne valuacije, pojam jediničnih klauza se mora uopštiti:

Definicija

Klauza c je jedinična u odnosu na parcijalnu valuaciju v akko sadrži literal l nedefinisan u v, dok su joj svi literali različiti od l netačni u v.

Čisti literali – uopštenje

Slično se može proširiti i pojam čistih literala:

Definicija

Literal l je čist u odnosu na parcijalnu valuaciju v ako je nedefinisan u v i svaka klauza $C \in \mathcal{F}$ ili tačna u v, ili ne sadrži literal $\overline{l}$.

DPLL procedura — prosleđivanje valuacija — pseudokod

Algoritam

```
function dpll (M : Valuation) : (SAT, UNSAT)
begin
   if M ⊨¬ F (i.e., there is c ∈ F s.t. M ⊨¬ c) then return UNSAT
   else if M ⊨ F then return SAT
   else if there is a unit clause (i.e., there is a clause
      l ∨ l1 ∨ ... ∨ lk in F s.t. l, l̄ ∉ M, l̄1, ..., l̄k ∈ M) then
      return dpll(M ∪ {l})
   else if there is a pure literal l in F then
      return dpll(M ∪ {l})
   else begin
      select a literal l s.t. l ∈ F, l, l̄ ∉ M
      if dpll(M ∪ {l}) = SAT then return SAT
      else return dpll(M ∪ {l̄})
   end
end
```

Provera tačnosti u parcijalnoj valuaciji?

Problem?

- Pokazuje se da postoje efikasni postupci za proveru da li u formuli postoji netačna ili jedinična klauza, ali ne postoji efikasan postupak kojim bi se utvrdilo da li je formula tačna u nekoj parcijalnoj valuaciji.
- Može li se taj test izbeći?

Stav

Ukoliko su sve promenljive iz $\mathcal{F}$ definisane u parcijalnoj valuaciji M, tada je $\mathcal{F}$ ili tačna ili netačna u M, tj. $M \models \mathcal{F}$ ili $M \models \neg \mathcal{F}$.

DPLL procedura — prosleđivanje valuacija — pseudokod

Algoritam

```
function dpll (M : Valuation) : (SAT, UNSAT)
begin
   if M ⊨¬ F (i.e., there is c ∈ F s.t. M ⊨¬ c) then return UNSAT
   else if M is total wrt. the variables of F then return SAT
   else there is a unit clause (i.e., there is a clause
      l ∨ l1 ∨ ... ∨ lk in F s.t. l, l̄ ∉ M, l̄1, ..., l̄k ∈ M) then
      return dpll(M ∪ {l})
   else if there is a pure literal l in F then
      return dpll(M ∪ {l})
   else begin
      select a literal l s.t. l ∈ F, l, l̄ ∉ M
      if dpll(M ∪ {l}) = SAT then return SAT
      else return dpll(M ∪ {l̄})
   end
end
```

Primer

Primer

$$\mathcal{F} = \{\{\neg x_1, \neg x_2\}, \{x_2\}, \{\neg x_2, x_3, \neg x_4\}, \{\neg x_3, x_4\}, \{x_4, x_5\}\}$$

$$v = \{\}$$

Unit propagate: *Klauza* $\{x_2\}$ *je jedinična.*

$$v = \{x_2\}$$

Unit propagate: *Klauza* $\{\neg x_1, \neg x_2\}$ *je jedinična.*

$$v = \{x_2, \neg x_1\}$$

Pure literal: *Literal* x_5 *je čist.*

$$v = \{x_2, \neg x_1, x_5\}$$

Tek sada kreće pretraga.

Primer

$$\mathcal{F} = \{\{\neg x_1, \neg x_2\}, \{x_2\}, \{\neg x_2, x_3, \neg x_4\}, \{\neg x_3, x_4\}, \{x_4, x_5\}, \{\neg x_3, \neg x_4, x_1\}\}$$

dpll($\{\}$)
dpll($\{x_2\}$) — *unitPropagate* ($c = \{x_2\}$)
dpll($\{x_2, \neg x_1\}$) — *unitPropagate* ($c = \{\neg x_1, \neg x_2\}$)
dpll($\{x_2, \neg x_1, x_5\}$) – *pureLiteral* (x_5)
dpll($\{x_2, \neg x_1, x_5, x_3\}$) — *split* ($x_3$)
dpll($\{x_2, \neg x_1, x_5, x_3, x_4\}$) — *unitPropagate* ($c = \{\neg x_3, x_4\}$)
return False — $c = \{\neg x_3, \neg x_4, x_1\}$

dpll($\{x_2, \neg x_1, x_5, \neg x_3\}$) — *split* ($x_3$) – *druga grana*
dpll($\{x_2, \neg x_1, x_5, \neg x_3, \neg x_4\}$) — *unitPropagate* ($c = \{\neg x_2, x_3, \neg x_4\}$)
return True

Iterativna implementacija

Predstavljanje parcijalne valuacije pomoću steka

Rekurzivna implementacija se može zameniti iterativnom, tako što se parcijalna valuacija predstavi stekom označenih literala:

- čuvanje parcijalne valuacije u obliku steka omogućava efikasno vraćanje unazad (engl. backtracking)
- literali na steku su označeni ili kao literali odlučivanja (engl. decision literal) ili kao izvedeni literali (engl. inferred literal)
- literali odlučivanja su rezultat naših odluka i njima započinju nivoi odlučivanja
- izvedeni literali su rezultat rezonovanja (jedinične propagacije, čisti literali) i posledica su odluka koje smo doneli
- stek označenih literala se često naziva i trag (engl. trail)

Apstraktni opis procedure

Apstraktni opis procedure

- Razmatranje procedure opisane na nivou koda je obično suviše komplikovano jer sadrži dosta implementacionih detalja
- Umesto koda, moguće je opisati tekuće stanje procedure i dati pravila koja opisuju kako se može preći iz stanja u stanje
- Ovakav način apstraktnog opisa procedura omogućava lakšu analizu i dokazivanje korektnosti procedure
- U slučaju CDCL procedure, stanje je predstavljeno stekom parcijalne valuacije (tragom)

Apstraktni opis procedure

Sistem pravila za promenu stanja

Stanje rešavača

- M - označena parcijalna valuacija u vidu steka (trag)

Decide:

$$\frac{l \in F \quad l, \bar{l} \notin M}{M := M \,|\, l}$$

UnitPropagate:

$$\frac{l \vee l_1 \vee \ldots \vee l_k \in F \qquad \bar{l}_1, \ldots, \bar{l}_k \in M \qquad l, \bar{l} \notin M}{M := M \; l}$$

Backtrack:

$$\frac{M \models \neg F \quad M = M' \,|\, l \, M'' \quad \text{decisions } M'' = [\,]}{M := M' \; \bar{l}}$$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	*M*
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	M
	UNDEF	
Decide ($l = +1$)	*UNDEF*	$\mid +1$
UnitProp ($c = [-1, +2]$, $l = +2$)	*UNDEF*	$\mid +1, +2$
UnitProp ($c = [-1, -3]$, $l = -3$)	*UNDEF*	$\mid +1, +2, -3$
Decide ($l = +4$)	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp ($c = [-4, +5]$, $l = +5$)	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack ($M \models \neg [+3, -4, -5]$)	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp ($c = [-2, +4, +5]$, $l = +5$)	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, (vars M) = (vars F)	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	*M*
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	*M*
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	*M*
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	*M*
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	M
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	M
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Primer

$F = [[-1, +2], [-1, -3], [-2, +4, +5], [+3, -4, -5], [-4, +5]]$

Primenjeno pravilo	*sat?*	*M*
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +1, +2$
UnitProp $(c = [-1, -3],\ l = -3)$	*UNDEF*	$\mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$\mid +1, +2, -3, \mid +4$
UnitProp $(c = [-4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, \mid +4, +5$
Backtrack $(M \models \neg [+3, -4, -5])$	*UNDEF*	$\mid +1, +2, -3, -4$
UnitProp $(c = [-2, +4, +5],\ l = +5)$	*UNDEF*	$\mid +1, +2, -3, -4, +5$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$\mid +1, +2, -3, -4, +5$

Efikasnost implementacije

Kako efikasno pronaći konfliktne i jedinične klauze?

- Skup klauza može biti veoma veliki
- Prolazak kroz skup svih klauza radi pronalaženja konfliktnih i jediničnih klauza može biti veoma skup
- Efikasna implementacija: shema dva posmatrana literala (engl. two-watched-literals scheme)
- Zasniva se na observaciji da dokle god u klauzi postoje bar dva literala čija je vrednost nedefinisana u tekućoj parcijalnoj valuaciji, ona ne može biti ni jedinična ni konfliktna

Shema dva posmatrana literala

Shema dva posmatrana literala

- Za svaku klauzu C se odaberu dva trenutno nedefinisana literala za posmatranje ($watch1(C)$ i $watch2(C)$)
- Za svaki literal l održavamo listu $watched(l)$ svih klauza u kojima je l jedan od posmatranih literala
- Kada se literal $\bar{l}$ doda u parcijalnu valuaciju, obilazi se lista $watched(l)$
- Za svaku klauzu $C \in watched(l)$ traži se alternativni posmatrani literal (tj. neki literal $l' \in C$ različit od $watch1(C)$ i $watch2(C)$ koji nije netačan u toj parcijalnoj valuaciji, ako postoji)
- Ako ga pronadjemo, tada tu klauzu prebacujemo u listu $watched(l')$
- U suprotnom, ako je drugi posmatrani literal takodje netačan u tekućoj parcijalnoj valuaciji, prijavljujemo konflikt
- Inače, propagiramo drugi posmatrani literal

Analiza konflikata

Šta je uzrok konflikta?

- Kada neka klauza postane netačna, uzrok često nije u poslednjem literalu odlučivanja
  - stoga, njegovom negacijom ponovo dobijamo konflikt
- Vrlo često je uzrok u nekom ranijem literalu odlučivanja, ali se to ne vidi odmah:
  - mi rezonovanjem donosimo zaključke isključivo na osnovu tekućeg skupa klauza
  - međutim, postoje klauze koje nisu u našem skupu, ali jesu logička posledica našeg skupa (npr. klauze koje bismo mogli dobiti pravilom rezolucije)
  - da su nam te klauze na raspolaganju, mnoge konflikte otkrili bismo znatno ranije
- Ideja je da, kada se već desi konflikt, probamo da otkrijemo njegov pravi uzrok (tj. izvedenu klauzu koja je postala netačna znatno ranije):
  - ovo će nam omogućiti da se vratimo na neki dublji nivo na steku
  - ovim se izbegava nepotrebni prolazak kroz grane prostora pretrage koje sigurno ne sadrže zadovoljavajuću valuaciju

Primer

$F_0 = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]]$.

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\mid +6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\mid +6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\mid +6, +1, +2, \mid +7$
Decide $(l = +3)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3, +4, +5$
Backtrack $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\mid +6, +1, +2, \mid +7, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, -3, +5$
Backtrack $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$\mid +6, +1, +2, -7$
Decide $(l = +3)$	*UNDEF*	$\mid +6, +1, +2, -7, \mid +3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\mid +6, +1, +2, -7, \mid +3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, -7, \mid +3, +4, +5$
Backtrack $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\mid +6, +1, +2, -7, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, -7, -3, +5$
Backtrack $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	-6

Primer

$F_0 = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]]$.

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6], l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2], l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4], l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5], l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Backtrack $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, -3$
UnitProp $(c = [-2, +3, +5, -6], l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, -3, +5$
Backtrack $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$\|+6, +1, +2, -7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, -7, \|+3$
UnitProp $(c = [-3, +4], l = +4)$	*UNDEF*	$\|+6, +1, +2, -7, \|+3, +4$
UnitProp $(c = [-1, -3, +5], l = +5)$	*UNDEF*	$\|+6, +1, +2, -7, \|+3, +4, +5$
Backtrack $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, -7, -3$
UnitProp $(c = [-2, +3, +5, -6], l = +5)$	*UNDEF*	$\|+6, +1, +2, -7, -3, +5$
Backtrack $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	-6

Problemi

Kako otkriti pravi uzrok konflikta?

- Niz koraka nakon pretpostavke $+7$ je pokazao da ni $+3$ ni -3 nisu kompatibilni sa prethodnim pretpostavkama.
- pravilo Backtrack menja uvek samo poslednju pretpostavku.
- Isti niz koraka pokazuje da ni ovaj put ni $+3$ ni -3 nisu kompatibilni sa prethodnim pretpostavkama.
- Pretpostavka $+7$ je potpuno irelevantna za konflikt koji se dogodio i ponavljanje postupka sa -7 je gubitak vremena.

Rešenje: analiza konflikata (eng. conflict analysis) i uvođenje povratnih skokova (eng. backjumping).

Problemi

Kako otkriti pravi uzrok konflikta?

- Niz koraka nakon pretpostavke $+7$ je pokazao da ni $+3$ ni -3 nisu kompatibilni sa prethodnim pretpostavkama.
- pravilo Backtrack menja uvek samo poslednju pretpostavku.
- Isti niz koraka pokazuje da ni ovaj put ni $+3$ ni -3 nisu kompatibilni sa prethodnim pretpostavkama.
- Pretpostavka $+7$ je potpuno irelevantna za konflikt koji se dogodio i ponavljanje postupka sa -7 je gubitak vremena.

Rešenje: analiza konflikata (eng. conflict analysis) i uvođenje povratnih skokova (eng. backjumping).

Primer

Primenjeno pravilo	*satFlag*	*M*
...		
Backtrack $(M \models \neg [-1, 3, -5, -6])$	*UNDEF*	-6
Decide $(l = 1)$	*UNDEF*	$-6, \lvert 1$
UnitProp $(c = [-1, 2],\ l = 2)$	*UNDEF*	$-6, \lvert 1, 2$
Decide $(l = 7)$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7$
Decide $(l = 3)$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7, \lvert 3$
UnitProp $(c = [-3, 4],\ l = 4)$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7, \lvert 3, 4$
UnitProp $(c = [-1, -3, 5],\ l = 5)$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7, \lvert 3, 4, 5$
Backtrack $(M \models \neg [-2, -4, -5])$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7, -3$
Decide $(l = 4)$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7, -3, \lvert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \lvert 1, 2, \lvert 7, -3, \lvert 4, -5$
$M \not\models \neg F_0$, $(\text{vars } M) = (\text{vars } F_0)$	*SAT*	$-6, \lvert 1, 2, \lvert 7, -3, \lvert 4, -5$

Primer

Primenjeno pravilo	*satFlag*	*M*
...		
Backtrack $(M \models \neg [-1, 3, -5, -6])$	*UNDEF*	-6
Decide $(l = 1)$	*UNDEF*	$-6, \|1$
UnitProp $(c = [-1, 2],\ l = 2)$	*UNDEF*	$-6, \|1, 2$
Decide $(l = 7)$	*UNDEF*	$-6, \|1, 2, \|7$
Decide $(l = 3)$	*UNDEF*	$-6, \|1, 2, \|7, \|3$
UnitProp $(c = [-3, 4],\ l = 4)$	*UNDEF*	$-6, \|1, 2, \|7, \|3, 4$
UnitProp $(c = [-1, -3, 5],\ l = 5)$	*UNDEF*	$-6, \|1, 2, \|7, \|3, 4, 5$
Backtrack $(M \models \neg [-2, -4, -5])$	*UNDEF*	$-6, \|1, 2, \|7, -3$
Decide $(l = 4)$	*UNDEF*	$-6, \|1, 2, \|7, -3, \|4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \|1, 2, \|7, -3, \|4, -5$
$M \not\models \neg F_0$, (vars M) = (vars F_0)	*SAT*	$-6, \|1, 2, \|7, -3, \|4, -5$

Problemi – nastavak

Kako da sprečimo da ponovo radimo istu stvar?

- Ista vrsta redundantnosti može da se javi i u nekom kontekstu (npr. sa literalom -6 umesto literala 6).

Rešenje: učenje iz ranijih konflikata (eng. learning).

Problemi – nastavak

Kako da sprečimo da ponovo radimo istu stvar?

- Ista vrsta redundantnosti može da se javi i u nekom kontekstu (npr. sa literalom -6 umesto literala 6).

Rešenje: učenje iz ranijih konflikata (eng. learning).

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

Zašto je došlo do konflikta? *Zbog literala $+2$, $+4$ i $+5$.*
A zbog čega je prisutan literal $+5$? *Zbog literala $+1$ i $+3$.*
A zbog čega je prisutan literal $+4$? *Zbog literala $+3$.*
Dakle, uz literale $+1$ i $+2$, ne ide literal $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

Zašto je došlo do konflikta? *Zbog literala $+2$, $+4$ i $+5$.*

A zbog čega je prisutan literal $+5$? *Zbog literala $+1$ i $+3$.*

A zbog čega je prisutan literal $+4$? *Zbog literala $+3$.*

Dakle, uz literale $+1$ i $+2$, ne ide literal $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+2, +4, +5\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+2, +4, +5\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+2, +4, +5\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+1, +2, +3, +4\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+1, +2, +3, +4\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+1, +2, +3, +4\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+1, +2, +3\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$

$$\{+1, +2, +3\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata

Primer

Primenjeno pravilo	*satFlag*	*M*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg[-2, -4, -5])$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$

$$\{+1, +2, +3\}$$

Zašto je došlo do konflikta? *Zbog literala* $+2$, $+4$ *i* $+5$.

A zbog čega je prisutan literal $+5$*?* *Zbog literala* $+1$ *i* $+3$.

A zbog čega je prisutan literal $+4$*?* *Zbog literala* $+3$.

Dakle, uz literale $+1$ *i* $+2$*, ne ide literal* $+3$.

Analiza konflikata kao rezolucija

Stablo rezolucije

$$\dfrac{\dfrac{[-2,-4,-5] \qquad [-1,-3,5]}{[-1,-2,-3,-4]} \qquad [-3,4]}{[-1,-2,-3]}$$

Dobijena klauza povratnog skoka je semantička posledica polaznog skupa klauza.

Analiza konflikata kao rezolucija

Stablo rezolucije

$$\dfrac{\dfrac{[-2,-4,-5] \qquad [-1,-3,5]}{[-1,-2,-3,-4]} \qquad [-3,4]}{[-1,-2,-3]}$$

Dobijena klauza povratnog skoka je semantička posledica polaznog skupa klauza.

Analiza konflikata kao rezolucija

Stablo rezolucije

$$\dfrac{\dfrac{[-2,-4,-5] \qquad [-1,-3,5]}{[-1,-2,-3,-4]} \qquad [-3,4]}{[-1,-2,-3]}$$

Dobijena klauza povratnog skoka je semantička posledica polaznog skupa klauza.

Analiza konflikata kao rezolucija

Stablo rezolucije

$$\dfrac{\dfrac{[-2,-4,-5] \qquad [-1,-3,5]}{[-1,-2,-3,-4]} \qquad [-3,4]}{[-1,-2,-3]}$$

Dobijena klauza povratnog skoka je semantička posledica polaznog skupa klauza.

Analiza konflikata kao rezolucija

Stablo rezolucije

$$\dfrac{\dfrac{[-2,-4,-5] \qquad [-1,-3,5]}{[-1,-2,-3,-4]} \qquad [-3,4]}{[-1,-2,-3]}$$

Dobijena klauza povratnog skoka je semantička posledica polaznog skupa klauza.

Sistem sa analizom konflikata

Stanje rešavača:

- (M, C)
- M - parcijalna valuacija
- C - konfliktna klauza (ili *no_cflt* ako nema konflikta)

Pravila:

Decide:
$$\frac{C = \mathit{no_cflt} \quad l \in F \quad l, \bar{l} \notin M}{M := M \,|\, l}$$

UnitPropagate:
$$\frac{C = \mathit{no_cflt} \quad l \vee l_1 \vee \ldots \vee l_k \in F \qquad \bar{l}_1, \ldots, \bar{l}_k \in M \qquad l, \bar{l} \notin M}{M := M\ l}$$

Conflict:
$$\frac{C = \mathit{no_cflt} \quad l_1 \vee \ldots l_k \in F \qquad \bar{l}_1, \ldots, \bar{l}_k \in M}{C := l_1 \vee \ldots \vee l_k}$$

Explain:
$$\frac{\bar{l} \in C \qquad l \vee l_1 \vee \ldots \vee l_k \in F \qquad \bar{l}_1, \ldots, \bar{l}_k \prec^M l}{C := (C \setminus \bar{l}) \vee l_1 \vee \ldots \vee l_k}$$

Backjump:
$$\frac{C = l \vee l_1 \vee \ldots \vee l_k \qquad \text{level } \bar{l} > m \geq \text{level } \bar{l}_i}{M := (\text{prefixToLevel } m\ M)\ l \quad C := \mathit{no_cflt}}$$

Učenje

Sistem sa učenjem

Stanje rešavača:

- (M, F, C)
- M - parcijalna valuacija
- F - formula koja se vremenom proširuje (učenjem klauza)
- C - konfliktna klauza (ili *no_cflt* ako nema konflikta)

Dodatno pravilo:

Learn:

$$\frac{F \models c}{F := F \wedge c}$$

NAPOMENA: Obično se uče isključivo klauze povratnog skoka.

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ ]$.

Primenjeno pravilo	*satFlag*	*M/C*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]]$.

Primenjeno pravilo	*satFlag*	*M/C*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6], l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2], l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4], l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5], l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5, c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4, c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6], l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5, c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3, c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2, c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1, c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	*M/C*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\mid +6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\mid +6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\mid +6, +1, +2, \mid +7$
Decide $(l = +3)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\mid +6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ ]$.

Primenjeno pravilo	*satFlag*	*M/C*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]$, $[-6]$].

Primenjeno pravilo	*satFlag*	*M/C*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\|+6$
UnitProp $(c = [+1, -6], l = +1)$	*UNDEF*	$\|+6, +1$
UnitProp $(c = [-1, +2], l = +2)$	*UNDEF*	$\|+6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\|+6, +1, +2, \|+7$
Decide $(l = +3)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3$
UnitProp $(c = [-3, +4], l = +4)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4$
UnitProp $(c = [-1, -3, +5], l = +5)$	*UNDEF*	$\|+6, +1, +2, \|+7, \|+3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5, c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4, c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\|+6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6], l = +5)$	*UNDEF*	$\|+6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5, c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3, c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2, c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1, c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7], [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	*M/C*
	UNDEF	
Decide $(l = +6)$	*UNDEF*	$\mid +6$
UnitProp $(c = [+1, -6],\ l = +1)$	*UNDEF*	$\mid +6, +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$\mid +6, +1, +2$
Decide $(l = +7)$	*UNDEF*	$\mid +6, +1, +2, \mid +7$
Decide $(l = +3)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3$
UnitProp $(c = [-3, +4],\ l = +4)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3, +4$
UnitProp $(c = [-1, -3, +5],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, \mid +7, \mid +3, +4, +5$
Conflict $(M \models \neg [-2, -4, -5])$	*UNDEF*	$[-2, -4, -5]$
Explain $(l = +5,\ c = [-1, -3, +5])$	*UNDEF*	$[-1, -2, -3, -4]$
Explain $(l = 4,\ c = [-3, +4])$	*UNDEF*	$[-1, -2, -3]$
Learn *(C = [-1, -2, -3])*	*UNDEF*	$[-1, -2, -3]$
Backjump $(C = [-1, -2, -3])$	*UNDEF*	$\mid +6, +1, +2, -3$
UnitProp $(c = [-2, +3, +5, -6],\ l = +5)$	*UNDEF*	$\mid +6, +1, +2, -3, +5$
Conflict $(M \models \neg [-1, +3, -5, -6])$	*UNDEF*	$[-1, +3, -5, -6]$
Explain $(l = +5,\ c = [-2, +3, +5, -6])$	*UNDEF*	$[-1, -2, +3, -6]$
Explain $(l = -3,\ c = [-1, -2, -3])$	*UNDEF*	$[-1, -2, -6]$
Explain $(l = +2,\ c = [-1, +2])$	*UNDEF*	$[-1, -6]$
Explain $(l = +1,\ c = [+1, -6])$	*UNDEF*	$[-6]$
Learn *(C = [-6])*	*UNDEF*	$[-6]$
Backjump $(C = [-6])$	*UNDEF*	-6

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \mid +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \mid +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \mid +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \mid +1, +2, -3, \mid 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \mid +1, +2, -3, \mid 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \mid +1, +2, -3, \mid 4, -5, \mid +7$
$M \not\models \neg F$, $(\text{vars } M) = (\text{vars } F)$	*SAT*	$-6, \mid +1, +2, -3, \mid 4, -5, \mid +7$

Primer

$F = [[-1, +2], [-3, +4], [-1, -3, +5], [-2, -4, -5], [-2, +3, +5, -6], [-1, +3, -5, -6], [+1, -6], [+1, +7]\ , [-1, -2, -3]\ , [-6]\ ]$.

Primenjeno pravilo	*satFlag*	M/C
Backjump $(C = [-6])$	*UNDEF*	-6
Decide $(l = +1)$	*UNDEF*	$-6, \vert +1$
UnitProp $(c = [-1, +2],\ l = +2)$	*UNDEF*	$-6, \vert +1, +2$
UnitProp $(c = [-1, -2, -3],\ l = -3)$	*UNDEF*	$-6, \vert +1, +2, -3$
Decide $(l = +4)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4$
UnitProp $(c = [-2, -4, -5],\ l = -5)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5$
Decide $(l = +7)$	*UNDEF*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$
$M \not\models \neg F$, (vars M) = (vars F)	*SAT*	$-6, \vert +1, +2, -3, \vert 4, -5, \vert +7$

Heuristike grananja

Kako odabrati literal odlučivanja?

- Izbor atoma za grananje: Variable State Independent Decaying Sum (VSIDS)
  - Prilikom analize konflikta svim literalima klauza koje učestvuju u konfliktu povećava se skor za 1
  - Svi skorovi se periodično množe nekom konstantom manjom od 1
  - Na ovaj način do izražaja dolaze atomi koji su učestvovali u skorijim konfliktima
- Izbor polariteta: Saved polarity
  - Biramo onaj polaritet koji je promenljiva imala prethodnog puta u parcijalnoj valuaciji

Zaboravljanje

Zaboravljanje naučenih klauza?

- Ukoliko broj naučenih klauza postane preveliki, pronalaženje jediničnih i konfliktnih klauza se usporava.
- Neke naučene klauze se posle određenog vremena uklanjaju.
- Zaboravljanje je kontrolisano heuristikama.

Otpočinjanje iznova

Ponovo pokretanje pretrage

- U nekim trenucima je korisno pretragu prekinuti i započeti iznova (engl. restart).
- Postoji nada da će nas klauze koje su u međuvremenu naučene odvesti u neku drugu (lakšu) granu stabla pretrage.
- Otpočinjanje iznova pokazuje dobra svojstva kada se koristi uz određenih procenat slučajnih odluka u toku pretrage.
- Otpočinjanje iznova je kontrolisano heuristikama.

Pouzdanost SAT rešavača

Kako verovati SAT rešavaču?

- CDCL zasnovani SAT rešavači su veoma kompleksan softver
- Na izlazu dobijamo odgovor *SAT* ili *UNSAT*:
  - bilo koja greška u implementaciji rešavača može proizvesti pogrešan odgovor za neki ulaz
  - kako mu možemo verovati?
- U slučaju SAT odgovora, dobijamo i zadovoljavajuću valuaciju
  - lako se može proveriti da li dobijena valuacija zaista zadovoljava formulu
- Šta u slučaju UNSAT odgovora?

Sertifikati

Sertifikat

- Tvrdnja da je formula nezadovoljiva se mora potkrepiti dokazom
- U slučaju CDCL SAT rešavača taj dokaz je rezolucijski
  - kada se desi konflikt na nultom nivou, može se izvršiti analiza konflikta koja će kao rezultat izvesti praznu klauzu
  - zajedno sa ranijim rezolucijskim izvodjenjima naučenih klauza, dobijamo rezolucijski dokaz da je polazni skup klauza nezadovoljiv
- Na zahtev, SAT rešavac može na izlazu dati ovaj rezolucijski dokaz u odgovarajućem formatu
- Ovaj dokaz se obično naziva sertifikat
  - sertifikat se može proveriti nezavisnim softverom koji je mnogo jednostavniji od SAT rešavača i lakše mu se može verovati
- Različiti formati (DRUP, DRAT, GRAT, GRIT, ...)
- Primer alata: drat-trim

Primer

Primer

$F = [[1, -3], [2, 3, -1], [-1, -3, 2], [-1, -2], [1, 3]]$

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\vert +1$
UnitProp $(c = [-1, -2], l = -2)$	*UNDEF*	$\vert +1, -2$
UnitProp $(c = [-1, -3, 2], l = -3)$	*UNDEF*	$\vert +1, -2, -3$
Conflict $(M \models \neg [2, 3, -1])$	*UNDEF*	$[2, 3, -1]$
Explain $(l = -3, c = [-1, -3, 2])$	*UNDEF*	$[-1, 2]$
Explain $(l = -2, c = [-1, -2])$	*UNDEF*	$[-1]$
Learn $(C = [-1])$	*UNDEF*	$[-1]$
Backjump $(C = [-1])$	*UNDEF*	-1
UnitProp $(c = [1, -3])$	*UNDEF*	$-1, -3$
Conflict $(M \models \neg [1, 3])$	*UNSAT*	$[1, 3]$
Explain $(l = -3, c = [1, -3])$	*UNSAT*	$[1]$
Explain $(l = -1, c = [-1])$	*UNSAT*	$[\,]$
Learn $(C = [\,])$	*UNSAT*	$[\,]$

Primer

Primer

$F = [[1, -3], [2, 3, -1], [-1, -3, 2], [-1, -2], [1, 3], [-1]]$

Primenjeno pravilo	*satFlag*	M / C
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, -2],\ l = -2)$	*UNDEF*	$\mid +1, -2$
UnitProp $(c = [-1, -3, 2],\ l = -3)$	*UNDEF*	$\mid +1, -2, -3$
Conflict $(M \models \neg [2, 3, -1])$	*UNDEF*	$[2, 3, -1]$
Explain $(l = -3,\ c = [-1, -3, 2])$	*UNDEF*	$[-1, 2]$
Explain $(l = -2,\ c = [-1, -2])$	*UNDEF*	$[-1]$
Learn $(C = [-1])$	*UNDEF*	$[-1]$
Backjump $(C = [-1])$	*UNDEF*	-1
UnitProp $(c = [1, -3])$	*UNDEF*	$-1, -3$
Conflict $(M \models \neg [1, 3])$	*UNSAT*	$[1, 3]$
Explain $(l = -3,\ c = [1, -3])$	*UNSAT*	$[1]$
Explain $(l = -1,\ c = [-1])$	*UNSAT*	$[\,]$
Learn $(C = [\,])$	*UNSAT*	$[\,]$

Primer

Primer

$F = [[1, -3], [2, 3, -1], [-1, -3, 2], [-1, -2], [1, 3], [-1], [\,]]$

Primenjeno pravilo	*satFlag*	M/C
	UNDEF	
Decide $(l = +1)$	*UNDEF*	$\mid +1$
UnitProp $(c = [-1, -2],\ l = -2)$	*UNDEF*	$\mid +1, -2$
UnitProp $(c = [-1, -3, 2],\ l = -3)$	*UNDEF*	$\mid +1, -2, -3$
Conflict $(M \models \neg [2, 3, -1])$	*UNDEF*	$[2, 3, -1]$
Explain $(l = -3,\ c = [-1, -3, 2])$	*UNDEF*	$[-1, 2]$
Explain $(l = -2,\ c = [-1, -2])$	*UNDEF*	$[-1]$
Learn $(C = [-1])$	*UNDEF*	$[-1]$
Backjump $(C = [-1])$	*UNDEF*	-1
UnitProp $(c = [1, -3])$	*UNDEF*	$-1, -3$
Conflict $(M \models \neg [1, 3])$	*UNSAT*	$[1, 3]$
Explain $(l = -3,\ c = [1, -3])$	*UNSAT*	$[1]$
Explain $(l = -1,\ c = [-1])$	*UNSAT*	$[\,]$
Learn $(C = [\,])$	*UNSAT*	$[\,]$

Primer - nastavak

Rezolucijski dokaz nezadovoljivosti

$$\dfrac{\dfrac{\dfrac{[2,3,-1] \qquad [-1,-3,2]}{[-1,2]} \qquad [-1,-2]}{[-1]} \qquad \dfrac{[1,3] \qquad [1,-3]}{[1]}}{[\,]}$$

U listovima su klauze polazne formule. Plavi deo predstavlja izvođenje naučene klauze povratnog skoka $[-1]$ prilikom prvog konflikta. Ostatak predstavlja izvođenje prazne klauze prilikom konflikta na nultom nivou.