Logic

Diana Gründlinger Aart Middeldorp Fabian Mitterwallner

Alexander Montag Johannes Niederhauser Daniel Rainer

Outline

1. **Summary of Previous Lecture**
2. **CTL***
3. **Intermezzo**
4. **SAT Solving**
5. **Sorting Networks**
6. **Further Reading**

Definitions

- path $s_1 \to s_2 \to \cdots$ is fair with respect to set C of CTL formulas if for all $\psi \in C$ $s_i \vDash \psi$ for infinitely many i
- A_C (E_C) denotes A (E) restricted to paths that are fair with respect to C

Lemma

$$\mathsf{E}_C[\varphi \,\mathsf{U}\, \psi] \equiv \mathsf{E}[\varphi \,\mathsf{U}\, (\psi \wedge \mathsf{E}_C\mathsf{G}\top)] \qquad \mathsf{E}_C\mathsf{X}\,\varphi \equiv \mathsf{EX}(\varphi \wedge \mathsf{E}_C\mathsf{G}\top)$$

Theorem

set of temporal connectives is adequate for CTL $\iff$

it contains $\begin{cases} \text{at least one of } \{\mathsf{AX}, \mathsf{EX}\} \\ \text{at least one of } \{\mathsf{EG}, \mathsf{AF}, \mathsf{AU}\} \\ \mathsf{EU} \end{cases}$

Theorem

- $\{X, U\}$, $\{X, W\}$ and $\{X, R\}$ are adequate sets of temporal connectives for LTL
- $\{U, R\}$, $\{U, W\}$, $\{U, G\}$, $\{F, W\}$ and $\{F, R\}$ are adequate sets of temporal connectives for LTL fragment consisting of negation-normal forms without X

LTL Model Checking

$\mathcal{M}, s \vDash \varphi$?

- construct labelled Büchi automaton $A_{\neg\varphi}$ for $\neg\varphi$
- combine $A_{\neg\varphi}$ and $\mathcal{M}$ into single automaton $A_{\neg\varphi} \times \mathcal{M}$
- determine whether there exists accepting path π in $A_{\neg\varphi} \times \mathcal{M}$ starting from s

Theorem

$\mathcal{M}, s \nvDash \varphi \iff$ exists accepting path in $A_{\neg\varphi} \times \mathcal{M}$ starting from state corresponding to s

Part I: Propositional Logic

algebraic normal forms, binary decision diagrams, conjunctive normal forms, DPLL, Horn formulas, natural deduction, Post's adequacy theorem, resolution, SAT, semantics, sorting networks, soundness and completeness, syntax, Tseitin's transformation

Part II: Predicate Logic

natural deduction, quantifier equivalences, resolution, semantics, Skolemization, syntax, undecidability, unification

Part III: Model Checking

adequacy, branching-time temporal logic, CTL^*, fairness, linear-time temporal logic, model checking algorithms, symbolic model checking

Part I: Propositional Logic

algebraic normal forms, binary decision diagrams, conjunctive normal forms, DPLL, Horn formulas, natural deduction, Post's adequacy theorem, resolution, SAT, semantics, sorting networks, soundness and completeness, syntax, Tseitin's transformation

Part II: Predicate Logic

natural deduction, quantifier equivalences, resolution, semantics, Skolemization, syntax, undecidability, unification

Part III: Model Checking

adequacy, branching-time temporal logic, CTL*, fairness, linear-time temporal logic, model checking algorithms, symbolic model checking

Outline

Definition

CTL* formulas consist of

- state formulas, which are evaluated in states:

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid \mathsf{A}[\alpha] \mid \mathsf{E}[\alpha]$$

Definition

CTL* formulas consist of

- state formulas, which are evaluated in states:

$$\varphi \ ::= \ \bot \mid \top \mid p \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid \mathsf{A}[\alpha] \mid \mathsf{E}[\alpha]$$

- path formulas, which are evaluated along paths:

$$\alpha \ ::= \ \varphi \mid (\neg \alpha) \mid (\alpha \wedge \alpha) \mid (\alpha \vee \alpha) \mid (\alpha \rightarrow \alpha) \mid (\mathsf{X}\, \alpha) \mid (\mathsf{F}\, \alpha) \mid (\mathsf{G}\, \alpha) \mid (\alpha\, \mathsf{U}\, \alpha)$$

Definition

CTL* formulas consist of

- state formulas, which are evaluated in states:

$$\varphi \ ::= \ \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \land \varphi) \mid (\varphi \lor \varphi) \mid (\varphi \to \varphi) \mid \mathsf{A}[\alpha] \mid \mathsf{E}[\alpha]$$

- path formulas, which are evaluated along paths:

$$\alpha \ ::= \ \varphi \mid (\neg\alpha) \mid (\alpha \land \alpha) \mid (\alpha \lor \alpha) \mid (\alpha \to \alpha) \mid (\mathsf{X}\,\alpha) \mid (\mathsf{F}\,\alpha) \mid (\mathsf{G}\,\alpha) \mid (\alpha\,\mathsf{U}\,\alpha)$$

Examples

$\mathsf{A}[\,(p\,\mathsf{U}\,r) \lor (q\,\mathsf{U}\,r)]$ $\qquad$ $\mathsf{A}[\mathsf{X}\,p \lor \mathsf{X}\,\mathsf{X}\,p]$ $\qquad$ $\mathsf{E}[\mathsf{G}\,\mathsf{F}\,p]$

$\mathsf{A}[\,(p \lor q)\,\mathsf{U}\,r]$ $\qquad$ $\mathsf{A}[\mathsf{X}\,p] \lor \mathsf{A}[\mathsf{X}\,\mathsf{A}[\mathsf{X}\,p]]$ $\qquad$ $\mathsf{E}[\mathsf{G}\,\mathsf{E}[\mathsf{F}\,p]]$

Definition

satisfaction of CTL* state formula φ in state $s \in S$ of model $\mathcal{M} = (S, \to, L)$

$$\begin{aligned}
\mathcal{M}, s &\nvDash \bot \\
\mathcal{M}, s &\vDash \top \\
\mathcal{M}, s &\vDash p && \iff && p \in L(s) \\
\mathcal{M}, s &\vDash \neg\varphi && \iff && \mathcal{M}, s \nvDash \varphi \\
\mathcal{M}, s &\vDash \varphi \wedge \psi && \iff && \mathcal{M}, s \vDash \varphi \text{ and } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s &\vDash \varphi \vee \psi && \iff && \mathcal{M}, s \vDash \varphi \text{ or } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s &\vDash \varphi \to \psi && \iff && \mathcal{M}, s \nvDash \varphi \text{ or } \mathcal{M}, s \vDash \psi
\end{aligned}$$

Definition

satisfaction of CTL* state formula φ in state $s \in S$ of model $\mathcal{M} = (S, \to, L)$

$$
\begin{aligned}
&\mathcal{M}, s \nvDash \bot \\
&\mathcal{M}, s \vDash \top \\
&\mathcal{M}, s \vDash p && \iff && p \in L(s) \\
&\mathcal{M}, s \vDash \neg\varphi && \iff && \mathcal{M}, s \nvDash \varphi \\
&\mathcal{M}, s \vDash \varphi \wedge \psi && \iff && \mathcal{M}, s \vDash \varphi \text{ and } \mathcal{M}, s \vDash \psi \\
&\mathcal{M}, s \vDash \varphi \vee \psi && \iff && \mathcal{M}, s \vDash \varphi \text{ or } \mathcal{M}, s \vDash \psi \\
&\mathcal{M}, s \vDash \varphi \to \psi && \iff && \mathcal{M}, s \nvDash \varphi \text{ or } \mathcal{M}, s \vDash \psi \\
&\mathcal{M}, s \vDash \mathrm{A}[\alpha] && \iff && \forall \text{ paths } \pi = s \to s_2 \to \cdots \quad \mathcal{M}, \pi \vDash \alpha
\end{aligned}
$$

Definition

satisfaction of CTL* state formula φ in state $s \in S$ of model $\mathcal{M} = (S, \to, L)$

$$
\begin{array}{lcl}
\mathcal{M}, s \nvDash \bot & & \\
\mathcal{M}, s \vDash \top & & \\
\mathcal{M}, s \vDash p & \iff & p \in L(s) \\
\mathcal{M}, s \vDash \neg\varphi & \iff & \mathcal{M}, s \nvDash \varphi \\
\mathcal{M}, s \vDash \varphi \wedge \psi & \iff & \mathcal{M}, s \vDash \varphi \text{ and } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s \vDash \varphi \vee \psi & \iff & \mathcal{M}, s \vDash \varphi \text{ or } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s \vDash \varphi \to \psi & \iff & \mathcal{M}, s \nvDash \varphi \text{ or } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s \vDash \mathsf{A}[\alpha] & \iff & \forall \text{ paths } \pi = s \to s_2 \to \cdots \quad \mathcal{M}, \pi \vDash \alpha \\
\mathcal{M}, s \vDash \mathsf{E}[\alpha] & \iff & \exists \text{ path } \pi = s \to s_2 \to \cdots \quad \mathcal{M}, \pi \vDash \alpha
\end{array}
$$

Definition

satisfaction of CTL* path formula α with respect to path $\pi = s_1 \to s_2 \to \cdots$ in $\mathcal{M} = (S, \to, L)$

$$\mathcal{M}, \pi \vDash \varphi \quad \Longleftrightarrow \quad \mathcal{M}, s_1 \vDash \varphi$$

Definition

satisfaction of CTL* path formula α with respect to path $\pi = s_1 \to s_2 \to \cdots$ in $\mathcal{M} = (S, \to, L)$

$$\mathcal{M}, \pi \vDash \varphi \iff \mathcal{M}, s_1 \vDash \varphi$$

$$\mathcal{M}, \pi \vDash \neg\alpha \iff \mathcal{M}, \pi \nvDash \alpha$$

$$\mathcal{M}, \pi \vDash \alpha \wedge \beta \iff \mathcal{M}, \pi \vDash \alpha \text{ and } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \alpha \vee \beta \iff \mathcal{M}, \pi \vDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \alpha \to \beta \iff \mathcal{M}, \pi \nvDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta$$

Definition

satisfaction of CTL* path formula α with respect to path $\pi = s_1 \to s_2 \to \cdots$ in $\mathcal{M} = (S, \to, L)$

$$\mathcal{M}, \pi \vDash \varphi \iff \mathcal{M}, s_1 \vDash \varphi$$

$$\mathcal{M}, \pi \vDash \neg\alpha \iff \mathcal{M}, \pi \nvDash \alpha$$

$$\mathcal{M}, \pi \vDash \alpha \wedge \beta \iff \mathcal{M}, \pi \vDash \alpha \text{ and } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \alpha \vee \beta \iff \mathcal{M}, \pi \vDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \alpha \to \beta \iff \mathcal{M}, \pi \nvDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \mathsf{X}\,\alpha \iff \mathcal{M}, \pi^2 \vDash \alpha$$

Definition

satisfaction of CTL* path formula α with respect to path $\pi = s_1 \to s_2 \to \cdots$ in $\mathcal{M} = (S, \to, L)$

$$\mathcal{M}, \pi \vDash \varphi \quad \Longleftrightarrow \quad \mathcal{M}, s_1 \vDash \varphi$$

$$\mathcal{M}, \pi \vDash \neg\alpha \quad \Longleftrightarrow \quad \mathcal{M}, \pi \nvDash \alpha$$

$$\mathcal{M}, \pi \vDash \alpha \wedge \beta \quad \Longleftrightarrow \quad \mathcal{M}, \pi \vDash \alpha \text{ and } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \alpha \vee \beta \quad \Longleftrightarrow \quad \mathcal{M}, \pi \vDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \alpha \to \beta \quad \Longleftrightarrow \quad \mathcal{M}, \pi \nvDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta$$

$$\mathcal{M}, \pi \vDash \mathsf{X}\,\alpha \quad \Longleftrightarrow \quad \mathcal{M}, \pi^2 \vDash \alpha$$

$$\mathcal{M}, \pi \vDash \mathsf{F}\,\alpha \quad \Longleftrightarrow \quad \exists\, i \geqslant 1 \;\; \mathcal{M}, \pi^i \vDash \alpha$$

$$\mathcal{M}, \pi \vDash \mathsf{G}\,\alpha \quad \Longleftrightarrow \quad \forall\, i \geqslant 1 \;\; \mathcal{M}, \pi^i \vDash \alpha$$

Definition

satisfaction of CTL* path formula α with respect to path $\pi = s_1 \to s_2 \to \cdots$ in $\mathcal{M} = (S, \to, L)$

$$\begin{aligned}
\mathcal{M}, \pi \vDash \varphi &\iff \mathcal{M}, s_1 \vDash \varphi \\
\mathcal{M}, \pi \vDash \neg\alpha &\iff \mathcal{M}, \pi \nvDash \alpha \\
\mathcal{M}, \pi \vDash \alpha \wedge \beta &\iff \mathcal{M}, \pi \vDash \alpha \text{ and } \mathcal{M}, \pi \vDash \beta \\
\mathcal{M}, \pi \vDash \alpha \vee \beta &\iff \mathcal{M}, \pi \vDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta \\
\mathcal{M}, \pi \vDash \alpha \to \beta &\iff \mathcal{M}, \pi \nvDash \alpha \text{ or } \mathcal{M}, \pi \vDash \beta \\
\mathcal{M}, \pi \vDash \mathsf{X}\,\alpha &\iff \mathcal{M}, \pi^2 \vDash \alpha \\
\mathcal{M}, \pi \vDash \mathsf{F}\,\alpha &\iff \exists\, i \geqslant 1\ \mathcal{M}, \pi^i \vDash \alpha \\
\mathcal{M}, \pi \vDash \mathsf{G}\,\alpha &\iff \forall\, i \geqslant 1\ \mathcal{M}, \pi^i \vDash \alpha \\
\mathcal{M}, \pi \vDash \alpha\,\mathsf{U}\,\beta &\iff \exists\, i \geqslant 1\ \mathcal{M}, \pi^i \vDash \beta \text{ and } \forall\, j < i\ \mathcal{M}, \pi^j \vDash \alpha
\end{aligned}$$

Theorem

satisfaction of CTL* formulas in finite models is decidable

Theorem

satisfaction of CTL* formulas in finite models is decidable

Definition

CTL* state (CTL, LTL) formulas φ and ψ are semantically equivalent if

$$\mathcal{M}, s \vDash \varphi \quad \Longleftrightarrow \quad \mathcal{M}, s \vDash \psi$$

for all models $\mathcal{M} = (S, \rightarrow, L)$ and states $s \in S$

Theorem

satisfaction of CTL* formulas in finite models is decidable

Definition

CTL* state (CTL, LTL) formulas φ and ψ are semantically equivalent if

$$\mathcal{M}, s \vDash \varphi \quad \Longleftrightarrow \quad \mathcal{M}, s \vDash \psi$$

for all models $\mathcal{M} = (S, \rightarrow, L)$ and states $s \in S$

Remarks

- LTL formula α is equivalent to CTL* formula A$[\alpha]$

Theorem

satisfaction of CTL* formulas in finite models is decidable

Definition

CTL* state (CTL, LTL) formulas φ and ψ are semantically equivalent if

$$\mathcal{M}, s \vDash \varphi \iff \mathcal{M}, s \vDash \psi$$

for all models $\mathcal{M} = (S, \to, L)$ and states $s \in S$

Remarks

- LTL formula α is equivalent to CTL* formula $\mathsf{A}[\alpha]$
- CTL is fragment of CTL* in which path formulas are "restricted" to

$$\alpha \ ::= \ \varphi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha \vee \alpha) \mid (\alpha \to \alpha) \mid (\mathsf{X}\,\varphi) \mid (\mathsf{F}\,\varphi) \mid (\mathsf{G}\,\varphi) \mid (\varphi\,\mathsf{U}\,\varphi)$$

Lemma

AG EF p is not expressible in LTL

Lemma

AG EF p is not expressible in LTL

Proof

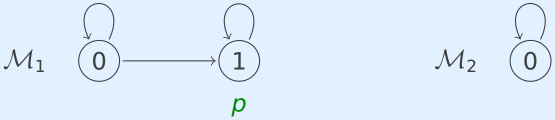
- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ

Lemma

AG EF p is not expressible in LTL

Proof

- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ
- consider models

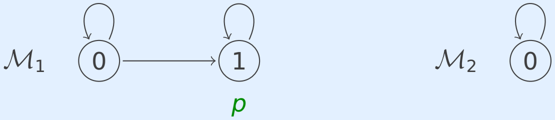


Lemma

AG EF p is not expressible in LTL

Proof

- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ
- consider models



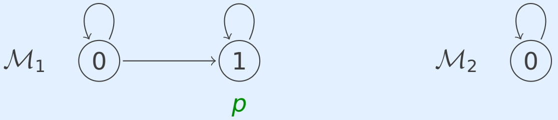
- $\mathcal{M}_1, 0 \vDash$ AG EF p

Lemma

AG EF p is not expressible in LTL

Proof

- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ
- consider models



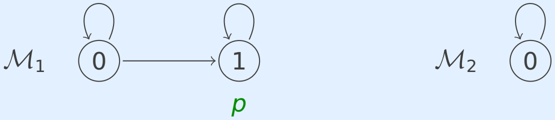
- $\mathcal{M}_1, 0 \vDash$ AG EF p
- $\mathcal{M}_1, 0 \vDash$ A$[\varphi]$

Lemma

AG EF p is not expressible in LTL

Proof

- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ
- consider models



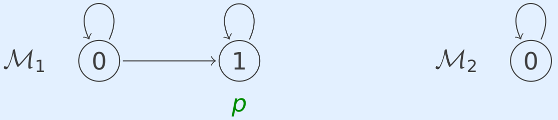
- $\mathcal{M}_1, 0 \vDash$ AG EF p
- $\mathcal{M}_1, 0 \vDash$ A$[\varphi]$
- $\mathcal{M}_2, 0 \nvDash$ AG EF p

Lemma

AG EF p is not expressible in LTL

Proof

- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ
- consider models



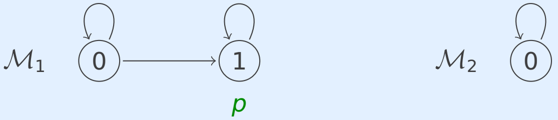
- $\mathcal{M}_1, 0 \vDash$ AG EF p
- $\mathcal{M}_1, 0 \vDash$ A$[\varphi]$
- $\mathcal{M}_2, 0 \nvDash$ AG EF p
- $\mathcal{M}_2, 0 \vDash$ A$[\varphi]$ because every path from 0 in $\mathcal{M}_2$ is also path in $\mathcal{M}_1$

Lemma

AG EF p is not expressible in LTL

Proof

- suppose AG EF $p \equiv$ A$[\varphi]$ for LTL formula φ
- consider models



- $\mathcal{M}_1, 0 \vDash$ AG EF p
- $\mathcal{M}_1, 0 \vDash$ A$[\varphi]$
- $\mathcal{M}_2, 0 \nvDash$ AG EF p
- $\mathcal{M}_2, 0 \vDash$ A$[\varphi]$ because every path from 0 in $\mathcal{M}_2$ is also path in $\mathcal{M}_1$ ↯

Lemma

- $\mathsf{A}[\mathsf{G}\,\mathsf{F}\,p \rightarrow \mathsf{F}\,q]$ is not expressible in CTL

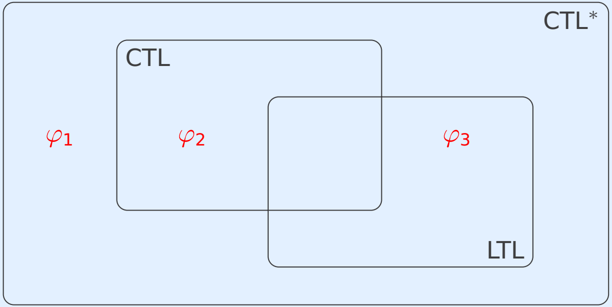
Lemma

- $\mathrm{A}[\mathrm{G}\,\mathrm{F}\,p \to \mathrm{F}\,q]$ is not expressible in CTL
- $\mathrm{E}[\mathrm{G}\,\mathrm{F}\,p]$ is expressible neither in CTL nor LTL

Lemma

- $A[G\,F\,p \rightarrow F\,q]$ is not expressible in CTL
- $E[G\,F\,p]$ is expressible neither in CTL nor LTL

Expressive Power



$$\varphi_1 = E[G\,F\,p]$$
$$\varphi_2 = AG\,EF\,p$$
$$\varphi_3 = A[G\,F\,p \rightarrow F\,q]$$

Outline

Question

Which of the following statements are true ?

A A set of LTL connectives which contains G cannot be adequate.

B The CTL formulas $\mathsf{AG}\,\neg p \to \mathsf{EF}\,q$ and $\mathsf{EF}(p \vee q)$ are equivalent.

C The CTL formula $p \wedge \mathsf{AX}\,\mathsf{AG}\,p$ is equivalent to the LTL formula $\mathsf{G}\,p$.

D The CTL* formulas $\mathsf{E}[\mathsf{G}\,\mathsf{E}[\mathsf{F}\,p]]$ and $\mathsf{E}[\mathsf{G}\,\mathsf{F}\,p]$ are equivalent.

Outline

1. **Summary of Previous Lecture**
2. **CTL***
3. **Intermezzo**
4. **SAT Solving**

   DPLL   Conflict Analysis

5. **Sorting Networks**
6. **Further Reading**

Remarks

- most state-of-the-art SAT solvers are based on variations of Davis – Putnam – Logemann – Loveland (DPLL) procedure (1960, 1962)

Remarks

- most state-of-the-art SAT solvers are based on variations of Davis – Putnam – Logemann – Loveland (DPLL) procedure (1960, 1962)
- abstract version of DPLL described in JACM paper of Nieuwenhuis, Oliveras, Tinelli (2006)

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$\| \;\; \neg 1 \vee \neg 2,\; 2 \vee 3,\; \neg 1 \vee \neg 3 \vee 4,\; 2 \vee \neg 3 \vee \neg 4,\; 1 \vee 4$$

initial state: empty assignment

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$\begin{array}{lrll} & \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\ \Longrightarrow & 1^{d} \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \end{array}$$

decide (guess): atom 1 is assumed to be true

Example

$$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$$

$$\begin{array}{rrcll}
 & & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\
\Longrightarrow & 1^{d} & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \\
\Longrightarrow & 1^{d}\ \neg 2 & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate}
\end{array}$$

unit propagation: atom 2 must be false

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$
\begin{array}{llll}
 & & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\
\Longrightarrow & 1^d & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \\
\Longrightarrow & 1^d\ \neg 2 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^d\ \neg 2\ 3 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate}
\end{array}
$$

unit propagation: atom 3 must be true

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$
\begin{array}{lrcll}
 & & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\
\Longrightarrow & 1^{d} & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \\
\Longrightarrow & 1^{d}\ \neg 2 & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^{d}\ \neg 2\ 3 & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^{d}\ \neg 2\ 3\ 4 & \| & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate}
\end{array}
$$

unit propagation: atom 4 must be true

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$
\begin{array}{llll}
 & \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\
\Longrightarrow & 1^d \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \\
\Longrightarrow & 1^d\ \neg 2 \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^d\ \neg 2\ 3 \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^d\ \neg 2\ 3\ 4 \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & \neg 1 \parallel & \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{backtrack}
\end{array}
$$

backtrack (previous decision was wrong): atom 1 must be false

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$
\begin{array}{lrll}
 & & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\
\Longrightarrow & 1^d & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \\
\Longrightarrow & 1^d\ \neg 2 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^d\ \neg 2\ 3 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & 1^d\ \neg 2\ 3\ 4 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & \neg 1 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{backtrack} \\
\Longrightarrow & \neg 1\ 4 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate}
\end{array}
$$

unit propagation: atom 4 must be true

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

$$
\begin{array}{rrll}
 & & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \\
\Longrightarrow & \overset{d}{1} & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ \neg 2 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ \neg 2\ 3 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ \neg 2\ 3\ 4 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & \neg 1 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{backtrack} \\
\Longrightarrow & \neg 1\ 4 & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{unit propagate} \\
\Longrightarrow & \neg 1\ 4\ \overset{d}{\neg 3} & \| \ \neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4 & \text{decide}
\end{array}
$$

decide (guess): atom 3 is assumed to be false

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

		$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	
$\Longrightarrow$	1^d	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	decide
$\Longrightarrow$	$1^d\ \neg 2$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$1^d\ \neg 2\ 3$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$1^d\ \neg 2\ 3\ 4$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$\neg 1$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	backtrack
$\Longrightarrow$	$\neg 1\ 4$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$\neg 1\ 4\ \neg 3^d$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	decide
$\Longrightarrow$	$\neg 1\ 4\ \neg 3^d\ 2$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate

unit propagation: atom 2 must be true

Example

$\varphi = (\neg 1 \vee \neg 2) \wedge (2 \vee 3) \wedge (\neg 1 \vee \neg 3 \vee 4) \wedge (2 \vee \neg 3 \vee \neg 4) \wedge (1 \vee 4)$

		$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	
$\Longrightarrow$	1^d	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	decide
$\Longrightarrow$	$1^d\ \neg 2$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$1^d\ \neg 2\ 3$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$1^d\ \neg 2\ 3\ 4$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$\neg 1$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	backtrack
$\Longrightarrow$	$\neg 1\ 4$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate
$\Longrightarrow$	$\neg 1\ 4\ \neg 3^d$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	decide
$\Longrightarrow$	$\neg 1\ 4\ \neg 3^d\ 2$	$\Vert$	$\neg 1 \vee \neg 2,\ 2 \vee 3,\ \neg 1 \vee \neg 3 \vee 4,\ 2 \vee \neg 3 \vee \neg 4,\ 1 \vee 4$	unit propagate

satisfying assignment: $\neg 1\ 2\ \neg 3\ 4$

Remarks

- most state-of-the-art SAT solvers are based on variations of Davis – Putnam – Logemann – Loveland (DPLL) procedure (1960, 1962)
- abstract version of DPLL described in JACM paper of Nieuwenhuis, Oliveras, Tinelli (2006)

Definition (Abstract DPLL)

- states $M \parallel F$ consist of
  - list M of (possibly annotated) non-complementary literals
  - CNF F

Remarks

- most state-of-the-art SAT solvers are based on variations of Davis – Putnam – Logemann – Loveland (DPLL) procedure (1960, 1962)
- abstract version of DPLL described in JACM paper of Nieuwenhuis, Oliveras, Tinelli (2006)

Definition (Abstract DPLL)

- states $M \parallel F$ consist of
  - list M of (possibly annotated) non-complementary literals
  - CNF F
- transition rules

$$M \parallel F \quad \Longrightarrow \quad M' \parallel F'$$

Remarks

- most state-of-the-art SAT solvers are based on variations of Davis – Putnam – Logemann – Loveland (DPLL) procedure (1960, 1962)
- abstract version of DPLL described in JACM paper of Nieuwenhuis, Oliveras, Tinelli (2006)

Definition (Abstract DPLL)

- states $M \parallel F$ consist of
  - list M of (possibly annotated) non-complementary literals
  - CNF F
- transition rules

$$M \parallel F \quad \Longrightarrow \quad M' \parallel F' \quad \text{or} \quad \text{fail-state}$$

Remarks

- most state-of-the-art SAT solvers are based on variations of Davis – Putnam – Logemann – Loveland (DPLL) procedure (1960, 1962)
- abstract version of DPLL described in JACM paper of Nieuwenhuis, Oliveras, Tinelli (2006)

Definition (Abstract DPLL)

- states $M \parallel F$ consist of
  - list M of (possibly annotated) non-complementary literals
  - CNF F
- transition rules

$$M \parallel F \quad \Longrightarrow \quad M' \parallel F' \quad \text{or} \quad \text{fail-state} \qquad (\text{this lecture: } F = F')$$

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M

Definition (Transition Rules)

- unit propagate $\qquad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M $\qquad$ unit clause

Definition (Transition Rules)

- unit propagate $\qquad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M $\qquad$ unit clause

- pure literal $\qquad M \parallel F \implies M\,\ell \parallel F$

  if ℓ occurs in F and ℓ^c does not occur in F and ℓ is undefined in M

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M $\quad$ unit clause

- pure literal $\quad M \parallel F \implies M\,\ell \parallel F$

  if ℓ occurs in F and ℓ^c does not occur in F and ℓ is undefined in M

- decide $\quad M \parallel F \implies M\,\overset{d}{\ell} \parallel F$

  if ℓ or ℓ^c occurs in F and ℓ is undefined in M

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M $\quad$ unit clause

- pure literal $\quad M \parallel F \implies M\,\ell \parallel F$

  if ℓ occurs in F and ℓ^c does not occur in F and ℓ is undefined in M

- decide $\quad M \parallel F \implies M\,\overset{d}{\ell} \parallel F$

  if ℓ or ℓ^c occurs in F and ℓ is undefined in M

- fail $\quad M \parallel F, C \implies$ fail-state

  if $M \vDash \neg C$ and M contains no decision literals

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M $\qquad$ unit clause

- pure literal $\quad M \parallel F \implies M\,\ell \parallel F$

  if ℓ occurs in F and ℓ^c does not occur in F and ℓ is undefined in M

- decide $\quad M \parallel F \implies M\,\overset{d}{\ell} \parallel F$

  if ℓ or ℓ^c occurs in F and ℓ is undefined in M

- fail $\quad M \parallel F, C \implies$ fail-state

  if $M \vDash \neg C$ and M contains no decision literals

- backtrack $\quad M\,\overset{d}{\ell}\,N \parallel F, C \implies M\,\ell^c \parallel F, C$

  if $M\,\overset{d}{\ell}\,N \vDash \neg C$ and N contains no decision literals

Outline

1. **Summary of Previous Lecture**
2. **CTL***
3. **Intermezzo**
4. **SAT Solving**

   DPLL   Conflict Analysis

5. **Sorting Networks**
6. **Further Reading**

Example

$$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$$

Example

$\varphi = (\neg 1 \lor 2) \land (\neg 3 \lor 4) \land (\neg 5 \lor \neg 6) \land (6 \lor \neg 5 \lor \neg 2)$

$$\| \; \neg 1 \lor 2, \; \neg 3 \lor 4, \; \neg 5 \lor \neg 6, \; 6 \lor \neg 5 \lor \neg 2$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$\begin{array}{lrll} & \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\ \Longrightarrow & 1^d \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \end{array}$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & 1^d \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & 1^d\ 2 \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate}
\end{array}
$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & & \| \; \neg 1 \vee 2,\; \neg 3 \vee 4,\; \neg 5 \vee \neg 6,\; 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & 1^d & \| \; \neg 1 \vee 2,\; \neg 3 \vee 4,\; \neg 5 \vee \neg 6,\; 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & 1^d\, 2 & \| \; \neg 1 \vee 2,\; \neg 3 \vee 4,\; \neg 5 \vee \neg 6,\; 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & 1^d\, 2\, 3^d & \| \; \neg 1 \vee 2,\; \neg 3 \vee 4,\; \neg 5 \vee \neg 6,\; 6 \vee \neg 5 \vee \neg 2 & \text{decide}
\end{array}
$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{llll}
 & & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & \overset{d}{1} & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2 & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate}
\end{array}
$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

		$\parallel$	$\neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$	
$\Longrightarrow$	$\overset{d}{1}$	$\parallel$	$\neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$	decide
$\Longrightarrow$	$\overset{d}{1}\ 2$	$\parallel$	$\neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$	unit propagate
$\Longrightarrow$	$\overset{d}{1}\ 2\ \overset{d}{3}$	$\parallel$	$\neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$	decide
$\Longrightarrow$	$\overset{d}{1}\ 2\ \overset{d}{3}\ 4$	$\parallel$	$\neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$	unit propagate
$\Longrightarrow$	$\overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}$	$\parallel$	$\neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$	decide

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & 1^d & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & 1^d\ 2 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & 1^d\ 2\ 3^d & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & 1^d\ 2\ 3^d\ 4 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & 1^d\ 2\ 3^d\ 4\ 5^d & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & 1^d\ 2\ 3^d\ 4\ 5^d\ \neg 6 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate}
\end{array}
$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & \overset{d}{1} \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2 \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \neg 5 \ \| & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{backtrack}
\end{array}
$$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & \overset{d}{1} & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2 & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 & \| \; \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate}
\end{array}
$$

conflict is due to $\overset{d}{1}\ 2$ and $\overset{d}{5}\ \neg 6$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & \overset{d}{1} \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2 \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 \parallel & \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate}
\end{array}
$$

conflict is due to $\overset{d}{1}\ 2$ and $\overset{d}{5}\ \neg 6$ hence $\overset{d}{1}$ is incompatible with $\overset{d}{5}$

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & \overset{d}{1} & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2 & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 & \parallel \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate}
\end{array}
$$

conflict is due to $\overset{d}{1}\ 2$ and $\overset{d}{5}\ \neg 6$ hence $\neg 1 \vee \neg 5$ can be inferred

Example

$\varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$
\begin{array}{rrll}
 & & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \\
\Longrightarrow & \overset{d}{1} & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{decide} \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{unit propagate} \\
\Longrightarrow & \overset{d}{1}\ 2\ \neg 5 & \| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2 & \text{backjump}
\end{array}
$$

conflict is due to $\overset{d}{1}\ 2$ and $\overset{d}{5}\ \neg 6$ hence $\neg 1 \vee \neg 5$ can be inferred

Definitions

- backtrack $\quad M \overset{d}{\ell} N \parallel F, C \implies M\, \ell^c \parallel F, C$

  if $M \overset{d}{\ell} N \vDash \neg C$ and N contains no decision literals

Definitions

- backtrack $\quad M \overset{d}{\ell} N \parallel F, C \implies M \ell^c \parallel F, C$

  if $M \overset{d}{\ell} N \vDash \neg C$ and N contains no decision literals

- backjump $\quad M \overset{d}{\ell} N \parallel F, C \implies M \ell' \parallel F, C$

  if $M \overset{d}{\ell} N \vDash \neg C$ and there exists clause $C' \vee \ell'$ such that

  - $F, C \vDash C' \vee \ell'$
  - $M \vDash \neg C'$
  - ℓ' is undefined in M
  - ℓ' or ℓ'^c occurs in F or in $M \overset{d}{\ell} N$

Definitions

- backtrack $\qquad M \overset{d}{\ell} N \parallel F, C \implies M \ell^c \parallel F, C$

  if $M \overset{d}{\ell} N \vDash \neg C$ and N contains no decision literals

- backjump $\qquad M \overset{d}{\ell} N \parallel F, C \implies M \ell' \parallel F, C$

  if $M \overset{d}{\ell} N \vDash \neg C$ and there exists clause $C' \vee \ell'$ such that

  - $F, C \vDash C' \vee \ell'$ $\qquad$ backjump clause
  - $M \vDash \neg C'$
  - ℓ' is undefined in M
  - ℓ' or ℓ'^c occurs in F or in $M \overset{d}{\ell} N$

Definitions

- backtrack $$M \overset{d}{\ell} N \parallel F, C \implies M\, \ell^c \parallel F, C$$
  if $M \overset{d}{\ell} N \vDash \neg C$ and N contains no decision literals
- backjump $$M \overset{d}{\ell} N \parallel F, C \implies M\, \ell' \parallel F, C$$
  if $M \overset{d}{\ell} N \vDash \neg C$ and there exists clause $C' \vee \ell'$ such that
  - $F, C \vDash C' \vee \ell'$ backjump clause
  - $M \vDash \neg C'$
  - ℓ' is undefined in M
  - ℓ' or ℓ'^c occurs in F or in $M \overset{d}{\ell} N$

Example (cont'd)

$\neg 1 \vee \neg 5$ and $\neg 2 \vee \neg 5$ are backjump clauses with respect to $\overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 \parallel \varphi$

Definition

basic DPLL $\mathcal{B}$ consists of transition rules

- unit propagate $\quad M \parallel F, C \vee \ell \implies M\,\ell \parallel F, C \vee \ell$

  if $M \vDash \neg C$ and ℓ is undefined in M

- decide $\quad M \parallel F \implies M\,\ell^{d} \parallel F$

  if ℓ or ℓ^{c} occurs in F and ℓ is undefined in M

- fail $\quad M \parallel F, C \implies$ fail-state

  if $M \vDash \neg C$ and M contains no decision literals

- backjump $\quad M\,\ell^{d}\,N \parallel F, C \implies M\,\ell' \parallel F, C$

  if $M\,\ell^{d}\,N \vDash \neg C$ and there exists clause $C' \vee \ell'$ such that

  - $F, C \vDash C' \vee \ell'$ and $M \vDash \neg C'$
  - ℓ' is undefined in M and ℓ' or ℓ'^{c} occurs in F or in $M\,\ell^{d}\,N$

Theorem

there are no infinite derivations $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Theorem

there are no infinite derivations $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M

Theorem

there are no infinite derivations $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M
- measure state $M_0 \, \ell_1^d \, M_1 \, \ell_2^d \, M_2 \dots \ell_k^d \, M_k \parallel F$ where $M_0, \dots, M_k$ contain no decision literals by tuple $(|M_0|, |M_1|, \dots, |M_k|)$

Theorem

there are no infinite derivations $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M
- measure state $M_0 \, \ell_1^d \, M_1 \, \ell_2^d \, M_2 \dots \ell_k^d \, M_k \parallel F$ where $M_0, \dots, M_k$ contain no decision literals by tuple $(|M_0|, |M_1|, \dots, |M_k|)$
- compare tuples lexicographically using standard order on $\mathbb{N}$

Theorem

there are no infinite derivations $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M
- measure state $M_0 \, \ell_1^d \, M_1 \, \ell_2^d \, M_2 \dots \ell_k^d \, M_k \parallel F$ where $M_0, \dots, M_k$ contain no decision literals by tuple $(|M_0|, |M_1|, \dots, |M_k|)$
- compare tuples lexicographically using standard order on $\mathbb{N}$
- every transition step strictly increases measure

Theorem

there are no infinite derivations $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M
- measure state $M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \dots \overset{d}{\ell_k} M_k \| F$ where $M_0, \dots, M_k$ contain no decision literals by tuple $(|M_0|, |M_1|, \dots, |M_k|)$
- compare tuples lexicographically using standard order on $\mathbb{N}$
- every transition step strictly increases measure
- measure is bounded by $(n + 1)$-tuple $(n, \dots, n)$ where n is total number of atoms

Example

$\| \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$

$$\Longrightarrow \quad 1^d \; \| \; \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad 1^d\, 2 \; \| \; \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d \; \| \; \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d\, 4 \; \| \; \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d\, 4\, 5^d \; \| \; \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d\, 4\, 5^d\, \neg 6 \; \| \; \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad 1^d\, 2\, \neg 5 \; \| \; \varphi \qquad \text{backjump}$$

Example

$$\parallel \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2) \qquad (0)$$

$$\Longrightarrow \quad \overset{d}{1} \parallel \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad \overset{d}{1}\ 2 \parallel \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3} \parallel \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3}\ 4 \parallel \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} \parallel \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 \parallel \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \neg 5 \parallel \varphi \qquad \text{backjump}$$

Example

$$\| \; \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2) \qquad (0)$$

$$\Longrightarrow \quad \overset{d}{1} \;\|\; \varphi \qquad \text{decide} \qquad (0,0)$$

$$\Longrightarrow \quad \overset{d}{1}\; 2 \;\|\; \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad \overset{d}{1}\; 2\; \overset{d}{3} \;\|\; \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad \overset{d}{1}\; 2\; \overset{d}{3}\; 4 \;\|\; \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad \overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5} \;\|\; \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad \overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5}\; \neg 6 \;\|\; \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad \overset{d}{1}\; 2\; \neg 5 \;\|\; \varphi \qquad \text{backjump}$$

Example

$$\| \ \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2) \qquad (0)$$

$$\Longrightarrow \quad 1^d \ \| \ \varphi \qquad \text{decide} \qquad (0,0)$$

$$\Longrightarrow \quad 1^d\, 2 \ \| \ \varphi \qquad \text{unit propagate} \qquad (0,1)$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d \ \| \ \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d\, 4 \ \| \ \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d\, 4\, 5^d \ \| \ \varphi \qquad \text{decide}$$

$$\Longrightarrow \quad 1^d\, 2\, 3^d\, 4\, 5^d\, \neg 6 \ \| \ \varphi \qquad \text{unit propagate}$$

$$\Longrightarrow \quad 1^d\, 2\, \neg 5 \ \| \ \varphi \qquad \text{backjump}$$

Example

$\| \; \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$ $\quad (0)$

$\Longrightarrow$	$\overset{d}{1} \; \| \; \varphi$	decide	$(0, 0)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \| \; \varphi$	unit propagate	$(0, 1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3} \; \| \; \varphi$	decide	$(0, 1, 0)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4 \; \| \; \varphi$	unit propagate	$(0, 1, 1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5} \; \| \; \varphi$	decide	$(0, 1, 1, 0)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5}\; \neg 6 \; \| \; \varphi$	unit propagate	$(0, 1, 1, 1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \neg 5 \; \| \; \varphi$	backjump	$(0, 2)$

Example

$$\| \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2) \qquad (0)$$

$\Longrightarrow$	$\overset{d}{1} \;\|\; \varphi$	decide	$(0, 0)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \;\|\; \varphi$	unit propagate	$(0, 1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3} \;\|\; \varphi$	decide	$(0, 1, 0)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4 \;\|\; \varphi$	unit propagate	$(0, 1, 1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5} \;\|\; \varphi$	decide	$(0, 1, 1, 0)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5}\; \neg 6 \;\|\; \varphi$	unit propagate	$(0, 1, 1, 1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \neg 5 \;\|\; \varphi$	backjump	$(0, 2)$

- decide $\quad (m_0, \dots, m_i) <_{\text{lex}} (m_0, \dots, m_i, 0)$

Example

$\| \; \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$ $\quad (0)$

$\Longrightarrow$	$\overset{d}{1} \; \| \; \varphi$	decide	$(0,0)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \| \; \varphi$	unit propagate	$(0,1)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \overset{d}{3} \; \| \; \varphi$	decide	$(0,1,0)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \overset{d}{3}\; 4 \; \| \; \varphi$	unit propagate	$(0,1,1)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \overset{d}{3}\; 4 \; \overset{d}{5} \; \| \; \varphi$	decide	$(0,1,1,0)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \overset{d}{3}\; 4 \; \overset{d}{5}\; \neg 6 \; \| \; \varphi$	unit propagate	$(0,1,1,1)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \neg 5 \; \| \; \varphi$	backjump	$(0,2)$

- decide $\quad (m_0, \ldots, m_i) <_{\text{lex}} (m_0, \ldots, m_i, 0)$
- unit propagate $\quad (m_0, \ldots, m_i) <_{\text{lex}} (m_0, \ldots, m_i + 1)$

Example

$$\| \; \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2) \qquad (0)$$

$\Longrightarrow$	$\overset{d}{1} \; \| \; \varphi$	decide	$(0,0)$
$\Longrightarrow$	$\overset{d}{1}\; 2 \; \| \; \varphi$	unit propagate	$(0,1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3} \; \| \; \varphi$	decide	$(0,1,0)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4 \; \| \; \varphi$	unit propagate	$(0,1,1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5} \; \| \; \varphi$	decide	$(0,1,1,0)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \overset{d}{3}\; 4\; \overset{d}{5}\; \neg 6 \; \| \; \varphi$	unit propagate	$(0,1,1,1)$
$\Longrightarrow$	$\overset{d}{1}\; 2\; \neg 5 \; \| \; \varphi$	backjump	$(0,2)$

- decide $(m_0, \dots, m_i) <_{\text{lex}} (m_0, \dots, m_i, 0)$
- unit propagate $(m_0, \dots, m_i) <_{\text{lex}} (m_0, \dots, m_i + 1)$
- backjump $(m_0, \dots, m_i) <_{\text{lex}} (m_0, \dots, m_j + 1)$ with $j < i$

Lemma

1. if $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F'$ then
   - $F = F'$

Lemma

1. if $\| F \Longrightarrow^*_{\mathcal{B}} M \| F'$ then
   - $F = F'$
   - M does not contain complementary literals

Lemma

1. if $\| F \Longrightarrow_{\mathcal{B}}^{*} M \| F'$ then
   - $F = F'$
   - M does not contain complementary literals
   - M consists of distinct literals

Lemma

1. if $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F'$ then
   - $F = F'$
   - M does not contain complementary literals
   - M consists of distinct literals
2. if $\parallel F \Longrightarrow^*_{\mathcal{B}} M_0 \, \overset{d}{\ell_1} \, M_1 \, \overset{d}{\ell_2} \, M_2 \cdots \overset{d}{\ell_k} \, M_k \parallel F$ with no decision literals in $M_0, \dots, M_k$

   then $F, \ell_1, \dots, \ell_i \vDash M_i$ for all $0 \leqslant i \leqslant k$

Theorem

if $\| \; F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable

Theorem

if $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Theorem

if $\| \ F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \ \| \ F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\| \ F \Longrightarrow_{\mathcal{B}}^{*} M \ \| \ F \Longrightarrow_{\text{fail}}$ fail-state

Theorem

if $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \| F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\| F \Longrightarrow_{\mathcal{B}}^{*} M \| F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F

Theorem

if $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \| F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\| F \Longrightarrow^*_{\mathcal{B}} M \| F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$

Theorem

if $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$

Theorem

if $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$

Theorem

if $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \| F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\| F \Longrightarrow^*_{\mathcal{B}} M \| F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

Theorem

if $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

② $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F' \not\Longrightarrow_{\mathcal{B}}$

Theorem

if $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. S_n = fail-state if and only if F is unsatisfiable
2. $S_n = M \| F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\| F \Longrightarrow^*_{\mathcal{B}} M \| F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

② $\| F \Longrightarrow^*_{\mathcal{B}} M \| F' \not\Longrightarrow_{\mathcal{B}}$

- $F = F'$

Theorem

if $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\parallel F \Longrightarrow^{*}_{\mathcal{B}} M \parallel F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

② $\parallel F \Longrightarrow^{*}_{\mathcal{B}} M \parallel F' \not\Longrightarrow_{\mathcal{B}}$

- $F = F'$ and all literals in F are defined in M, otherwise decide is applicable

Theorem

if $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

❶ $S_n =$ fail-state if and only if F is unsatisfiable

❷ $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

② $\parallel F \Longrightarrow^*_{\mathcal{B}} M \parallel F' \not\Longrightarrow_{\mathcal{B}}$

- $F = F'$ and all literals in F are defined in M, otherwise decide is applicable
- F contains no clause such that $M \vDash \neg C$, otherwise backjump or fail is applicable

Theorem

if $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \parallel F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\parallel F \Longrightarrow^{*}_{\mathcal{B}} M \parallel F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

② $\parallel F \Longrightarrow^{*}_{\mathcal{B}} M \parallel F' \not\Longrightarrow_{\mathcal{B}}$

- $F = F'$ and all literals in F are defined in M, otherwise decide is applicable
- F contains no clause such that $M \vDash \neg C$, otherwise backjump or fail is applicable
- $M \vDash F$

Theorem

if $\| \; F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ then

1. $S_n =$ fail-state if and only if F is unsatisfiable
2. $S_n = M \; \| \; F'$ only if F is satisfiable and $M \vDash F$

Proof

① (only if) $\| \; F \Longrightarrow^*_{\mathcal{B}} M \; \| \; F \Longrightarrow_{\text{fail}}$ fail-state

- M contains no decision literals and $M \vDash \neg C$ for some C in F
- $F \vDash C$ and $F \vDash M$ and thus $F \vDash \neg C$ and thus F is unsatisfiable

② $\| \; F \Longrightarrow^*_{\mathcal{B}} M \; \| \; F' \not\Longrightarrow_{\mathcal{B}}$

- $F = F'$ and all literals in F are defined in M, otherwise decide is applicable
- F contains no clause such that $M \vDash \neg C$, otherwise backjump or fail is applicable
- $M \vDash F$ and thus F is satisfiable

Lemma

backjump can simulate backtrack

Lemma

backjump can simulate backtrack

Proof

- suppose $\| F \Longrightarrow_{\mathcal{B}}^{*} M \overset{d}{\ell} N \| F \Longrightarrow_{\text{backtrack}} M \ell^c \| F$

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow_{\mathcal{B}}^{*} M\,\overset{d}{\ell}\,N \parallel F \Longrightarrow_{\text{backtrack}} M\,\ell^{c} \parallel F$
- $M\,\overset{d}{\ell}\,N \vDash \neg C$ for some C in F and N contains no decision literals

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow_{\mathcal{B}}^{*} M \overset{d}{\ell} N \parallel F \Longrightarrow_{\text{backtrack}} M \ell^c \parallel F$
- $M \overset{d}{\ell} N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \cdots \overset{d}{\ell_k} M_k$ with all decision literals displayed

Lemma

backjump can simulate backtrack

Proof

- suppose $\| \; F \Longrightarrow_{\mathcal{B}}^{*} M \overset{d}{\ell} N \; \| \; F \Longrightarrow_{\text{backtrack}} M \, \ell^c \; \| \; F$
- $M \overset{d}{\ell} N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \cdots \overset{d}{\ell_k} M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow_{\mathcal{B}}^{*} M \overset{d}{\ell} N \parallel F \Longrightarrow_{\text{backtrack}} M \ell^c \parallel F$
- $M \overset{d}{\ell} N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \cdots \overset{d}{\ell_k} M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause:
  - $F, \ell_1, \ldots, \ell_k, \ell \vDash \neg C$

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow^*_{\mathcal{B}} M \overset{d}{\ell} N \parallel F \Longrightarrow_{\text{backtrack}} M \ell^c \parallel F$
- $M \overset{d}{\ell} N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \cdots \overset{d}{\ell_k} M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause:
  - $F, \ell_1, \dots, \ell_k, \ell \vDash \neg C \quad \Longrightarrow \quad F, \ell_1, \dots, \ell_k, \ell$ is unsatisfiable

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow^*_{\mathcal{B}} M\,\overset{d}{\ell}\,N \parallel F \Longrightarrow_{\text{backtrack}} M\,\ell^c \parallel F$
- $M\,\overset{d}{\ell}\,N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0\,\overset{d}{\ell_1}\,M_1\,\overset{d}{\ell_2}\,M_2 \cdots \overset{d}{\ell_k}\,M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause:
  - $F, \ell_1, \dots, \ell_k, \ell \vDash \neg C \quad\Longrightarrow\quad F, \ell_1, \dots, \ell_k, \ell$ is unsatisfiable $\quad\Longrightarrow\quad F \vDash \ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow^*_{\mathcal{B}} M\,\overset{d}{\ell}\,N \parallel F \Longrightarrow_{\text{backtrack}} M\,\ell^c \parallel F$
- $M\,\overset{d}{\ell}\,N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0\,\overset{d}{\ell_1}\,M_1\,\overset{d}{\ell_2}\,M_2 \cdots \overset{d}{\ell_k}\,M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause:
  - $F, \ell_1, \dots, \ell_k, \ell \vDash \neg C \quad \Longrightarrow \quad F, \ell_1, \dots, \ell_k, \ell$ is unsatisfiable $\quad \Longrightarrow \quad F \vDash \ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$
  - $M \vDash \ell_1 \wedge \cdots \wedge \ell_k$

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow^*_{\mathcal{B}} M \overset{d}{\ell} N \parallel F \Longrightarrow_{\text{backtrack}} M \, \ell^c \parallel F$
- $M \overset{d}{\ell} N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \cdots \overset{d}{\ell_k} M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause:
  - $F, \ell_1, \ldots, \ell_k, \ell \vDash \neg C \quad \Longrightarrow \quad F, \ell_1, \ldots, \ell_k, \ell$ is unsatisfiable $\quad \Longrightarrow \quad F \vDash \ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$
  - $M \vDash \ell_1 \wedge \cdots \wedge \ell_k$ and ℓ^c is undefined in M

Lemma

backjump can simulate backtrack

Proof

- suppose $\parallel F \Longrightarrow^*_{\mathcal{B}} M \overset{d}{\ell} N \parallel F \Longrightarrow_{\text{backtrack}} M\, \ell^c \parallel F$
- $M \overset{d}{\ell} N \vDash \neg C$ for some C in F and N contains no decision literals
- write $M = M_0 \overset{d}{\ell_1} M_1 \overset{d}{\ell_2} M_2 \cdots \overset{d}{\ell_k} M_k$ with all decision literals displayed
- $\ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$ is backjump clause:
  - $F, \ell_1, \dots, \ell_k, \ell \vDash \neg C \quad \Longrightarrow \quad F, \ell_1, \dots, \ell_k, \ell$ is unsatisfiable $\quad \Longrightarrow \quad F \vDash \ell_1^c \vee \cdots \vee \ell_k^c \vee \ell^c$
  - $M \vDash \ell_1 \wedge \cdots \wedge \ell_k$ and ℓ^c is undefined in M
- $M \overset{d}{\ell} N \parallel F \Longrightarrow_{\text{backjump}} M\, \ell^c \parallel F$

Terminology

non-chronological backtracking or conflict-driven backtracking

Terminology

non-chronological backtracking or conflict-driven backtracking

Question

how to find good backjump clauses ?

Terminology

non-chronological backtracking or conflict-driven backtracking

Question

how to find good backjump clauses ?

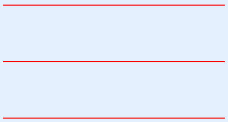
Answer

use conflict graph (lecture 13)

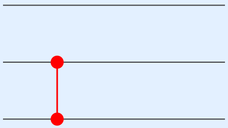
Outline

1. Summary of Previous Lecture
2. CTL*
3. Intermezzo
4. SAT Solving
5. **Sorting Networks**
6. Further Reading

Wires

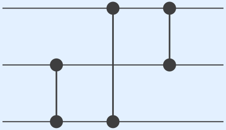


Comparator



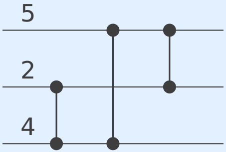
Comparator

Comparator Network

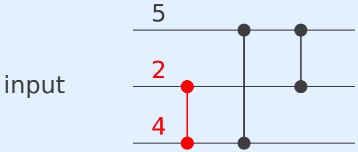


Comparator Network

input

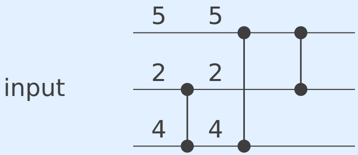


Comparator Network



$4 > 2$

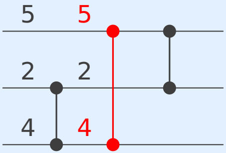
Comparator Network



$4 > 2$

Comparator Network

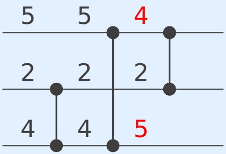
input



$4 > 2$ $\quad$ $4 \not> 5$

Comparator Network

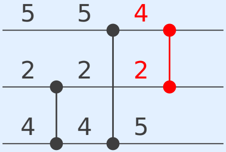
input



$4 > 2$ $4 \not> 5$

Comparator Network

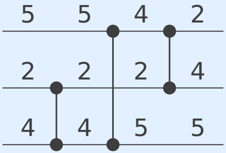
input



$4 > 2$ $4 \not> 5$ $2 \not> 4$

Comparator Network

input

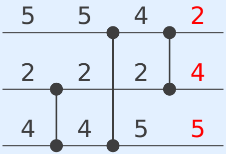


output

$4 > 2$ $4 \not> 5$ $2 \not> 4$

Sorting Network

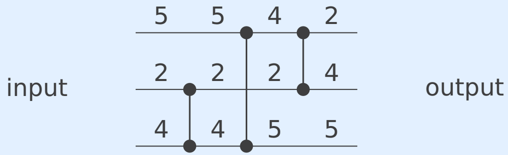
input



output

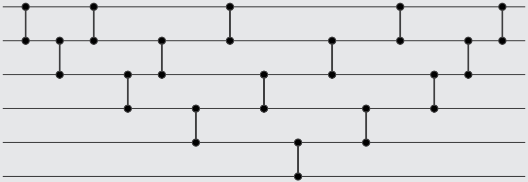
$4 > 2$ $\quad$ $4 \not> 5$ $\quad$ $2 \not> 4$

Sorting Network



$4 > 2$ $4 \not> 5$ $2 \not> 4$

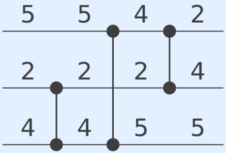
Example



- size $(=$ number of comparators$)$: 15

Sorting Network

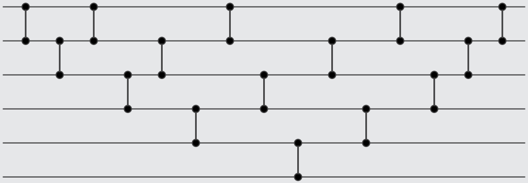
input



output

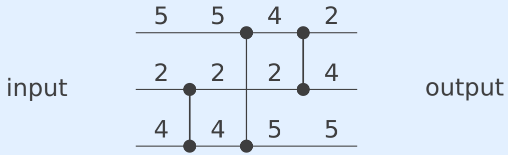
$4 > 2$ $4 \not> 5$ $2 \not> 4$

Example



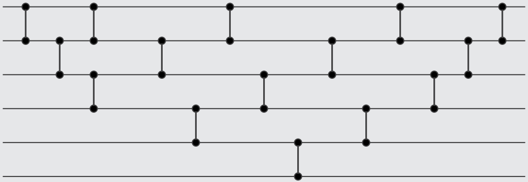
- size $(=$ number of comparators$)$: 15
- depth

Sorting Network



$4 > 2$ $\quad 4 \not> 5$ $\quad 2 \not> 4$

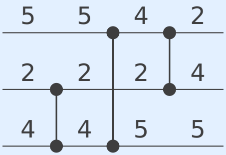
Example



- size ($=$ number of comparators): 15
- depth

Sorting Network

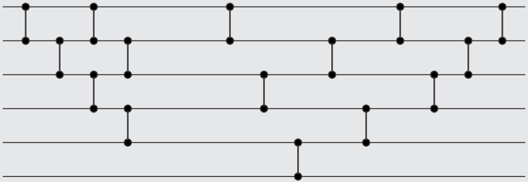
input



output

$4 > 2$ $4 \not> 5$ $2 \not> 4$

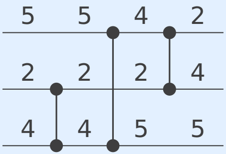
Example



- size $(=$ number of comparators$)$: 15
- depth

Sorting Network

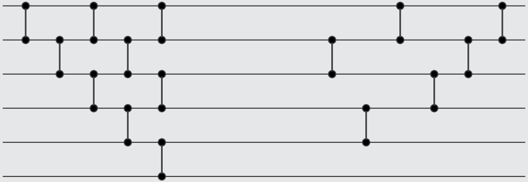
input



output

$4 > 2$ $4 \not> 5$ $2 \not> 4$

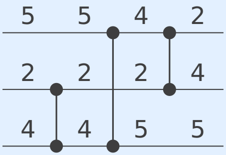
Example



- size $(=$ number of comparators$)$: 15
- depth

Sorting Network

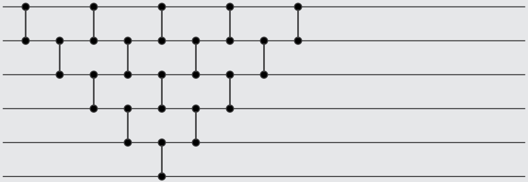
input



output

4 > 2 4 $\not>$ 5 2 $\not>$ 4

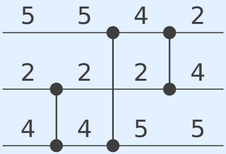
Example



- size (= number of comparators): 15
- depth

Sorting Network

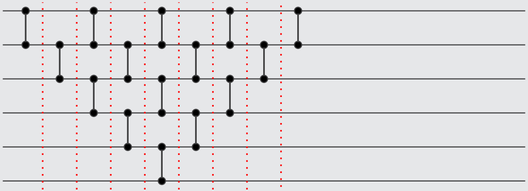
input



output

$4 > 2 \qquad 4 \not> 5 \qquad 2 \not> 4$

Example



- size (= number of comparators): 15
- depth: 9

Definition

sorting network is comparator network that transforms any input sequence $a = (a_1, \ldots, a_n)$ of natural numbers into sorted output sequence $b = (b_1, \ldots, b_n)$

Definition

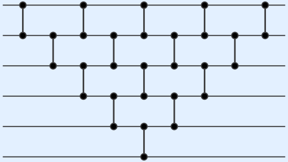
sorting network is comparator network that transforms any input sequence $a = (a_1, \ldots, a_n)$ of natural numbers into sorted output sequence $b = (b_1, \ldots, b_n)$:

b is permutation of a and $b_1 \leqslant \cdots \leqslant b_n$

Definition

sorting network is comparator network that transforms any input sequence $a = (a_1, \dots, a_n)$ of natural numbers into **sorted** output sequence $b = (b_1, \dots, b_n)$:

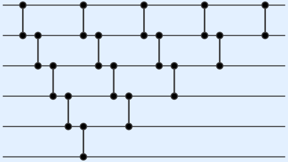
b is permutation of a and $b_1 \leqslant \cdots \leqslant b_n$



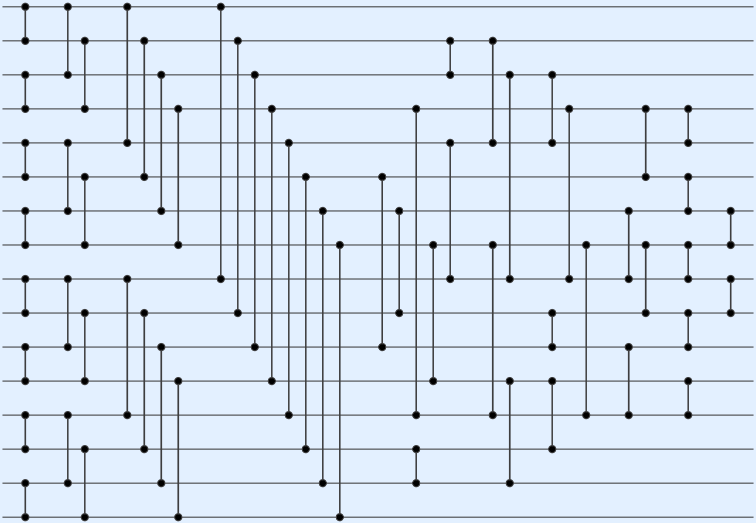
Definition

sorting network is comparator network that transforms any input sequence $a = (a_1, \dots, a_n)$ of natural numbers into **sorted** output sequence $b = (b_1, \dots, b_n)$:

b is permutation of a and $b_1 \leqslant \cdots \leqslant b_n$



Sorting Network ?



Questions

1. how to check that comparator network is sorting network ?

Questions

① how to check that comparator network is sorting network ?

② how to find optimal (with respect to size or depth) sorting networks ?

Questions

① how to check that comparator network is sorting network ?

② how to find optimal (with respect to size or depth) sorting networks ?

Answers

① testing all $n!$ permutations of $1, \ldots, n$ for network with n wires suffices

Questions

① how to check that comparator network is sorting network ?

② how to find optimal (with respect to size or depth) sorting networks ?

Answers

① testing all $n!$ permutations of $1, \dots, n$ for network with n wires suffices

Questions

① how to check that comparator network is sorting network ?

② how to find optimal (with respect to size or depth) sorting networks ?

Answers

① testing all $n!$ permutations of $1, \ldots, n$ for network with n wires suffices

② very difficult problem . . .

Outline

Huth and Ryan

- Section 3.5

DPLL

- Section 2 of Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T)
  Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli
  Journal of the ACM 53(6), pp. 937–977, 2006
  doi: 10.1145/1217856.1217859

Huth and Ryan

- Section 3.5

DPLL

- Section 2 of Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T)
  Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli
  Journal of the ACM 53(6), pp. 937–977, 2006
  doi: 10.1145/1217856.1217859

Sorting Networks

- Wikipedia [accessed December 14, 2022]
- Section 5.3.4 of The Art of Computer Programming
  Donald Knuth

Important Concepts

- abstract DPLL
- basic DPLL
- backjump
- backtrack
- comparator network
- CTL^*
- decide
- depth
- fail-state
- path formula
- pure literal
- size
- sorting network
- state formula
- unit propagation

Important Concepts

- abstract DPLL
- basic DPLL
- backjump
- backtrack
- comparator network
- CTL*
- decide
- depth
- fail-state
- path formula
- pure literal
- size
- sorting network
- state formula
- unit propagation

homework for June 13

Important Concepts

- abstract DPLL
- basic DPLL
- backjump
- backtrack
- comparator network
- CTL*
- decide
- depth
- fail-state
- path formula
- pure literal
- size
- sorting network
- state formula
- unit propagation

homework for June 13

evaluation SS 2024