universität innsbruck

SS 2024 lecture 1

Constraint Solving

René Thiemann and Fabian Mitterwallner

based on a previous course by Aart Middeldorp

Outline

Important Information

- LVA 703304 (VO 2) + 703305 (PS 2)

Important Information

- LVA 703304 (VO 2) + 703305 (PS 2)
- http://cl-informatik.uibk.ac.at/teaching/ss24/cs

Important Information

- LVA 703304 (VO 2) + 703305 (PS 2)
- `http://cl-informatik.uibk.ac.at/teaching/ss24/cs`
- Please register for both VO and PS

Important Information

- LVA 703304 (VO 2) + 703305 (PS 2)
- `http://cl-informatik.uibk.ac.at/teaching/ss24/cs`
- Please register for both VO and PS
- OLAT link for VO (contains recordings)

Important Information

- LVA 703304 (VO 2) + 703305 (PS 2)
- `http://cl-informatik.uibk.ac.at/teaching/ss24/cs`
- Please register for both VO and PS
- OLAT link for VO (contains recordings)

Time and Place

VO	Tuesday	8:15 – 10:00	SR 13	RT
PS	Friday	8:30 – 10:00	3W04	FM

Consultation Hours

René Thiemann	3M09	Tuesday	10:15 – 11:15
Fabian Mitterwallner	3M03	Thursday	10:30 – 12:00

Consultation Hours

René Thiemann	3M09	Tuesday	10:15 – 11:15
Fabian Mitterwallner	3M03	Thursday	10:30 – 12:00

Schedule – one deviation: PS in week 11 on June 3, 17:15–18:45

week 1	05.03 & 08.03	week 6	23.04 & 26.04	week 11	28.05 & 03.06 at 17:15
week 2	12.03 & 15.03	week 7	30.04 & 03.05	week 12	04.06 & 07.06
week 3	19.03 & 22.03	week 8	07.05 & 10.05	week 13	11.06 & 14.06
week 4	09.04 & 12.04	week 9	14.05 & 17.05	week 14	18.06 & 21.06
week 5	16.04 & 19.04	week 10	21.05 & 24.05	week 15	25.06

Consultation Hours

René Thiemann	3M09	Tuesday	10:15 – 11:15
Fabian Mitterwallner	3M03	Thursday	10:30 – 12:00

Schedule – one deviation: PS in week 11 on June 3, 17:15–18:45

week 1	05.03 & 08.03	week 6	23.04 & 26.04	week 11	28.05 & 03.06 at 17:15	
week 2	12.03 & 15.03	week 7	30.04 & 03.05	week 12	04.06 & 07.06	
week 3	19.03 & 22.03	week 8	07.05 & 10.05	week 13	11.06 & 14.06	
week 4	09.04 & 12.04	week 9	14.05 & 17.05	week 14	18.06 & 21.06	Q & A
week 5	16.04 & 19.04	week 10	21.05 & 24.05	week 15	25.06	first exam

Grading — Vorlesung

- first exam on June 25
- registration starts 5 weeks before exam and ends 1 week before exam
- de-registration is possible until 23:59 on June 23
- second and third exams in September and February (on demand)

Grading — Vorlesung

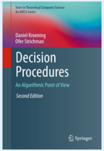
- first exam on June 25
- registration starts 5 weeks before exam and ends 1 week before exam
- de-registration is possible until 23:59 on June 23
- second and third exams in September and February (on demand)

Grading — Proseminar

- solved exercises must be marked in OLAT before 8 am on Friday
- 10 points per PS
- attendance is compulsory; unexcused absence is allowed twice
  (email solutions to get some points in such cases)
- grading: 80 % for marked exercises, 20 % for presentation of solutions

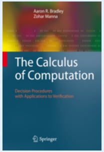
Literature

Daniel Kröning and Ofer Strichman
Decision Procedures – An Algorithmic Point of View (Second Edition)
Springer, 2016 online version via Ebook Central



`http://www.decision-procedures.org`

Aaron Bradley and Zohar Manna
The Calculus of Computation – Decision Procedures with Applications to Verification
Springer, 2007 online version via Ebook Central

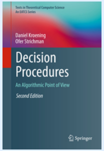


`http://theory.stanford.edu/~arbrad/book.html`

Literature

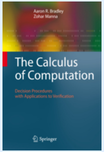
Daniel Kröning and Ofer Strichman
Decision Procedures – An Algorithmic Point of View (Second Edition)
Springer, 2016 online version via Ebook Central

http://www.decision-procedures.org



Aaron Bradley and Zohar Manna
The Calculus of Computation – Decision Procedures with Applications to Verification
Springer, 2007 online version via Ebook Central

http://theory.stanford.edu/~arbrad/book.html



Online Material

slides and additional reading material are available from uibk.ac.at domain

Topics

- propositional logic (SAT)
  cardinality constraints, DPLL, maxSAT, NP-completeness, unsatisfiable cores
- satisfiability modulo theories (SMT)
  DPLL(T), Nelson–Oppen combination method
- equality logic and uninterpreted functions (EUF)
  congruence closure, graph-based reduction
- linear arithmetic (LIA and LRA)
  branch and bound, cubes and equalities, difference logic, simplex algorithm, small model property
- bit vectors (BV) and floating points
  bit-vector arithmetic, bounded model checking, fixed-point arithmetic, flattening
- arrays and pointers
  array properties, lazy encoding, pointer logic
- quantified formulas
  Cooper's method, Ferrante–Rackoff's method, PSPACE-completeness, quantified boolean formulas (QBF)

Topics

- propositional logic (SAT)
  cardinality constraints, DPLL, maxSAT, NP-completeness, unsatisfiable cores
- satisfiability modulo theories (SMT)
  DPLL(T), Nelson–Oppen combination method
- equality logic and uninterpreted functions (EUF)
  congruence closure, graph-based reduction
- linear arithmetic (LIA and LRA)
  branch and bound, cubes and equalities, difference logic, simplex algorithm, small model property
- bit vectors (BV) and floating points
  bit-vector arithmetic, bounded model checking, fixed-point arithmetic, flattening
- arrays and pointers
  array properties, lazy encoding, pointer logic
- quantified formulas
  Cooper's method, Ferrante–Rackoff's method, PSPACE-completeness, quantified boolean formulas (QBF)

Outline

Concepts of Propositional Logic

- formula
- assignment
- satisfiability
- validity
- negation normal form (NNF)
- conjunctive normal form (CNF)
- disjunctive normal form (DNF)
- literal

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$
  - top,bottom $\top, \bot$

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$
  - top,bottom $\top, \bot$
  - negation $\neg$ $\neg p$ "not p"

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$
  - top,bottom $\top, \bot$
  - negation $\neg$ $\neg p$ "not p"
  - conjunction $\wedge$ $p \wedge q$ "p and q"
  - disjunction $\vee$ $p \vee q$ "p or q"
  - implication $\rightarrow$ $p \rightarrow q$ "if p then q"
  - equivalence $\leftrightarrow$ $p \leftrightarrow q$ "p if and only if q"

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$
  - top,bottom $\top, \bot$
  - negation $\neg$ — $\neg p$ — "not p"
  - conjunction $\wedge$ — $p \wedge q$ — "p and q"
  - disjunction $\vee$ — $p \vee q$ — "p or q"
  - implication $\rightarrow$ — $p \rightarrow q$ — "if p then q"
  - equivalence $\leftrightarrow$ — $p \leftrightarrow q$ — "p if and only if q"

  according to BNF grammar $\varphi ::= p \mid \bot \mid \top \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi)$

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms — $p, q, r, p_1, p_2, \ldots$
  - top,bottom — $\top, \bot$
  - negation — $\neg$ — $\neg p$ — "not p"
  - conjunction — $\wedge$ — $p \wedge q$ — "p and q"
  - disjunction — $\vee$ — $p \vee q$ — "p or q"
  - implication — $\rightarrow$ — $p \rightarrow q$ — "if p then q"
  - equivalence — $\leftrightarrow$ — $p \leftrightarrow q$ — "p if and only if q"

  according to BNF grammar $\varphi ::= p \mid \bot \mid \top \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi)$
- notational conventions:
  - binding precedence $\neg > \wedge > \vee > \rightarrow, \leftrightarrow$

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$
  - top,bottom $\top, \bot$
  - negation $\neg$ $\neg p$ "not p"
  - conjunction $\wedge$ $p \wedge q$ "p and q"
  - disjunction $\vee$ $p \vee q$ "p or q"
  - implication $\rightarrow$ $p \rightarrow q$ "if p then q"
  - equivalence $\leftrightarrow$ $p \leftrightarrow q$ "p if and only if q"

  according to BNF grammar $\varphi ::= p \mid \bot \mid \top \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi)$
- notational conventions:
  - binding precedence $\neg > \wedge > \vee > \rightarrow, \leftrightarrow$ omit outer parentheses

Definition (Propositional Logic: Syntax)

- propositional formulas are built from
  - atoms $p, q, r, p_1, p_2, \ldots$
  - top,bottom $\top, \bot$
  - negation $\neg$ $\neg p$ "not p"
  - conjunction $\wedge$ $p \wedge q$ "p and q"
  - disjunction $\vee$ $p \vee q$ "p or q"
  - implication $\rightarrow$ $p \rightarrow q$ "if p then q"
  - equivalence $\leftrightarrow$ $p \leftrightarrow q$ "p if and only if q"

  according to BNF grammar $\varphi ::= p \mid \bot \mid \top \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi)$
- notational conventions:
  - binding precedence $\neg > \wedge > \vee > \rightarrow, \leftrightarrow$ omit outer parentheses
  - $\rightarrow$, $\wedge$, $\vee$ are right-associative: $p \rightarrow q \rightarrow r$ denotes $p \rightarrow (q \rightarrow r)$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v \colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$

  truth values

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$
  truth values
- extension to formulas:

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$
  truth values
- extension to formulas:
  - $v(\top) = \mathsf{T}$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$
  truth values
- extension to formulas:
  - $v(\top) = \mathsf{T}$
  - $v(\bot) = \mathsf{F}$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$ truth values
- extension to formulas:
  - $v(\top) = \mathsf{T}$
  - $v(\bot) = \mathsf{F}$
  - $v(\neg\varphi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = \mathsf{F} \\ \mathsf{F} & \text{otherwise} \end{cases}$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{\, p \mid p \text{ is atom} \,\} \to \{\, \mathsf{T}, \mathsf{F} \,\}$

  truth values
- extension to formulas:
  - $v(\top) = \mathsf{T}$
  - $v(\bot) = \mathsf{F}$
  - $v(\neg\varphi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = \mathsf{F} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \wedge \psi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = v(\psi) = \mathsf{T} \\ \mathsf{F} & \text{otherwise} \end{cases}$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$

  truth values
- extension to formulas:
  - $v(\top) = \mathsf{T}$
  - $v(\bot) = \mathsf{F}$
  - $v(\neg\varphi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = \mathsf{F} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \wedge \psi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = v(\psi) = \mathsf{T} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \vee \psi) = \begin{cases} \mathsf{F} & \text{if } v(\varphi) = v(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T}, \mathsf{F}\}$

  truth values
- extension to formulas:
  - $v(\top) = \mathsf{T}$
  - $v(\bot) = \mathsf{F}$
  - $v(\neg\varphi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = \mathsf{F} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \wedge \psi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = v(\psi) = \mathsf{T} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \vee \psi) = \begin{cases} \mathsf{F} & \text{if } v(\varphi) = v(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$
  - $v(\varphi \to \psi) = \begin{cases} \mathsf{F} & \text{if } v(\varphi) = \mathsf{T} \text{ and } v(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$

Definition (Propositional Logic: Semantics)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \to \{\mathsf{T},\mathsf{F}\}$ (truth values)
- extension to formulas:

  - $v(\top) = \mathsf{T}$
  - $v(\bot) = \mathsf{F}$
  - $v(\neg\varphi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = \mathsf{F} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \wedge \psi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = v(\psi) = \mathsf{T} \\ \mathsf{F} & \text{otherwise} \end{cases}$
  - $v(\varphi \vee \psi) = \begin{cases} \mathsf{F} & \text{if } v(\varphi) = v(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$
  - $v(\varphi \to \psi) = \begin{cases} \mathsf{F} & \text{if } v(\varphi) = \mathsf{T} \text{ and } v(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$
  - $v(\varphi \leftrightarrow \psi) = \begin{cases} \mathsf{T} & \text{if } v(\varphi) = v(\psi) \\ \mathsf{F} & \text{otherwise} \end{cases}$

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

if $v(\psi) = \mathrm{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathrm{T}$, for every valuation v

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

if $v(\psi) = \mathrm{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathrm{T}$, for every valuation v

- formula φ is valid if $v(\varphi) = \mathrm{T}$ for every valuation v

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

  if $v(\psi) = \mathrm{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathrm{T}$, for every valuation v

- formula φ is valid if $v(\varphi) = \mathrm{T}$ for every valuation v
- formula φ is satisfiable if $v(\varphi) = \mathrm{T}$ for some valuation v

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \models \psi$$

  if $v(\psi) = \mathrm{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathrm{T}$, for every valuation v
- formula φ is valid if $v(\varphi) = \mathrm{T}$ for every valuation v
- formula φ is satisfiable if $v(\varphi) = \mathrm{T}$ for some valuation v
- formulas φ and ψ are equivalent ($\varphi \equiv \psi$) if $v(\varphi) = v(\psi)$ for every valuation v

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

  if $v(\psi) = \mathsf{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathsf{T}$, for every valuation v
- formula φ is valid if $v(\varphi) = \mathsf{T}$ for every valuation v
- formula φ is satisfiable if $v(\varphi) = \mathsf{T}$ for some valuation v
- formulas φ and ψ are equivalent ($\varphi \equiv \psi$) if $v(\varphi) = v(\psi)$ for every valuation v
- formulas φ and ψ are equisatisfiable ($\varphi \approx \psi$) if

$$\varphi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

  if $v(\psi) = \mathsf{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathsf{T}$, for every valuation v
- formula φ is valid if $v(\varphi) = \mathsf{T}$ for every valuation v
- formula φ is satisfiable if $v(\varphi) = \mathsf{T}$ for some valuation v
- formulas φ and ψ are equivalent ($\varphi \equiv \psi$) if $v(\varphi) = v(\psi)$ for every valuation v
- formulas φ and ψ are equisatisfiable ($\varphi \approx \psi$) if

$$\varphi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Theorem

- formula φ is valid $\iff$ $\neg\varphi$ is unsatisfiable

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

  if $v(\psi) = \mathsf{T}$ whenever $v(\varphi_1) = v(\varphi_2) = \cdots = v(\varphi_n) = \mathsf{T}$, for every valuation v

- formula φ is valid if $v(\varphi) = \mathsf{T}$ for every valuation v
- formula φ is satisfiable if $v(\varphi) = \mathsf{T}$ for some valuation v
- formulas φ and ψ are equivalent ($\varphi \equiv \psi$) if $v(\varphi) = v(\psi)$ for every valuation v
- formulas φ and ψ are equisatisfiable ($\varphi \approx \psi$) if

$$\varphi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Theorem

- formula φ is valid $\iff$ $\neg\varphi$ is unsatisfiable
- validity and satisfiability are decidable

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms
- literal is atom p or negation $\neg p$ of atom

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms
- literal is atom p or negation $\neg p$ of atom
- clause is disjunction of literals

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms
- literal is atom p or negation $\neg p$ of atom
- clause is disjunction of literals
- conjunctive normal form (CNF) is conjunction of clauses

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms
- literal is atom p or negation $\neg p$ of atom
- clause is disjunction of literals
- conjunctive normal form (CNF) is conjunction of clauses
- disjunctive normal form (DNF) is disjunction of conjunctions of literals

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms
- literal is atom p or negation $\neg p$ of atom
- clause is disjunction of literals
- conjunctive normal form (CNF) is conjunction of clauses
- disjunctive normal form (DNF) is disjunction of conjunctions of literals

Theorem

$\forall$ formula φ $\exists$ CNF ψ $\exists$ DNF χ such that $\varphi \equiv \psi \equiv \chi$

Definitions

- negation normal form (NNF) is formula without implication and equivalence, and with negation only applied to atoms
- literal is atom p or negation $\neg p$ of atom
- clause is disjunction of literals
- conjunctive normal form (CNF) is conjunction of clauses
- disjunctive normal form (DNF) is disjunction of conjunctions of literals

Theorem

$\forall$ formula φ $\exists$ CNF ψ $\exists$ DNF χ such that $\varphi \equiv \psi \equiv \chi$

Satisfiability (SAT)

instance: (propositional) formula φ

question: is φ satisfiable?

Theorem

SAT is NP-complete

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

DIMACS Input Format

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

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

DIMACS Input Format

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

4 atoms

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

DIMACS Input Format

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

4 atoms and 3 clauses

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

DIMACS Input Format

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

4 atoms and 3 clauses

$x_1 \vee \neg x_2 \vee x_4$

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

DIMACS Input Format

`c`	
`c comments`	
`c`	
`p cnf 4 3`	4 atoms and 3 clauses
`1 -2 4 0`	$x_1 \lor \neg x_2 \lor x_4$
`-1 2 -3 -4 0`	$\neg x_1 \lor x_2 \lor \neg x_3 \lor \neg x_4$
`3 -2 0`	

Theorem

SAT is NP-complete, even for CNF formulas

Remark

most SAT solvers require CNF as input

DIMACS Input Format

`c`	
`c comments`	
`c`	
`p cnf 4 3`	4 atoms and 3 clauses
`1 -2 4 0`	$x_1 \vee \neg x_2 \vee x_4$
`-1 2 -3 -4 0`	$\neg x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4$
`3 -2 0`	$x_3 \vee \neg x_2$

SAT Applications

Applications of SAT

- Encoding games
  `http://cl-informatik.uibk.ac.at/software/puzzles/`
- Strategies and configurations
- Cryptanalysis
- Many graph problems
- Component of reasoning in more complex logics

Outline

Remarks

- translation from arbitrary formula to equivalent CNF is expensive

Remarks

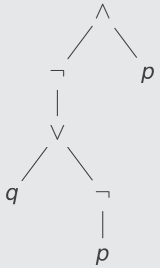
- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Remarks

- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Example (Tseitin's Transformation)

- $\varphi = \neg(q \vee \neg p) \wedge p$



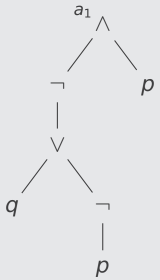
Remarks

- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Example (Tseitin's Transformation)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- introduce new variable for each propositional connective:

$$a_1 \quad \neg(q \vee \neg p) \wedge p$$



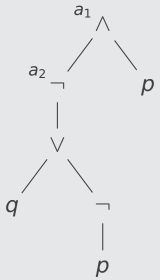
Remarks

- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Example (Tseitin's Transformation)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- introduce new variable for each propositional connective:

$$a_1 \quad \neg(q \vee \neg p) \wedge p$$
$$a_2 \quad \neg(q \vee \neg p)$$



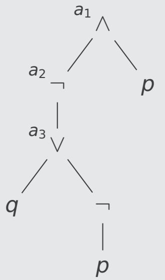
Remarks

- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Example (Tseitin's Transformation)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- introduce new variable for each propositional connective:

a_1	$\neg(q \vee \neg p) \wedge p$	a_3	$q \vee \neg p$
a_2	$\neg(q \vee \neg p)$		



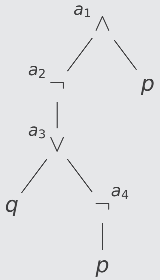
Remarks

- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Example (Tseitin's Transformation)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- introduce new variable for each propositional connective:

a_1	$\neg(q \vee \neg p) \wedge p$	a_3	$q \vee \neg p$
a_2	$\neg(q \vee \neg p)$	a_4	$\neg p$



Remarks

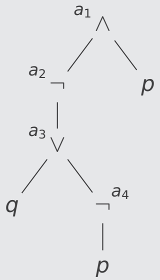
- translation from arbitrary formula to equivalent CNF is expensive
- Tseitin's transformation is linear-time translation to equisatisfiable CNF
- in lecture: only consider formulas without $\rightarrow$ and $\leftrightarrow$

Example (Tseitin's Transformation)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- introduce new variable for each propositional connective:

a_1	$\neg(q \vee \neg p) \wedge p$	a_3	$q \vee \neg p$
a_2	$\neg(q \vee \neg p)$	a_4	$\neg p$

- $\varphi \approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Proof

1.

φ	ψ	$\varphi \leftrightarrow \neg\psi$	$(\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
T	T	F	F
T	F	T	T
F	T	T	T
F	F	F	F

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Example (cont'd)

$$\varphi \approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Example (cont'd)

$$\begin{aligned}\varphi &\approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p) \\ &\equiv a_1 \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p)\end{aligned}$$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Example (cont'd)

$$\begin{aligned}\varphi &\approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p) \\ &\equiv a_1 \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p) \wedge (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3)\end{aligned}$$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Example (cont'd)

$$\begin{aligned}
\varphi &\approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p) \\
&\equiv a_1 \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p) \wedge (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3) \\
&\qquad \wedge (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (\neg a_3 \vee q \vee a_4)
\end{aligned}$$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Example (cont'd)

$$\begin{aligned}
\varphi \approx\ & a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p) \\
\equiv\ & a_1 \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p) \wedge (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3) \\
& \wedge (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (\neg a_3 \vee q \vee a_4) \wedge (a_4 \vee p) \wedge (\neg a_4 \vee \neg p)
\end{aligned}$$

Lemma

1. $(\varphi \leftrightarrow \neg\psi) \equiv (\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
2. $(\varphi \leftrightarrow \psi \wedge \chi) \equiv (\neg\varphi \vee \psi) \wedge (\neg\varphi \vee \chi) \wedge (\varphi \vee \neg\psi \vee \neg\chi)$
3. $(\varphi \leftrightarrow \psi \vee \chi) \equiv (\varphi \vee \neg\psi) \wedge (\varphi \vee \neg\chi) \wedge (\neg\varphi \vee \psi \vee \chi)$

Example (cont'd)

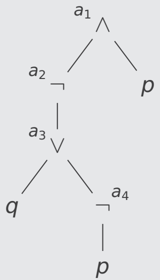
$$\begin{aligned}
\varphi &\approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p) \\
&\equiv a_1 \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p) \wedge (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3) \\
&\quad \wedge (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (\neg a_3 \vee q \vee a_4) \wedge (a_4 \vee p) \wedge (\neg a_4 \vee \neg p)
\end{aligned}$$

Improvement (Plaisted & Greenbaum 1986)

replace equivalence ($\leftrightarrow$) by implication ($\rightarrow$ or $\leftarrow$) based on polarity of subformulas

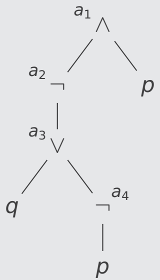
Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



Example (cont'd)

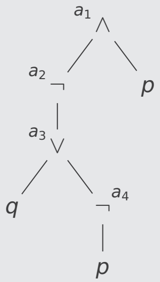
- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \leftrightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



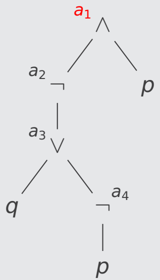
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \leftrightarrow \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



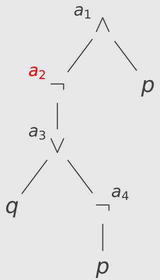
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \to a_2 \wedge p) \wedge (a_2 \to \neg a_3) \wedge (a_3 \leftrightarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



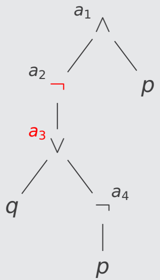
replace $a \leftrightarrow \psi$ by $a \to \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftrightarrow \neg p)$



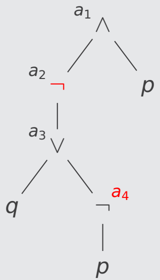
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftarrow \neg p)$



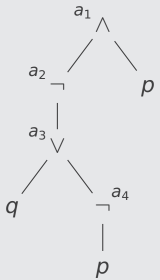
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftarrow \neg p)$
- $a_1 \rightarrow a_2 \wedge p \equiv (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p)$



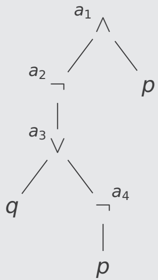
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftarrow \neg p)$
- $a_1 \rightarrow a_2 \wedge p \equiv (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p)$
- $a_2 \rightarrow \neg a_3 \equiv (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3)$



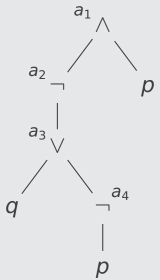
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftarrow \neg p)$
- $a_1 \rightarrow a_2 \wedge p \equiv (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p)$
- $a_2 \rightarrow \neg a_3 \equiv (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3)$
- $a_3 \leftarrow q \vee a_4 \equiv (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (\neg a_3 \vee q \vee a_4)$



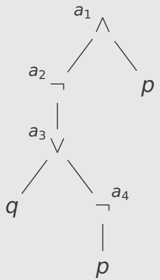
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftarrow \neg p)$
- $a_1 \rightarrow a_2 \wedge p \;\equiv\; (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p)$
- $a_2 \rightarrow \neg a_3 \;\equiv\; (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3)$
- $a_3 \leftarrow q \vee a_4 \;\equiv\; (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (\neg a_3 \vee q \vee a_4)$
- $a_4 \leftarrow \neg p \;\equiv\; (a_4 \vee p) \wedge (\neg a_4 \vee \neg p)$



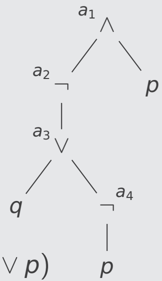
replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Example (cont'd)

- $\varphi = \neg(q \vee \neg p) \wedge p$
- $\varphi \approx a_1 \wedge (a_1 \rightarrow a_2 \wedge p) \wedge (a_2 \rightarrow \neg a_3) \wedge (a_3 \leftarrow q \vee a_4) \wedge (a_4 \leftarrow \neg p)$
- $a_1 \rightarrow a_2 \wedge p \equiv (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (a_1 \vee \neg a_2 \vee \neg p)$
- $a_2 \rightarrow \neg a_3 \equiv (a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3)$
- $a_3 \leftarrow q \vee a_4 \equiv (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (\neg a_3 \vee q \vee a_4)$
- $a_4 \leftarrow \neg p \equiv (a_4 \vee p) \wedge (\neg a_4 \vee \neg p)$
- $\varphi \approx a_1 \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee p) \wedge (\neg a_2 \vee \neg a_3) \wedge (a_3 \vee \neg q) \wedge (a_3 \vee \neg a_4) \wedge (a_4 \vee p)$



replace $a \leftrightarrow \psi$ by $a \rightarrow \psi$ if ψ occurs only positively, and by $a \leftarrow \psi$ if ψ never occurs positively

Definition

subformula ψ occurs positively in formula φ if number of negations on path from root of φ to root of ψ in parse tree of φ is even

Outline

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

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

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

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

Example

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

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

unit propagation: atom 2 must be false

Example

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

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

unit propagation: atom 3 must be true

Example

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

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

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

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

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

unit propagation: atom 4 must be true

Example

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

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

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

Example

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

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

unit propagation: atom 2 must be true

Example

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

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

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

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee l \quad \Longrightarrow \quad M\, l \parallel F, C \vee l$

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

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee l \quad \Longrightarrow \quad M\, l \parallel F, C \vee l$

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

Definition (Transition Rules)

- unit propagate $\quad M \parallel F, C \vee l \quad \Longrightarrow \quad M\, l \parallel F, C \vee l$

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

- pure literal $\quad M \parallel F \quad \Longrightarrow \quad M\, l \parallel F$

  if l occurs in F and l^c (complement of l) does not occur in F and l is undefined in M

Definition (Transition Rules)

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

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

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

  if l occurs in F and l^c (complement of l) does not occur in F and l is undefined in M

- decide $\qquad M \parallel F \implies M\, \overset{d}{l} \parallel F$

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

Definition (Transition Rules)

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

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

- pure literal $M \parallel F \implies M\, l \parallel F$

  if l occurs in F and l^c (complement of l) does not occur in F and l is undefined in M

- decide $M \parallel F \implies M\, l^d \parallel F$

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

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

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

Definition (Transition Rules)

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

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

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

  if l occurs in F and l^c (complement of l) does not occur in F and l is undefined in M

- decide $\qquad M \parallel F \implies M\, \overset{d}{l} \parallel F$

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

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

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

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

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

Outline

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

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

$$\| \ \neg 1 \vee 2,\ \neg 3 \vee 4,\ \neg 5 \vee \neg 6,\ 6 \vee \neg 5 \vee \neg 2$$

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

Example

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

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

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

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

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

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

Example

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

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

conflict is due to $\overset{d}{1}\ 2$ and $\overset{d}{5}\ \neg 6$ 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 $$M \overset{d}{l} N \parallel F, C \implies M\, l^c \parallel F, C$$
  if $M \overset{d}{l} N \models \neg C$ and N contains no decision literals

Definitions

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

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

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

  if $M \overset{d}{l} N \models \neg C$ and $\exists$ clause $C' \vee l'$ such that

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

Definitions

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

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

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

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

  backjump clause

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

Definitions

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

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

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

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

  backjump clause

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

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 $\quad M \parallel F, C \vee l \quad \Longrightarrow \quad M\, l \parallel F, C \vee l$

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

- decide $\quad M \parallel F \quad \Longrightarrow \quad M\, l^{d} \parallel F$

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

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

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

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

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

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

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 $\| F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M
- measure state $M_0 \, l_1^d \, M_1 \, l_2^d \, M_2 \dots l_k^d \, M_k \| F$ 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 $M_0 \, l_1^d \, M_1 \, l_2^d \, M_2 \dots l_k^d \, M_k \parallel F$ 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 $M_0 \overset{d}{l_1} M_1 \overset{d}{l_2} M_2 \dots \overset{d}{l_k} M_k \parallel F$ 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 $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} S_2 \Longrightarrow_{\mathcal{B}} \cdots$

Proof

- for list of distinct literals M, $|M|$ is length of M
- measure state $M_0 \overset{d}{l_1} M_1 \overset{d}{l_2} M_2 \ldots \overset{d}{l_k} M_k \parallel F$ where $M_0, \ldots, M_k$ contain no decision literals by tuple $(|M_0|, |M_1|, \ldots, |M_k|)$
- compare tuples lexicographically using standard order on $\mathbb{N}$
- every transition step strictly increases measure
- measure is bounded by $(n+1)$-tuple $(n, \ldots, n)$ where n is total number of atoms

Example

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

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

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

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

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

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

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

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

Example

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

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

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

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

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

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

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

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

Example

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

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

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

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

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

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

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

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

Example

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

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

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

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

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

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

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

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

Example

$$
\begin{array}{rrll}
\| \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2) & & & (0) \\
\Longrightarrow & \overset{d}{1} \| \varphi & \text{decide} & (0, 0) \\
\Longrightarrow & \overset{d}{1}\ 2 \| \varphi & \text{unit propagate} & (0, 1) \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3} \| \varphi & \text{decide} & (0, 1, 0) \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4 \| \varphi & \text{unit propagate} & (0, 1, 1) \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} \| \varphi & \text{decide} & (0, 1, 1, 0) \\
\Longrightarrow & \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5}\ \neg 6 \| \varphi & \text{unit propagate} & (0, 1, 1, 1) \\
\Longrightarrow & \overset{d}{1}\ 2\ \neg 5 \| \varphi & \text{backjump} & (0, 2)
\end{array}
$$

Example

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

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

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

Example

$\| \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$			(0)
$\Longrightarrow$	$1^d \| \varphi$	decide	$(0,0)$
$\Longrightarrow$	$1^d\ 2 \| \varphi$	unit propagate	$(0,1)$
$\Longrightarrow$	$1^d\ 2\ 3^d \| \varphi$	decide	$(0,1,0)$
$\Longrightarrow$	$1^d\ 2\ 3^d\ 4 \| \varphi$	unit propagate	$(0,1,1)$
$\Longrightarrow$	$1^d\ 2\ 3^d\ 4\ 5^d \| \varphi$	decide	$(0,1,1,0)$
$\Longrightarrow$	$1^d\ 2\ 3^d\ 4\ 5^d\ \neg 6 \| \varphi$	unit propagate	$(0,1,1,1)$
$\Longrightarrow$	$1^d\ 2\ \neg 5 \| \varphi$	backjump	$(0,2)$

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

Example

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

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

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

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3} \| \varphi \quad \text{decide} \quad (0,1,0)$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3}\ 4 \| \varphi \quad \text{unit propagate} \quad (0,1,1)$$

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \overset{d}{3}\ 4\ \overset{d}{5} \| \varphi \quad \text{decide} \quad (0,1,1,0)$$

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

$$\Longrightarrow \quad \overset{d}{1}\ 2\ \neg 5 \| \varphi \quad \text{backjump} \quad (0,2)$$

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

Lemma

1. if $\| F \Longrightarrow_{\mathcal{B}}^{*} M \| F'$ then
   - $F = F'$

Lemma

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

Lemma

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

Lemma

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

2. if $\Vert F \Longrightarrow^*_{\mathcal{B}} M_0\, l_1^d\, M_1\, l_2^d\, M_2 \cdots l_k^d\, M_k \Vert F$ with no decision literals in $M_0, \dots, M_k$

   then $F, l_1, \dots, l_i \vDash M_i$ 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 $\parallel F \Longrightarrow_{\mathcal{B}} S_1 \Longrightarrow_{\mathcal{B}} \cdots \Longrightarrow_{\mathcal{B}} S_n \not\Longrightarrow_{\mathcal{B}}$ 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

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

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
   - $F \vDash C$

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 $F \vDash M$

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 $F \vDash \neg C$

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

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. $\| F \Longrightarrow_{\mathcal{B}}^{*} M \| F' \not\Longrightarrow_{\mathcal{B}}$

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. $\parallel F \Longrightarrow_{\mathcal{B}}^{*} M \parallel F' \not\Longrightarrow_{\mathcal{B}}$
   - $F = F'$

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

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 C 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 C such that $M \vDash \neg C$, otherwise backjump or fail is applicable
   - $M \vDash F$

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 C such that $M \vDash \neg C$, otherwise backjump or fail is applicable
   - $M \vDash F$ and thus F is satisfiable

Remarks

- finding backjump clauses is done via conflict graphs

Remarks

- finding backjump clauses is done via conflict graphs
- extension of DPLL: Conflict Driven Clause Learning (CDCL)

Remarks

- finding backjump clauses is done via conflict graphs
- extension of DPLL: Conflict Driven Clause Learning (CDCL)
  - backjump clauses can be learned and are added to formula

Remarks

- finding backjump clauses is done via conflict graphs
- extension of DPLL: Conflict Driven Clause Learning (CDCL)
  - backjump clauses can be learned and are added to formula
  - restart rule erases current list M, but keeps learned formulas

Remarks

- finding backjump clauses is done via conflict graphs
- extension of DPLL: Conflict Driven Clause Learning (CDCL)
  - backjump clauses can be learned and are added to formula
  - restart rule erases current list M, but keeps learned formulas
- restarts do not compromise completeness if number of steps between consecutive restarts strictly increases

Remarks

- finding backjump clauses is done via conflict graphs
- extension of DPLL: Conflict Driven Clause Learning (CDCL)
  - backjump clauses can be learned and are added to formula
  - restart rule erases current list M, but keeps learned formulas
- restarts do not compromise completeness if number of steps between consecutive restarts strictly increases
- modern SAT solvers additionally incorporate
  - heuristics for selecting next decision literal
  - special data structures that allow for efficient unit propagation

Outline

1. Introduction
2. Propositional Logic – Review
3. Tseitin's Transformation
4. DPLL
5. **Further Reading**

Kröning and Strichmann

- Chapter 1
- Chapter 2

Kröning and Strichmann

- Chapter 1
- Chapter 2

Further Reading

- David A. Plaisted and Steven Greenbaum
  A Structure-Preserving Clause Form Translation
  Journal of Symbolic Computation 2(3), pp. 293 – 304, 1986
- Martin Davis and Hilary Putnam
  A Computing Procedure for Quantification Theory
  Journal of the ACM 7(3), pp. 201 – 215, 1960
- Martin Davis, George Logemann, and Donald Loveland
  A Machine Program for Theorem-Proving
  Communications of the ACM 5(7), pp. 394 – 397, 1962

Further Reading (cont'd)

- Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli
  Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T)
  Journal of the ACM 53(6), pp. 937 – 977, 2006
- Moshe Y. Vardi
  Boolean Satisfiability: Theory and Engineering
  Communications of the ACM 57(3), editor's letter, 2014

Important Concepts

- abstract DPLL
- atom
- basic DPLL
- backjump
- backtrack
- bottom
- clause
- complementary literals
- conflict graph
- conjunction
- conjunctive normal form
- decide
- disjunction
- disjunctive normal form
- equisatisfiability
- fail-state
- implication
- literal
- negation
- polarity
- pure literal
- restart
- right-associativity
- satisfiability
- semantic entailment
- semantic equivalence
- Tseitin's transformation
- tautology
- top
- truth table
- truth values
- unit propagation
- validity
- valuation