



Logic

Luca Campa Philipp Dablander Aaron Groß Aart Middeldorp
Alexander Montag Johannes Niederhauser Vera Schmitt

Definitions

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

Lemma

$$E_C[\varphi U \psi] \equiv E[\varphi U (\psi \wedge E_C G T)] \qquad E_C X \varphi \equiv EX(\varphi \wedge E_C G T)$$

Theorem

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

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

Outline

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

Theorem

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

LTL Model Checking

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

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

Theorem

$\mathcal{M}, s \not\models \varphi \iff$ exists **accepting** path in $A_{\neg\varphi} \times \mathcal{M}$ starting from state corresponding to s

Part I: Propositional Logic

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

Part II: Predicate Logic

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

Part III: Model Checking

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

Online Evaluation in Presence

<https://lv-analyse.uibk.ac.at/evasys/public/online/index>



Outline

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

Outline

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

Definition

CTL* formulas consist of

- ▶ **state formulas**, which are evaluated in states:

$$\varphi ::= \perp \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid A[\alpha] \mid E[\alpha]$$

- ▶ **path formulas**, which are evaluated along paths:

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

Examples

$$A[(p U r) \vee (q U r)]$$

$$A[Xp \vee XXp]$$

$$E[GFp]$$

$$A[(p \vee q) U r]$$

$$A[Xp] \vee A[XA[Xp]]$$

$$E[GE[Fp]]$$

Definition

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

$$\mathcal{M}, s \not\models \perp$$

$$\mathcal{M}, s \models \top$$

$$\mathcal{M}, s \models p \iff p \in L(s)$$

$$\mathcal{M}, s \models \neg\varphi \iff \mathcal{M}, s \not\models \varphi$$

$$\mathcal{M}, s \models \varphi \wedge \psi \iff \mathcal{M}, s \models \varphi \text{ and } \mathcal{M}, s \models \psi$$

$$\mathcal{M}, s \models \varphi \vee \psi \iff \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi$$

$$\mathcal{M}, s \models \varphi \rightarrow \psi \iff \mathcal{M}, s \not\models \varphi \text{ or } \mathcal{M}, s \models \psi$$

$$\mathcal{M}, s \models A[\alpha] \iff \forall \text{ paths } \pi = s \rightarrow s_2 \rightarrow \dots \quad \mathcal{M}, \pi \models \alpha$$

$$\mathcal{M}, s \models E[\alpha] \iff \exists \text{ path } \pi = s \rightarrow s_2 \rightarrow \dots \quad \mathcal{M}, \pi \models \alpha$$

Definition

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

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

$$\mathcal{M}, \pi \models \neg\alpha \iff \mathcal{M}, \pi \not\models \alpha$$

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

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

$$\mathcal{M}, \pi \models \alpha \rightarrow \beta \iff \mathcal{M}, \pi \not\models \alpha \text{ or } \mathcal{M}, \pi \models \beta$$

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

$$\mathcal{M}, \pi \models F\alpha \iff \exists i \geq 1 \quad \mathcal{M}, \pi^i \models \alpha$$

$$\mathcal{M}, \pi \models G\alpha \iff \forall i \geq 1 \quad \mathcal{M}, \pi^i \models \alpha$$

$$\mathcal{M}, \pi \models \alpha U \beta \iff \exists i \geq 1 \quad \mathcal{M}, \pi^i \models \beta \text{ and } \forall j < i \quad \mathcal{M}, \pi^j \models \alpha$$

Theorem

satisfaction of CTL* formulas in finite models is **decidable**

Definition

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

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

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

Remarks

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

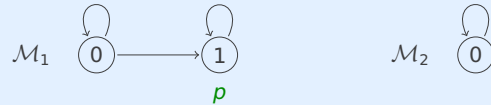
$$\alpha ::= \varphi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha \vee \alpha) \mid (\alpha \rightarrow \alpha) \mid (X\varphi) \mid (F\varphi) \mid (G\varphi) \mid (\varphi U \varphi)$$

Lemma

AG EF p is not expressible in LTL

Proof

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

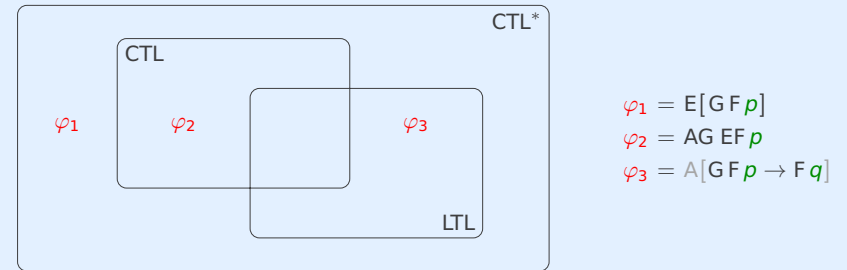


- ▶ $\mathcal{M}_1, 0 \models \text{AG EF } p$
- ▶ $\mathcal{M}_1, 0 \models \text{A}[\varphi]$
- ▶ $\mathcal{M}_2, 0 \not\models \text{AG EF } p$
- ▶ $\mathcal{M}_2, 0 \not\models \text{A}[\varphi]$ because every path from 0 in \mathcal{M}_2 is also path in \mathcal{M}_1 ⚡

Lemma

- ▶ $\text{A}[\text{GF } p \rightarrow \text{F } q]$ is not expressible in CTL
- ▶ $\text{E}[\text{GF } p]$ is expressible neither in CTL nor LTL

Expressive Power



Outline

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

articify with session ID 6893 6178

Question

Which of the following statements are true ?

- A** The CTL formula $\text{AF AG } p \rightarrow \text{AG AF } p$ is valid.
- B** The LTL formula $\text{F } G p$ is expressible in CTL.
- C** The CTL formula $\text{AG AX } p$ is equivalent to the LTL formula $\text{G } X p$.
- D** The CTL* formulas $\text{A}[\text{FA}[\text{G } p]]$ and $\text{A}[\text{F } G p]$ are equivalent.



Outline

1. Summary of Previous Lecture

2. Evaluation

3. CTL*

4. Intermezzo

5. SAT Solving

DPLL Conflict Analysis McGregor Map

6. Sorting Networks

7. Further Reading

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

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

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

Definition (Transition Rules)

- ▶ **unit propagate** $M \parallel F, C \vee \ell \implies M \ell \parallel F, C \vee \ell$
if $M \models \neg C$ and ℓ is undefined in M **unit clause**
- ▶ **pure literal** $M \parallel F \implies M \ell \parallel F$
if ℓ occurs in F and ℓ^c does not occur in F and ℓ is undefined in M
- ▶ **decide** $M \parallel F \implies M \overset{d}{\ell} \parallel F$
if ℓ or ℓ^c occurs in F and ℓ is undefined in M
- ▶ **fail** $M \parallel F, C \implies$ fail-state
if $M \models \neg C$ and M contains no decision literals
- ▶ **backtrack** $M \overset{d}{\ell} N \parallel F, C \implies M \ell^c \parallel F, C$
if $M \overset{d}{\ell} N \models \neg C$ and N contains no decision literals

Outline

1. Summary of Previous Lecture

2. Evaluation

3. CTL*

4. Intermezzo

5. SAT Solving

DPLL Conflict Analysis McGregor Map

6. Sorting Networks

7. Further Reading

Example

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

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

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

Definitions

▶ **backtrack** $M \overset{d}{\ell} N \parallel F, C \Rightarrow M \ell^c \parallel F, C$

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

▶ **backjump** $M \overset{d}{\ell} N \parallel F, C \Rightarrow M \ell' \parallel F, C$

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

▶ $F, C \models C' \vee \ell'$ **backjump clause**

▶ $M \models \neg C'$

▶ ℓ' is undefined in M

▶ ℓ' or ℓ'^c occurs in F or in $M \overset{d}{\ell} N$

Example (cont'd)

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

Definition

basic DPLL \mathcal{B} consists of transition rules

▶ **unit propagate** $M \parallel F, C \vee \ell \Rightarrow M \ell \parallel F, C \vee \ell$

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

▶ **decide** $M \parallel F \Rightarrow M \overset{d}{\ell} \parallel F$

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

▶ **fail** $M \parallel F, C \Rightarrow \text{fail-state}$

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

▶ **backjump** $M \overset{d}{\ell} N \parallel F, C \Rightarrow M \ell' \parallel F, C$

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

▶ $F, C \models C' \vee \ell'$ and $M \models \neg C'$

▶ ℓ' is undefined in M and ℓ' or ℓ'^c occurs in F or in $M \overset{d}{\ell} N$

Theorem

there are no infinite derivations $\parallel F \Rightarrow_B S_1 \Rightarrow_B S_2 \Rightarrow_B \dots$

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
- ▶ measure is **bounded** by $(n+1)$ -tuple (n, \dots, n) where n is total number of atoms

Example

$\parallel \varphi = (\neg 1 \vee 2) \wedge (\neg 3 \vee 4) \wedge (\neg 5 \vee \neg 6) \wedge (6 \vee \neg 5 \vee \neg 2)$	(0)
$\Rightarrow \quad \quad \quad \overset{d}{1} \parallel \varphi$	decide (0, 0)
$\Rightarrow \quad \quad \quad \overset{d}{1} \overset{d}{2} \parallel \varphi$	unit propagate (0, 1)
$\Rightarrow \quad \quad \quad \overset{d}{1} \overset{d}{2} \overset{d}{3} \parallel \varphi$	decide (0, 1, 0)
$\Rightarrow \quad \quad \quad \overset{d}{1} \overset{d}{2} \overset{d}{3} \overset{d}{4} \parallel \varphi$	unit propagate (0, 1, 1)
$\Rightarrow \quad \quad \quad \overset{d}{1} \overset{d}{2} \overset{d}{3} \overset{d}{4} \overset{d}{5} \parallel \varphi$	decide (0, 1, 1, 0)
$\Rightarrow \quad \overset{d}{1} \overset{d}{2} \overset{d}{3} \overset{d}{4} \overset{d}{5} \neg 6 \parallel \varphi$	unit propagate (0, 1, 1, 1)
$\Rightarrow \quad \quad \quad \overset{d}{1} \overset{d}{2} \neg 5 \parallel \varphi$	backjump (0, 2)

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

Lemma

- if $\parallel F \Rightarrow_B^* M \parallel F'$ then
 - ▶ $F = F'$
 - ▶ M does not contain complementary literals
 - ▶ M consists of distinct literals
- if $\parallel F \Rightarrow_B^* M_0 \overset{d}{l_1} M_1 \overset{d}{l_2} M_2 \dots \overset{d}{l_k} M_k \parallel F$ with no decision literals in M_0, \dots, M_k then $F, l_1, \dots, l_i \models M_i$ for all $0 \leq i \leq k$

Theorem

- if $\parallel F \Rightarrow_B S_1 \Rightarrow_B \dots \Rightarrow_B S_n \not\Rightarrow_B$ then
- $S_n = \text{fail-state}$ if and only if F is unsatisfiable
 - $S_n = M \parallel F'$ only if F is satisfiable and $M \models F$

Proof

- (only if) $\parallel F \Rightarrow_B^* M \parallel F \Rightarrow_{\text{fail}}$ fail-state
 - ▶ M contains no decision literals and $M \models \neg C$ for some C in F
 - ▶ $F \models C$ and $F \models M$ and thus $F \models \neg C$ and thus F is unsatisfiable
- $\parallel F \Rightarrow_B^* M \parallel F' \not\Rightarrow_B$
 - ▶ $F = F'$ and all literals in F are defined in M , otherwise **decide** is applicable
 - ▶ F contains no clause C such that $M \models \neg C$, otherwise **backjump** or **fail** is applicable
 - ▶ $M \models F$ and thus F is satisfiable

Lemma

backjump can simulate backtrack

Proof

- ▶ suppose $\| F \Rightarrow_{\mathcal{B}}^* M^d \ell N \| F \Rightarrow_{\text{backtrack}} M \ell^c \| F$
- ▶ $M^d \ell N \models \neg C$ for some C in F and N contains no decision literals
- ▶ write $M = M_0 \ell_1^d M_1 \ell_2^d M_2 \dots \ell_k^d M_k$ with all decision literals displayed
- ▶ $\ell_1^c \vee \dots \vee \ell_k^c \vee \ell^c$ is backjump clause:
 - ▶ $F, \ell_1, \dots, \ell_k, \ell \models \neg C \implies F, \ell_1, \dots, \ell_k, \ell$ is unsatisfiable $\implies F \models \ell_1^c \vee \dots \vee \ell_k^c \vee \ell^c$
 - ▶ $M \models \ell_1 \wedge \dots \wedge \ell_k$ and ℓ^c is undefined in M
- ▶ $M^d \ell N \| F \Rightarrow_{\text{backjump}} M \ell^c \| F$

Terminology

non-chronological backtracking or conflict-driven backtracking

Question

how to find good backjump clauses ?

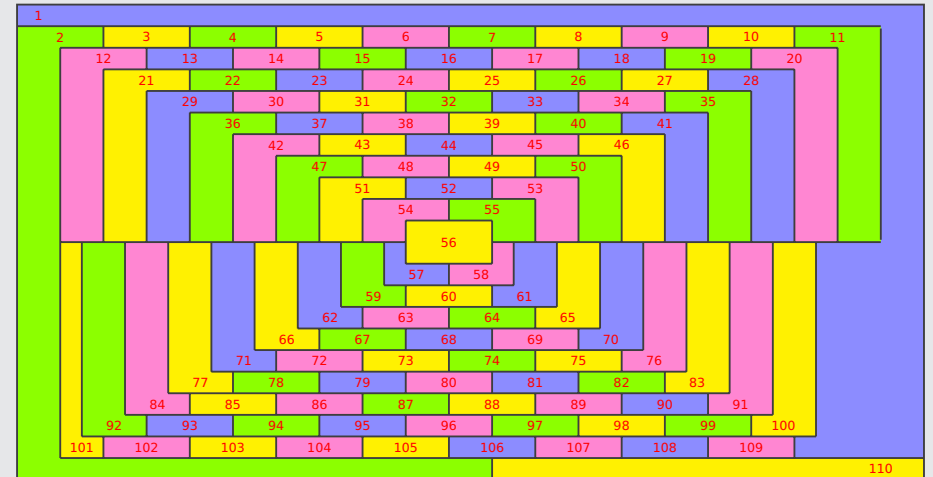
Answer

use conflict graph (lecture 13)

Outline

1. Summary of Previous Lecture
2. Evaluation
3. CTL*
4. Intermezzo
5. SAT Solving
 - DPLL
 - Conflict Analysis
 - McGregor Map
6. Sorting Networks
7. Further Reading

Example (McGregor map)



Background

- ▶ are four colors sufficient to color any planar map ? (four color conjecture, Guthrie 1852)
- ▶ every map can be colored using no more than five colors (Heywood 1890)
- ▶ McGregor map is presupposed counterexample to four color conjecture (Gardner 1975)
- ▶ four colors are sufficient to color any planar map (Appel and Haken 1977)
- ▶ formalized proof of four color theorem in proof assistant Coq (Gonthier 2005)

Example (McGregor map, cont'd)

- ▶ use SAT to find a coloring for McGregor map using four colors
- ▶ atoms x_{rc} with $r \in \{1, \dots, 110\}$ denoting region and $c \in \{1, \dots, 4\}$ denoting color

Example (McGregor map, cont'd)

- ▶ two types of constraints:

- ① every region receives exactly one color:

$$(x_{r1} \vee x_{r2} \vee x_{r3} \vee x_{r4}) \wedge (\neg x_{r1} \vee \neg x_{r2}) \wedge (\neg x_{r1} \vee \neg x_{r3}) \wedge (\neg x_{r1} \vee \neg x_{r4}) \\ \wedge (\neg x_{r2} \vee \neg x_{r3}) \wedge (\neg x_{r2} \vee \neg x_{r4}) \wedge (\neg x_{r3} \vee \neg x_{r4})$$

for all $r \in \{1, \dots, 110\}$

- ② neighbouring regions receive different colors:

$$(\neg x_{r1} \vee \neg x_{s1}) \wedge (\neg x_{r2} \vee \neg x_{s2}) \wedge (\neg x_{r3} \vee \neg x_{s3}) \wedge (\neg x_{r4} \vee \neg x_{s4})$$

for all $(r, s) \in \{(1, 2), (1, 3), \dots, (109, 110)\}$

- ▶ for DIMACS format atoms x_{rc} are simply encoded as number rc
- ▶ resulting DIMACS input consists of 2058 clauses and is easily solved by SAT solver

- ▶ 1 =  2 =  3 =  4 = 

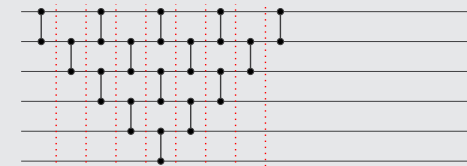
Outline

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

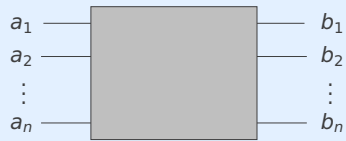
Sorting Network



Example

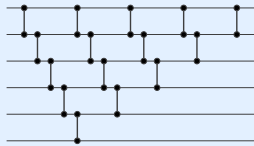


- ▶ size (= number of comparators): 15
- ▶ depth: 9



Definition

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



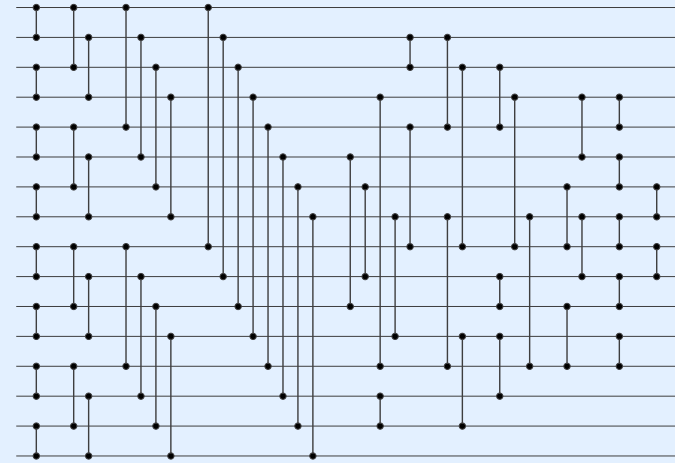
Questions

- ① how to check that comparator network is sorting network ?
- ② how to find optimal (with respect to size or depth) sorting networks ?

Answers

- ① testing all $n!$ permutations of $1, \dots, n$ for network with n wires suffices
- ② very difficult problem ...

Sorting Network ?



Outline

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

Huth and Ryan

- ▶ Section 3.5

DPLL

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

Sorting Networks

- ▶ Wikipedia [accessed December 28, 2024]
- ▶ Section 5.3.4 of The Art of Computer Programming
Donald Knuth

Important Concepts

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

homework for June 18