The Complexity of Generalized Satisfiability for Linear Temporal Logic

Michael Bauland[1] Thomas Schneider[2] Henning Schnoor[1]
Ilka Schnoor[1] Heribert Vollmer[1]

[1]Institut für Theoretische Informatik
Leibniz University, Hannover, Germany

[2]Institut für Informatik
Friedrich Schiller University, Jena, Germany

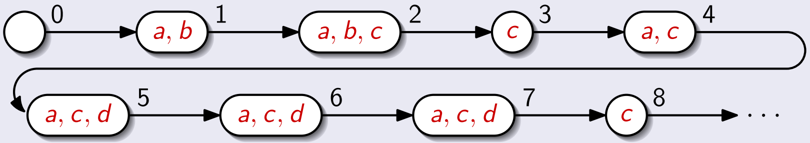
FoSSaCS 2007
26–28 March

And now for . . .

What is Linear Temporal Logic?

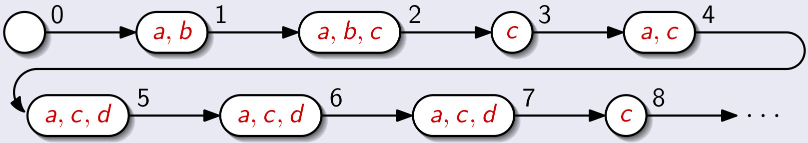
LTL = propositional logic plus temporal operators;
speaks about linear structures, for example:

The structure Σ



Structures and formulae

The structure Σ

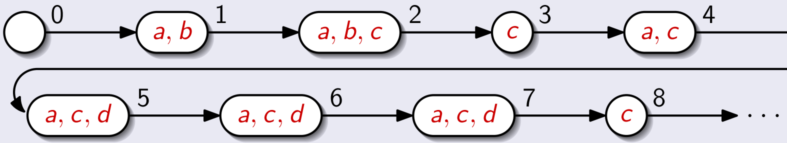


The following kinds of statements can be formulated in LTL.

- now $\qquad \Sigma, 2 \models (a \wedge \neg b) \vee c$
- at some time in the Future $\qquad \Sigma, 0 \models \mathsf{F}d$
- always Going to $\qquad \Sigma, 3 \models \mathsf{G}\neg b$
- neXt time $\qquad \Sigma, 1 \models \mathsf{X}(a \longrightarrow b)$
- Until $\qquad \Sigma, 5 \models c\mathsf{U}(\neg a)$
- Since $\qquad \Sigma, 2 \models b\mathsf{S}\neg b$
- $\Sigma, 0 \models \mathsf{F}(c \wedge \neg b) \wedge \mathsf{G}\big[(c \wedge \neg a) \longrightarrow [\mathsf{X}(a \wedge \mathsf{X}d) \wedge (d \longrightarrow a \wedge c)\mathsf{U}c]\big]$

Structures and formulae

The structure Σ

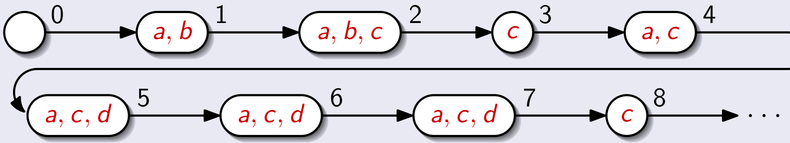


The following kinds of statements can be formulated in LTL.

- now $\Sigma, 2 \models (a \wedge \neg b) \vee c$
- at some time in the Future $\Sigma, 0 \models \mathsf{F} d$
- always Going to $\Sigma, 3 \models \mathsf{G} \neg b$
- neXt time $\Sigma, 1 \models \mathsf{X}(a \longrightarrow b)$
- Until $\Sigma, 5 \models c \mathsf{U} (\neg a)$
- Since $\Sigma, 2 \models b \mathsf{S} \neg b$
- $\Sigma, 0 \models \mathsf{F}(c \wedge \neg b) \wedge \mathsf{G}\big[(c \wedge \neg a) \longrightarrow [\mathsf{X}(a \wedge \mathsf{X} d) \wedge (d \longrightarrow a \wedge c) \mathsf{U} c]\big]$

Structures and formulae

The structure Σ



The following kinds of statements can be formulated in LTL.

- now — $\Sigma, 2 \models (a \wedge \neg b) \vee c$
- at some time in the Future — $\Sigma, 0 \models \mathsf{F}d$
- always Going to — $\Sigma, 3 \models \mathsf{G}\neg b$
- neXt time — $\Sigma, 1 \models \mathsf{X}(a \longrightarrow b)$
- Until — $\Sigma, 5 \models c\mathsf{U}(\neg a)$
- Since — $\Sigma, 2 \models b\mathsf{S}\neg b$
- $\Sigma, 0 \models \mathsf{F}(c \wedge \neg b) \wedge \mathsf{G}\big[(c \wedge \neg a) \longrightarrow [\mathsf{X}(a \wedge \mathsf{X}d) \wedge (d \longrightarrow a \wedge c)\mathsf{U}c]\big]$

The satisfiability problem for LTL

Definition

1. A formula φ is *satisfiable* iff
there is a structure Σ and an $i \in \mathbb{N}$ such that $\Sigma, i \models \varphi$.

Let $\mathcal{T} \subseteq \{\mathsf{F}, \mathsf{G}, \mathsf{X}, \mathsf{U}, \mathsf{S}\}$.

2. $\mathrm{LTL}(\mathcal{T})$ = set of all LTL-formulae with operators from $\mathcal{T}$.
3. $\mathrm{SAT}(\mathcal{T}) = \{\varphi \in \mathrm{LTL}(\mathcal{T}) \mid \varphi \text{ is satisfiable}\}$.

The satisfiability problem for LTL

Definition

1. A formula φ is *satisfiable* iff there is a structure Σ and an $i \in \mathbb{N}$ such that $\Sigma, i \models \varphi$.

Let $\mathcal{T} \subseteq \{\mathsf{F}, \mathsf{G}, \mathsf{X}, \mathsf{U}, \mathsf{S}\}$.

2. $\mathrm{LTL}(\mathcal{T})$ = set of all LTL-formulae with operators from $\mathcal{T}$.
3. $\mathrm{SAT}(\mathcal{T}) = \{\varphi \in \mathrm{LTL}(\mathcal{T}) \mid \varphi \text{ is satisfiable}\}$.

The complexity of LTL-satisfiability

Theorem (Sistla, Clarke 1985)

1. SAT({F}), SAT({G}), *and* SAT({F, G}) *are* NP*-complete.*
2. SAT({F, X}), SAT({G, X}), SAT({U}), *and* SAT({F, G, U, S, X}) *are* PSPACE*-complete.*

This is an almost complete classification of the complexity of LTL-satisfiability with respect to many combinations of temporal operators.

The complexity of LTL-satisfiability

Theorem (Sistla, Clarke 1985)

1. $\mathrm{SAT}(\{\mathrm{F}\})$, $\mathrm{SAT}(\{\mathrm{G}\})$, *and* $\mathrm{SAT}(\{\mathrm{F},\mathrm{G}\})$ *are* NP*-complete.*
2. $\mathrm{SAT}(\{\mathrm{F},\mathrm{X}\})$, $\mathrm{SAT}(\{\mathrm{G},\mathrm{X}\})$, $\mathrm{SAT}(\{\mathrm{U}\})$, *and* $\mathrm{SAT}(\{\mathrm{F},\mathrm{G},\mathrm{U},\mathrm{S},\mathrm{X}\})$ *are* PSPACE*-complete.*

This is an almost complete classification of the complexity of LTL-satisfiability with respect to many combinations of temporal operators.

And now for . . .

Generalized LTL satisfiability

- NP- and PSPACE-completeness are very high complexities.
- Restricting the set of *propositional* operators shows promise for "relief".

  (example: only monotonic formulae)

- $\left.\begin{array}{l}\text{set } \mathcal{T} \text{ of temporal operators} \\ \text{set } \mathcal{B} \text{ of Boolean operators}\end{array}\right\} \rightsquigarrow \mathrm{SAT}(\mathcal{T}, \mathcal{B})$

Example

Sistla and Clarke have examined the problems $\mathrm{SAT}(\mathcal{T}, \{\wedge, \neg\})$ for several $\mathcal{T} \subseteq \{\mathrm{F}, \mathrm{G}, \mathrm{X}, \mathrm{U}, \mathrm{S}\}$.

Generalized LTL satisfiability

- NP- and PSPACE-completeness are very high complexities.
- Restricting the set of *propositional* operators shows promise for "relief".

  (example: only monotonic formulae)
- $\left.\begin{array}{l}\text{set } \mathcal{T} \text{ of temporal operators}\\ \text{set } \mathcal{B} \text{ of Boolean operators}\end{array}\right\} \rightsquigarrow \mathrm{SAT}(\mathcal{T}, \mathcal{B})$

Example

Sistla and Clarke have examined the problems $\mathrm{SAT}(\mathcal{T}, \{\wedge, \neg\})$ for several $\mathcal{T} \subseteq \{\mathsf{F}, \mathsf{G}, \mathsf{X}, \mathsf{U}, \mathsf{S}\}$.

Generalized LTL satisfiability

- NP- and PSPACE-completeness are very high complexities.
- Restricting the set of *propositional* operators shows promise for "relief".

  (example: only monotonic formulae)
- $\left.\begin{array}{l}\text{set } \mathcal{T} \text{ of temporal operators} \\ \text{set } \mathcal{B} \text{ of Boolean operators}\end{array}\right\} \rightsquigarrow \mathrm{SAT}(\mathcal{T}, \mathcal{B})$

Example

Sistla and Clarke have examined the problems $\mathrm{SAT}(\mathcal{T}, \{\wedge, \neg\})$ for several $\mathcal{T} \subseteq \{\mathrm{F}, \mathrm{G}, \mathrm{X}, \mathrm{U}, \mathrm{S}\}$.

Generalized LTL satisfiability

Goal

- Classification of the complexity of SAT$(\mathcal{T}, \mathcal{B})$ for *all* possible combinations of $\mathcal{T}$ and $\mathcal{B}$.
- Exhibit combinations leading to NP-hard or PSPACE-hard fragments.

Post's lattice

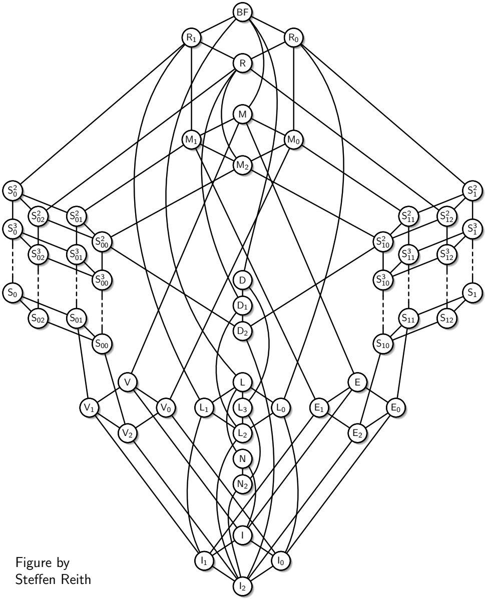
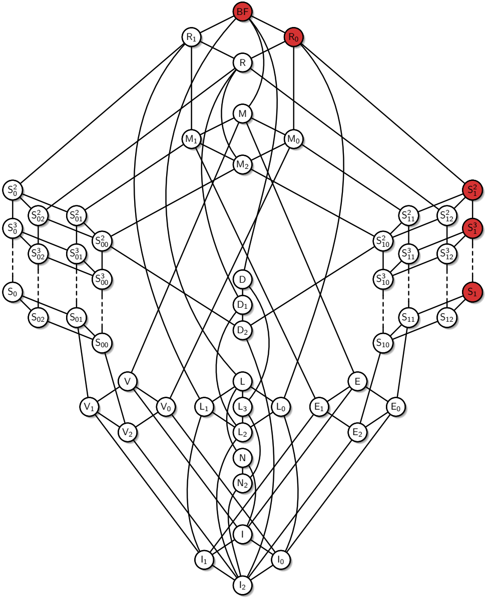


Figure by
Steffen Reith

Post's lattice

established 1941
by Emil Post

Generalized satisfiability in the literature



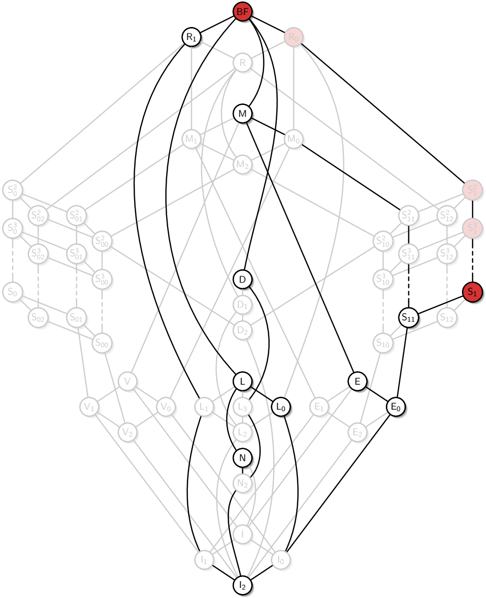
Theorem (H. R. Lewis 1979)

$\mathrm{SAT}(\emptyset, \mathcal{B})$ is:

- ● NP-complete
- ○ in P

$$(S_1:\quad \neg(x \longrightarrow y))$$

Generalized satisfiability in the literature



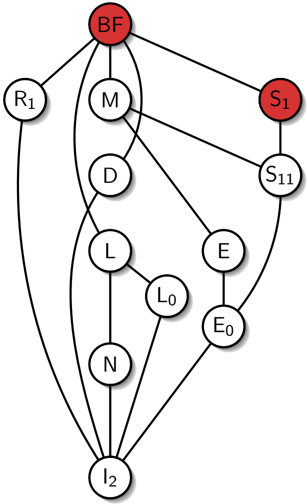
Theorem (H. R. Lewis 1979)

$\mathrm{SAT}(\emptyset, \mathcal{B})$ is:

- NP-complete (red)
- in P (white)

$(S_1: \ \neg(x \longrightarrow y))$

Generalized satisfiability in the literature



Theorem (H. R. Lewis 1979)

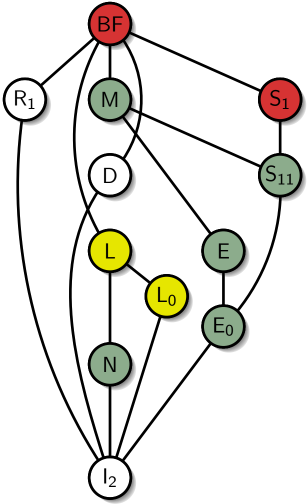
$\mathrm{SAT}(\emptyset, \mathcal{B})$ is:

- NP-complete
- in P

$(S_1\colon\ \neg(x \longrightarrow y))$

And now for . . .

An overview of our results



Theorem 1

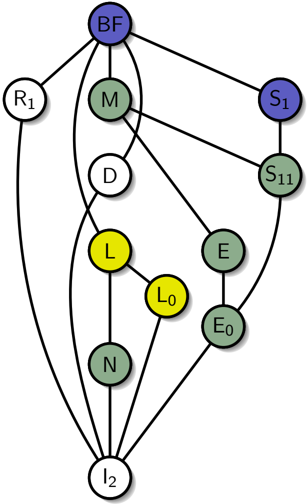
If $\mathcal{T} \subseteq \{F, G\}$ or $\mathcal{T} = \{X\}$, then $SAT(\mathcal{T}, \mathcal{B})$ is:

- (red) NP-complete
- (yellow) ?
- (green) in P
- (white) trivial

S_1: $\neg(x \longrightarrow y)$

L_0, L: $x \oplus y$, 0, (1)

An overview of our results



Theorem 2

For *all remaining* $\mathcal{T}$,

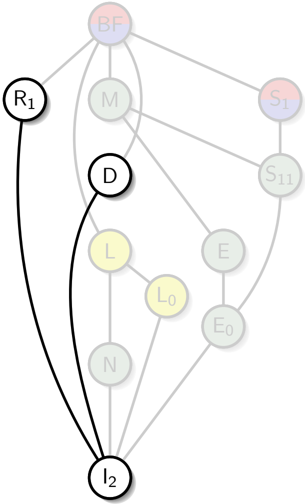
SAT$(\mathcal{T}, \mathcal{B})$ is:

- PSPACE-complete
- ?
- in P
- trivial

S_1: $\neg(x \longrightarrow y)$

L_0, L: $x \oplus y$, 0, (1)

Our results in detail: trivial cases



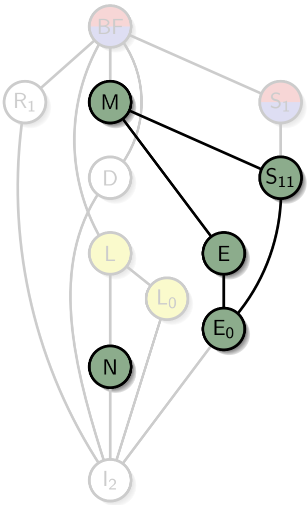
Lemma

Every LTL($\mathcal{T}$, R_1)-formula and every LTL($\mathcal{T}$, D)-formula is satisfiable.

R_1: $f(1,\ldots,1)=1$

D: $f(a_1,\ldots,a_n) = \overline{f(\overline{a_1},\ldots,\overline{a_n})}$

Our results in detail: polynomial-time solvable cases



Lemma

- $\mathrm{SAT}(\mathcal{T}, \mathrm{M}) \in P$
- $\mathrm{SAT}(\mathcal{T}, \mathrm{N}) \in P$

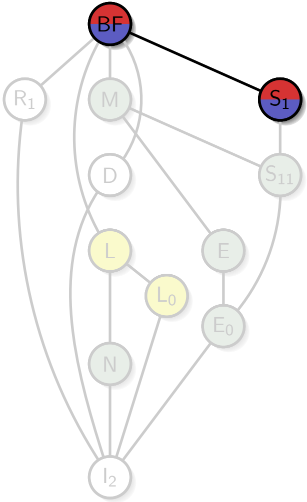
M: monotonic functions ($\wedge$, $\vee$, 0, 1)

N: $\neg$, 0, 1

example:

$\neg j \mathrm{U}(\mathrm{X}(\ i \mathrm{S}(\neg(f \mathrm{U} n))))$

Our results in detail: hard cases



Lemma

If $\mathcal{B} \supseteq S_1$,

then $SAT(\mathcal{T}, \mathcal{B})$ is complete for NP or PSPACE.

$$S_1: \quad f(x, y) = \neg(x \rightarrow y)$$

Our results in detail: hard cases

- Upper bounds follow from those for SAT($\mathcal{T}$, BF).
- NP-hardness follows from NP-hardness of SAT($\emptyset$, S_1).
- Remains to show: PSPACE-hardness of

(1) SAT($\{F, X\}$, S_1)

(2) SAT($\{G, X\}$, S_1)

(3) SAT($\{U\}$, S_1)

(4) SAT($\{S\}$, S_1)

Our results in detail: hard cases

- Upper bounds follow from those for SAT($\mathcal{T}$, BF).
- NP-hardness follows from NP-hardness of SAT($\emptyset$, S_1).
- Remains to show: PSPACE-hardness of

(1) SAT($\{F, X\}$, S_1)

(2) SAT($\{G, X\}$, S_1)

(3) SAT($\{U\}$, S_1)

(4) SAT($\{S\}$, S_1)

Our results in detail: hard cases

(4) First show PSPACE-hardness of SAT($\{S\}$, BF).

- QBF $\leq_m^{\log}$ SAT($\{S\}$, BF)
- Consider QBF-instance $\varphi = Q_1 x_1 \ldots Q_n x_n \, \psi(x_1, \ldots, x_n)$, $Q_i \in \{\forall, \exists\}$.
- Let $x_{p_1}, \ldots, x_{p_k}$ be all $\forall$-quantified variables.
- Construct LTL($\{S\}$, BF)-formula φ' that enforces that
  - the state satisfying φ' has $3 \cdot 2^k$ predecessor states:

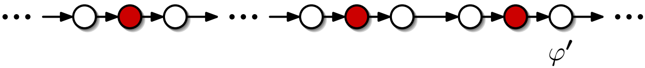
  - each combination of the x_{p_i} appears in the red states;
  - the truth values of the remaining x_j in the red states depend only of the values of $x_0, \ldots, x_{j-1}$;
  - ψ is true in each red state.

(SAT($\{S\}$, S_1) requires a modification of this construction.)

Our results in detail: hard cases

(4) First show PSPACE-hardness of SAT({S}, BF).

- QBF $\leq_m^{\log}$ SAT({S}, BF)
- Consider QBF-instance $\varphi = Q_1 x_1 \dots Q_n x_n \, \psi(x_1, \dots, x_n)$, $Q_i \in \{\forall, \exists\}$.
- Let $x_{p_1}, \dots, x_{p_k}$ be all $\forall$-quantified variables.
- Construct LTL({S}, BF)-formula φ' that enforces that
  - the state satisfying φ' has $3 \cdot 2^k$ predecessor states:



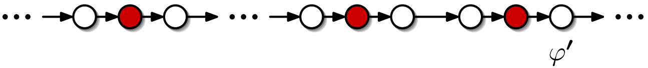
  - each combination of the x_{p_i} appears in the red states;
  - the truth values of the remaining x_j in the red states depend only of the values of $x_0, \dots, x_{j-1}$;
  - ψ is true in each red state.

(SAT({S}, S_1) requires a modification of this construction.)

Our results in detail: hard cases

(4) First show PSPACE-hardness of SAT({S}, BF).

- QBF $\leq_m^{\log}$ SAT({S}, BF)
- Consider QBF-instance $\varphi = Q_1 x_1 \ldots Q_n x_n \, \psi(x_1, \ldots, x_n)$, $Q_i \in \{\forall, \exists\}$.
- Let $x_{p_1}, \ldots, x_{p_k}$ be all $\forall$-quantified variables.
- Construct LTL({S}, BF)-formula φ' that enforces that
  - the state satisfying φ' has $3 \cdot 2^k$ predecessor states:



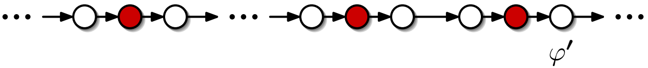
  - each combination of the x_{p_i} appears in the red states;
  - the truth values of the remaining x_j in the red states depend only of the values of $x_0, \ldots, x_{j-1}$;
  - ψ is true in each red state.

(SAT({S}, S_1) requires a modification of this construction.)

Our results in detail: hard cases

(4) First show PSPACE-hardness of SAT({S}, BF).

- QBF $\leq_{\mathrm{m}}^{\log}$ SAT({S}, BF)
- Consider QBF-instance $\varphi = \mathrm{Q}_1 x_1 \ldots \mathrm{Q}_n x_n \, \psi(x_1, \ldots, x_n)$, $\mathrm{Q}_i \in \{\forall, \exists\}$.
- Let $x_{p_1}, \ldots, x_{p_k}$ be all $\forall$-quantified variables.
- Construct LTL({S}, BF)-formula φ' that enforces that
  - the state satisfying φ' has $3 \cdot 2^k$ predecessor states:

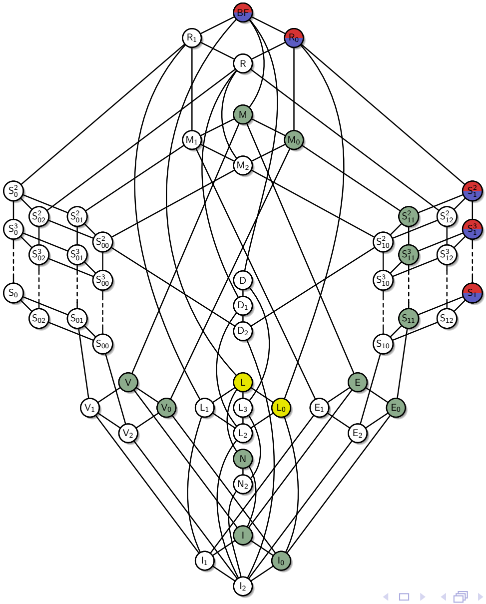


  - each combination of the x_{p_i} appears in the red states;
  - the truth values of the remaining x_j in the red states depend only of the values of $x_0, \ldots, x_{j-1}$;
  - ψ is true in each red state.

(SAT({S}, S_1) requires a modification of this construction.)

And now for . . .

Conclusion


BF
R_1
R_0
R
M
M_1
M_0
M_2
S_0^2
S_{02}^2
S_{01}^2
S_{00}^2
S_0^3
S_{02}^3
S_{01}^3
S_{00}^3
S_0
S_{02}
S_{01}
S_{00}
S_1^2
S_{11}^2
S_{12}^2
S_{10}^2
S_1^3
S_{11}^3
S_{12}^3
S_{10}^3
S_1
S_{11}
S_{12}
S_{10}
D
D_1
D_2
V
V_1
V_0
V_2
L
L_1
L_3
L_0
L_2
E
E_1
E_0
E_2
N
N_2
I
I_1
I_0
I_2

Conclusion

- achieved: classification of the complexity of SAT($\mathcal{T}, \mathcal{B}$) for almost all combinations of $\mathcal{T}$ and $\mathcal{B}$
- open cases: L_0, L ("xor" function)
- noticeable:
  Hard cases are those containing $\neg(x \longrightarrow y)$.
  (These are the same as in Lewis's classification of SAT($\emptyset, \mathcal{B}$).)

Conclusion

- achieved: classification of the complexity of $\mathrm{SAT}(\mathcal{T}, \mathcal{B})$ for almost all combinations of $\mathcal{T}$ and $\mathcal{B}$
- open cases: L_0, L ("xor" function)
- noticeable:
  Hard cases are those containing $\neg(x \longrightarrow y)$.
  (These are the same as in Lewis's classification of $\mathrm{SAT}(\emptyset, \mathcal{B})$.)

Perspectives

- Solve the "xor" cases!
- Further classify the P-cases.
- Similarly classify the model-checking problem:

> Given a formula φ, a directed graph, and a state z,
> is there a path starting at z that satisfies φ?

- Similarly classify decision problems for other temporal logics:
  - CTL, CTL* (Computation Tree Logic)
  - hybrid temporal logics

Thank you!

Perspectives

- Solve the "xor" cases!
- Further classify the P-cases.
- Similarly classify the model-checking problem:

> Given a formula φ, a directed graph, and a state z,
> is there a path starting at z that satisfies φ?

- Similarly classify decision problems for other temporal logics:
  - CTL, CTL* (Computation Tree Logic)
  - hybrid temporal logics

Thank you!

Perspectives

- Solve the "xor" cases!
- Further classify the P-cases.
- Similarly classify the model-checking problem:

> Given a formula φ, a directed graph, and a state z,
> is there a path starting at z that satisfies φ?

- Similarly classify decision problems for other temporal logics:
  - CTL, CTL* (Computation Tree Logic)
  - hybrid temporal logics

Thank you!

Perspectives

- Solve the "xor" cases!
- Further classify the P-cases.
- Similarly classify the model-checking problem:

> Given a formula φ, a directed graph, and a state z,
> is there a path starting at z that satisfies φ?

- Similarly classify decision problems for other temporal logics:
  - CTL, CTL* (Computation Tree Logic)
  - hybrid temporal logics

Thank you!

References (Post's lattice)

E. Post.
The two-valued iterative systems of mathematical logic.
Annals of Mathematical Studies, 5: 1–122, 1941.

E. Böhler, N. Creignou, S. Reith, H. Vollmer.
Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory.
In: *ACM SIGACT-Newsletter*, 34 (4): 38–52, 2003.

References (logic)

H. R. Lewis.
Satisfiability problems for propositional calculi.
Mathematical Systems Theory, 13: 45–53, 1979.

A. P. Sistla, E. M. Clarke.
The complexity of propositional linear temporal logics.
Journal of the ACM, 32 (3): 733–749, 1985.

M. Bauland, E. Hemaspaandra, H. Schnoor, I. Schnoor.
Generalized modal satisfiability.
In: *STACS*, *LNCS 3884*, pg. 500–511, Springer, 2006.

My e-mail address is . . .

`schneider@cs.uni-jena.de`