Complexity and Expressivity of Propositional Logics with Team Semantics

ESSLLI 2024 course

Arne Meier[1] Jonni Virtema[2]

[1] Leibniz Universität Hannover, Germany
[2] University of Sheffield, UK

Version of 7th August 2024

Lecture 4: Complexity of propositional dependence logic and beyond

Literature: [Vir17; Han+18]

Canonical complete problems: SAT and QBF

SAT [Coo71]		QBF (Stockmeyer and Meyer, 1973)	
Input:	Boolean formula θ	Input:	Quantified Boolean formula $\phi := Q_1 p_1 \ldots Q_n p_n \theta$
Question:	Is θ satisfiable?	Question:	Is ϕ true?
Complete for:	NP (Thm. 5)	Complete for:	PSPACE

W.l.o.g. θ in 3CNF

$$\theta = (p_1 \vee p_2 \vee \neg p_3) \wedge (\neg p_2 \vee \neg p_4 \vee p_5) \wedge \ldots$$

$\forall p_0 \exists p_1 \forall p_2 \exists p_3 \exists p_4 \forall p_5 (\theta)$

Model Checking for Dependence Logic

Theorem 41 ([Loh12, Theorem 4.13])

$\mathrm{PL}[\mathrm{dep}]\text{-}\mathrm{MC}$ *is* NP-*complete.*

Proof ideas:

Membership: Use nondeterminism for splitjunctions.

Hardness: reduce from 3SAT.

Membership in NP

FORMULA $\varphi \in PL[dep]$ TEAM T

$T \models \varphi$?

$O(n)$

$O(nL)$

$\varphi = p$ $\qquad T \models p \iff \forall s \in T \quad s(p) = 1$

$\neg p \qquad s(p) = 0$

$\varphi = dep(p_1 \ldots, p_n, q) \qquad T \models dep(p_1, \ldots, p_n, q)$

GO THROUGH ALL PAIRS $(s, s') \in T \times T$

if $s(p_1, \ldots, p_n) = s'(p_1, \ldots, p_n)$ then $s(q) = s'(q)$

Membership in NP

$T \models \varphi \wedge \psi$? $\quad T \models \varphi \quad T \models \psi$

$T \models \varphi \vee \psi \iff \exists T_1 \, \exists T_2$ s.t. $T_1 \cup T_2 = T$

$T_1 \models \varphi$ and $T_2 \models \psi$

USE NONDETERMINISM TO GUESS

$T_1, T_2 \subseteq T$ s.t. $T_1 \cup T_2 = T$

AND $T_1 \models \varphi$ and $T_2 \models \psi$

NP Lower Bound

We give a reduction from SAT.

3-CNF formula:

$$\bigwedge_{i=1}^{m} (l_{i,1} \vee l_{i,2} \vee l_{i,3})$$

over variables $\{x_1, \ldots, x_n\}$

$l_{i,j}$

x

$\neg x$

Define a team

$T = \{s_1, \ldots, s_m\}$ over variables

$\{v_1, \ldots, v_n, p_1, \ldots, p_n\}$

NP Lower Bound

$S_i(v_j) = 1$ iff x_j OCCURS IN THE i:th CLAUSE WITH PARITY $S_i(p_j)$

FORMALLY:

$$S_i(v_j) = \begin{cases} 1 & \text{if } x_j \text{ OCCURS IN THE } i\text{th CLAUSE} \\ 0 & \text{OTHERWISE} \end{cases}$$

$$S_i(p_j) = \begin{cases} 1 & \text{if } x_j \text{ OCCURS POSITIVELY IN THE } i\text{th CLAUSE} \\ 0 & \text{OTHERWISE} \end{cases}$$

NP Lower Bound

DEF $\varphi := \bigvee_{j=1}^{n} \left(v_j \wedge \mathrm{dep}(p_j) \right)$

CLAIM: φ IS SATISFIABLE

iff $T \models \varphi$

Canonical complete problems: DQBF

DQBF (Peterson, Reif, Azhar, 2001)	
Input:	Dependency Quantified Boolean formula $\phi := \forall p_1 \dots \forall p_m \exists q_1 \dots \exists q_n \theta$ and constraints $\vec{c}_1, \dots, \vec{c}_n$
Question:	Is ϕ true?
Complete for:	NEXPTIME

- The constraint $\vec{c}_i$ is a tuple of the universally quantified variables of which the existentially quantified variable q_i may depend on.

Canonical complete problems: DQBF

DQBF (Peterson, Reif, Azhar, 2001)	
Input:	Dependency Quantified Boolean formula $\phi := \forall p_1 \ldots \forall p_m \exists q_1 \ldots \exists q_n \theta$ and constraints $\vec{c}_1, \ldots, \vec{c}_n$
Question:	Is ϕ true?
Complete for:	NEXPTIME

- The constraint $\vec{c}_i$ is a tuple of the universally quantified variables of which the existentially quantified variable q_i may depend on.
- A DQBF formula $\forall p_1 \ldots \forall p_m \exists q_1 \ldots \exists q_n \theta$ with constraints $\vec{c}_1, \ldots, \vec{c}_n$ is true, if the the following formula with Boolean function quantification

$$\exists f_1 \ldots f_n \forall p_1 \ldots \forall p_m \theta(f_1(\vec{c}_1)/q_1, \ldots f_n(\vec{c}_n)/q_n)$$

is true. Note that f_i is a Boolean function (Skolem function) which is used to interpret q_i given the values of the variables in $\vec{c}_i$.

Canonical complete problems: DQBF

DQBF (Peterson, Reif, Azhar, 2001)	
Input:	Dependency Quantified Boolean formula $\phi := \forall p_1 \dots \forall p_m \exists q_1 \dots \exists q_n \theta$ and constraints $\vec{c}_1, \dots, \vec{c}_n$
Question:	Is ϕ true?

Complete for: NEXPTIME

- The constraint $\vec{c}_i$ is a tuple of the universally quantified variables of which the existentially quantified variable q_i may depend on.
- A DQBF formula $\forall p_1 \dots \forall p_m \exists q_1 \dots \exists q_n \theta$ with constraints $\vec{c}_1, \dots, \vec{c}_n$ is true, if the the following formula with Boolean function quantification

$$\underline{\exists f_1 \dots f_n} \forall p_1 \dots \forall p_m \theta(f_1(\vec{c}_1)/q_1, \dots f_n(\vec{c}_n)/q_n)$$

  is true. Note that f_i is a Boolean function (Skolem function) which is used to interpret q_i given the values of the variables in $\vec{c}_i$.
- Note how close the above is to $\mathrm{dep}(\vec{c}_1, q_1) \wedge \dots \wedge \mathrm{dep}(\vec{c}_n, q_n) \wedge \theta$!

The validity problem for PD is in NEXPTIME

If $D \subseteq \mathrm{PROP}$, we denote by 2^D the set of all assignments $s\colon D \to \{0,1\}$.

Lemma 42

A $\mathrm{PL}[\mathrm{dep}]$*-formula* φ *with proposition symbols in D is valid iff* $2^D \models \varphi$*.*

Proof.

Left-to-right direction is trivial and the converse follows from downward closure. $\square$

The validity problem for PD is in NEXPTIME

If $D \subseteq \mathrm{PROP}$, we denote by 2^D the set of all assignments $s\colon D \to \{0,1\}$.

Lemma 42

A $\mathrm{PL}[\mathrm{dep}]$-formula φ with proposition symbols in D is valid iff $2^D \models \varphi$.

Proof.

Left-to-right direction is trivial and the converse follows from downward closure. $\square$

Lemma 43

The validity problem for $\mathrm{PL}[\mathrm{dep}]$ is in NEXPTIME.

Proof.

Let $\varphi \in \mathrm{PL}[\mathrm{dep}]$ whose variables are in D. By Lemma 42, φ is valid iff $2^D \models \varphi$. The size of 2^D is $2^{|D|} \leq 2^{|\varphi|}$. Therefore 2^D can be constructed from φ in exponential time.

The validity problem for PD is in NEXPTIME

If $D \subseteq \text{PROP}$, we denote by 2^D the set of all assignments $s\colon D \to \{0,1\}$.

Lemma 42

A $\mathrm{PL}[\mathrm{dep}]$-formula φ with proposition symbols in D is valid iff $2^D \models \varphi$.

Proof.

Left-to-right direction is trivial and the converse follows from downward closure. $\square$

Lemma 43

The validity problem for $\mathrm{PL}[\mathrm{dep}]$ is in NEXPTIME.

Proof.

Let $\varphi \in \mathrm{PL}[\mathrm{dep}]$ whose variables are in D. By Lemma 42, φ is valid iff $2^D \models \varphi$. The size of 2^D is $2^{|D|} \leq 2^{|\varphi|}$. Therefore 2^D can be constructed from φ in exponential time. By Theorem 41, there exists an NP algorithm (with respect to $|2^D| + |\varphi|$) for checking whether $2^D \models \varphi$. Clearly this algorithm is in NEXPTIME with respect to $|\varphi|$. $\square$

The validity problem for PD is NEXPTIME-hard

We will associate each DQBF-formula μ with a corresponding $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ. Let

$$\mu = \big(\forall p_1 \ldots \forall p_n \exists q_1 \ldots \exists q_k\, \theta, (\vec{c}_1, \ldots, \vec{c}_k)\big)$$

be a DQBF-formula and denote by D_μ the set of propositional variables in μ, i.e., $D_\mu := \{p_1, \ldots, p_n, q_1, \ldots, q_k\}$.

The validity problem for PD is NEXPTIME-hard

We will associate each DQBF-formula μ with a corresponding $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ. Let

$$\mu = \big(\forall p_1 \ldots \forall p_n \exists q_1 \ldots \exists q_k\, \theta, (\vec{c}_1, \ldots, \vec{c}_k)\big)$$

be a DQBF-formula and denote by D_μ the set of propositional variables in μ, i.e., $D_\mu := \{p_1, \ldots, p_n, q_1, \ldots, q_k\}$. For each tuple of propositional variables $\vec{c}_i$, $i \leq k$, we stipulate that $\vec{c}_i = (p_{i_1}, \ldots, p_{i_{n_i}})$. Thus n_i denotes the lenth of $\vec{c}_i$. Define

$$\varphi_\mu := \theta \vee \bigvee_{i \leq k} \mathrm{dep}(p_{i_1}, \ldots, p_{i_{n_i}}, q_i).$$

$\mathrm{dep}(\bar{c}_i, q_i)$

We will show that μ is true if and only if the $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ is valid.

The validity problem for PD is NEXPTIME-hard

We will associate each DQBF-formula μ with a corresponding $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ. Let

$$\mu = \big(\forall p_1 \ldots \forall p_n \exists q_1 \ldots \exists q_k \, \theta, (\vec{c}_1, \ldots, \vec{c}_k)\big)$$

be a DQBF-formula and denote by D_μ the set of propositional variables in μ, i.e., $D_\mu := \{p_1, \ldots, p_n, q_1, \ldots, q_k\}$. For each tuple of propositional variables $\vec{c}_i$, $i \leq k$, we stipulate that $\vec{c}_i = (p_{i_1}, \ldots, p_{i_{n_i}})$. Thus n_i denotes the lenth of $\vec{c}_i$. Define

$$\varphi_\mu := \Big(\theta \wedge \bigwedge_{1 \leq k} \mathrm{dep}(\vec{c}_i, q_i)\Big) \vee \bigvee_{i \leq k} \mathrm{dep}(\vec{c}_i, q_i)$$

$$\varphi_\mu := \theta \vee \bigvee_{i \leq k} \mathrm{dep}(p_{i_1}, \ldots, p_{i_{n_i}}, q_i).$$

We will show that μ is true if and only if the $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ is valid. By Lemma 42, it suffices to show that μ is valid if and only if $2^{D_\mu} \models \varphi_\mu$.

The validity problem for PD is NEXPTIME-hard

We will associate each DQBF-formula μ with a corresponding $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ. Let

$$\mu = \big(\forall p_1 \ldots \forall p_n \exists q_1 \ldots \exists q_k\, \theta, (\vec{c}_1, \ldots, \vec{c}_k)\big)$$

be a DQBF-formula and denote by D_μ the set of propositional variables in μ, i.e., $D_\mu := \{p_1, \ldots, p_n, q_1, \ldots, q_k\}$. For each tuple of propositional variables $\vec{c}_i$, $i \leq k$, we stipulate that $\vec{c}_i = (p_{i_1}, \ldots, p_{i_{n_i}})$. Thus n_i denotes the lenth of $\vec{c}_i$. Define

$$\varphi_\mu := \theta \vee \bigvee_{i \leq k} \mathrm{dep}(p_{i_1}, \ldots, p_{i_{n_i}}, q_i).$$

We will show that μ is true if and only if the $\mathrm{PL}[\mathrm{dep}]$-formula φ_μ is valid. By Lemma 42, it suffices to show that μ is valid if and only if $2^{D_\mu} \models \varphi_\mu$. Since DQBF is NEXPTIME-complete and φ_μ is polynomial with respect to μ, it follows that the validity problem for $\mathrm{PL}[\mathrm{dep}]$ is NEXPTIME-hard.

A logic to rule them all

The extension of PL with the contradictory negation $\mathrm{PL}[\sim]$

$$X \models \sim\varphi \iff X \not\models \varphi$$

is very expressive and all connectives studied in team sematics can be defined in it.

A logic to rule them all

The extension of PL with the contradictory negation $\mathrm{PL}[\sim]$

$$X \models \sim\varphi \iff X \not\models \varphi \qquad \sim(\sim\varphi \vee \sim\psi)$$

is very expressive and all connectives studied in team sematics can be defined in it.

The connectives below can be defined in $\mathrm{PL}[\sim]$ with polynomial blow up.

$$\begin{aligned}
X \models \varphi \varovee \psi \quad &\Leftrightarrow \quad X \models \varphi \text{ or } X \models \psi, \\
X \models \varphi \otimes \psi \quad &\Leftrightarrow \quad \forall\, Y, Z \subseteq X: \text{ if } Y \cup Z = X, \text{ then } Y \models \varphi \text{ or } Z \models \psi, \\
X \models \varphi \to \psi \quad &\Leftrightarrow \quad \forall\, Y \subseteq X: \text{if } Y \models \varphi, \text{ then } Y \models \psi, \\
X \models \mathbf{max}(p_1, \dots, p_n) \quad &\Leftrightarrow \quad \{(s(p_1), \dots, s(p_n)) \mid s \in X\} = \{0,1\}^n.
\end{aligned}$$

Also dependence/inclusion/independence atoms can be expressed in $\mathrm{PL}[\sim]$ with polynomial blow up [LV19].

Expression	Defining $\mathrm{PL}[\sim]$-formula
$\varphi \otimes \psi$	$\sim(\sim\varphi \vee \sim\psi)$
$\varphi \owedge \psi$	$\sim(\sim\varphi \wedge \sim\psi)$
$\varphi \to \psi$	$(\sim\varphi \owedge \psi) \otimes \sim(p \vee \neg p)$
$\mathrm{dep}(p)$	$p \owedge \neg p$
$\mathrm{dep}(p_1, \ldots, p_n, q)$	$\bigwedge_{i=1}^{n} \mathrm{dep}(p_i) \to \mathrm{dep}(q)$
$\mathbf{max}(p_1, \ldots, p_n)$	$\sim \bigvee_{i=1}^{n} \mathrm{dep}(p_i)$

PTIME **Reductions Between Validity and Satisfiability**

Note: $X \models \sim(p \wedge \neg p)$ iff X is non-empty.

For $\varphi \in \mathrm{PL}[\mathcal{C}, \sim]$, define

$$\varphi_{\mathrm{SAT}} := \boxed{\mathbf{max}(\vec{x})} \to \overbrace{((p \vee \neg p) \vee (\varphi \wedge \sim(p \wedge \neg p)))}^{\psi},$$

$$\varphi_{\mathrm{VAL}} := \mathbf{max}(\vec{x}) \wedge \underbrace{(\sim(p \wedge \neg p) \to \varphi)}_{\psi^*},$$

where $\vec{x}$ lists the variables of φ

φ_{SAT} is VALID

iff $T \models \varphi_{SAT}$ for all T

iff $2^D \models \varphi_{SAT}$

iff $2^D \models \psi$

iff $T \models \varphi$ for some $T \subseteq 2^D$

$\boxed{\varphi_{VAL}}$ SATISFIABLE ?

iff $2^D \models \psi^*$

iff $\forall\, T \subseteq 2^D$ $T \neq \emptyset$ $T \models \varphi$

PTIME Reductions Between Validity and Satisfiability

Note: $X \models \sim(p \wedge \neg p)$ iff X is non-empty.

For $\varphi \in \mathrm{PL}[\mathcal{C}, \sim]$, define

$$\begin{aligned}
\varphi_{\mathrm{SAT}} &:= \mathbf{max}(\vec{x}) \to (\overbrace{(p \vee \neg p)}^{\text{ANY}} \vee (\overset{\overset{\varphi}{\downarrow}}{\varphi} \wedge \overbrace{\sim(p \wedge \neg p)}^{\neq \emptyset})),\\
\varphi_{\mathrm{VAL}} &:= \mathbf{max}(\vec{x}) \wedge (\sim(p \wedge \neg p) \to \varphi),
\end{aligned}$$

where $\vec{x}$ lists the variables of φ

Theorem 44

- *φ is satisfiable iff φ_{SAT} is valid.*
- *φ is valid iff φ_{VAL} is satisfiable.*

Oracle Turing Machines

The exponential-time hierarchy corresponds to the class of problems that can be recognized by an exponential-time alternating Turing machine with constantly many alternations.

In 1983 Orponen characterized the classes Σ_k^{EXP} and Π_k^{EXP} of the exponential time hierarchy by polynomial-time constant-alternation oracle Turing machines that query to k oracles.

Orponen's characterization can be generalised to exponential-time alternating Turing machines with polynomially many alternations (i.e. the class AEXPTIME(poly)) by allowing queries to polynomially many oracles.

Complexity of $\mathrm{PL}[\sim]$

Theorem 45

$\mathrm{SAT}(\mathrm{PL}[\sim])$ *is* AEXPTIME(poly)*-complete.*

Proof.

Hardness: By simulating polynomial time alternating oracle Turing machines.
Membership: Guess a possibly exponential-size team T and do APTIME model checking. $\square$

Complexity of $\mathrm{PL}[\sim]$

Theorem 45

$\mathrm{SAT}(\mathrm{PL}[\sim])$ *is* AEXPTIME(poly)*-complete.*

Proof.

Hardness: By simulating polynomial time alternating oracle Turing machines.
Membership: Guess a possibly exponential-size team T and do APTIME model checking. $\square$

Corollary 46

$\mathrm{VAL}(\mathrm{PL}[\sim])$ *is* AEXPTIME(poly)*-complete.*

Complexity of $\mathrm{PL}[\sim]$

Theorem 45

$\mathrm{SAT}(\mathrm{PL}[\sim])$ *is* AEXPTIME(poly)*-complete.*

Proof.

Hardness: By simulating polynomial time alternating oracle Turing machines.
Membership: Guess a possibly exponential-size team T and do APTIME model checking. $\square$

Corollary 46

$\mathrm{VAL}(\mathrm{PL}[\sim])$ *is* AEXPTIME(poly)*-complete.*

Theorem 47

$\mathrm{MC}(\mathrm{PL}[\sim])$ *is* PSPACE*-complete*

Complexity Results

Logic	SAT	VAL	MC
PL	NP [0]	coNP [0]	NC[1] [1]
$\mathrm{PL}[\mathrm{dep}]$	NP [3]	NEXPTIME [4]	NP [2]
$\mathrm{PL}[\bot_{\mathrm{c}}]$	NP [7]	in $\mathrm{coNEXPTIME}^{\mathrm{NP}}$ [7]	NP [7]
$\mathrm{PL}[\subseteq]$	EXP [5]	coNP [7]	in P [6]
$\mathrm{PL}[\sim]$	AEXPTIME(poly) [7]	AEXPTIME(poly) [7]	PSPACE [8]

[0] Cook 1971, Levin 1973, [1] Buss 1987, [2] Ebbing, Lohmann 2012,
[3] Lohmann, Vollmer 2013, [4] Virtema 2014, [5] Hella, Kuusisto, Meier, Vollmer 2015,
[6] Hella, Kuusisto, Meier and Virtema 2019,
[7] Hannula, Kontinen, Virtema, Vollmer 2018, [8] Müller 2014.

Conclusion of Lecture 4

- DQBF is a canonical NEXPTIME-complete problem.
- $\mathrm{SAT}(\mathrm{PL}[\mathrm{dep}])$ and $\mathrm{MC}(\mathrm{PL}[\mathrm{dep}])$ are NP-complete.
- $\mathrm{VAL}(\mathrm{PL}[\mathrm{dep}])$ is NEXPTIME-complete.
- $\mathrm{SAT}(\mathrm{PL}[\sim])$ and $\mathrm{VAL}(\mathrm{PL}[\sim])$ are AEXPTIME(poly)-complete.
- $\mathrm{MC}(\mathrm{PL}[\sim])$ are PSPACE-complete.

Complexity and Expressivity of Propositional Logics with Team Semantics
Arne Meier, Jonni Virtema
9th of August

Lecture 5: Recent Trends: Hyperproperties

Literature: [Vir+21; Gut+22]

Bibliography i

[BRV01] Patrick Blackburn, Maarten de Rijke and Yde Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. DOI: 10.1017/CBO9781107050884.

[CES86] E. Clarke, E. Allen Emerson and A. Sistla. 'Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications'. In: *ACM Transactions on Programming Languages and Systems* 8.2 (1986), pp. 244–263.

[Coo71] Stephen A. Cook. 'The Complexity of Theorem-Proving Procedures'. In: *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*. Ed. by Michael A. Harrison, Ranan B. Banerji and Jeffrey D. Ullman. ACM, 1971, pp. 151–158. DOI: 10.1145/800157.805047. URL: https://doi.org/10.1145/800157.805047.

[EFT94] Heinz-Dieter Ebbinghaus, Jörg Flum and Wolfgang Thomas. *Mathematical logic (2. ed.)* Undergraduate texts in mathematics. Springer, 1994.

[EJ99] E. Allen Emerson and Charanjit S. Jutla. 'The Complexity of Tree Automata and Logics of Programs'. In: *SIAM J. Comput.* 29.1 (1999), pp. 132–158.

[ES84] E. Allen Emerson and A. Prasad Sistla. 'Deciding Full Branching Time Logic'. In: *Inf. Control.* 61.3 (1984), pp. 175–201.

[FL79] Michael J. Fischer and Richard E. Ladner. 'Propositional Dynamic Logic of Regular Programs'. In: *J. Comput. Syst. Sci.* 18.2 (1979), pp. 194–211.

[GHR95] Raymond Greenlaw, H. James Hoover and Walter L. Ruzzo. *Limits to Parallel Computation: P-completeness Theory*. New York, NY, USA: Oxford University Press, Inc., 1995. ISBN: 0-19-508591-4.

[Gol77] L. M. Goldschlager. 'The monotone and planar circuit value problems are log-space complete for P'. In: *SIGACT News* 9 (1977), pp. 25–29.

[Gut+22] Jens Oliver Gutsfeld, Arne Meier, Christoph Ohrem and Jonni Virtema. 'Temporal Team Semantics Revisited'. In: *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*. Ed. by Christel Baier and Dana Fisman. ACM, 2022, 44:1–44:13. DOI: 10.1145/3531130.3533360. URL: https://doi.org/10.1145/3531130.3533360.

[Han+18] Miika Hannula, Juha Kontinen, Jonni Virtema and Heribert Vollmer. 'Complexity of Propositional Logics in Team Semantic'. In: *ACM Trans. Comput. Log.* 19.1 (2018), 2:1–2:14.

[Han19] Miika Hannula. 'Validity and Entailment in Modal and Propositional Dependence Logics'. In: *Logical Methods in Computer Science* Volume 15, Issue 2 (Apr. 2019). DOI: 10.23638/LMCS-15(2:4)2019. URL: https://lmcs.episciences.org/5403.

[Hel+14] Lauri Hella, Kerkko Luosto, Katsuhiko Sano and Jonni Virtema. 'The Expressive Power of Modal Dependence Logic'. In: *Advances in Modal Logic*. College Publications, 2014, pp. 294–312.

[Hel+19] Lauri Hella, Antti Kuusisto, Arne Meier and Jonni Virtema. 'Model checking and validity in propositional and modal inclusion logics'. In: *J. Log. Comput.* 29.5 (2019), pp. 605–630.

[Hel+20] Lauri Hella, Antti Kuusisto, Arne Meier and Heribert Vollmer. 'Satisfiability of Modal Inclusion Logic: Lax and Strict Semantics'. In: *ACM Trans. Comput. Log.* 21.1 (2020), 7:1–7:18.

[HS15] Lauri Hella and Johanna Stumpf. 'The expressive power of modal logic with inclusion atoms'. In: *GandALF*. Vol. 193. EPTCS. 2015, pp. 129–143.

[KV85] Gabriel M. Kuper and Moshe Y. Vardi. 'On the Expressive Power of the Logical Data Model (Preliminary Report)'. In: *SIGMOD Conference*. ACM Press, 1985, pp. 180–187.

[Lev73] Leonid A. Levin. 'Universal sequential search problems'. In: *Problemy Peredachi Informatsii* 9.3 (1973).

[Loh12] Peter Lohmann. 'Computational Aspects of Dependence Logic'. PhD thesis. Leibniz Universität Hannover, 2012. arXiv: `1206.4564`. URL: `http://arxiv.org/abs/1206.4564`.

[LV19] Martin Lück and Miikka Vilander. 'On the Succinctness of Atoms of Dependency'. In: *Log. Methods Comput. Sci.* 15.3 (2019).

[Pap07] Christos H. Papadimitriou. *Computational complexity*. Academic Internet Publ., 2007.

[Pra80] V. R. Pratt. 'A near-optimal method for reasoning about action'. In: *Journal of Computer and System Sciences* 20.2 (1980), pp. 231–254.

[SC85] A. Prasad Sistla and Edmund M. Clarke. 'The Complexity of Propositional Linear Temporal Logics'. In: *J. ACM* 32.3 (1985), pp. 733–749.

[Sch02] P. Schnoebelen. 'The Complexity of Temporal Logic Model Checking'. In: *Advances in Modal Logic*. Vol. 4. 2002, pp. 393–436.

[Sip97] Michael Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997.

[Var09] Moshe Y. Vardi. 'From Philosophical to Industrial Logics'. In: *ICLA*. Vol. 5378. Lecture Notes in Computer Science. Springer, 2009, pp. 89–115.

[Vir+21] Jonni Virtema, Jana Hofmann, Bernd Finkbeiner, Juha Kontinen and Fan Yang. 'Linear-Time Temporal Logic with Team Semantics: Expressivity and Complexity'. In: *FSTTCS*. Vol. 213. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 52:1–52:17.

[Vir17] Jonni Virtema. 'Complexity of validity for propositional dependence logics'. In: *Inf. Comput.* 253 (2017), pp. 224–236.

[YV17] Fan Yang and Jouko Väänänen. 'Propositional team logics'. In: *Ann. Pure Appl. Log.* 168.7 (2017), pp. 1406–1441. DOI: 10.1016/J.APAL.2017.01.007. URL: https://doi.org/10.1016/j.apal.2017.01.007.