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 4th August 2024

Preliminary skirmish

Organisational information about the course

Material for the course:

- Lecture notes
- Slides (in advance) and writings into slides (afterwards)

About the lecturers

Arne Meier (Leibniz University Hannover)
Research Interests: Complexity Theory, Foundations of AI, Non-Classical Logics, Enumeration
https://arnemeier.github.io

Jonni Virtema (University of Sheffield)
Research Interests: Finite Model Theory, Temporal Logics for Hyperproperties, Logical Foundations of Neural Networks, Complexity Theory.
http://www.virtema.fi/

Prerequisites and requirements

- Complexity theory foundations, e.g., [Pap07; Sip97]
- Propositional Logic foundations, e.g., [EFT94]
- Modal Logic (only relevant for last lecture), e.g., [BRV01]

Course outline

Monday, 5th of August Syntax and Semantics, Properties, Problems.

Tuesday, 6th of August Expressivity and succintness

Wednesday, 7th of August Inclusion Logic: P-complete MC, coNP-complete VAL

Thursday, 8th of August Dependence. Show MC(PDL) is NP-complete. DQBF, VAL(PDL) is NEXP-complete

Friday, 9th of August Hyperproperties, Temporal Aspects. TeamLTL(inclusion, dep) is undecidable.

Complexity and Expressivity of Propositional Logics with Team Semantics
Arne Meier, Jonni Virtema
5th of August

Lecture 1: Propositional Logics with Team Semantics

Literature: [YV17]

Dependence and independence

What means "x depends on y" or "x and y are independent"?

Compare it to: "x divides y"

Here: fix structure $\mathcal{A}$, with well-defined division and find an assignment $s\colon \{x, a\} \to A$.

Then: Check Tarskian semantics of $\mathcal{A} \models_s$"x divides y"

What means "x depends on y" or "x and y are independent"?

Compare it to: "x divides y"

Here: fix structure $\mathcal{A}$, with well-defined division and find an assignment $s\colon \{x, a\} \to A$.

Then: Check Tarskian semantics of $\mathcal{A} \models_s$"x divides y"

Caution: (In-)dependence is different. It does not manifest itself in single assignments, but in

- tables or relations
- sets of rounds of a game
- sets of assignments $\rightsquigarrow$ teams

(In-)Dependence Logics: Henkin-Quantifiers (1959)

$\exists x \forall y \exists z \forall u \exists v \varphi(x,y,z,u,v)$ in FO

$$\varphi = \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} P(x,y,u,v)$$

Semantics: over Skolem functions or via games with imperfect information

Leon A. Henkin
(1921–2006)

$(A,P) \models \varphi$, if there are functions $f, g\colon A \to A$ such that for all $a, c \in A$

$$P(a, f(a), c, g(c))$$

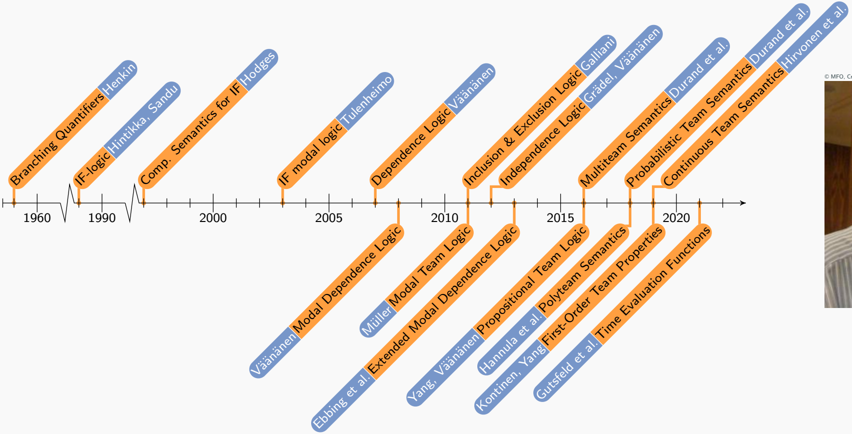
(In-)Dependence Logics: Independence-friendly logic (1989)

- First logic with quantifiers that are annotated with independence
- Quantification: φ formula, x variable, W finite set of variables yields expressions $(\exists x/W)\varphi$ and $(\forall x/W)\varphi$
- Game-theoretic Semantics: In the evaluation game for $(\partial x/W)\varphi$, the value x has to be chosen independent of the values in W
- At two positions $((\partial x/W)\varphi, s)$ and $((\partial x/W)\varphi, s')$ with $s(y) \neq s'(y)$ for all $y \in W$, the same value for x has to be chosen

Jaakko Hintikka
(1929-2005)

Gabriel Sandu
(* 1954)

Dependence Logic: Historically



Jouko Väänänen
(* 1950)

Dependence Logic: A Bit of Motivation

Primary key (docent, time)

docent	time	room	lecture
Antti	09:00	A.10	Genetics
Antti	11:00	A.10	Biochemistry
Antti	15:00	B.20	Ecology
Jonni	10:00	C.30	Bio-LAB
Juha	10:00	C.30	Bio-LAB
Juha	13:00	A.10	Biochemistry
⋮	⋮	⋮	⋮

Dependence Logic: A Bit of Motivation

Primary key (docent, time)

docent	time	room	lecture
Antti	09:00	A.10	Genetics
Antti	11:00	A.10	Biochemistry
Antti	15:00	B.20	Ecology
Jonni	10:00	C.30	Bio-LAB
Juha	10:00	C.30	Bio-LAB
Juha	13:00	A.10	Biochemistry
⋮	⋮	⋮	⋮

Task: Consistency check of a timetable.
{docent, time} functionally determines {room, lecture}, where {room, time} does not functionally determine {docent}.

Dependence Logic: Applications

Dependence Atom

- Models functional dependencies in sets of assignments
- Semantics: y depends on x, i.e., y is uniquely determined by x

x	y	z
0	1	0
0	1	1

$\models \mathrm{dep}(\{x\}; \{y\})$

x	y	z
0	1	0
0	0	1

$\not\models \mathrm{dep}(\{x\}; \{y\})$

Dependence Logic: Applications

Dependence Atom

- Models functional dependencies in sets of assignments
- Semantics: y depends on x, i.e., y is uniquely determined by x

x	y	z
0	1	0
0	1	1

$\models \mathrm{dep}(\{x\}; \{y\})$

x	y	z
0	1	0
0	0	1

$\not\models \mathrm{dep}(\{x\}; \{y\})$

Applications: Modelling of...

- database schemes
- deterministic behaviour
- specifications
- ...

Team-Based Propositional Logic

Definition 1

Let PROP be a countably infinite set of propositions.

- Propositional Team Logic (PL):

  only atomic negation!

$$\varphi ::= x \mid \neg x \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

  where $x \in \mathrm{PROP}$ ~~and $P, Q \subseteq \mathrm{PROP}$~~.

- A team over PROP is a set of assignments, i.e., an element of $\mathcal{P}(2^{\mathrm{PROP}})$

assignment $s : \mathrm{PROP} \to \{0,1\}$

For assignmt. s: PROP $\to \{0,1\}$ and $P \subseteq$ PROP, $s{\restriction}P$ is assignmt. restricted to P.

Definition 2

Let T be a team and $\varphi, \psi \in \mathrm{PL}[\mathrm{dep}]$. We define $T \models \varphi$ recursively via:

$$
\begin{array}{lll}
T \models x & \text{iff} & s(x) = 1 \quad \forall s \in T, \\
T \models \neg x & \text{iff} & s(x) = 0 \quad \forall s \in T, \\
T \models \varphi \wedge \psi & \text{iff} & T \models \varphi \ \text{ and } T \models \psi, \\
T \models \varphi \vee \psi & \text{iff} & \exists T_1 \exists T_2 (T = T_1 \cup T_2) \text{ s.t. } T_1 \models \varphi \ \text{ and } T_2 \models \psi.
\end{array}
$$

Caution: Only atomic negation here.

$P, Q \subseteq \mathrm{PROP}$

$\mathrm{dep}(x_1, \dots, x_n; y) = (x_1, \dots, x_n; y)$ $\quad$ $\mathrm{dep}(x_1, \dots, x_n; y_1, \dots, y_n)$

$$T \models \mathrm{dep}(P; Q) \quad \text{iff} \quad \forall s, t \in T : s{\restriction}P = t{\restriction}P \Rightarrow s{\restriction}Q = t{\restriction}Q.$$

Notation: $\mathrm{PL}[\mathrm{dep}]$ for propositional logic with dependence atoms.

Observations:

$$T \models \mathrm{dep}(;Q) \quad \text{iff} \quad \forall s, t \in T : s{\restriction}Q = t{\restriction}Q \quad \text{(Constancy Atom)},$$
$$T \models \neg\mathrm{dep}(P, Q) \quad \text{iff} \quad T = \emptyset.$$

Team Semantics from the Database Perspective

propositions $\triangleq$ attributes

assignment $\triangleq$ **entries**

docent	time	room	lecture
Antti	09:00	A.10	Genetics
Antti	11:00	A.10	Biochemistry
Antti	15:00	B.20	Ecology
Jonni	10:00	C.30	Bio-LAB
Juha	10:00	C.30	Bio-LAB
Juha	13:00	A.10	Biochemistry
⋮	⋮	⋮	⋮

team $\triangleq$ table

Back to the Initial Example

primary key (docent, time)

docent	time	room	lecture
Antti	09:00	A.10	Genetics
Antti	11:00	A.10	Biochemistry
Antti	15:00	B.20	Ecology
Jonni	10:00	C.30	Bio-LAB
Juha	10:00	C.30	Bio-LAB
Juha	13:00	A.10	Biochemistry
⋮	⋮	⋮	⋮

$\Rightarrow$ $\{\texttt{docent}, \texttt{time}\}$ functionally determines $\{\texttt{room}, \texttt{lecture}\}$.

Back to the Initial Example

docent	time	room	lecture
Antti	09:00	A.10	Genetics
Antti	11:00	A.10	Biochemistry
Antti	15:00	B.20	Ecology
Jonni	10:00	C.30	Bio-LAB
Juha	10:00	C.30	Bio-LAB
Juha	13:00	A.10	Biochemistry
⋮	⋮	⋮	⋮

(docent and time together are marked as the primary key)

$\Rightarrow$ $\{\texttt{docent}, \texttt{time}\}$ functionally determines $\{\texttt{room}, \texttt{lecture}\}$.

How do you express this in $\mathrm{PL}[\mathrm{dep}]$?

Back to the Initial Example

primary key			
docent	time	room	lecture
Antti	09:00	A.10	Genetics
Antti	11:00	A.10	Biochemistry
Antti	15:00	B.20	Ecology
Jonni	10:00	C.30	Bio-LAB
Juha	10:00	C.30	Bio-LAB
Juha	13:00	A.10	Biochemistry
⋮	⋮	⋮	⋮

$\Rightarrow$ $\{\texttt{docent}, \texttt{time}\}$ functionally determines $\{\texttt{room}, \texttt{lecture}\}$.

How do you express this in $\mathrm{PL}[\mathrm{dep}]$?

$$\mathrm{dep}(\underbrace{\{\texttt{docent}, \texttt{time}\}}_{P}, \underbrace{\{\texttt{room}, \texttt{lecture}\}}_{Q})$$

We only consider Propositional Team Logic here

Caution: encode all entries in binary (Propositional Logic vs. FO)

docent	room	time	lecture
Antti	A.10	09.00	Genetics
Antti	A.10	11.00	Biochemistry
Antti	B.20	15.00	Ecology
Jonni	C.30	10.00	Bio-Lab
Juha	C.30	10.00	Bio-Lab
Juha	A.10	13.00	Biochemistry

$i_1 i_2$	$r_1 r_2$	$t_1 t_2 t_3$	$c_1 c_2$
00	11	110	11
00	11	111	00
00	00	000	01
01	01	001	10
10	01	001	10
10	11	010	00

(Left) Sample database with 4 attributes and universe size 15.
(Right) Encoding with $\lceil \log_2(3) \rceil + \lceil \log_2(3) \rceil + \lceil \log_2(5) \rceil + \lceil \log_2(4) \rceil$-many propositions.

Interesting and Important Properties of such Logics

Comes a little bit later

property	definition	dep	$\subseteq$	$\perp$	$\mid$
Downward closure	$T \models \varphi$ and $T' \subseteq T$ implies $T' \models \varphi$	$\checkmark$	$\times$	$\times$	$\checkmark$
Union closure	$T \models \varphi$ and $T' \models \varphi$ implies $T \cup T' \models \varphi$	$\times$	$\checkmark$	$\checkmark$	$\times$

All logics considered here are:

flat: $T \models \varphi \iff \forall s \in T : \{s\} \models \varphi$

and satisfy the

Empty team property: $\emptyset \models \varphi$.

Downward Closure of $\mathrm{PL}[\mathrm{dep}]$

Lemma 3

$\mathrm{PL}[\mathrm{dep}]$ *is downward closed.*

Proof. By structural induction over φ. Let T be any team.

IB 1) $\varphi = x \in PROP$. If $T \models \varphi$ then $T \models x$. Hence $\forall s \in T : s(x) = 1$.

$\Rightarrow \forall T' \subseteq T : \forall s' \in T' \; s'(x) = 1 \Rightarrow T' \models \varphi$.

2) $\varphi = dep(P; Q)$. Semantics: $\forall s,t \in T : s \upharpoonright P = t \upharpoonright P \Rightarrow s \upharpoonright Q = t \upharpoonright Q$.

$\Rightarrow \forall T' \subseteq T : \forall s',t' \in T' \; s' \upharpoonright P = t' \upharpoonright P \Rightarrow s' \upharpoonright Q = t' \upharpoonright Q \Rightarrow T' \models dep(P;Q)$

IS 1) $\varphi = \neg x$, $x \in PROP$, then similarly to IB 1) we get $T' \models \varphi$.

Lemma 3

PL[dep] *is downward closed.*

2) $\varphi = \alpha \wedge \beta$. By $T \vDash \varphi$ we get $T \vDash \alpha$ and $T \vDash \beta$ (semantics).
By IH $\forall T' \subseteq T : T' \vDash \alpha$, By IH $\forall T' \subseteq T : T' \vDash \beta$
Hence: $\forall T' \subseteq T : T' \vDash \alpha \wedge \beta$.

3) $\varphi = \alpha \vee \beta$. By semantics, we get $\exists T_1, T_2$ with $T = T_1 \cup T_2$ s.t.
$T_1 \vDash \alpha$ and $T_2 \vDash \beta$.
By IH: $\forall T_1' \subseteq T_1 : T_1' \vDash \alpha$
By IH: $\forall T_2' \subseteq T_2 : T_2' \vDash \beta$
$\Big\}$ $\forall T' \subseteq T \, \exists T_1', T_2', T_1' \cup T_2' = T'$
$T' \vDash \alpha \vee \beta = \varphi$

$\square$

Decision Problems

Problem: $\mathrm{PL[dep]\text{-}MC}$ — the model checking problem
Input: A $\mathrm{PL[dep]}$-formula φ, a team T over **Vars**(φ)
Question: Is $T \models \varphi$ true?

Problem: $\mathrm{PL[dep]\text{-}SAT}$ — the satisfiability problem
Input: A $\mathrm{PL[dep]}$-formula φ
Question: Exists a non-empty team T over **Vars**(φ) with $T \models \varphi$?

Theorem 4 ([Loh12, Theorem 4.13], Proof: Thursday)

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

Satisfiability Does not Become Harder Than in the Classical Case

Complexity of Classical SAT

Theorem 5 ([Coo71; Lev73])

PL[dep]-SAT *is* NP-*complete.*

We know: PL[dep] is downward closed. This gives:

Lemma A PL[dep]-formula is satisfiable iff a singleton team satisfies it.

Note: dependence atoms are always satisfied by singleton teams.

Lemma For any $\varphi \in$ PL[dep] and singleton team $\{s\}$, we have that

$\{s\} \models \varphi$ iff $\{s\} \models \varphi^*$, where φ^* is φ but every dep(P,Q)

is replaced by constant true, e.g., prop. p ∨ ¬p

Proof of Thm 5

NP-hardness: follows for singletons wrt. classical NP-hardness

NP-membership: two Lemmas of above yield reduction to classical SAT. □

Inclusion (Wednesday)

$p_i, q_i \in \mathrm{PROP}$ $\quad\quad \bar{p} = p_1 \cdots p_k$

Inspired by "inclusion dependencies" from database theory.

$$T \models p_1 \cdots p_k \subseteq q_1 \cdots q_k \text{ iff } \forall u \in T \, \exists v \in T : u(\bar{p}) = v(\bar{q})$$

Theorem 6 ([Hel+20, Cor. 3.6])

$\mathrm{PL}[\subseteq]$-SAT *is* EXP*-complete.* $\mathrm{TIME}(2^{n^{O(1)}})$

Theorem 7 ([Hel+19, Thm. 13])

$\mathrm{PL}[\subseteq]$-MC *is* P*-complete.*

Looking Beyond the Horizon: Independence

Caution: Also exists in stochastics; two events are independent if the occurrence of one does not influence the probability of the other occurring

Looking Beyond the Horizon: Independence

Caution: Also exists in stochastics; two events are independent if the occurrence of one does not influence the probability of the other occurring

But:
- logical independence compatible with this
- every possible pattern for (x, y) occurs, but how often does not matter
- knowing only x/y gives no information about the other

Looking Beyond the Horizon: Independence

Caution: Also exists in stochastics; two events are independent if the occurrence of one does not influence the probability of the other occurring

But:

- logical independence compatible with this
- every possible pattern for (x, y) occurs, but how often does not matter
- knowing only x/y gives no information about the other

$$T \models p_1 \cdots p_{k_1} \perp_{r_1 \cdots r_{k_2}} q_1 \cdots q_{k_3} \text{ iff } \forall (u, v) \in T \times T \text{ s.t. } u(\bar{r}) = v(\bar{r})$$

$$\exists w \in T : u(\bar{p}\bar{r}) = w(\bar{p}\bar{r}) \wedge w(\bar{q}) = v(\bar{q})$$

Looking Beyond the Horizon: Independence

Caution: Also exists in stochastics; two events are independent if the occurrence of one does not influence the probability of the other occurring

But:
- logical independence compatible with this
- every possible pattern for (x, y) occurs, but how often does not matter
- knowing only x/y gives no information about the other

$$T \models p_1 \cdots p_k \perp_{r_1 \cdots r_k} q_1 \cdots q_k \text{ iff } \forall\, (u, v) \in T \times T \text{ mit } u(\bar{r}) = v(\bar{r})$$
$$\exists\, w \in T : u(\bar{p}\bar{r}) = w(\bar{p}\bar{r}) \wedge w(\bar{q}) = v(\bar{q})$$

“the variables in $\bar{p}$ are completely independent of $\bar{q}$ for each constant value of $\bar{r}$”

Theorem 8 ([Han+18])

$\mathrm{PL}[\perp]$-SAT *and* $\mathrm{PL}[\perp]$-MC *are* NP-*complete.*

Looking Beyond the Horizon: Exclusion

$$T \models p_1 \cdots p_k \mid q_1 \cdots q_k \text{ iff } \forall (u, v) \in T \times T : u(\bar{p}) \neq v(\bar{q})$$

Theorem 9 (by vanilla SAT, [Coo71; Lev73])

$\mathrm{PL}[|]$-*SAT is* NP-*complete.*

classical problem is coNP-complete.

φ is valid iff $1 \to \varphi$ is in IMP.

A formula φ entails a formula ψ if and only if every team that satisfies φ also satisfies ψ, written $\varphi \models \psi$. A set of formulas Σ entails a formula φ if and only if every team that satisfies all formulas in Σ also satisfies φ, written $\Sigma \models \varphi$.

Problem: $\mathrm{PL[dep]}$-IMP — the entailment problem for $\mathrm{PL[dep]}$
Eingabe: a set of $\mathrm{PL[dep]}$-formulas Σ, a $\mathrm{PL[dep]}$-formula φ
Frage: Is $\Sigma \models \varphi$ true?

Theorem 10 ([Han19, Thm. 5.6, Thm. 6.1])

$\mathrm{PL[dep]}$-IMP *is* coNEXPTIME$^{\mathrm{NP}}$*-complete.*

such calls are charged O(1) time units

base machine in coNEXPTIME, oracle calls to an NP-machine

Conclusion of Lecture 1

- Team semantics
- Dependence Atoms
- Inclusion, Exclusion, Independence
- Complexity of Satisfiability of $\mathrm{PL}[\mathrm{dep}]$
- Properties of $\mathrm{PL}[\mathrm{dep}]$ Downward closure

Complexity and Expressivity of Propositional Logics with Team Semantics
Arne Meier, Jonni Virtema
6th of August

Lecture 2: Expressive power of team-based logics

Literature: [YV17; Hel+14]

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.

Bibliography v

[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.