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 5th August 2024

www.thi.uni-hannover.de/de/esslli24

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]

How to characterise expressivity – Tarski's semantics

Definition 11

If φ is formula of propositional logic, with variables $p_1 \ldots, p_n$, one can say that φ defines the n-ary Boolean function $f_\varphi : \{0,1\}^n \to \{0,1\}$ defined

$$s \mapsto s(\varphi),$$

where s is an assignment for the variables $p_1 \ldots, p_n$.

One can then ask, which Boolean functions can be expressed in propositional logic.

How to characterise expressivity – Tarski's semantics

Definition 11

If φ is formula of propositional logic, with variables $p_1 \ldots, p_n$, one can say that φ defines the n-ary Boolean function $f_\varphi : \{0,1\}^n \to \{0,1\}$ defined

$$s \mapsto s(\varphi),$$

where s is an assignment for the variables $p_1 \ldots, p_n$.

One can then ask, which Boolean functions can be expressed in propositional logic. In fact, propositional logic is expressively complete (in the standard Tarskian setting).

Proposition 12

Every Boolean function can be defined in propositional logic.

How to characterise expressivity – Team semantics – definitions

In team semantics setting, a propositional formula defines a set of teams that satisfy it.

Definition 13

We define

$$\mathrm{Teams}(\varphi) := \{T \mid T \models \varphi\}$$

We then want to know, what are the families of teams that can be written as $\mathrm{Teams}(\varphi)$ by some formula φ.

$$T \models \varphi \iff T\upharpoonright_{VAR(\varphi)} \models \varphi$$

How to characterise expressivity – Team semantics – definitions

In team semantics setting, a propositional formula defines a set of teams that satisfy it.

Definition 13

We define

$$\mathrm{Teams}(\varphi) := \{T \mid T \models \varphi\}$$

We then want to know, what are the families of teams that can be written as $\mathrm{Teams}(\varphi)$ by some formula φ.
Definitions of downward/union closure and flatness generalise to families of teams.

Definition 14

A family of teams $\mathcal{T}$ is

- downward closed, if ($T \in \mathcal{T}$ and $S \subseteq T$) implies $S \in \mathcal{T}$.
- union closed, if $T, S \in \mathcal{T}$ implies $T \cup S \in \mathcal{T}$.
- flat, if $T \in \mathcal{T}$ if and only if $\{t\} \in \mathcal{T}$, for all $t \in T$.

Properties of families of teams

$T \models \varphi \iff \forall s \in T \; \{s\} \models \varphi$

Proposition 15

A family of teams $\mathcal{T}$ is flat if and only if it is union & downward closed and $\emptyset \in \mathcal{T}$.

Proof.

Left-to-right direction if trivial.

Properties of families of teams

Proposition 15

A family of teams $\mathcal{T}$ is flat if and only if it is union & downward closed and $\emptyset \in \mathcal{T}$.

Proof.

Left-to-right direction if trivial. For the right-to-left direction, assume that $\mathcal{T}$ is union & downward closed and that $\emptyset \in \mathcal{T}$. Now the left-to-right direction of

$$T \in \mathcal{T} \iff \forall t \in T : \{t\} \in \mathcal{T}$$

follows from downward closure, while the converse direction follows from union closure. The empty team property is required to omit the special case of $\mathcal{T} = \emptyset$. □

Proposition 16

Let $\mathcal{T}$ be a flat family of teams. Then $T \in \mathcal{T}$ if and only if $T \subseteq \bigcup \mathcal{T}$.

Proof.

Left-to-right direction is trivial and follows directly from the definition of a union.

Proposition 16

Let $\mathcal{T}$ be a flat family of teams. Then $T \in \mathcal{T}$ if and only if $T \subseteq \bigcup \mathcal{T}$.

Proof.

Left-to-right direction is trivial and follows directly from the definition of a union.
Right-to-left direction: By Proposition 15, $\mathcal{T}$ is union closed and downward closed.

Proposition 16

Let $\mathcal{T}$ be a flat family of teams. Then $T \in \mathcal{T}$ if and only if $T \subseteq \bigcup \mathcal{T}$.

Proof.

Left-to-right direction is trivial and follows directly from the definition of a union.
Right-to-left direction: By Proposition 15, $\mathcal{T}$ is union closed and downward closed.
From union closure of $\mathcal{T}$ it follows that $\bigcup \mathcal{T} \in \mathcal{T}$.

Proposition 16

Let $\mathcal{T}$ be a flat family of teams. Then $T \in \mathcal{T}$ if and only if $T \subseteq \bigcup \mathcal{T}$.

Proof.

Left-to-right direction is trivial and follows directly from the definition of a union. Right-to-left direction: By Proposition 15, $\mathcal{T}$ is union closed and downward closed. From union closure of $\mathcal{T}$ it follows that $\bigcup \mathcal{T} \in \mathcal{T}$. Now since $\mathcal{T}$ is downward closed and $T \subseteq \bigcup \mathcal{T}$, if follows that $T \in \mathcal{T}$. □

How to characterise expressivity – Team semantics – results

We have already seen (and partly proved) the following closure results:

Proposition 17

- *A family of teams defined by a PL-formula is flat.*
- $\mathrm{PL}[\mathrm{dep}]$-*definable team families are downward closed and include the empty team.*

How to characterise expressivity – Team semantics – results

We have already seen (and partly proved) the following closure results:

Proposition 17

$T \models \varphi \iff \forall s \in T \;\{s\} \models \varphi$

- *A family of teams defined by a PL-formula is flat.*
- $\mathrm{PL}[\mathrm{dep}]$*-definable team families are downward closed and include the empty team.*

Proof.

Flatness is proven by structural induction. The cases for atomic formulae follow directly from their semantics. The case for $\wedge$ is trivial.

$p, \neg p$

How to characterise expressivity – Team semantics – results

We have already seen (and partly proved) the following closure results:

Proposition 17

- *A family of teams defined by a PL-formula is flat.*
- $\mathrm{PL}[\mathrm{dep}]$*-definable team families are downward closed and include the empty team.*

Proof.

Flatness is proven by structural induction. The cases for atomic formulae follow directly from their semantics. The case for $\wedge$ is trivial. Assume flatness holds for φ and ψ.

$$T \models \varphi \vee \psi \iff T_1 \models \varphi \text{ and } T_2 \models \psi \text{ for some } T_1 \cup T_2 = T$$

How to characterise expressivity – Team semantics – results

We have already seen (and partly proved) the following closure results:

Proposition 17

- *A family of teams defined by a PL-formula is flat.*
- $\mathrm{PL}[\mathrm{dep}]$*-definable team families are downward closed and include the empty team.*

Proof.

Flatness is proven by structural induction. The cases for atomic formulae follow directly from their semantics. The case for $\wedge$ is trivial. Assume flatness holds for φ and ψ.

$$T \models \varphi \vee \psi \iff T_1 \models \varphi \text{ and } T_2 \models \psi \text{ for some } T_1 \cup T_2 = T$$

By IH, the right-hand side is equivalent to: $\forall t \in T : \{t\} \models \varphi$ or $\{t\} \models \psi$. This is again equivalent to $\forall t \in T : \{t\} \models \varphi \vee \psi$, due to the empty team property. □

How to characterise expressivity – Team semantics – results

We have already seen (and partly proved) the following closure results:

Proposition 17

- *A family of teams defined by a PL-formula is flat.*
- $\mathrm{PL}[\mathrm{dep}]$*-definable team families are downward closed and include the empty team.*

Proof.

Flatness is proven by structural induction. The cases for atomic formulae follow directly from their semantics. The case for $\wedge$ is trivial. Assume flatness holds for φ and ψ.

$$T \models \varphi \vee \psi \iff T_1 \models \varphi \text{ and } T_2 \models \psi \text{ for some } T_1 \cup T_2 = T$$

By IH, the right-hand side is equivalent to: $\forall t \in T : \{t\} \models \varphi$ or $\{t\} \models \psi$. This is again equivalent to $\forall t \in T : \{t\} \models \varphi \vee \psi$, due to the empty team property. $\square$

Interestingly the above results can be strengthened to if and only if!

Expressivity of PL with team semantics

Proposition 18

For every flat family $\mathcal{T}$ there exists a PL-formula φ such that $\mathcal{T} = \mathrm{Teams}(\varphi)$.

Proof.

Expressivity of PL with team semantics

Proposition 18

For every flat family $\mathcal{T}$ there exists a PL-formula φ such that $\mathcal{T} = \mathrm{Teams}(\varphi)$.

Proof.

Let $\mathcal{T}$ be a flat family of teams using proposition symbols $p_1, \ldots, p_n$. For every assignment s over the propositions $p_1, \ldots, p_n$, let φ_s be a PL-formula whose only satisfying assignment is s. This exists by Proposition 12.

Expressivity of PL with team semantics

Proposition 18

For every flat family $\mathcal{T}$ there exists a PL-formula φ such that $\mathcal{T} = \mathrm{Teams}(\varphi)$.

Proof.

Let $\mathcal{T}$ be a flat family of teams using proposition symbols $p_1, \ldots, p_n$. For every assignment s over the propositions $p_1, \ldots, p_n$, let φ_s be a PL-formula whose only satisfying assignment is s. This exists by Proposition 12. We will then define

$$\Phi := \bigvee_{s \in \bigcup \mathcal{T}} \varphi_s$$

and claim that $\mathcal{T} = \mathrm{Teams}(\Phi)$.

Expressivity of PL with team semantics

Proposition 18

For every flat family $\mathcal{T}$ there exists a PL-formula φ such that $\mathcal{T} = \mathrm{Teams}(\varphi)$.

Proof.

Let $\mathcal{T}$ be a flat family of teams using proposition symbols $p_1, \ldots, p_n$. For every assignment s over the propositions $p_1, \ldots, p_n$, let φ_s be a PL-formula whose only satisfying assignment is s. This exists by Proposition 12. We will then define

$$\Phi := \bigvee_{s \in \bigcup \mathcal{T}} \varphi_s$$

and claim that $\mathcal{T} = \mathrm{Teams}(\Phi)$. It is easy to check that $T \models \Phi$ if and only if $T \subseteq \bigcup \mathcal{T}$.

Expressivity of PL with team semantics

Proposition 18

For every flat family $\mathcal{T}$ there exists a PL-formula φ such that $\mathcal{T} = \mathrm{Teams}(\varphi)$.

Proof.

Let $\mathcal{T}$ be a flat family of teams using proposition symbols $p_1, \ldots, p_n$. For every assignment s over the propositions $p_1, \ldots, p_n$, let φ_s be a PL-formula whose only satisfying assignment is s. This exists by Proposition 12. We will then define

$$\Phi := \bigvee_{s \in \bigcup \mathcal{T}} \varphi_s$$

and claim that $\mathcal{T} = \mathrm{Teams}(\Phi)$. It is easy to check that $T \models \Phi$ if and only if $T \subseteq \bigcup \mathcal{T}$. By Proposition 16, the latter holds if and only if $T \in \mathcal{T}$. □

Expressivity of PL with team semantics

Proposition 18

For every flat family $\mathcal{T}$ there exists a PL-formula φ such that $\mathcal{T} = \mathrm{Teams}(\varphi)$.

Proof.

Let $\mathcal{T}$ be a flat family of teams using proposition symbols $p_1, \ldots, p_n$. For every assignment s over the propositions $p_1, \ldots, p_n$, let φ_s be a PL-formula whose only satisfying assignment is s. This exists by Proposition 12. We will then define

$$\Phi := \bigvee_{s \in \bigcup \mathcal{T}} \varphi_s$$

and claim that $\mathcal{T} = \mathrm{Teams}(\Phi)$. It is easy to check that $T \models \Phi$ if and only if $T \subseteq \bigcup \mathcal{T}$. By Proposition 16, the latter holds if and only if $T \in \mathcal{T}$. □

Theorem 19

A family of teams is definable in PL if and only if the family is flat.

Expressivity: the downward closed case

Let's consider an extension $\mathrm{PL}[\varovee]$ of PL with the so-called Boolean disjunction

$$T \models \varphi \varovee \psi \text{ if and only if } T \models \varphi \text{ or } T \models \psi.$$

$T \models p \varovee \neg p$

Proposition 20

$\mathrm{PL}[\varovee]$ *is downward closed and has the empty team property.*

$T \models p \vee \neg p$

Expressivity: the downward closed case

Let's consider an extension $\mathrm{PL}[\varovee]$ of PL with the so-called Boolean disjunction

$$T \models \varphi \varovee \psi \text{ if and only if } T \models \varphi \text{ or } T \models \psi.$$

Proposition 20

$\mathrm{PL}[\varovee]$ *is downward closed and has the empty team property.* PL(DEP)

It is easy to note that dependence atoms can be expressed in $\mathrm{PL}[\varovee]$:

$$T \models \mathrm{dep}(p_1, \ldots, p_n, q) \text{ if and only if } T \models \bigvee_{b \in \{\bot, \top\}^n} \left(p_1^{b_1} \wedge \cdots \wedge p_n^{b_n} \wedge (q \varovee \neg q)\right),$$

where $p^\bot := \neg p$ and $p^\top := p$.

Expressive power of $\mathrm{PL}[\varovee]$

Theorem 21

A family of teams is definable in $\mathrm{PL}[\varovee]$ if and only if the family is downward closed and includes the empty team.

Expressive power of $\mathrm{PL}[\varovee]$

Theorem 21

A family of teams is definable in $\mathrm{PL}[\varovee]$ *if and only if the family is downward closed and includes the empty team.*

Proof.

Let $\mathcal{T}$ be a family of teams with the aforementioned properties. Define

$$\Phi := \bigovee_{T \in \mathcal{T}} \bigvee_{s \in T} \varphi_s, \text{ where } \varphi_s \text{ is a formula whose only satisfying assignment is } s.$$

$s \models \Phi$

We claim that $\mathrm{Teams}(\Phi) = \mathcal{T}$.

Expressive power of $\mathrm{PL}[\varovee]$

Theorem 21

A family of teams is definable in $\mathrm{PL}[\varovee]$ if and only if the family is downward closed and includes the empty team.

Proof.

Let $\mathcal{T}$ be a family of teams with the aforementioned properties. Define

$$\Phi := \bigovee_{T \in \mathcal{T}} \bigvee_{s \in T} \varphi_s, \text{ where } \varphi_s \text{ is a formula whose only satisfying assignment is } s.$$

We claim that $\mathrm{Teams}(\Phi) = \mathcal{T}$. If $S \models \Phi$, there is some $T \in \mathcal{T}$ s.t. $S \models \bigvee_{s \in T} \varphi_s$.

Expressive power of $\mathrm{PL}[\varovee]$

Theorem 21

A family of teams is definable in $\mathrm{PL}[\varovee]$ *if and only if the family is downward closed and includes the empty team.*

Proof.

Let $\mathcal{T}$ be a family of teams with the aforementioned properties. Define

$$\Phi := \bigovee_{T \in \mathcal{T}} \bigvee_{s \in T} \varphi_s, \text{ where } \varphi_s \text{ is a formula whose only satisfying assignment is } s.$$

We claim that $\mathrm{Teams}(\Phi) = \mathcal{T}$. If $S \models \Phi$, there is some $T \in \mathcal{T}$ s.t. $S \models \bigvee_{s \in T} \varphi_s$. Thus $S \subseteq T$ and hence $S \in \mathcal{T}$, for $\mathcal{T}$ is downward closed.

Expressive power of $\mathrm{PL}[\varovee]$

Theorem 21

A family of teams is definable in $\mathrm{PL}[\varovee]$ *if and only if the family is downward closed and includes the empty team.*

Proof.

Let $\mathcal{T}$ be a family of teams with the aforementioned properties. Define

$$\Phi := \bigovee_{T \in \mathcal{T}} \bigvee_{s \in T} \varphi_s, \text{ where } \varphi_s \text{ is a formula whose only satisfying assignment is } s.$$

We claim that $\mathrm{Teams}(\Phi) = \mathcal{T}$. If $S \models \Phi$, there is some $T \in \mathcal{T}$ s.t. $S \models \bigvee_{s \in T} \varphi_s$. Thus $S \subseteq T$ and hence $S \in \mathcal{T}$, for $\mathcal{T}$ is downward closed. Conversely, if $S \in \mathcal{T}$ then $S \models \bigvee_{s \in S} \varphi_s$, and thus $S \models \Phi$. □

Expressive power of $\mathrm{PL}[\varovee]$

Theorem 21

A family of teams is definable in $\mathrm{PL}[\varovee]$ if and only if the family is downward closed and includes the empty team.

Proof.

Let $\mathcal{T}$ be a family of teams with the aforementioned properties. Define

$$\Phi := \bigvarovee_{T \in \mathcal{T}} \bigvee_{s \in T} \varphi_s, \text{ where } \varphi_s \text{ is a formula whose only satisfying assignment is } s.$$

We claim that $\mathrm{Teams}(\Phi) = \mathcal{T}$. If $S \models \Phi$, there is some $T \in \mathcal{T}$ s.t. $S \models \bigvee_{s \in T} \varphi_s$. Thus $S \subseteq T$ and hence $S \in \mathcal{T}$, for $\mathcal{T}$ is downward closed. Conversely, if $S \in \mathcal{T}$ then $S \models \bigvee_{s \in S} \varphi_s$, and thus $S \models \Phi$. □

Can you make the formula a bit shorter?

Types and characterising formulae

We define some auxiliary notation and formulae:

- $\mathrm{Type}_{\Psi}(s) := \{\varphi \in \Psi \mid s \models \varphi\}$, for a set of PL-formulae Ψ and an assignment s.
- For $\Gamma \subseteq \Psi$, define $\theta_{\Gamma} := \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi \in \Psi \setminus \Gamma} \neg\psi$,

It is easy to check that $\mathrm{Type}_{\Psi}(s) = \Gamma$ if and only if $s \models \theta_{\Gamma}$.

Types and characterising formulae

We define some auxiliary notation and formulae:

- $\mathrm{Type}_\Psi(s) := \{\varphi \in \Psi \mid s \models \varphi\}$, for a set of PL-formulae Ψ and an assignment s.
- For $\Gamma \subseteq \Psi$, define $\theta_\Gamma := \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi \in \Psi \setminus \Gamma} \neg\psi$,

It is easy to check that $\mathrm{Type}_\Psi(s) = \Gamma$ if and only if $s \models \theta_\Gamma$.

- $\mathrm{Type}_\Psi(T) := \{\mathrm{Type}_\Psi(s) \mid s \in T\}$, for a team T.

Types and characterising formulae

We define some auxiliary notation and formulae:

- $\mathrm{Type}_\Psi(s) := \{\varphi \in \Psi \mid s \models \varphi\}$, for a set of PL-formulae Ψ and an assignment s.
- For $\Gamma \subseteq \Psi$, define $\theta_\Gamma := \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi \in \Psi \setminus \Gamma} \neg\psi$,

It is easy to check that $\mathrm{Type}_\Psi(s) = \Gamma$ if and only if $s \models \theta_\Gamma$.

- $\mathrm{Type}_\Psi(T) := \{\mathrm{Type}_\Psi(s) \mid s \in T\}$, for a team T.

Lemma 22

Assume that T and S be teams and let Ψ be a finite set of PL-formulae.

1. *For each $\psi \in \Psi$, $T \models \psi$ if and only if $\psi \in \bigcap \mathrm{Type}_\Psi(T)$.*
2. *If $T \models \bigovee \Psi$ and $\mathrm{Type}_\Psi(S) \subseteq \mathrm{Type}_\Psi(T)$, then $S \models \bigovee \Psi$.*

Case 1. follows by flatness of PL,

Types and characterising formulae

We define some auxiliary notation and formulae:

- $\mathrm{Type}_\Psi(s) \coloneqq \{\varphi \in \Psi \mid s \models \varphi\}$, for a set of PL-formulae Ψ and an assignment s.
- For $\Gamma \subseteq \Psi$, define $\theta_\Gamma \coloneqq \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi \in \Psi \setminus \Gamma} \neg\psi$,

It is easy to check that $\mathrm{Type}_\Psi(s) = \Gamma$ if and only if $s \models \theta_\Gamma$.

- $\mathrm{Type}_\Psi(T) \coloneqq \{\mathrm{Type}_\Psi(s) \mid s \in T\}$, for a team T.

Lemma 22

Assume that T and S be teams and let Ψ be a finite set of PL-formulae.

1. *For each $\psi \in \Psi$, $T \models \psi$ if and only if $\psi \in \bigcap \mathrm{Type}_\Psi(T)$.*
2. *If $T \models \bigovee \Psi$ and $\mathrm{Type}_\Psi(S) \subseteq \mathrm{Type}_\Psi(T)$, then $S \models \bigovee \Psi$.*

Case 1. follows by flatness of PL, and 2. uses 1. together with the definition of $\ovee$. Intuitively, it follows due to downward closure.

Consider next the formula stating that the truth value w.r.t. a set of propositions $\Psi \subseteq \text{PROP}$ is constant:

$$\gamma := \bigwedge_{p \in \Psi} \text{dep}(p).$$

$S \models \gamma \vee \gamma$

Hence $T \models \gamma$ if and only if $|\text{Type}_{\Psi}(T)| \leq 1$.

Expressive power of $\mathrm{PL}[\mathrm{dep}]$

Consider next the formula stating that the truth value w.r.t. a set of propositions $\Psi \subseteq \mathrm{PROP}$ is constant:

$$\gamma := \bigwedge_{p \in \Psi} \mathrm{dep}(p).$$

Hence $T \models \gamma$ if and only if $|\mathrm{Type}_{\Psi}(T)| \leq 1$. Define now recursively

$$\gamma^0 := p \wedge \neg p, \qquad \gamma^{k+1} := (\gamma^k \vee \gamma).$$

It is easy to show by induction that $T \models \gamma^k$ if and only if $|\mathrm{Type}_{\Psi}(T)| \leq k$.

Lemma 23

If $\Psi \subseteq \mathrm{PROP}$ *is a finite set of propositions and* $T \neq \emptyset$ *a team, there is a* $\xi_T \in \mathrm{PL}[\mathrm{dep}]$ *s.t. for every* S

$$S \models \xi_T \iff \mathrm{Type}_{\Psi}(T) \not\subseteq \mathrm{Type}_{\Psi}(S).$$

Lemma 23

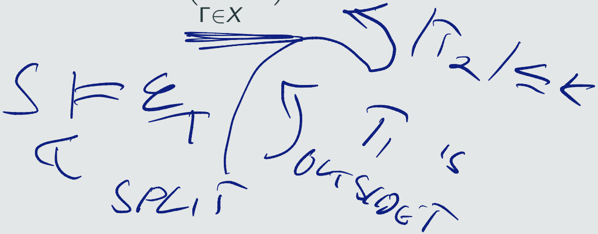
If $\Psi \subseteq \text{PROP}$ *is a finite set of propositions and* $T \neq \emptyset$ *a team, there is a* $\xi_T \in \mathrm{PL}[\mathrm{dep}]$ *s.t. for every* S

$$S \models \xi_T \iff \mathrm{Type}_\Psi(T) \not\subseteq \mathrm{Type}_\Psi(S).$$

Proof.

Let $|\mathrm{Type}_\Psi(T)| = k + 1$. Recall θ_Γ is a characterisic formula of Γ. We define

$$\xi_T := \left(\bigvee_{\Gamma \in X} \theta_\Gamma \right) \vee \gamma^k, \text{ where } X = \mathcal{P}(\Psi) \setminus \mathrm{Type}_\Psi(T).$$



Lemma 23

If $\Psi \subseteq \mathrm{PROP}$ *is a finite set of propositions and* $T \neq \emptyset$ *a team, there is a* $\xi_T \in \mathrm{PL}[\mathrm{dep}]$ *s.t. for every* S

$$S \models \xi_T \iff \mathrm{Type}_\Psi(T) \not\subseteq \mathrm{Type}_\Psi(S).$$

Proof.

Let $|\mathrm{Type}_\Psi(T)| = k+1$. Recall θ_Γ is a characterisic formula of Γ. We define

$$\xi_T := \left(\bigvee_{\Gamma \in X} \theta_\Gamma \right) \vee \gamma^k, \text{ where } X = \mathcal{P}(\Psi) \setminus \mathrm{Type}_\Psi(T).$$

Now given a team S we have

$$\begin{aligned} S \models \xi_T &\iff \text{there are } T_1, T_2 \text{ s.t. } T_1 \cup T_2 = S, \mathrm{Type}_\Psi(T_1) \subseteq X, |\mathrm{Type}_\Psi(T_2)| \leq k \\ &\iff |\mathrm{Type}_\Psi(T) \cap \mathrm{Type}_\Psi(S)| \leq k \\ &\iff \mathrm{Type}_\Psi(T) \not\subseteq \mathrm{Type}_\Psi(S). \quad \square \end{aligned}$$

Theorem 24

$\mathrm{PL}[\varovee]$ *is equi-expressive with* $\mathrm{PL}[\mathrm{dep}]$.

Theorem 24

$\mathrm{PL}[\varovee]$ *is equi-expressive with* $\mathrm{PL}[\mathrm{dep}]$.

Proof.

$\mathrm{PL}[\varovee] \leq \mathrm{PL}[\mathrm{dep}]$ direction: Let $\varphi = \bigcirc\!\!\!\!\vee\, \Psi$ be a $\mathrm{PL}[\varovee]$-formula in a normal form, where $\Psi \subseteq \mathrm{PL}$. Define

$$\eta := \bigwedge_{T \notin \mathrm{Teams}(\varphi)} \xi_T, \text{ where } \xi_T \text{ is as in Lemma 23.}$$

Intuitively $S \models \eta$ iff no falsifying team of φ is completely subsumed by S.

Theorem 24

$\mathrm{PL}[\varovee]$ *is equi-expressive with* $\mathrm{PL}[\mathrm{dep}]$.

Proof.

$\mathrm{PL}[\varovee] \leq \mathrm{PL}[\mathrm{dep}]$ direction: Let $\varphi = \bigovee \Psi$ be a $\mathrm{PL}[\varovee]$-formula in a normal form, where $\Psi \subseteq \mathrm{PL}$. Define

$$\eta := \bigwedge_{T \notin \mathrm{Teams}(\varphi)} \xi_T, \text{ where } \xi_T \text{ is as in Lemma 23.}$$

Intuitively $S \models \eta$ iff no falsifying team of φ is completely subsumed by S.
By definition η is a $\mathrm{PL}[\mathrm{dep}]$-formula. To prove that $\mathrm{Teams}(\eta) = \mathrm{Teams}(\varphi)$, assume first that $S \in \mathrm{Teams}(\varphi)$, and consider any $T \notin \mathrm{Teams}(\varphi)$.

Theorem 24

$\mathrm{PL}[\varovee]$ *is equi-expressive with* $\mathrm{PL}[\mathrm{dep}]$.

Proof.

$\mathrm{PL}[\varovee] \leq \mathrm{PL}[\mathrm{dep}]$ direction: Let $\varphi = \bigvee\!\!\!\!\bigcirc \Psi$ be a $\mathrm{PL}[\varovee]$-formula in a normal form, where $\Psi \subseteq \mathrm{PL}$. Define

$$\eta := \bigwedge_{T \notin \mathrm{Teams}(\varphi)} \xi_T, \text{ where } \xi_T \text{ is as in Lemma 23.}$$

Intuitively $S \models \eta$ iff no falsifying team of φ is completely subsumed by S.
By definition η is a $\mathrm{PL}[\mathrm{dep}]$-formula. To prove that $\mathrm{Teams}(\eta) = \mathrm{Teams}(\varphi)$, assume first that $S \in \mathrm{Teams}(\varphi)$, and consider any $T \notin \mathrm{Teams}(\varphi)$. It follows from Lemma 22 that $\mathrm{Type}_\Psi(T) \not\subseteq \mathrm{Type}_\Psi(S)$. Hence by Lemma 23, $S \models \xi_T$. Thus $S \in \mathrm{Teams}(\eta)$.

Theorem 24

$\mathrm{PL}[\varovee]$ *is equi-expressive with* $\mathrm{PL}[\mathrm{dep}]$.

Proof.

$\mathrm{PL}[\varovee] \leq \mathrm{PL}[\mathrm{dep}]$ direction: Let $\varphi = \bigcirc\!\!\!\!\vee\, \Psi$ be a $\mathrm{PL}[\varovee]$-formula in a normal form, where $\Psi \subseteq \mathrm{PL}$. Define

$$\eta := \bigwedge_{T \notin \mathrm{Teams}(\varphi)} \xi_T, \text{ where } \xi_T \text{ is as in Lemma 23.}$$

Intuitively $S \models \eta$ iff no falsifying team of φ is completely subsumed by S.
By definition η is a $\mathrm{PL}[\mathrm{dep}]$-formula. To prove that $\mathrm{Teams}(\eta) = \mathrm{Teams}(\varphi)$, assume first that $S \in \mathrm{Teams}(\varphi)$, and consider any $T \notin \mathrm{Teams}(\varphi)$. It follows from Lemma 22 that $\mathrm{Type}_\Psi(T) \not\subseteq \mathrm{Type}_\Psi(S)$. Hence by Lemma 23, $S \models \xi_T$. Thus $S \in \mathrm{Teams}(\eta)$.

Assume then that $S \notin \mathrm{Teams}(\varphi)$. Since $\mathrm{Type}_\Psi(S) \subseteq \mathrm{Type}_\Psi(S)$, it follows from Lemma 23 that $S \not\models \xi_S$. Thus $S \notin \mathrm{Teams}(\eta)$. □

Dimensions of team families

Definition 25

The lower dimension $\dim(\varphi)$ of a formula φ to is the least n such that

$$T \models \varphi \iff S \models \varphi \text{ for all } S \subseteq T \text{ s.t. } |S| \leq n.$$

The lower dimension of a flat formula is 1, and for a dependence atom it is 2. The lower dimension is not easy to approximate compositionally,

Dimensions of team families

Definition 25

The lower dimension $\dim(\varphi)$ of a formula φ to is the least n such that

$$T \models \varphi \iff S \models \varphi \text{ for all } S \subseteq T \text{ s.t. } |S| \leq n.$$

The lower dimension of a flat formula is 1, and for a dependence atom it is 2. The lower dimension is not easy to approximate compositionally, for that we define the notion of upper dimension. Define $M(\varphi)$ as the set of subset maximal teams satisfying φ.

Definition 26

The upper dimension $\mathrm{Dim}(\varphi)$ of a formula φ is the cardinality of $M(\varphi)$.

Interestingly, $\mathrm{Dim}(\varphi)$ can be given sharp compositional estimates, and it can be shown that $\dim(\varphi) \leq \mathrm{Dim}(\varphi)$.

Estimates for the upper dimension

Lemma 27

We have the following upper dimension estimates for $\varphi, \psi \in \mathrm{PL}[\mathbin{\varovee}]$:

1. $\mathsf{Dim}(p) = \mathsf{Dim}(\neg p) = 1$.
2. $\mathsf{Dim}(\varphi \wedge \psi) \leq \mathsf{Dim}(\varphi)\,\mathsf{Dim}(\psi)$.
3. $\mathsf{Dim}(\varphi \vee \psi) \leq \mathsf{Dim}(\varphi)\,\mathsf{Dim}(\psi)$.
4. $\mathsf{Dim}(\varphi \mathbin{\varovee} \psi) \leq \mathsf{Dim}(\varphi) + \mathsf{Dim}(\psi)$.

Proof.

We omit the cases for (1) and (3), since (1) is trivial, and (3) is analogous to (2).

Estimates for the upper dimension

Lemma 27

We have the following upper dimension estimates for $\varphi, \psi \in \mathrm{PL}[\varovee]$*:*

1. $\mathsf{Dim}(p) = \mathsf{Dim}(\neg p) = 1$.
2. $\mathsf{Dim}(\varphi \wedge \psi) \leq \mathsf{Dim}(\varphi)\,\mathsf{Dim}(\psi)$.
3. $\mathsf{Dim}(\varphi \vee \psi) \leq \mathsf{Dim}(\varphi)\,\mathsf{Dim}(\psi)$.
4. $\mathsf{Dim}(\varphi \varovee \psi) \leq \mathsf{Dim}(\varphi) + \mathsf{Dim}(\psi)$.

Proof.

We omit the cases for (1) and (3), since (1) is trivial, and (3) is analogous to (2). We defer the proof of (2) to the lecture notes.
Case (4): For the Boolean disjunction, it holds that

$$M(\varphi \varovee \psi) \subseteq M(\varphi) \cup M(\psi)$$

and the right-hand side of the inclusion generates the family $\mathrm{Teams}(\varphi \varovee \psi)$. The dimension estimate follows immediately. □

What are dimensions good for?

Proposition 28

$\mathrm{Dim}(\mathrm{dep}(p_1, \ldots, p_n, q)) = 2^{2^n}$.

Proposition 29

For $\varphi \in \mathrm{PL}[\oslash]$, $\mathrm{Dim}(\varphi) \leq 2^k$, where k is the number of occurrences of $\oslash$ in φ.

What are dimensions good for?

Proposition 28

$\text{Dim}(\mathrm{dep}(p_1, \ldots, p_n, q)) = 2^{2^n}$.

Proposition 29

For $\varphi \in \mathrm{PL}[\varovee]$, $\text{Dim}(\varphi) \leq 2^k$, *where* k *is the number of occurrences of* $\varovee$ *in* φ.

Theorem 30

Let $\varphi \in \mathrm{PL}[\varovee]$ *such that* $\mathrm{Teams}(\varphi) = \mathrm{Teams}(\mathrm{dep}(p_1, \ldots, p_n, q))$. *Then* φ *contains more than* 2^n *symbols.*

Proof.

By Prop 28, $\text{Dim}(\varphi) = \text{Dim}(\mathrm{dep}(p_1, \ldots, p_n, q)) = 2^{2^n}$. Thus $2^{2^n} \leq 2^{\mathrm{occ}_{\varovee}(\varphi)}$ by Prop. 29, implying $2^n \leq \mathrm{occ}_{\varovee}(\varphi)$. Hence φ has at least 2^n Boolean disjunctions. □

What are dimensions good for?

Proposition 28

$\mathrm{Dim}(\mathrm{dep}(p_1, \ldots, p_n, q)) = 2^{2^n}$.

Proposition 29

For $\varphi \in \mathrm{PL}[\varovee]$, $\mathrm{Dim}(\varphi) \leq 2^k$, *where* k *is the number of occurrences of* $\varovee$ *in* φ.

Theorem 30

Let $\varphi \in \mathrm{PL}[\varovee]$ *such that* $\mathrm{Teams}(\varphi) = \mathrm{Teams}(\mathrm{dep}(p_1, \ldots, p_n, q))$. *Then* φ *contains more than* 2^n *symbols.*

Proof.

By Prop 28, $\mathrm{Dim}(\varphi) = \mathrm{Dim}(\mathrm{dep}(p_1, \ldots, p_n, q)) = 2^{2^n}$. Thus $2^{2^n} \leq 2^{\mathrm{occ}_{\varovee}(\varphi)}$ by Prop. 29, implying $2^n \leq \mathrm{occ}_{\varovee}(\varphi)$. Hence φ has at least 2^n Boolean disjunctions. □

Thus, any translation from $\mathrm{PL}[\mathrm{dep}]$ to $\mathrm{PL}[\varovee]$ leads to an exponential blow-up.

Expressivity of propositional inclusion logic

Theorem 31

A family of teams is definable in $\mathrm{PL}[\subseteq]$ *if and only if it is union closed and includes the empty team.*

Proof.

We will omit the proof, which combines ideas from the characterisation of $\mathrm{PL}[\varovee]$ and its equivalence with $\mathrm{PL}[\mathrm{dep}]$. The result was first shown in [HS15]. □

Conclusion of Lecture 2

- Properties of families of teams.
- Expressivity characterisation of $\mathrm{PL}[\varovee]$.
- Equivalence of $\mathrm{PL}[\varovee]$ and $\mathrm{PL}[\mathrm{dep}]$.
- Expressivity characterisation of $\mathrm{PL}[\subseteq]$.

Complexity and Expressivity of Propositional Logics with Team Semantics
Arne Meier, Jonni Virtema
7th of August

Lecture 3: Inclusion Logic

Literature: [Hel+19; Hel+20]

[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.

Bibliography iii

[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`.