Efficient program transformers for translating LCC *to* PDL

P. Pardo E. Sarrión Morillo
F. Soler Toscano F. Velázquez Quesada

Grupo de Investigación en Lógica, Lenguaje e Información
Universidad de Sevilla
{ppardo1, esarrion, fsoler, frvelazquezquesada}@us.es

Workshop "Logic, Language and Information"
November 4th 2014, Málaga

Outline

Outline

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

is a multi-agent dynamic epistemic logic (*DEL*)

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

is a multi-agent dynamic epistemic logic (*DEL*)

. . . with public/private/secret { communication among agents / observation / factic change

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

is a multi-agent dynamic epistemic logic (*DEL*)

. . . with public/private/secret { communication among agents / observation / factic change

- Verification of (secure) communication protocols in *DEL* (e.g. Russian Cards Problems, Muddy Children Problem).

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

is a multi-agent dynamic epistemic logic (*DEL*)

. . . with public/private/secret { communication among agents / observation / factic change }

- Verification of (secure) communication protocols in *DEL* (e.g. Russian Cards Problems, Muddy Children Problem).
- Generation of plans or protocols in *DEL* planning.

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

- Validity checking in LCC makes use of:

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

- Validity checking in LCC makes use of:
  - A translation of LCC to PDL (which requires program transformers).

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

- Validity checking in LCC makes use of:
  - A translation of LCC to PDL (which requires program transformers).
  - Some PDL checker.

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

- Validity checking in LCC makes use of:
  - A translation of LCC to PDL (which requires program transformers).
  - Some PDL checker.
- The translation is also used to generate a complete set of reduction of axioms, together with those of PDL.

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

- Validity checking in LCC makes use of:
  - A translation of LCC to PDL (which requires program transformers).
  - Some PDL checker.
- The translation is also used to generate a complete set of reduction of axioms, together with those of PDL.
- (LCC 2006) uses an inefficient translation based on Kleene's translation of *finite automata* to *regular languages*.

Introduction

Logic of communication and change (LCC)

[J. van Benthem, B. Kooi and J. van Eijck, 2006]

- Validity checking in LCC makes use of:
  - A translation of LCC to PDL (which requires program transformers).
  - Some PDL checker.
- The translation is also used to generate a complete set of reduction of axioms, together with those of PDL.
- (LCC 2006) uses an inefficient translation based on Kleene's translation of *finite automata* to *regular languages*.
- Our proposal: a new translation with lower complexity based on a matrix treatment of Brzozowski's equational method.

Outline

LCC models

Let Var $= \{p, q, \ldots\}$ and Ag $= \{a, b, \ldots\}$ be sets of atoms and agents.

LCC models

Let $\mathrm{Var} = \{p, q, \ldots\}$ and $\mathrm{Ag} = \{a, b, \ldots\}$ be sets of atoms and agents.

Definition (Epistemic model)

A triple $M = (W, \langle R_a \rangle_{a \in \mathrm{Ag}}, V)$ with:

Example (Agents *b* and *c* know that *a* knows whether *p*)

LCC models

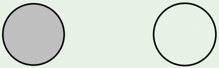
Let $\text{Var} = \{p, q, \ldots\}$ and $\text{Ag} = \{a, b, \ldots\}$ be sets of atoms and agents.

Definition (Epistemic model)

A triple $M = (W, \langle R_a \rangle_{a \in \text{Ag}}, V)$ with:

- A non-empty set of worlds $W = \{w_0, w_1, \ldots\}$.

Example (Agents *b* and *c* know that *a* knows whether *p*)



LCC models

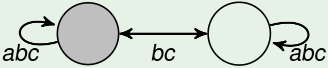
Let Var $= \{p, q, \ldots\}$ and Ag $= \{a, b, \ldots\}$ be sets of atoms and agents.

Definition (Epistemic model)

A triple $M = (W, \langle R_a \rangle_{a \in \mathrm{Ag}}, V)$ with:

- A non-empty set of worlds $W = \{w_0, w_1, \ldots\}$.
- An accessibility relations $R_a \subseteq W \times W$ for each agent a.

Example (Agents *b* and *c* know that *a* knows whether *p*)



LCC models

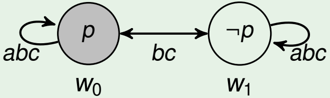
Let Var $= \{p, q, \ldots\}$ and Ag $= \{a, b, \ldots\}$ be sets of atoms and agents.

Definition (Epistemic model)

A triple $M = (W, \langle R_a \rangle_{a \in \mathrm{Ag}}, V)$ with:

- A non-empty set of worlds $W = \{w_0, w_1, \ldots\}$.
- An accessibility relations $R_a \subseteq W \times W$ for each agent a.
- A valuation $V : \mathrm{Var} \to \wp(W)$.

Example (Agents *b* and *c* know that *a* knows whether *p*)



LCC models

Definition (Action model)

For a language $\mathcal{L}$ upon Var and Ag that can be interpreted over relational models, a 4-tuple $\mathsf{U} = (\mathsf{E}, \langle \mathsf{R}_a \rangle_{a \in \mathsf{Ag}}, \mathsf{pre}, \mathsf{sub})$ with:

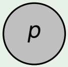
LCC models

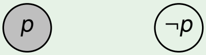
Definition (Action model)

For a language $\mathcal{L}$ upon Var and Ag that can be interpreted over relational models, a 4-tuple $\mathsf{U} = (\mathsf{E}, \langle \mathsf{R}_a \rangle_{a \in \mathsf{Ag}}, \mathsf{pre}, \mathsf{sub})$ with:

- $\mathsf{E} = \{\mathsf{e}_0, \ldots, \mathsf{e}_{n-1}\}$ is a finite non-empty set of actions.

Example (Public change to $\neg p$. | Private comm. by a to b about p.)





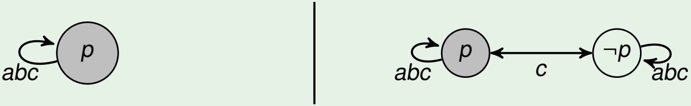
LCC models

Definition (Action model)

For a language $\mathcal{L}$ upon Var and Ag that can be interpreted over relational models, a 4-tuple $\mathsf{U} = (\mathsf{E}, \langle \mathsf{R}_a \rangle_{a \in \mathsf{Ag}}, \mathsf{pre}, \mathsf{sub})$ with:

- $\mathsf{E} = \{\mathsf{e}_0, \dots, \mathsf{e}_{n-1}\}$ is a finite non-empty set of actions.
- $\mathsf{R}_a \subseteq \mathsf{E} \times \mathsf{E}$ is an accessibility relation for each agent *a*.

Example (Public change to ¬*p*. | Private comm. by *a* to *b* about *p*.)



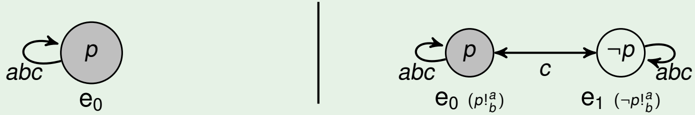
LCC models

Definition (Action model)

For a language $\mathcal{L}$ upon Var and Ag that can be interpreted over relational models, a 4-tuple $\mathsf{U} = (\mathsf{E}, \langle \mathsf{R}_a \rangle_{a \in \mathsf{Ag}}, \mathsf{pre}, \mathsf{sub})$ with:

- $\mathsf{E} = \{\mathsf{e}_0, \ldots, \mathsf{e}_{n-1}\}$ is a finite non-empty set of actions.
- $\mathsf{R}_a \subseteq \mathsf{E} \times \mathsf{E}$ is an accessibility relation for each agent a.
- $\mathsf{pre} : \mathsf{E} \to \mathcal{L}$ is a precondition map.

Example (Public change to $\neg p$. | Private comm. by a to b about p.)



LCC models

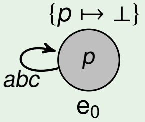
Definition (Action model)

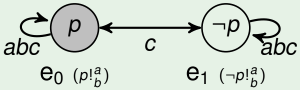
For a language $\mathcal{L}$ upon Var and Ag that can be interpreted over relational models, a 4-tuple $\mathsf{U} = (\mathsf{E}, \langle \mathsf{R}_a \rangle_{a \in \mathsf{Ag}}, \mathsf{pre}, \mathsf{sub})$ with:

- $\mathsf{E} = \{\mathsf{e}_0, \ldots, \mathsf{e}_{n-1}\}$ is a finite non-empty set of actions.
- $\mathsf{R}_a \subseteq \mathsf{E} \times \mathsf{E}$ is an accessibility relation for each agent a.
- $\mathsf{pre} : \mathsf{E} \to \mathcal{L}$ is a precondition map.
- $\mathsf{sub} : (\mathsf{E} \times \mathsf{Var}) \to \mathcal{L}$ is a postcondition map $(\mathsf{e}, p) \to \varphi$.

[Notation: $p^{\mathsf{sub}(\mathsf{e})} := sub(\mathsf{e}, p) = \varphi$.]

Example (Public change to $\neg p$. | Private comm. by a to b about p.)





LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathsf{U}, \mathsf{e}]\varphi$ for an LCC pointed action model (U, e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathsf{U}, \mathsf{e}]\varphi$$

$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathrm{U},\mathrm{e}]\varphi$ for an LCC pointed action model (U,e):

$$\begin{array}{lcl}\varphi & ::= & \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U},\mathrm{e}]\varphi \\ \pi & ::= & a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*\end{array}$$

Example (LCC modalities)

$[a]$ agent a knows/believes that ...

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathrm{U}, \mathrm{e}]\varphi$ for an LCC pointed action model (U, e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U}, \mathrm{e}]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

Example (LCC modalities)

$[a]$ agent a knows/believes that ...

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathsf{U}, \mathsf{e}]\varphi$ for an LCC pointed action model (U, e):

$$\begin{array}{lcl} \varphi & ::= & \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathsf{U}, \mathsf{e}]\varphi \\ \pi & ::= & a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \end{array}$$

Example (LCC modalities)

$[a]$	agent a knows/believes that ...
$[a;b]$	agent a knows/believes that b knows/believes that ...

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathrm{U},\mathrm{e}]\varphi$ for an LCC pointed action model (U,e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U},\mathrm{e}]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

Example (LCC modalities)

$[a]$	agent a knows/believes that ...
$[a;b]$	agent a knows/believes that b knows/believes that ...
$[a \cup b]$	both agents a, b know/believe that ...

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathsf{U}, \mathsf{e}]\varphi$ for an LCC pointed action model (U, e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathsf{U}, \mathsf{e}]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

Example (LCC modalities)

$[a]$	agent a knows/believes that ...
$[a;b]$	agent a knows/believes that b knows/believes that ...
$[a \cup b]$	both agents a, b know/believe that ...
$[(a \cup b)^*]$	it is common knowledge among a, b that ...

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathsf{U}, \mathsf{e}]\varphi$ for an LCC pointed action model (U, e):

$$\begin{array}{lcl} \varphi & ::= & \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathsf{U}, \mathsf{e}]\varphi \\ \pi & ::= & a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \end{array}$$

Example (LCC modalities)

$[a]$	agent a knows/believes that ...
$[a; b]$	agent a knows/believes that b knows/believes that ...
$[a \cup b]$	both agents a, b know/believe that ...
$[(a \cup b)^*]$	it is common knowledge among a, b that ...
$[(a \cup b); (a \cup b)^*]$	it is common belief among a, b that ...

LCC syntax

Definition (LCC language)

Extend PDL language with formula $[\mathsf{U},\mathsf{e}]\varphi$ for an LCC pointed action model (U,e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathsf{U},\mathsf{e}]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

Example (LCC modalities)

$[a]$	agent a knows/believes that ...
$[a;b]$	agent a knows/believes that b knows/believes that ...
$[a \cup b]$	both agents a, b know/believe that ...
$[(a \cup b)^*]$	it is common knowledge among a, b that ...
$[(a \cup b);(a \cup b)^*]$	it is common belief among a, b that ...
$[\mathsf{U},\mathsf{e}]$	after executing action e of U it necessarily holds that ...

LCC semantics

Definition (Language LCC)

Extend PDL language with formula $[\mathrm{U}, \mathrm{e}]\varphi$ for an LCC pointed action model (U, e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U}, \mathrm{e}]\varphi$$

$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

LCC semantics

Definition (Language LCC)

Extend PDL language with formula $[\mathrm{U}, \mathrm{e}]\varphi$ for an LCC pointed action model (U, e):

$$\begin{array}{lcl} \varphi & ::= & \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U}, \mathrm{e}]\varphi \\ \pi & ::= & a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \end{array}$$

Definition (Semantics of LCC formulas and LCC programs)

The function $||\_||^M$ for some LCC epistemic model $M = (W, \langle R_a \rangle_{a \in \mathrm{Ag}}, V)$ is:

LCC semantics

Definition (Language LCC)

Extend PDL language with formula $[\mathrm{U}, \mathrm{e}]\varphi$ for an LCC pointed action model (U, e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U}, \mathrm{e}]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

Definition (Semantics of LCC formulas and LCC programs)

The function $||\_||^M$ for some LCC epistemic model $M = (W, \langle R_a \rangle_{a \in \mathrm{Ag}}, V)$ is:

LCC semantics

Definition (Language LCC)

Extend PDL language with formula $[\mathrm{U},\mathrm{e}]\varphi$ for an LCC pointed action model (U,e):

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid [\mathrm{U},\mathrm{e}]\varphi$$
$$\pi \quad ::= \quad a \mid ?\varphi \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$

Definition (Semantics of LCC formulas and LCC programs)

The function $||\_||^M$ for some LCC epistemic model $M = (W, \langle R_a \rangle_{a \in \mathrm{Ag}}, V)$ is:

$$||\top||^M = W \qquad ||a||^M = R_a$$
$$||p||^M = V(p) \qquad ||?\varphi||^M = \mathrm{Id}_{||\varphi||^M}$$
$$||\neg\varphi||^M = W \setminus ||\varphi||^M \qquad ||\pi_1;\pi_2||^M = ||\pi_1||^M \circ ||\pi_2||^M$$
$$||\varphi_1 \wedge \varphi_2||^M = ||\varphi_1||^M \cap ||\varphi_2||^M \qquad ||\pi_1 \cup \pi_2||^M = ||\pi_1||^M \cup ||\pi_2||^M$$
$$||[\pi]\varphi||^M = \{w \in W \mid \forall v((w,v) \in ||\pi||^M \Rightarrow v \in ||\varphi||^M)\} \qquad ||\pi^*||^M = (||\pi||^M)^*$$
$$||[\mathrm{U},\mathrm{e}]\varphi||^M = \{w \in W \mid w \in ||\mathrm{pre}(\mathrm{e})||^M \Rightarrow (w,\mathrm{e}) \in ||\varphi||^{M \otimes \mathrm{U}}\}$$

LCC semantics

Definition (Update execution)

An epistemic model $M \otimes \mathrm{U} = (W^{M \otimes \mathrm{U}}, \langle R_a^{M \otimes \mathrm{U}} \rangle_{a \in \mathrm{Ag}}, V^{M \otimes \mathrm{U}})$ with:

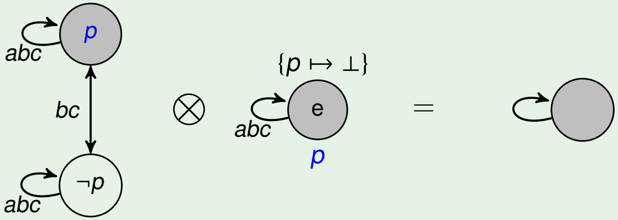
LCC semantics

Definition (Update execution)

An epistemic model $M \otimes \mathsf{U} = (W^{M\otimes\mathsf{U}}, \langle R_a^{M\otimes\mathsf{U}}\rangle_{a\in\mathsf{Ag}}, V^{M\otimes\mathsf{U}})$ with:

$$W^{M\otimes\mathsf{U}} \quad = \quad \text{the pairs } (w, \mathsf{e}) \text{ such that } M, w \models \mathsf{pre}(\mathsf{e})$$

Example (After a public change to $\neg p$, it is common knowl. that $\neg p$.)



LCC semantics

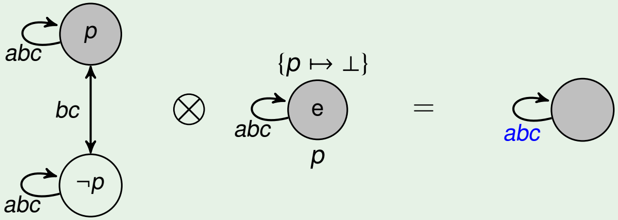
Definition (Update execution)

An epistemic model $M \otimes \mathrm{U} = (W^{M\otimes \mathrm{U}}, \langle R_a^{M\otimes \mathrm{U}} \rangle_{a\in \mathrm{Ag}}, V^{M\otimes \mathrm{U}})$ with:

$$W^{M\otimes \mathrm{U}} = \text{the pairs } (w, \mathrm{e}) \text{ such that } M, w \models \mathrm{pre}(\mathrm{e})$$

$$R_a^{M\otimes \mathrm{U}} = \text{the pairs } ((w, \mathrm{e}), (v, \mathrm{f})) \text{ such that } wR_a v \text{ and } \mathrm{e}\, \mathrm{R}_a \mathrm{f}$$

Example (After a public change to $\neg p$, it is common knowl. that $\neg p$.)



LCC semantics

Definition (Update execution)

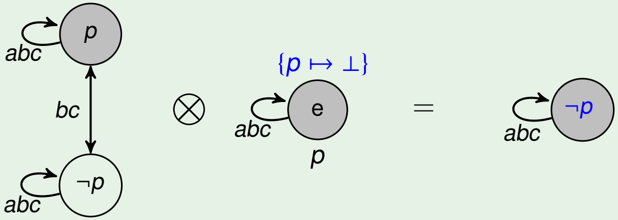
An epistemic model $M \otimes \mathsf{U} = (W^{M\otimes \mathsf{U}}, \langle R_a^{M\otimes \mathsf{U}} \rangle_{a\in \mathsf{Ag}}, V^{M\otimes \mathsf{U}})$ with:

$$W^{M\otimes \mathsf{U}} = \text{the pairs } (w, \mathsf{e}) \text{ such that } M, w \models \mathsf{pre}(\mathsf{e})$$

$$R_a^{M\otimes \mathsf{U}} = \text{the pairs } ((w, \mathsf{e}), (v, \mathsf{f})) \text{ such that } wR_a v \text{ and } \mathsf{e}\,\mathsf{R}_a \mathsf{f}$$

$$V^{M\otimes \mathsf{U}}(p) = \text{the pairs } (w, \mathsf{e}) \text{ such that } M, w \models p^{\mathsf{sub}(\mathsf{e})}$$

Example (After a public change to $\neg p$, it is common knowl. that $\neg p$.)



LCC axioms

Definition (LCC = PDL + reduction axioms for $[\mathsf{U},\mathsf{e}]$)

	Propositional tautologies		
(K)	$[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$	(top)	$[\mathsf{U},\mathsf{e}]\top \leftrightarrow \top$
$(test)$	$[?\varphi_1]\varphi_2 \leftrightarrow (\varphi_1 \to \varphi_2)$	$(atoms)$	$[\mathsf{U},\mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to p^{\mathsf{sub}(e)})$
] $(seq.)$	$[\pi_1;\pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$	$(neg.)$	$[\mathsf{U},\mathsf{e}]\neg\varphi \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to \neg[\mathsf{U},\mathsf{e}]\varphi)$
$(choice)$	$[\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$	$(conj.)$	$[\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2) \leftrightarrow ([\mathsf{U},\mathsf{e}]\varphi_1 \wedge [\mathsf{U},\mathsf{e}]\varphi_2)$
(mix)	$[\pi^*]\varphi \leftrightarrow \varphi \wedge [\pi][\pi^*]\varphi$	$(prog.)$	$[\mathsf{U},\mathsf{e}_i][\pi]\varphi \leftrightarrow \bigwedge_{j=0}^{n-1}[T_{ij}^{\mathsf{U}}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi$
$(ind.)$	$\varphi \wedge [\pi^*](\varphi \to [\pi]\varphi)) \to [\pi^*]\varphi$	(MP)	$\vdash \varphi_1$ and $\vdash \varphi_1 \to \varphi_2$ imply $\vdash \varphi_2$
(Nec_π)	$\vdash \varphi$ implies $\vdash [\pi]\varphi$.	(Nec_{U})	$\vdash \varphi$ implies $\vdash [\mathsf{U},\mathsf{e}]\varphi$

LCC program transformers [J. van Benthem et al., 2006]

Definition (Program transformers T^{U}_{ij})

Given some U with $\mathrm{E} = \{\mathrm{e}_0, \dots, \mathrm{e}_{n-1}\}$, the T^{U}_{ij} function ($0 \leq i, j \leq n-1$) is:

$$T^{\mathrm{U}}_{ij}(a) = \begin{cases} ?\mathrm{pre}(\mathrm{e}_i); a & \text{if } \mathrm{e}_i \mathrm{R}_a \mathrm{e}_j \\ ?\bot & \text{otherwise} \end{cases} \qquad T^{\mathrm{U}}_{ij}(?\varphi) = \begin{cases} ?(\mathrm{pre}(\mathrm{e}_i) \wedge [\mathrm{U}, \mathrm{e}_i]\varphi) & \text{if } i = j \\ ?\bot & \text{if } i \neq j \end{cases}$$

$$T^{\mathrm{U}}_{ij}(\pi_1; \pi_2) = \bigcup_{k=0}^{n-1} (T^{\mathrm{U}}_{ik}(\pi_1); T^{\mathrm{U}}_{kj}(\pi_2)) \qquad T^{\mathrm{U}}_{ij}(\pi_1 \cup \pi_2) = T^{\mathrm{U}}_{ij}(\pi_1) \cup T^{\mathrm{U}}_{ij}(\pi_2)$$

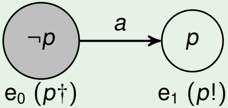
$$T^{\mathrm{U}}_{ij}(\pi^*) = K^{\mathrm{U}}_{ijn}(\pi) \quad \text{where } K^{\mathrm{U}}_{ijn} \text{ is inductively defined as:}$$

$$K^{\mathrm{U}}_{ij0}(\pi) = \begin{cases} ?\top \cup T^{\mathrm{U}}_{ij}(\pi) & \text{if } i = j \\ T^{\mathrm{U}}_{ij}(\pi) & \text{otherwise} \end{cases}$$

$$K^{\mathrm{U}}_{ij(k+1)}(\pi) = \begin{cases} (K^{\mathrm{U}}_{kkk}(\pi))^* & \text{if } i = k = j \\ (K^{\mathrm{U}}_{kkk}(\pi))^*; K^{\mathrm{U}}_{kjk}(\pi) & \text{if } i = k \neq j \\ K^{\mathrm{U}}_{ikk}(\pi); (K^{\mathrm{U}}_{kkk}(\pi))^* & \text{if } i \neq k = j \\ K^{\mathrm{U}}_{ijk}(\pi) \cup (K^{\mathrm{U}}_{ikk}(\pi); (K^{\mathrm{U}}_{kkk}(\pi))^*; K^{\mathrm{U}}_{kjk}(\pi)) & \text{if } i \neq k \neq j \end{cases}$$

LCC program transformers [J. van Benthem et al., 2006]

Example ($T^{\mathsf{U}}_{ij}(a)$)

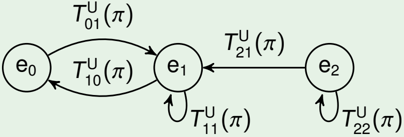


$[\mathsf{U}, \mathsf{e}_0][a]p$	path in $M \otimes \mathsf{U}$
$[T^{\mathsf{U}}_{00}(a)][\mathsf{U}, \mathsf{e}_0]p \wedge [T^{\mathsf{U}}_{01}(a)][\mathsf{U}, \mathsf{e}_1]p$	
$[?\bot][\mathsf{U}, \mathsf{e}_0]p \wedge [?\neg p; a][\mathsf{U}, \mathsf{e}_1]p$	path in M

LCC program transformers [J. van Benthem et al., 2006]

Example ($T^{\mathsf{U}}_{10}(\pi^*) = K^{\mathsf{U}}_{103}(\pi)$)

$$K^{\mathsf{U}}_{103}(\pi) = K^{\mathsf{U}}_{102}(\pi) \cup (K^{\mathsf{U}}_{122}(\pi) \,;\, (K^{\mathsf{U}}_{222}(\pi))^* \,;\, K^{\mathsf{U}}_{202}(\pi))$$



$$\begin{aligned} &= ((K^{\mathsf{U}}_{111}(\pi))^* \,;\, K^{\mathsf{U}}_{101}(\pi)) \cup \\ &((K^{\mathsf{U}}_{111}(\pi))^* \,;\, K^{\mathsf{U}}_{101}(\pi) \,; \\ &(K^{\mathsf{U}}_{221}(\pi) \cup (K^{\mathsf{U}}_{211}(\pi) \,;\, (K^{\mathsf{U}}_{111}(\pi))^* \,;\, K^{\mathsf{U}}_{121}(\pi)))^* \,; \\ &(K^{\mathsf{U}}_{201}(\pi) \cup (K^{\mathsf{U}}_{211}(\pi) \,;\, (K^{\mathsf{U}}_{111}(\pi)^* \,;\, K^{\mathsf{U}}_{101}(\pi)))) = \ldots \end{aligned}$$

Translation of $\mathcal{L}_{\mathrm{LCC}}$ to $\mathcal{L}_{\mathrm{PDL}}$ from (LCC 2006).

Definition (Translation functions t and r.)

$$
\begin{aligned}
t(\top) &= \top \\
t(p) &= p \\
t(\neg\varphi) &= \neg t(\varphi) \\
t(\varphi_1 \wedge \varphi_2) &= t(\varphi_1) \wedge t'(\varphi_2) \\
t([\pi]\varphi) &= [r(\pi)]t(\varphi) \\
t([\mathrm{U},\mathrm{e}]\top) &= \top \\
t([\mathrm{U},\mathrm{e}]p) &= t(\mathrm{pre}(\mathrm{e})) \to t(p^{\mathrm{sub}(\mathrm{e})}) \\
t([\mathrm{U},\mathrm{e}]\neg\varphi) &= t(\mathrm{pre}(\mathrm{e})) \to \neg t([\mathrm{U},\mathrm{e}]\varphi) \\
t([\mathrm{U},\mathrm{e}](\varphi_1 \wedge \varphi_2)) &= t([\mathrm{U},\mathrm{e}]\varphi) \wedge t([\mathrm{U},\mathrm{e}]\varphi_2) \\
t([\mathrm{U},\mathrm{e}_i][\pi]\varphi) &= \bigwedge_{j=0}^{n-1} [r(T_{ij}^{\mathrm{U}}(\pi))]t([\mathrm{U},\mathrm{e}_j]\varphi) \\
t([\mathrm{U},\mathrm{e}][\mathrm{U}',\mathrm{e}']\varphi) &= t([\mathrm{U},\mathrm{e}]t([\mathrm{U}',\mathrm{e}']\varphi))
\end{aligned}
$$

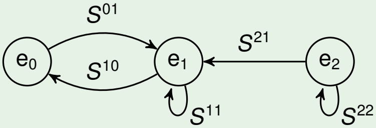
$$
\begin{aligned}
r(a) &= a \\
r(B) &= B \\
r(?\varphi) &= ?t(\varphi) \\
r(\pi_1;\pi_2) &= r(\pi_1); r(\pi_2) \\
r(\pi_1 \cup \pi_2) &= r(\pi_1) \cup r(\pi_2) \\
r(\pi^*) &= (r(\pi))^*
\end{aligned}
$$

Outline

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \rightarrow e_j$, denoted X^{ij})

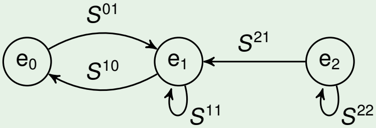
Let S^{ij} be the transformed direct π path $e_i \rightarrow e_j$ in U.



Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

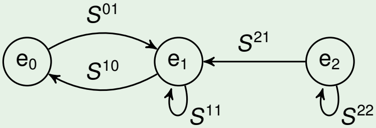
Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \rightarrow e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \rightarrow e_j$ in U.

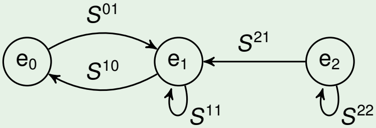


$$X^{00} = \quad ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \rightarrow e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \rightarrow e_j$ in U.

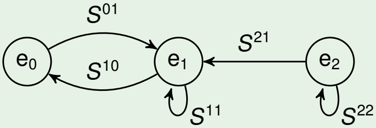


$$X^{00} = \ ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.

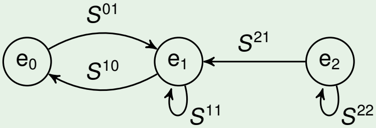


$$X^{00} = ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$
$$X^{20} = (S^{22}; X^{20}) \cup (S^{21}; X^{10})$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



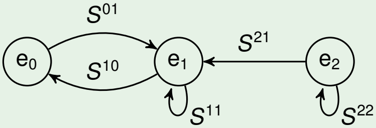
$$X^{00} = \ ?\text{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$
$$X^{20} = (S^{22}; X^{20}) \cup (S^{21}; X^{10})$$

Solve X^{ij} by substitution and Arden theorem: $X = AX \cup B \ \Rightarrow \ X = A^*B$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



$$X^{00} = \ ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$
$$X^{20} = (S^{22}; X^{20}) \cup (S^{21}; X^{10})$$

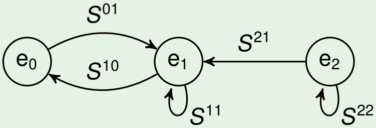
Solve X^{ij} by substitution and Arden theorem: $X = AX \cup B \ \Rightarrow \ X = A^*B$

$$X^{10} = (S^{10}; (?\mathrm{pre}(e_0) \cup (S^{01}; X^{10}))) \cup (S^{11}; X^{10})$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



$$X^{00} = \ ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$
$$X^{20} = (S^{22}; X^{20}) \cup (S^{21}; X^{10})$$

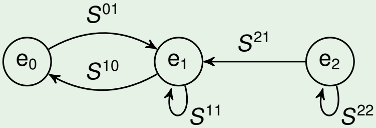
Solve X^{ij} by substitution and Arden theorem: $X = AX \cup B \ \Rightarrow \ X = A^*B$

$$X^{10} = (S^{10}; (?\mathrm{pre}(e_0) \cup (S^{01}; X^{10}))) \cup (S^{11}; X^{10})$$
$$= (S^{10}; ?\mathrm{pre}(e_0)) \cup (S^{10}; S^{01}; X^{10}) \cup (S^{11}; X^{10})$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



$$X^{00} = ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$
$$X^{20} = (S^{22}; X^{20}) \cup (S^{21}; X^{10})$$

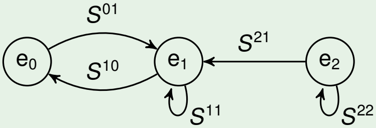
Solve X^{ij} by substitution and Arden theorem: $X = AX \cup B \;\Rightarrow\; X = A^*B$

$$\begin{aligned} X^{10} &= (S^{10}; (?\mathrm{pre}(e_0) \cup (S^{01}; X^{10}))) \cup (S^{11}; X^{10}) \\ &= (S^{10}; ?\mathrm{pre}(e_0)) \cup (S^{10}; S^{01}; X^{10}) \cup (S^{11}; X^{10}) \\ &= (S^{10}; ?\mathrm{pre}(e_0)) \cup (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \end{aligned}$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



$$X^{00} = \ ?\mathrm{pre}(e_0) \cup (S^{01}; X^{10})$$
$$X^{10} = (S^{10}; X^{00}) \cup (S^{11}; X^{10})$$
$$X^{20} = (S^{22}; X^{20}) \cup (S^{21}; X^{10})$$

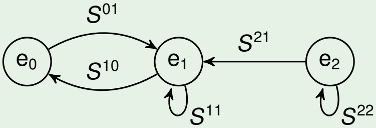
Solve X^{ij} by substitution and Arden theorem: $X = AX \cup B \ \Rightarrow \ X = A^*B$

$$\begin{aligned} X^{10} &= (S^{10}; (?\mathrm{pre}(e_0) \cup (S^{01}; X^{10}))) \cup (S^{11}; X^{10}) \\ &= (S^{10}; ?\mathrm{pre}(e_0)) \cup (S^{10}; S^{01}; X^{10}) \cup (S^{11}; X^{10}) \\ &= (S^{10}; ?\mathrm{pre}(e_0)) \cup (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \\ &= (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \cup (S^{10}; ?\mathrm{pre}(e_0)) \end{aligned}$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



$$
\begin{aligned}
X^{00} &= \ ?pre(e_0) \cup (S^{01}; X^{10}) \\
X^{10} &= (S^{10}; X^{00}) \cup (S^{11}; X^{10}) \\
X^{20} &= (S^{22}; X^{20}) \cup (S^{21}; X^{10})
\end{aligned}
$$

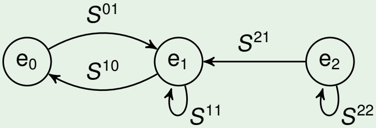
Solve X^{ij} by substitution and Arden theorem: $X = AX \cup B \;\Rightarrow\; X = A^*B$

$$
\begin{aligned}
X^{10} &= (S^{10}; (?pre(e_0) \cup (S^{01}; X^{10}))) \cup (S^{11}; X^{10}) \\
&= (S^{10}; ?pre(e_0)) \cup (S^{10}; S^{01}; X^{10}) \cup (S^{11}; X^{10}) \\
&= (S^{10}; ?pre(e_0)) \cup (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \\
&= (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \cup (S^{10}; ?pre(e_0)) \\
&= ((S^{10}; S^{01}) \cup S^{11})^*; S^{10}; ?pre(e_0) \qquad \text{(by Arden Theorem)}
\end{aligned}
$$

Brzozowski's equational method.

Example (The transformations of π^*-paths $e_i \to e_j$, denoted X^{ij})

Let S^{ij} be the transformed direct π path $e_i \to e_j$ in U.



$$
\begin{aligned}
X^{00} &= \;?\mathrm{pre}(e_0) \cup (S^{01}; X^{10}) \\
X^{10} &= (S^{10}; X^{00}) \cup (S^{11}; X^{10}) \\
X^{20} &= (S^{22}; X^{20}) \cup (S^{21}; X^{10})
\end{aligned}
$$

Solve X^{ij} by substitution and Arden theorem: $X = A X \cup B \;\Rightarrow\; X = A^* B$

$$
\begin{aligned}
X^{10} &= (S^{10}; (?\mathrm{pre}(e_0) \cup (S^{01}; X^{10}))) \cup (S^{11}; X^{10}) \\
&= (S^{10}; ?\mathrm{pre}(e_0)) \cup (S^{10}; S^{01}; X^{10}) \cup (S^{11}; X^{10}) \\
&= (S^{10}; ?\mathrm{pre}(e_0)) \cup (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \\
&= (((S^{10}; S^{01}) \cup S^{11}); X^{10}) \cup (S^{10}; ?\mathrm{pre}(e_0)) \\
&= ((S^{10}; S^{01}) \cup S^{11})^*; S^{10}; ?\mathrm{pre}(e_0) \qquad \text{(by Arden Theorem)}
\end{aligned}
$$

New program transformers $\mu^{\mathsf{U}}(\pi)[i,j]$

Definition (Program transformers for π-paths $\mathsf{e}_i \to \mathsf{e}_j$)

$$\mu^{\mathsf{U}}(a)[i,j] = \begin{cases} ?\mathsf{pre}(\mathsf{e}_i); a & \text{if } \mathsf{e}_i \mathsf{R}_a \mathsf{e}_j \\ ?\bot & \text{otherwise} \end{cases} \qquad \mu^{\mathsf{U}}(?\varphi)[i,j] = \begin{cases} ?(\mathsf{pre}(\mathsf{e}_i) \wedge [\mathsf{U}, \mathsf{e}_i]\varphi) & \text{if } i = j \\ ?\bot & \text{if } i \neq j \end{cases}$$

$$\mu^{\mathsf{U}}(\pi_1 \cup \pi_2)[i,j] = \bigoplus \left\{ \mu^{\mathsf{U}}(\pi_1)[i,j],\ \mu^{\mathsf{U}}(\pi_2)[i,j] \right\} \text{ where } \bigoplus\Gamma = \begin{cases} \bigcup (\Gamma \setminus \{?\bot\}) & \text{if } \varnothing \neq \Gamma \neq \{?\bot\} \\ ?\bot & \text{otherwise} \end{cases}$$

$$\mu^{\mathsf{U}}(\pi_1;\pi_2)[i,j] = \bigoplus \left\{ \mu^{\mathsf{U}}(\pi_1)[i,k] \odot \mu^{\mathsf{U}}(\pi_2)[k,j] \mid 0 \leq k \leq n-1 \right\}$$

$$\text{where } \sigma \odot \rho = \begin{cases} \sigma;\rho & \text{if } \sigma \neq ?\bot \neq \rho \\ ?\bot & \text{otherwise} \end{cases}$$

$$\mu^{\mathsf{U}}(\pi^*) = S_0^{\mathsf{U}}\left(\mu^{\mathsf{U}}(\pi) \mid A^{\mathsf{U}}\right)$$ defined next.

Program transformers $\mu^{\mathsf{U}}(\pi^*)[i,j]$

Definition ((cont'd))

$$\mu^{\mathsf{U}}(\pi^*) = S_0^{\mathsf{U}}(\mu^{\mathsf{U}}(\pi) \mid A^{\mathsf{U}}) \qquad \text{where}$$

$(\mu^{\mathsf{U}}(\pi) \mid A^{\mathsf{U}})$ is an $n \times 2n$ matrix with $A^{\mathsf{U}}[i,j] = \begin{cases} ?\mathrm{pre}(\mathrm{e}_i) & \text{if } i = j \\ ?\bot & \text{otherwise} \end{cases}$

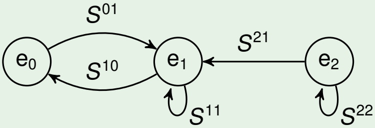
$S_n^{\mathsf{U}}(M \mid A) = A$ and $S_k^{\mathsf{U}}(M \mid A) = S_{k+1}^{\mathsf{U}}(Subs_k(Ard_k(M \mid A)))$, where

$$Ard_k(N)[i,j] = \begin{cases} N[i,j] & \text{if } i \neq k \\ ?\bot & \text{if } i = k = j \\ N[i,j] & \text{if } i = k \neq j \text{ and } N[k,k] = ?\bot \\ N[k,k]^* \odot N[i,j] & \text{otherwise} \end{cases}$$

$$Subs_k(N)[i,j] = \begin{cases} N[i,j] & \text{if } i = k \\ ?\bot & \text{if } i \neq k = j \\ \bigoplus\{N[i,k] \odot N[k,j], N[i,j]\} & \text{otherwise} \end{cases}$$

New program transformers

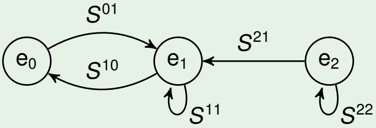
Example ((cont'd) 1/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?\mathrm{pre}(e_0)$	$?\bot$	$?\bot$
e_1	S^{10}	S^{11}	$?\bot$	$?\bot$	$?\mathrm{pre}(e_1)$	$?\bot$
e_2	$?\bot$	S^{21}	S^{22}	$?\bot$	$?\bot$	$?\mathrm{pre}(e_2)$

New program transformers

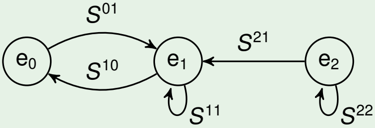
Example ((cont'd) 2/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?pre(e_0)$	$?\bot$	$?\bot$
e_1	$(S^{11})^*; S^{10}$	$?\bot$	$?\bot$	$?\bot$	$(S^{11})^*; ?pre(e_1)$	$?\bot$
e_2	$?\bot$	S^{21}	S^{22}	$?\bot$	$?\bot$	$?pre(e_2)$

New program transformers

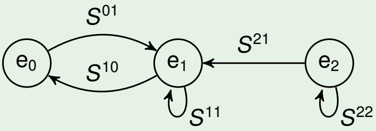
Example ((cont'd) 3/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?pre(e_0)$	$?\bot$	$?\bot$
e_1	$(S^{11})^*; S^{10}$	$?\bot$	$(S^{11})^*;?\bot$	$(S^{11})^*;?\bot$	$(S^{11})^*; ?pre(e_1)$	$(S^{11})^*;?\bot$
e_2	$?\bot$	S^{21}	S^{22}	$?\bot$	$?\bot$	$?pre(e_2)$

New program transformers

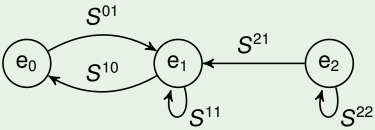
Example ((cont'd) 4/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?\mathrm{pre}(e_0)$	$?\bot$	$?\bot$
e_1	$(S^{11})^*; S^{10}$	$?\bot$	$?\bot$	$?\bot$	$(S^{11})^*; ?\mathrm{pre}(e_1)$	$?\bot$
e_2	$(S^{21}; (S^{11})^*; S^{10})$	$?\bot$	S^{22}	$?\bot$	$(S^{21}; (S^{11})^*; ?\mathrm{pre}(e_1))$	$?\mathrm{pre}(e_2)$

New program transformers

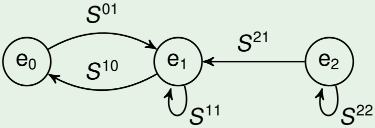
Example ((cont'd) 5/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?pre(e_0)$	$?\bot$	$?\bot$
e_1	$(S^{11})^*; S^{10}$	$?\bot$	$?\bot$	$?\bot$	$(S^{11})^*; ?pre(e_1)$	$?\bot$
e_2	$(S^{21}; (S^{11})^*; S^{10}) \cup ?\bot$	$?\bot$	$(S^{21}; ?\bot) \cup S^{22}$	$?\bot$	$?\bot$	$?pre(e_2)$

New program transformers

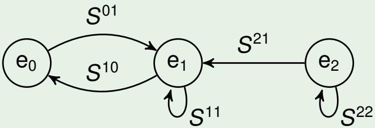
Example ((cont'd) 6/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?pre(e_0)$	$?\bot$	$?\bot$
e_1	$(S^{11})^*; S^{10}$	$?\bot$	$?\bot$	$?\bot$	$(S^{11})^*; ?pre(e_1)$	$?\bot$
e_2	$(S^{21}; (S^{11})^*; S^{10})$	$?\bot$	S^{22}	$?\bot$	$?\bot$	$?pre(e_2)$

New program transformers

Example ((cont'd) 7/7)



	e_0	e_1	e_2	e_0	e_1	e_2
e_0	$?\bot$	S^{01}	$?\bot$	$?\mathrm{pre}(e_0)$	$?\bot$	$?\bot$
e_1	$(S^{11})^*; S^{10}$	$?\bot$	$?\bot$	$?\bot$	$(S^{11})^*; ?\mathrm{pre}(e_1)$	$?\bot$
e_2	$(S^{21}; (S^{11})^*; S^{10})$	$?\bot$	S^{22}	$(S^{21}; ?\bot) \cup ?\bot$	$(S^{21}; (S^{11})^*; ?\mathrm{pre}(e_1)) \cup ?\bot$	$(S^{21}; ?\bot) \cup ?\mathrm{pre}(e_2)$

A new translation $\mathcal{L}_{\mathrm{LCC}} \to \mathcal{L}_{\mathrm{PDL}}$.

Definition (Translation functions t', r')

$$
\begin{aligned}
t'(\top) &= \top & r'(a) &= a\\
t'(p) &= p & r'(B) &= B\\
t'(\neg\varphi) &= \neg t'(\varphi) & r'(?\varphi) &= ?t'(\varphi)\\
t'(\varphi_1 \wedge \varphi_2) &= t'(\varphi_1) \wedge t'(\varphi_2) & r'(\pi_1;\pi_2) &= r'(\pi_1); r'(\pi_2)\\
t'([\pi]\varphi) &= [r'(\pi)]t'(\varphi) & r'(\pi_1 \cup \pi_2) &= r'(\pi_1) \cup r'(\pi_2)\\
t'([\mathrm{U},\mathrm{e}]\top) &= \top & r'(\pi^*) &= (r'(\pi))^*\\
t'([\mathrm{U},\mathrm{e}]p) &= t'(\mathrm{pre}(\mathrm{e})) \to t'(p^{\mathrm{sub}(\mathrm{e})})\\
t'([\mathrm{U},\mathrm{e}]\neg\varphi) &= t'(\mathrm{pre}(\mathrm{e})) \to \neg t'([\mathrm{U},\mathrm{e}]\varphi)\\
t'([\mathrm{U},\mathrm{e}](\varphi_1 \wedge \varphi_2)) &= t'([\mathrm{U},\mathrm{e}]\varphi) \wedge t'([\mathrm{U},\mathrm{e}]\varphi_2)\\
t'([\mathrm{U},\mathrm{e}_i][\pi]\varphi) &= \bigwedge_{\substack{0\le j\le n-1\\ \mu^{\mathrm{U}}(\pi)[i,j]\neq ?\bot}} [r'(\mu^{\mathrm{U}}(\pi)[i,j])]t'([\mathrm{U},\mathrm{e}_j]\varphi)\\
t'([\mathrm{U},\mathrm{e}][\mathrm{U}',\mathrm{e}']\varphi) &= t'([\mathrm{U},\mathrm{e}]t'([\mathrm{U}',\mathrm{e}']\varphi))
\end{aligned}
$$

Correctness of the new translation

Lemma

Let $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ *be an action model with* $\mathsf{e}_i, \mathsf{e}_j \in \mathsf{E}$*; let* π *be an* LCC *program. For any epistemic model M,*

$$\|T_{ij}^{\mathsf{U}}(\pi)\|^M \quad = \quad \|\mu^{\mathsf{U}}(\pi)[i,j]\|^M$$

Correctness of the new translation

Lemma

Let $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ *be an action model with* $\mathsf{e}_i, \mathsf{e}_j \in \mathsf{E}$*; let* π *be an* LCC *program. For any epistemic model M,*

$$||T_{ij}^{\mathsf{U}}(\pi)||^M \quad = \quad ||\mu^{\mathsf{U}}(\pi)[i,j]||^M$$

Corollary

The translation functions t', r' *reduce the language of* LCC *to that of* PDL*. This translation is correct.*

Correctness of the new translation

Lemma

Let U = (E, R, pre, sub) *be an action model with* $e_i, e_j \in E$*; let* π *be an* LCC *program. For any epistemic model M,*

$$\|T_{ij}^{U}(\pi)\|^{M} \quad = \quad \|\mu^{U}(\pi)[i,j]\|^{M}$$

Corollary

The translation functions t', r' *reduce the language of* LCC *to that of* PDL*. This translation is correct.*

Fact

The complexity of $T_{ij}^{U}(\pi)$ *in (LCC, 2006) is exponential. The complexity of* $\mu^{U}(\pi)$ *is* $O(g \cdot n^3)$*, where g is the number of subprograms*

New axioms for LCC; soundness and completeness.

Definition (LCC = PDL + reduction axioms)

Propositional tautologies

(*K*)	$[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$	(*top*)	$[\mathsf{U},\mathsf{e}]\top \leftrightarrow \top$
(*test*)	$[?\varphi_1]\varphi_2 \leftrightarrow (\varphi_1 \to \varphi_2)$	(*atoms*)	$[\mathsf{U},\mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to p^{\mathsf{sub}(e)})$
(*seq.*)	$[\pi_1;\pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$	(*neg.*)	$[\mathsf{U},\mathsf{e}]\neg\varphi \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to \neg[\mathsf{U},\mathsf{e}]\varphi)$
(*choice*)	$[\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$	(*conj.*)	$[\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2) \leftrightarrow ([\mathsf{U},\mathsf{e}]\varphi_1 \wedge [\mathsf{U},\mathsf{e}]\varphi_2)$
(*mix*)	$[\pi^*]\varphi \leftrightarrow \varphi \wedge [\pi][\pi^*]\varphi$	(*prog.*)	$[\mathsf{U},\mathsf{e}_i][\pi]\varphi \leftrightarrow \bigwedge_{\substack{0\le j\le n-1 \\ \mu^{\mathsf{U}}(\pi)[i,j]\neq ?\bot}} [\mu^{\mathsf{U}}(\pi)[i,j]][\mathsf{U},\mathsf{e}_j]\varphi$
(*ind.*)	$\varphi \wedge [\pi^*](\varphi \to [\pi]\varphi) \to [\pi^*]\varphi$	(*MP*)	$\vdash \varphi_1$ and $\vdash \varphi_1 \to \varphi_2$ imply $\vdash \varphi_2$
(*Nec*$_\pi$)	$\vdash \varphi$ implies $\vdash [\pi]\varphi$.	(*Nec*$_{\mathsf{U}}$)	$\vdash \varphi$ implies $\vdash [\mathsf{U},\mathsf{e}]\varphi$

New axioms for LCC; soundness and completeness.

Definition (LCC = PDL + reduction axioms)

Propositional tautologies

(*K*)	$[\pi](\varphi \rightarrow \psi) \rightarrow ([\pi]\varphi \rightarrow [\pi]\psi)$	(*top*)	$[\mathsf{U},\mathsf{e}]\top \leftrightarrow \top$
(*test*)	$[?\varphi_1]\varphi_2 \leftrightarrow (\varphi_1 \rightarrow \varphi_2)$	(*atoms*)	$[\mathsf{U},\mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \rightarrow p^{\mathsf{sub}(e)})$
(*seq.*)	$[\pi_1;\pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$	(*neg.*)	$[\mathsf{U},\mathsf{e}]\neg\varphi \leftrightarrow (\mathsf{pre}(\mathsf{e}) \rightarrow \neg[\mathsf{U},\mathsf{e}]\varphi)$
(*choice*)	$[\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$	(*conj.*)	$[\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2) \leftrightarrow ([\mathsf{U},\mathsf{e}]\varphi_1 \wedge [\mathsf{U},\mathsf{e}]\varphi_2)$
(*mix*)	$[\pi^*]\varphi \leftrightarrow \varphi \wedge [\pi][\pi^*]\varphi$	(*prog.*)	$[\mathsf{U},\mathsf{e}_i][\pi]\varphi \leftrightarrow \bigwedge_{\substack{0 \le j \le n-1 \\ \mu^{\mathsf{U}}(\pi)[i,j] \neq ?\bot}} [\mu^{\mathsf{U}}(\pi)[i,j]][\mathsf{U},\mathsf{e}_j]\varphi$
(*ind.*)	$\varphi \wedge [\pi^*](\varphi \rightarrow [\pi]\varphi)) \rightarrow [\pi^*]\varphi$	(*MP*)	$\vdash \varphi_1$ and $\vdash \varphi_1 \rightarrow \varphi_2$ imply $\vdash \varphi_2$
(*Nec*$_\pi$)	$\vdash \varphi$ implies $\vdash [\pi]\varphi$.	(*Nec*$_\mathsf{U}$)	$\vdash \varphi$ implies $\vdash [\mathsf{U},\mathsf{e}]\varphi$

Corollary

The new axiom system for LCC *is sound and complete.*

Outline

Summary

- We presented an alternative definition of the LCC program transformers

Summary

- We presented an alternative definition of the LCC program transformers

  instead of Kleene's method, we used Brzozowski's equational method.

Summary

- We presented an alternative definition of the LCC program transformers

  instead of Kleene's method, we used Brzozowski's equational method.

- Our proposal generates:

Summary

- We presented an alternative definition of the LCC program transformers

  instead of Kleene's method, we used Brzozowski's equational method.

- Our proposal generates:
  - A more efficient translation LCC $\rightarrow$ PDL.

Summary

- We presented an alternative definition of the LCC program transformers

  instead of Kleene's method, we used Brzozowski's equational method.
- Our proposal generates:
  - A more efficient translation LCC $\rightarrow$ PDL.
  - A new set of reduction axioms for LCC.

Summary

- We presented an alternative definition of the LCC program transformers

  instead of Kleene's method, we used Brzozowski's equational method.
- Our proposal generates:
  - A more efficient translation LCC $\rightarrow$ PDL.
  - A new set of reduction axioms for LCC.
  - A more elegant and simpler implementation to be used with PDL checkers.

Future Work

- Simplify some of the definitions used in program transformers

$$\text{e.g.}\qquad \sigma \odot \rho = \begin{cases} \sigma & \text{if } \sigma \neq ?\top = \rho \\ \rho & \text{if } \sigma = ?\top \end{cases}$$

Future Work

- Simplify some of the definitions used in program transformers

$$\text{e.g.} \quad \sigma \odot \rho = \begin{cases} \sigma & \text{if } \sigma \neq ?\top = \rho \\ \rho & \text{if } \sigma = ?\top \end{cases}$$

- Simplify the algorithm for the Ard_k and $Subs_k$ functions

Future Work

- Simplify some of the definitions used in program transformers

$$\text{e.g.} \quad \sigma \odot \rho = \begin{cases} \sigma & \text{if } \sigma \neq ?\top = \rho \\ \rho & \text{if } \sigma = ?\top \end{cases}$$

- Simplify the algorithm for the Ard_k and $Subs_k$ functions

  disregard the $N[i,j] = ?\bot$ elements with $j < k$ or $j > n + k$, for any $n \times 2n$ matrix $N[i,j]$

Future Work

- Simplify some of the definitions used in program transformers

$$\text{e.g.} \quad \sigma \odot \rho = \begin{cases} \sigma & \text{if } \sigma \neq ?\top = \rho \\ \rho & \text{if } \sigma = ?\top \end{cases}$$

- Simplify the algorithm for the Ard_k and $Subs_k$ functions

  disregard the $N[i,j] = ?\bot$ elements with $j < k$ or $j > n + k$, for any $n \times 2n$ matrix $N[i,j]$

- An implementation in Prolog of the proposed translation, to be combined with e.g. pdlProver, to be applied to:

Future Work

- Simplify some of the definitions used in program transformers

$$\text{e.g.} \quad \sigma \odot \rho = \begin{cases} \sigma & \text{if } \sigma \neq ?\top = \rho \\ \rho & \text{if } \sigma = ?\top \end{cases}$$

- Simplify the algorithm for the Ard_k and $Subs_k$ functions

  disregard the $N[i,j] = ?\bot$ elements with $j < k$ or $j > n + k$, for any $n \times 2n$ matrix $N[i,j]$

- An implementation in Prolog of the proposed translation, to be combined with e.g. pdlProver, to be applied to:
  - Verification of epistemic protocols (Russian Cards Problems).

Future Work

- Simplify some of the definitions used in program transformers

$$\text{e.g.} \quad \sigma \odot \rho = \begin{cases} \sigma & \text{if } \sigma \neq ?\top = \rho \\ \rho & \text{if } \sigma = ?\top \end{cases}$$

- Simplify the algorithm for the Ard_k and $Subs_k$ functions

  disregard the $N[i,j] = ?\bot$ elements with $j < k$ or $j > n + k$, for any $n \times 2n$ matrix $N[i,j]$

- An implementation in Prolog of the proposed translation, to be combined with e.g. pdlProver, to be applied to:
  - Verification of epistemic protocols (Russian Cards Problems).
  - Planning algorithms for LCC.

Thank you for your attention!

Bibliography

1. Johan van Benthem and Jan van Eijck and Barteld Kooi, *Logics of communication and change*, Information and Computation, 11(204) 1620–1662, (2006)

Bibliography

1. Johan van Benthem and Jan van Eijck and Barteld Kooi, *Logics of communication and change*, Information and Computation, 11(204) 1620–1662, (2006)
2. John H. Conway, *Regular Algebra and Finite Machines*, Chapman and Hall, (1971)

Bibliography

1. Johan van Benthem and Jan van Eijck and Barteld Kooi, *Logics of communication and change*, Information and Computation, 11(204) 1620–1662, (2006)
2. John H. Conway, *Regular Algebra and Finite Machines*, Chapman and Hall, (1971)
3. Janusz A. Brzozowski, *Derivatives of Regular Expressions*, Journal of the ACM, 11(4), 481–494, (1964),

Bibliography

1. Johan van Benthem and Jan van Eijck and Barteld Kooi, *Logics of communication and change*, Information and Computation, 11(204) 1620–1662, (2006)
2. John H. Conway, *Regular Algebra and Finite Machines*, Chapman and Hall, (1971)
3. Janusz A. Brzozowski, *Derivatives of Regular Expressions*, Journal of the ACM, 11(4), 481–494, (1964),
4. Dean N. Arden, *Delayed-logic and finite-state machines*, SWCT (FOCS), 133-151, (1961)

Bibliography

1. Johan van Benthem and Jan van Eijck and Barteld Kooi, *Logics of communication and change*, Information and Computation, 11(204) 1620–1662, (2006)
2. John H. Conway, *Regular Algebra and Finite Machines*, Chapman and Hall, (1971)
3. Janusz A. Brzozowski, *Derivatives of Regular Expressions*, Journal of the ACM, 11(4), 481–494, (1964),
4. Dean N. Arden, *Delayed-logic and finite-state machines*, SWCT (FOCS), 133-151, (1961)
5. S.C. Kleene, *Representation of Events in Nerve Nets and Finite Automata*, 3–41 in: Automata Studies (Claude E. Shannon and John McCarthy, eds.), Princeton University Press, (1956)