Forgetting and Uniform Interpolation in Multi-Agent Modal Logics

Liangda Fang[1,3], Yongmei Liu[1] and Hans van Ditmarsch[2]

[1]Dept. of Computer Science, Sun Yat-sen University, China
[2]LORIA, CNRS – Université de Lorraine, France
[3]Dept. of Computer Science, Jinan University, China

Mar 24, 2018

Introduction

- $\mathcal{P}$: a finite set of propositions;
- $p \in \mathcal{P}$;
- φ: the original formula;
- ψ: a result of forgetting p in φ, *i.e.*, the strongest consequence of φ without p:
  1. ψ does not contain p;
  2. For any query η which does not contain p, $\varphi \models \eta$ iff $\psi \models \eta$.

Introduction

- $\mathcal{P}$: a finite set of propositions;
- $p \in \mathcal{P}$;
- φ: the original formula;
- ψ: a result of forgetting p in φ, *i.e.*, the strongest consequence of φ without p:
  1. ψ does not contain p;
  2. For any query η which does not contain p, $\varphi \models \eta$ iff $\psi \models \eta$.
- ψ: the uniform interpolant of ϕ on $\mathcal{P} \setminus \{p\}$.

Introduction

- $\mathcal{P}$: a finite set of propositions;
- $p \in \mathcal{P}$;
- φ: the original formula;
- ψ: a result of forgetting p in φ, *i.e.*, the strongest consequence of φ without p:
  1. ψ does not contain p;
  2. For any query η which does not contain p, $\varphi \models \eta$ iff $\psi \models \eta$.
- ψ: the uniform interpolant of ϕ on $\mathcal{P} \setminus \{p\}$.

How to compute ψ?

A brute-force approach to propositional logic

1. Transform φ into an equivalent principal DNF $\bigvee_{t\in\Phi} t$;
2. Obtain t^p by eliminating any occurrence of p or $\neg p$;
3. $\bigvee_{t\in\Phi} t^p$ is a result of forgetting p in φ.

A brute-force approach to propositional logic

1. Transform φ into an equivalent principal DNF $\bigvee_{t\in\Phi} t$;
2. Obtain t^p by eliminating any occurrence of p or $\neg p$;
3. $\bigvee_{t\in\Phi} t^p$ is a result of forgetting p in φ.

Example

- $\varphi = (p \wedge q) \vee (\neg p \wedge \neg r)$;
- $\varphi \equiv (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$;
- $\psi = (q \wedge r) \vee (q \wedge \neg r) \vee (\neg q \wedge \neg r)$.

A brute-force approach to propositional logic

1. Transform φ into an equivalent principal DNF $\bigvee_{t \in \Phi} t$;
2. Obtain t^p by eliminating any occurrence of p or $\neg p$;
3. $\bigvee_{t \in \Phi} t^p$ is a result of forgetting p in φ.

Example

- $\varphi = (p \wedge q) \vee (\neg p \wedge \neg r)$;
- $\varphi \equiv (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$;
- $\psi = (q \wedge r) \vee (q \wedge \neg r) \vee (\neg q \wedge \neg r)$.

How about multi-agent modal logics?

- Modal logic: propositional logic + $\mathbf{K}_i$ operators;
- $\mathbf{K}_i\varphi$: agent i knows φ.

Outline

Outline

Modal language: $\mathcal{L}_{\mathbf{C}}^{\mathbf{K}}$

Definition (Syntax of $\mathcal{L}_{\mathbf{C}}^{\mathbf{K}}$)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{K}_i\varphi \mid \mathbf{C}\varphi,$$

where $p \in \mathcal{P}$, $i \in \mathcal{A}$, and $\varphi \in \mathcal{L}_{\mathbf{C}}^{\mathbf{K}}$.

Modal language: $\mathcal{L}^{\mathbf{K}}_{\mathbf{C}}$

Definition (Syntax of $\mathcal{L}^{\mathbf{K}}_{\mathbf{C}}$)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{K}_i\varphi \mid \mathbf{C}\varphi,$$

where $p \in \mathcal{P}$, $i \in \mathcal{A}$, and $\varphi \in \mathcal{L}^{\mathbf{K}}_{\mathbf{C}}$.

Definition (Two sublanguages)

1. $\mathcal{L}^{\mathbf{K}}_{n}$: $\mathcal{L}^{\mathbf{K}}_{\mathbf{C}}$ without $\mathbf{C}$ operator;
2. $\mathcal{L}^{\mathbf{K}}_{\mathbf{PC}}$: $\mathcal{L}^{\mathbf{K}}_{\mathbf{C}}$ with propositional common knowledge, *i.e.*, any φ appearing in $\mathbf{C}\varphi$ must be propositional.

Kripke models

Definition (Kripke models)

A Kripke model is a tuple $\langle S, R, V\rangle$ where

- S: a non-empty set of states,
- R: for each $i \in \mathcal{A}$, $R_i \subseteq S \times S$ is a relation on states.
- V: $S \to 2^{\mathcal{P}}$ is a function assigning to each proposition in a subset of states.

A pair (M, s) is called a pointed model.

Kripke models

Definition (Kripke models)

A Kripke model is a tuple $\langle S, R, V\rangle$ where

- S: a non-empty set of states,
- R: for each $i \in \mathcal{A}$, $R_i \subseteq S \times S$ is a relation on states.
- V: $S \to 2^{\mathcal{P}}$ is a function assigning to each proposition in a subset of states.

A pair (M, s) is called a pointed model.

L	K	D	T	K4	S4	K45	KD45	S5
Serial		✓	✓		✓		✓	✓
Reflexive			✓		✓			✓
Transitive				✓	✓	✓	✓	✓
Euclidean						✓	✓	✓

Cover modalites

Definition (Cover modalities)

1. $\nabla_i \Phi = \mathbf{K}_i(\bigvee_{\varphi \in \Phi} \varphi) \wedge \bigwedge_{\varphi \in \Phi} \hat{\mathbf{K}}_i \varphi$;
2. $\nabla \Phi = \mathbf{C}(\bigvee_{\varphi \in \Phi} \varphi) \wedge \bigwedge_{\varphi \in \Phi} \hat{\mathbf{C}} \varphi$.

where $\hat{\mathbf{K}}_i \doteq \neg \mathbf{K}_i \neg$ and $\hat{\mathbf{C}} \doteq \neg \mathbf{C} \neg$.

We can use ∇_i (resp. ∇) modality instead of $\mathbf{K}_i$ (resp. $\mathbf{C}$) modality.

Canonical formulas

Definition (Canonical formulas)

Let $P \subseteq \mathcal{P}$ be finite. We inductively define the set E_k^P as follows:

- $E_0^P = \{\bigwedge_{p \in \mathcal{X}} p \wedge \bigwedge_{p \in P \setminus \mathcal{X}} \neg p \mid \mathcal{X} \subseteq P\}$;
- $E_{k+1}^P = \{\delta_0 \wedge \bigwedge_{i \in \mathcal{A}} \nabla_i \Phi_i \mid \delta_0 \in E_0^P$ and $\Phi_i \subseteq E_k^P\}$.

$\delta_k \in E_k$:

- completely characterizes a Kripke model up to depth k;
- a minterm in modal logics.

Canonical formulas

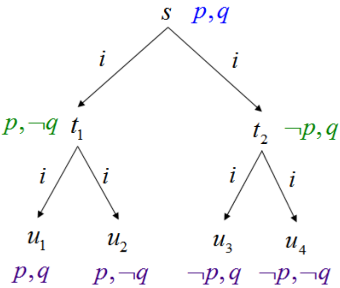


Figure: Kripke model

1. $p \wedge q$;
2. $p \wedge q \wedge \nabla_i\{p \wedge \neg q, \neg p \wedge q\}$;
3. $p \wedge q \wedge$
   $\nabla_i\{p \wedge \neg q \wedge \nabla_i\{p \wedge q, p \wedge \neg q\} \wedge$
   $\neg p \wedge q \wedge \nabla_i\{\neg p \wedge q, \neg p \wedge \neg q\}\}$

Canonical formulas

Proposition (Moss, 2007)

Any formula in $\mathcal{L}_n^{\mathbf{K}}$ can be equivalently transformed into a disjunction of satisfiable canonical formulas.

Outline

Computation of forgetting

1. Transform φ to a disjunction of satisfiable canonical formulas $\bigvee_{\delta \in \Phi} \delta$;
2. Obtain δ^p by eliminating any occurrence of p or $\neg p$;
3. $\bigvee_{\delta \in \Phi} \delta^p$ is a result of forgetting p in φ.

Computation of forgetting

1. Transform φ to a disjunction of satisfiable canonical formulas $\bigvee_{\delta \in \Phi} \delta$;
2. Obtain δ^p by eliminating any occurrence of p or $\neg p$;
3. $\bigvee_{\delta \in \Phi} \delta^p$ is a result of forgetting p in φ.

Example

- $\varphi = \hat{\mathbf{K}}_i p \wedge \hat{\mathbf{K}}_i \neg p$;
- $\varphi \equiv \delta_1 \vee \delta_2$;
- $\delta_1 = p \wedge \nabla_i \{p, \neg p\}$;
- $\delta_2 = \neg p \wedge \nabla_i \{p, \neg p\}$;
- $\delta_1^p \vee \delta_2^p \equiv \top \wedge \nabla_i \{\top\} \equiv \hat{\mathbf{K}}_i \top$.

Main theorem

Theorem

Let L *be* $\mathsf{K_n}$, $\mathsf{D_n}$, $\mathsf{T_n}$, $\mathsf{K45_n}$, $\mathsf{KD45_n}$ *or* $\mathsf{S5_n}$.
Let δ *be a canonical formula satisfiable in* L.
Then, δ^p *is a result of forgetting* p *in* δ.

Main theorem

Theorem

Let L *be* $\mathsf{K_n}$, $\mathsf{D_n}$, $\mathsf{T_n}$, $\mathsf{K45_n}$, $\mathsf{KD45_n}$ *or* $\mathsf{S5_n}$.
Let δ *be a canonical formula satisfiable in* L.
Then, δ^p *is a result of forgetting* p *in* δ.

Corollary

$\mathsf{K_n}$, $\mathsf{D_n}$, $\mathsf{T_n}$, $\mathsf{K45_n}$, $\mathsf{KD45_n}$ *and* $\mathsf{S5_n}$ *are closed under forgetting.*

Main theorem

Definition (Uniform interpolation)

A logic L has uniform interpolation: In the logic L, for any formula φ and any proposition p, there is a formula ψ that is a uniform interpolant of φ on $\mathcal{P} \setminus p$.

Main theorem

Definition (Uniform interpolation)

A logic L has uniform interpolation: In the logic L, for any formula φ and any proposition p, there is a formula ψ that is a uniform interpolant of φ on $\mathcal{P} \setminus p$.

Corollary

$\mathsf{K_n}$, $\mathsf{D_n}$, $\mathsf{T_n}$, $\mathsf{K45_n}$, $\mathsf{KD45_n}$ *and* $\mathsf{S5_n}$ *have uniform interpolation.*

Common knowledge case

- Negative result: KC is not closed under forgetting. [Studer, 2009]
- We consider the propositional common knowledge case, *i.e.*, $\mathcal{L}^{\mathbf{K}}_{\mathbf{PC}}$ where any φ appearing in $\mathbf{C}\varphi$ must be propositional.

Pc-canonical formulas

Definition (Pc-canonical formulas)

Let $P \subseteq \mathcal{P}$ be finite. We inductively define the set C_k^P as follows:

- $C_0^P = \{\theta \wedge \nabla\Phi_{\mathcal{A}} \mid \theta \in E_0^P \text{ and } \Phi_{\mathcal{A}} \subseteq E_0^P\}$;
- $C_{k+1}^P = \{\theta \wedge (\bigwedge_{i\in\mathcal{A}} \nabla_i\Phi_i) \wedge \nabla\Phi_{\mathcal{A}} \mid \theta \in E_0^P, \Phi_i \subseteq C_k^P \text{ and } \Phi_{\mathcal{A}} \subseteq E_0^P\}$.

Pc-canonical formulas

Definition (Pc-canonical formulas)

Let $P \subseteq \mathcal{P}$ be finite. We inductively define the set C_k^P as follows:

- $C_0^P = \{\theta \wedge \nabla\Phi_\mathcal{A} \mid \theta \in E_0^P$ and $\Phi_\mathcal{A} \subseteq E_0^P\}$;
- $C_{k+1}^P = \{\theta \wedge (\bigwedge_{i\in\mathcal{A}} \nabla_i\Phi_i) \wedge \nabla\Phi_\mathcal{A} \mid \theta \in E_0^P, \Phi_i \subseteq C_k^P$ and $\Phi_\mathcal{A} \subseteq E_0^P\}$.

Proposition

Any formula in $\mathcal{L}_{\mathbf{PC}}^{\mathbf{K}}$ can be equivalently transformed into a disjunction of satisfiable pc-canonical formulas.

Main theorem 2

Theorem

Let L *be* KC, DC, TC, K45C, KD45C *or* S5C.
Let δ *be a pc-canonical formula satisfiable in* L.
Then, δ^p *is a result of forgetting* p *in* δ.

Corollary

KPC, DPC, TPC, K45PC, KD45PC *and* S5PC *are closed under forgetting, and have uniform interpolation.*

Conclusions

- Prove that K_n, D_n, T_n, $K45_n$, $KD45_n$ and $S5_n$ are closed under forgetting.
- Extend the above results to propositional common knowledge case.

Current Results

L	K	D	T	K4	S4	K45	KD45	S5
$\mathcal{L}_1^{\mathbf{K}}$	$\checkmark^1$	$\checkmark^7$	$\checkmark^5$	✗[5]	✗[2]	$\checkmark$	$\checkmark$	$\checkmark^3$
$\mathcal{L}_n^{\mathbf{K}}$	$\checkmark^4$	$\checkmark^7$	$\checkmark^{3,5}$	✗[5]	✗[2]	$\checkmark$	$\checkmark$	$\checkmark^3$
$\mathcal{L}_{\mathbf{PC}}^{\mathbf{K}}$	$\checkmark$	$\checkmark$	$\checkmark$	✗[5]	✗[2]	$\checkmark$	$\checkmark$	$\checkmark$
$\mathcal{L}_{\mathbf{C}}^{\mathbf{K}}$	✗[6]	?	?	✗[5]	✗[2]	?	?	?

1. [Ghilardi, 1995]
2. [Ghilardi and Zawadowski, 1995]
3. [Wolter, F., 1998]
4. [D'Agostino and Lenzi, 2005]
5. [Bílková, 2007]
6. [Studer, 2009]
7. [Pattinson, 2013]

Future work

1. A practical approach for computing forgetting;
   - Identify a tractable form (DNF counterpart of modal logics);
   - Resolution methods: [Herzig and Mengin, 2008].

Future work

1. A practical approach for computing forgetting;
    - Identify a tractable form (DNF counterpart of modal logics);
    - Resolution methods: [Herzig and Mengin, 2008].
2. More general cases of common knowledge: any φ appearing in $\mathbf{C}\varphi$ can be in $\mathcal{L}_n^{\mathbf{K}}$;

Future work

1. A practical approach for computing forgetting;
   - Identify a tractable form (DNF counterpart of modal logics);
   - Resolution methods: [Herzig and Mengin, 2008].
2. More general cases of common knowledge: any φ appearing in $\mathbf{C}\varphi$ can be in $\mathcal{L}_n^{\mathbf{K}}$;
3. Distributed knowledge: the sum of the knowledge in a group
   - $\mathsf{K_D}$, $\mathsf{D_D}$ and $\mathsf{T_D}$: ✓;
   - $\mathsf{K45_D}$, $\mathsf{KD45_D}$ and $\mathsf{S5_D}$: ?

Future work

1. A practical approach for computing forgetting;
   - Identify a tractable form (DNF counterpart of modal logics);
   - Resolution methods: [Herzig and Mengin, 2008].
2. More general cases of common knowledge: any φ appearing in $\mathbf{C}\varphi$ can be in $\mathcal{L}_n^{\mathbf{K}}$;
3. Distributed knowledge: the sum of the knowledge in a group
   - K_D, D_D and T_D: ✓;
   - $K45_D$, $KD45_D$ and $S5_D$: ?
4. Monotone Modal Logic: $\mathbf{K}_i(p \wedge q) \rightarrow \mathbf{K}_i p$
   - M: ✓ [Santocanale and Venema, 2010];
   - M extended by classical axioms: ?.

Future work

1. A practical approach for computing forgetting;
   - Identify a tractable form (DNF counterpart of modal logics);
   - Resolution methods: [Herzig and Mengin, 2008].
2. More general cases of common knowledge: any φ appearing in $\mathbf{C}\varphi$ can be in $\mathcal{L}_n^{\mathbf{K}}$;
3. Distributed knowledge: the sum of the knowledge in a group
   - $\mathsf{K_D}$, $\mathsf{D_D}$ and $\mathsf{T_D}$: ✓;
   - $\mathsf{K45_D}$, $\mathsf{KD45_D}$ and $\mathsf{S5_D}$: ?
4. Monotone Modal Logic: $\mathbf{K}_i(p \wedge q) \rightarrow \mathbf{K}_i p$
   - M: ✓ [Santocanale and Venema, 2010];
   - M extended by classical axioms: ?.
5. Progression and diagnose in multi-agent settings.
   - Progression in the Situation Calculus: [Fang, *et al.*, 2015];
   - Diagnose in propositional logic: [Lin, 2001] and [Lang, 2008].

Thank you!

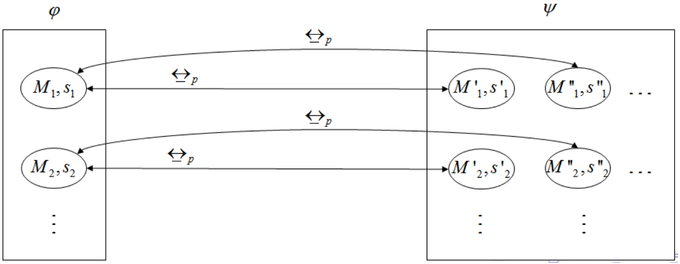
Outline

Model-theoretic definition of forgetting

Definition

Consider the context of a modal system L. Let $\varphi \in \mathcal{L}^{\mathbf{K}}_{\mathbf{C}}$. We call ψ is the result of forgetting p from φ, if the following conditions hold:

- Forth: for any model (M, s) of φ, if $(M,' s')$ is a model s.t. $(M, s)\underline{\leftrightarrow}_p(M', s')$, then $M', s' \models \psi$; (Easy: by induction)
- Back: for any model $(M,' s')$ of ψ, there exists a model (M, s) s.t. $M, s \models \varphi$ and $(M, s)\underline{\leftrightarrow}_p(M', s')$. (Very difficult!)



Proof of back condition

- (M', s'): an L-model of δ^p;
- Construct (M, s) s.t.
  1. (M, s) is an L-model;
  2. $M, s \models \varphi$;
  3. $(M, s) \underline{\leftrightarrow}_p (M', s')$.

- $\delta \in E_0$:
  1. Let (M, s) be the copy (M', s').
  2. Modify the valuation on s s.t. $V'(s') \models \delta$.
- $\delta \in E_{k+1}$: $\delta = \theta \wedge \bigwedge_{i \in \mathcal{A}} \nabla_i \Phi_i$. By induction?

K_n and D_n: by induction

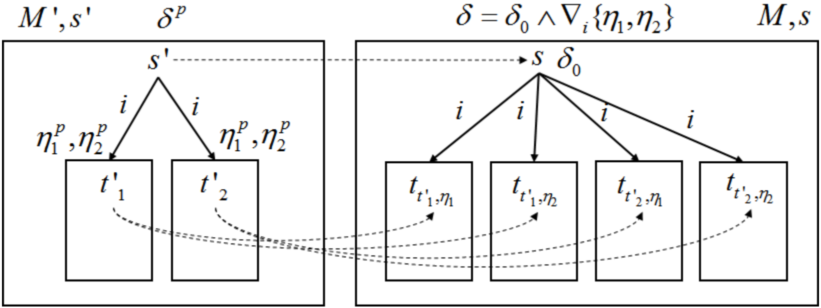


Figure: Illustration for the proof of K_n and D_n cases

T_n: add reflexive edge

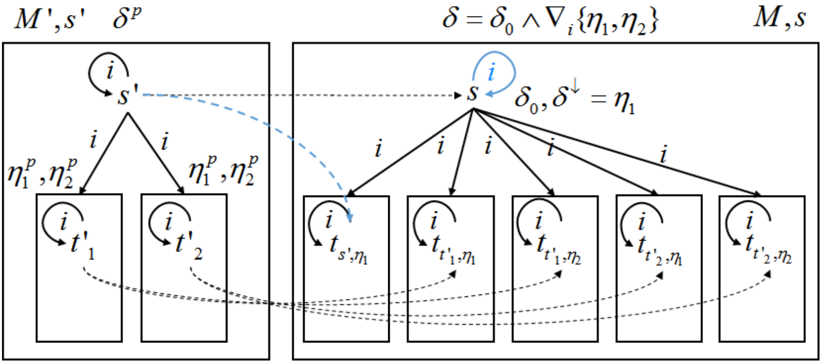


Figure: Illustration for the proof of T_n case

K45$_n$ and KD45$_n$: multi-pointed models

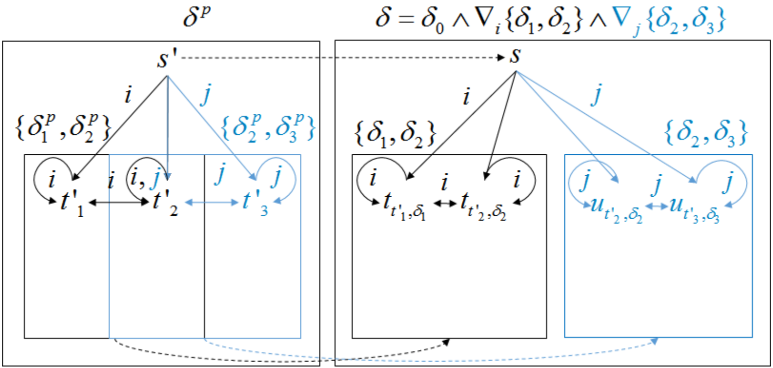


Figure: Illustration for the proof of K45$_n$ and KD45$_n$ cases

$K45_n$ and $KD45_n$: multi-pointed models

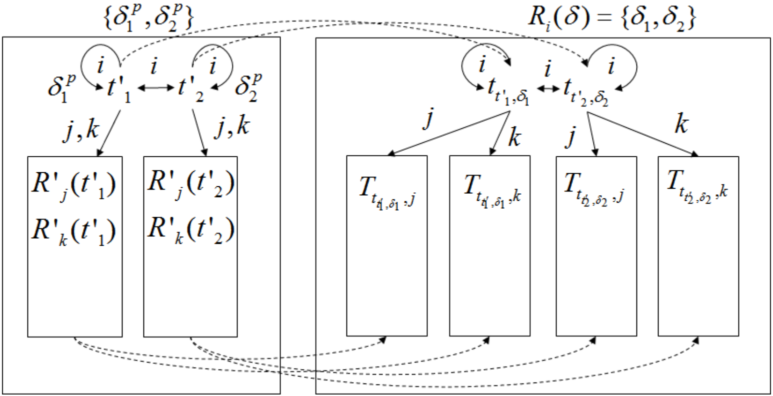


Figure: Illustration for the proof of $K45_n$ and $KD45_n$ cases

$S5_n$: add reflexive and symmetric edges

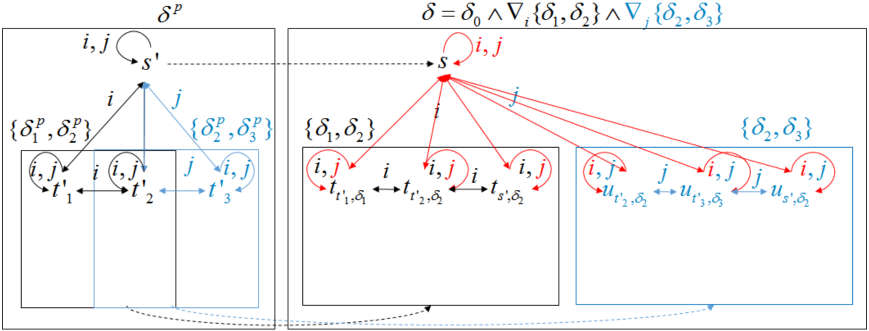


Figure: Illustration for the proof of $S5_n$ case

S5$_n$: add reflexive and symmetric edges

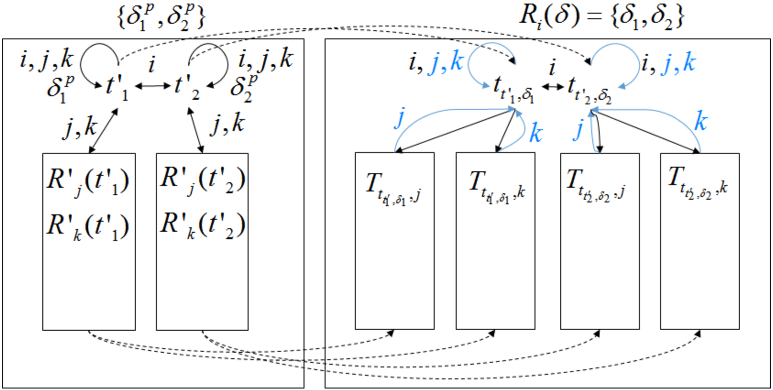


Figure: Illustration for the proof of S5$_n$ case

Extension to $\mathcal{L}^{\mathrm{K}}_{\mathrm{PC}}$: split a world into several copies

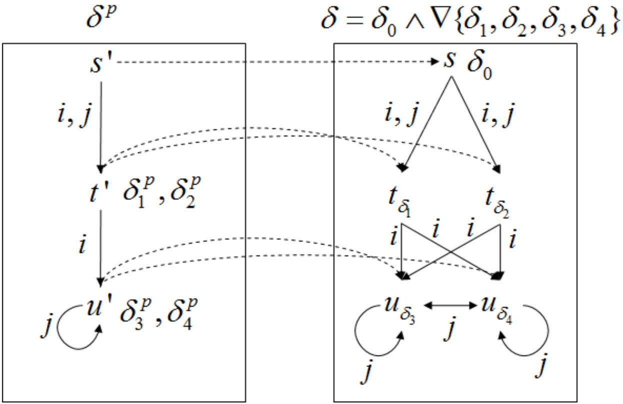


Figure: Illustration for the proof of KPC basic case