Pattern models:

A Dynamic Epistemic Logic For Distributed System Analysis

Armando Castañeda, Hans van Ditmarsch,
David A. Rosenblueth & Diego A. Velázquez

Agenda

1. Context
2. Pattern models
3. Parametrized pattern models

Context
Preliminaries

- Some authors in distributed computing still refer to knowledge only informally
  - It is important to develop tools to formalize such a concept

Context

Preliminaries

- Some authors in distributed computing still refer to knowledge only informally
  - It is important to develop tools to formalize such a concept
- Interpreted systems
  - Halpern & Moses (80's)

Context
Preliminaries

- Some authors in distributed computing still refer to knowledge only informally
  - It is important to develop tools to formalize such a concept
- Interpreted systems
  - Halpern & Moses (80's)
- Connection of distributed computing & dynamic epistemic logic through action models
  - Pfleger & Schmid (2018)
  - Goubault, Ledent & Rajsbaum (since 2018)

Context

Preliminaries

- Some authors in distributed computing still refer to knowledge only informally
  - It is important to develop tools to formalize such a concept
- Interpreted systems
  - Halpern & Moses (80's)
- Connection of distributed computing & dynamic epistemic logic through action models
  - Pfleger & Schmid (2018)
  - Goubault, Ledent & Rajsbaum (since 2018)
    - Kripke frames cat. and simplicial complex cat. are equivalent

Context

Epistemic logic

Syntax

$$\mathcal{L}_K \ni \varphi ::= p \mid \neg\varphi \mid \ \varphi \wedge \varphi \mid K_a\varphi$$

where $a \in A$, and $p \in P$

No update modality

Context

Epistemic logic

Syntax

$$\mathcal{L}_K \ni \varphi ::= p \mid \neg\varphi \mid \ \varphi \wedge \varphi \mid K_a\varphi$$

where $a \in A$, and $p \in P$

No update modality

Semantics

$$\begin{array}{ll}
M, w \models p & \text{iff } \ p \in L(w) \\
M, w \models \neg\varphi & \text{iff } M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & \text{iff } M, w \models \varphi \text{ and } M, w \models \psi \\
M, w \models K_a\varphi & \text{iff } M, w' \models \varphi \text{ for all } w' \text{ such that } \ w \ \sim_a \ w'
\end{array}$$

Context

Distributed computing models

Agents (processes)

- State machines
- Private input (local state)
- Execute a protocol of communication
  - All gathered information is sent (full-information)

Context

Distributed computing models

Agents (processes)

- State machines
- Private input (local state)
- Execute a protocol of communication
  - All gathered information is sent (full-information)

Dynamic-network models

- Synchronous (round closed) communication
- An adversary decides who communicates with whom
  - Picks a communication graph in every round

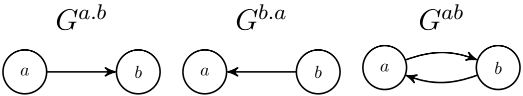
    $a \rightarrow b$ *reflexive relation
  - Oblivious
    - any communication graph in a given set of communication graphs may occur in any round

Context

Iterated immediate snapshot model (IIS)

IIS model

- processes write to a shared memory and then take a snapshot
  - Concurrent read

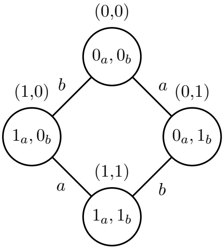


Context

Iterated immediate snapshot model (IIS)

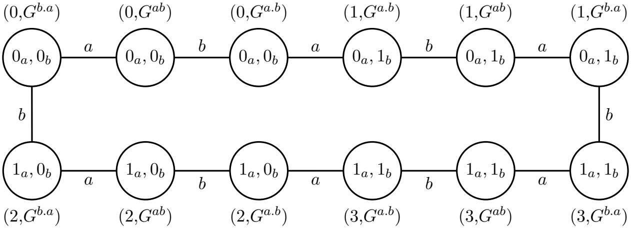
Epistemic model (Kripke models with equivalence relations)

- A triple $(W, \sim, L)$
  - Worlds
  - Indistinguishability relations
  - True valued propositions



Context

Iterated immediate snapshot model (IIS)



Action models

- Update mechanism

Action models

- Update mechanism
  - Action model (Structure) $M = (E, R, Pre)$
    - Events
    - Indistinguishability relations
    - Precondition formulas

Action models

- Update mechanism
  - Action model (Structure) $\mathsf{M} = (\mathsf{E}, \mathsf{R}, \mathsf{Pre})$
    - Events
    - Indistinguishability relations
    - Precondition formulas
  - Restricted modal product
    - $M \otimes \mathsf{M} = M'$

Context

Action models

$M' = (W', \sim', L') = M \otimes \mathsf{M}$ is defined as follows:

Context

Action models

$M' = (W', \sim', L') = M \otimes \mathsf{M}$ is defined as follows:

- $W' = \{(w, \mathsf{e}) \in W \times \mathsf{E} \mid M, w \models \mathsf{Pre}(\mathsf{e})\}$

Context

Action models

$M' = (W', \sim', L') = M \otimes \mathsf{M}$ is defined as follows:

- $W' = \{(w, \mathsf{e}) \in W \times \mathsf{E} \mid M, w \models \mathsf{Pre}(\mathsf{e})\}$
- $\sim'_a = \{((w, \mathsf{e}), (w', \mathsf{e}')) \in W' \times W' \mid w \sim_a w' \text{ and } \mathsf{e}\, R_a\, \mathsf{e}'\}$

Context

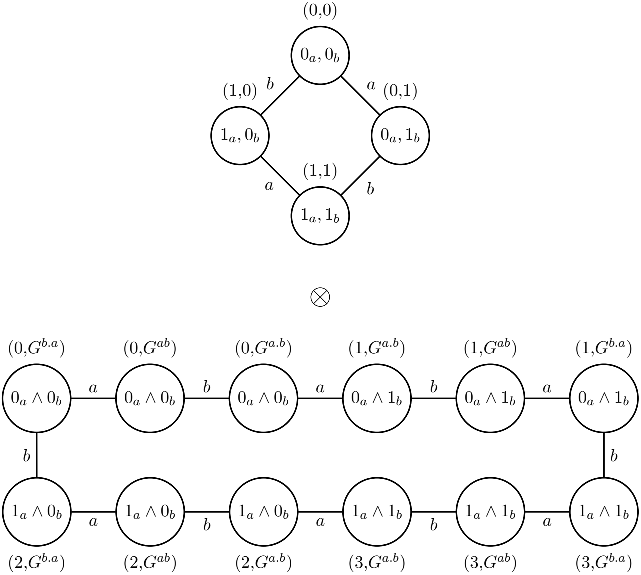
Action models

$M' = (W', \sim', L') = M \otimes \mathsf{M}$ is defined as follows:

- $W' = \{(w, \mathsf{e}) \in W \times \mathsf{E} \mid M, w \models \mathsf{Pre}(\mathsf{e})\}$
- $\sim'_a = \{((w, \mathsf{e}), (w', \mathsf{e}')) \in W' \times W' \mid w \sim_a w' \;\text{ and }\; \mathsf{e}\, R_a\, \mathsf{e}'\}$
- $L'(w, e) = L(w)$

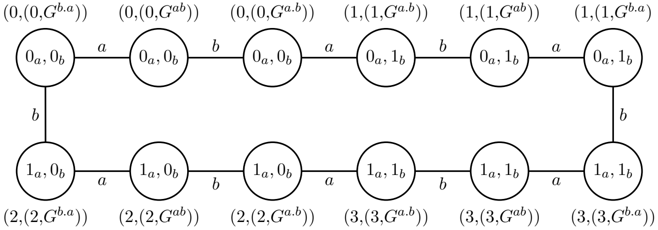
Context

Action models



Context

Iterated immediate snapshot model (IIS)



Context

Action models

Modeling epistemic change with action models through rounds of communication has drawbacks

- Direct application is simple but inefficient

Context

Action models

Modeling epistemic change with action models through rounds of communication has drawbacks

- Direct application is simple but inefficient
- Finding compact action models is not clear

Agenda

1. Context
2. Pattern models
3. Parametrized pattern models

Pattern models

We propose *pattern models*, a dynamic epistemic logic

- Similar to reading event models (Baltag and Smets)

Pattern models

We propose *pattern models*, a dynamic epistemic logic

- Similar to reading event models (Baltag and Smets)
  - The events are communication graphs

Pattern models

We propose *pattern models*, a dynamic epistemic logic

- Similar to reading event models (Baltag and Smets)
  - The events are communication graphs
  - The restricted modal product allows preconditions

Pattern models

We propose *pattern models*, a dynamic epistemic logic

- Similar to reading event models (Baltag and Smets)
  - The events are communication graphs
  - The restricted modal product allows preconditions
  - A protocol can be a parameter of the product (not only full-information)

Pattern models

Pattern Models

- Update mechanism

Pattern models

Pattern Models

- Update mechanism
  - Pattern model (structure) $\mathcal{P} = (\mathbf{G}, Pre)$
    - Set of communication graphs
    - Precondition formulas

Pattern models

Pattern Models

- Update mechanism
  - Pattern model (structure) $\mathcal{P} = (\mathbf{G}, Pre)$
    - Set of communication graphs
    - Precondition formulas
  - Restricted modal product
    - $M \odot \mathcal{P} = M'$

Pattern models

Let $Ga = \{b \in A \mid bGa\}$ be the in-neighbourhood of a in G

$(W', \sim', L') = M' = M \odot \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$

Pattern models

Let $Ga = \{b \in A \mid bGa\}$ be the in-neighbourhood of a in G

$(W', \sim', L') = M' = M \odot \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid$

Pattern models

Let $Ga = \{b \in A \mid bGa\}$ be the in-neighbourhood of a in G

$(W', \sim', L') = M' = M \odot \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid$
  $\underline{Ga = G'a}$ (same in-neighbourhood)

Pattern models

Let $Ga = \{b \in A \mid bGa\}$ be the in-neighbourhood of a in G

$(W', \sim', L') = M' = M \odot \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid$
  $\underline{Ga = G'a}$ (same in-neighbourhood)
  and $\underline{w \sim_a w' \ \forall a \in Ga}\}$

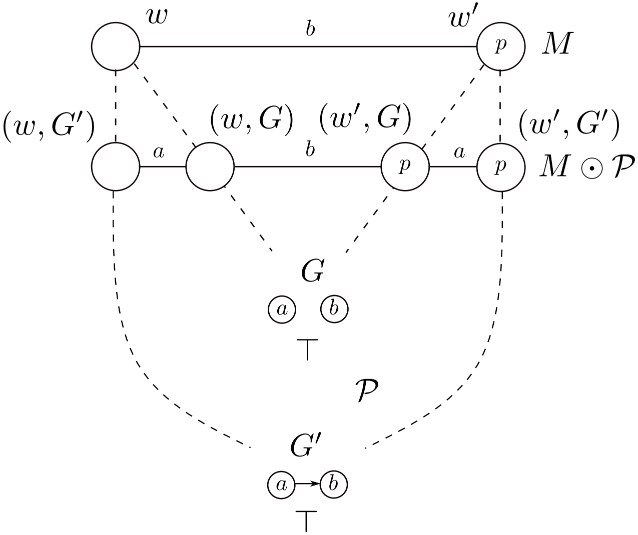
Pattern models

Let $Ga = \{b \in A \mid bGa\}$ be the in-neighbourhood of a in G

$(W', \sim', L') = M' = M \odot \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid$
  $\underline{Ga = G'a}$ (same in-neighbourhood)
  and $\underline{w \sim_a w' \; \forall a \in Ga}\}$
- $L'(w, G) = L(w)$

Pattern models



Pattern models

- The full-information protocol is *explicit* in the product definition
  - This non-parametrized version is useful for studying computability

Pattern models

- The full-information protocol is *explicit* in the product definition
  - This non-parametrized version is useful for studying computability
- The full-information protocol is not practical in real systems

Agenda

1. Context
2. Pattern models
3. Parametrized pattern models

Parametrized pattern models

Loc, set of local states
Msg, set of message contents

Parametrized pattern models

Loc, set of local states
Msg, set of message contents
A protocol $\pi = (\mu, \lambda)$,

Parametrized pattern models

Loc, set of local states
Msg, set of message contents
A protocol $\pi = (\mu, \lambda)$,

- $\mu : Loc \rightarrow (Msg \cup \{\perp\})^{|A|}$

Parametrized pattern models

Loc, set of local states
Msg, set of message contents
A protocol $\pi = (\mu, \lambda)$,

- $\mu : Loc \to (Msg \cup \{\perp\})^{|A|}$
- $\lambda : Loc \times (Msg \cup \{\perp\})^{|A|} \to Loc$

Parametrized pattern models

Epistemic models for distributed systems

Parametrized pattern models

Epistemic models for distributed systems

- A tuple $(W, \sim, L, S)$

Parametrized pattern models

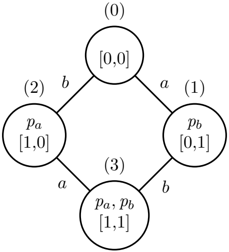
Epistemic models for distributed systems

- A tuple $(W, \sim, L, S)$
  - $S : W \times A \to Loc$

Parametrized pattern models

Epistemic models for distributed systems

- A tuple $(W, \sim, L, S)$
  - $S : W \times A \to Loc$



Parametrized pattern models

$M_a^w \mid_G$ is the set of messages that a gets when G occurs in w

$(W', \sim', L', S') = M' = M \odot_\pi \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid$
  $\lambda(S(w, a), M_a^w \mid_G) = \lambda(S(w', a), M_a^{w'} \mid_{G'})$
- $L'(w, G) = L(w)$
- $S'(a, (w, G)) = \lambda(S(w, a), M_a^w \mid_G)$

Parametrized pattern models

$M_a^w \mid_G$ is the set of messages that a gets when G occurs in w

$(W', \sim', L', S') = M' = M \odot_\pi \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid$
  $\underline{\lambda(S(w, a), M_a^w \mid_G) = \lambda(S(w', a), M_a^{w'} \mid_{G'})}\}$
- $L'(w, G) = L(w)$
- $S'(a, (w, G)) = \lambda(S(w, a), M_a^w \mid_G)$

Parametrized pattern models

$M_a^w \mid_G$ is the set of messages that a gets when G occurs in w

$(W', \sim', L', S') = M' = M \odot_\pi \mathcal{P}$ is defined as follows:

- $W' = \{(w, G) \in W \times \mathbf{G} \mid M, w \models Pre(G)\}$
- $\sim'_a = \{((w, G), (w', G')) \in W' \times W' \mid \underline{\lambda(S(w,a), M_a^w \mid_G) = \lambda(S(a,w'), M_a^{w'} \mid_{G'})}$
- $L'(w, G) = L(w)$
- $S'(a, (w, G)) = \lambda(S(w, a), M_a^w \mid_G)$

Parametrized pattern models

$$A = \{a\}$$

Parametrized pattern models

$A = \{a\}$
$\mathcal{I} = \{0, 1\}$

Parametrized pattern models

$A = \{a\}$
$\mathcal{I} = \{0, 1\}$
$Msg = Loc = \mathcal{I} \cup \{-\}$

Parametrized pattern models

$A = \{a\}$
$\mathcal{I} = \{0, 1\}$
$Msg = Loc = \mathcal{I} \cup \{-\}$

- $\pi = (\mu, \lambda)$ (Forget)

Parametrized pattern models

$A = \{a\}$
$\mathcal{I} = \{0, 1\}$
$Msg = Loc = \mathcal{I} \cup \{-\}$

- $\pi = (\mu, \lambda)$ (Forget)
    - $\mu(x) = (x)$

Parametrized pattern models

$A = \{a\}$
$\mathcal{I} = \{0, 1\}$
$Msg = Loc = \mathcal{I} \cup \{-\}$

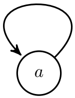
- $\pi = (\mu, \lambda)$ (Forget)
  - $\mu(x) = (x)$
  - $\lambda(x') = -$ (Constant function)

Parametrized pattern models

$A = \{a\}$
$\mathcal{I} = \{0, 1\}$
$Msg = Loc = \mathcal{I} \cup \{-\}$

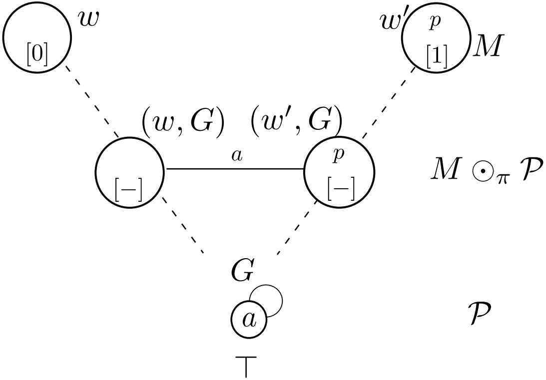
- $\pi = (\mu, \lambda)$ (Forget)
  - $\mu(x) = (x)$
  - $\lambda(x') = -$ (Constant function)

G



$\top$

Parametrized pattern models



Pattern models

- This parametrized version is useful for designing efficient protocols
  - Automated formal verification of protocols

Results

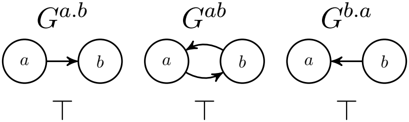
- Systematic construction of pattern models for each round of communication given an arbitrary adversary

Results

- Systematic construction of pattern models for each round of communication given an arbitrary adversary
  - Proof of correctness of such pattern models
    - The correctness is still valid with the parametrized version

Results

- Systematic construction of pattern models for each round of communication given an arbitrary adversary
  - Proof of correctness of such pattern models
    - The correctness is still valid with the parametrized version
  - Oblivious dynamic-network models requires constant space



рахмат
danke
謝謝
ngiyabonga
spas
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
tapadh leat
kiitos
dankie
nanni
nandri
dhanyavad
thank you
gracias
хвала
asante
manana
obrigada
tenki
bedankt
bayarlalaa
gràcie
hvala
mauruuru
köszönöm
dankon
akun
aciu
murakoze
enkosi
djiere dieuf
tau
mochchakkeram
дякую
mamnun
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
chokrane
sobodi
dėkuji
sagolun
sukriya
kop khun krap
taiku
obrigado
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
arigatô
takk
dakujem
trugarez
terima kasih
rahmet
dhanyavadagalu
shukriya
mercè
мерси
kam sah hamnida
rahmat
ευχαριστώ
diolch
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci