MuBDDy Children.

Symbolic Model Checking DEL.

Malvin Gattinger

ILLC, Amsterdam

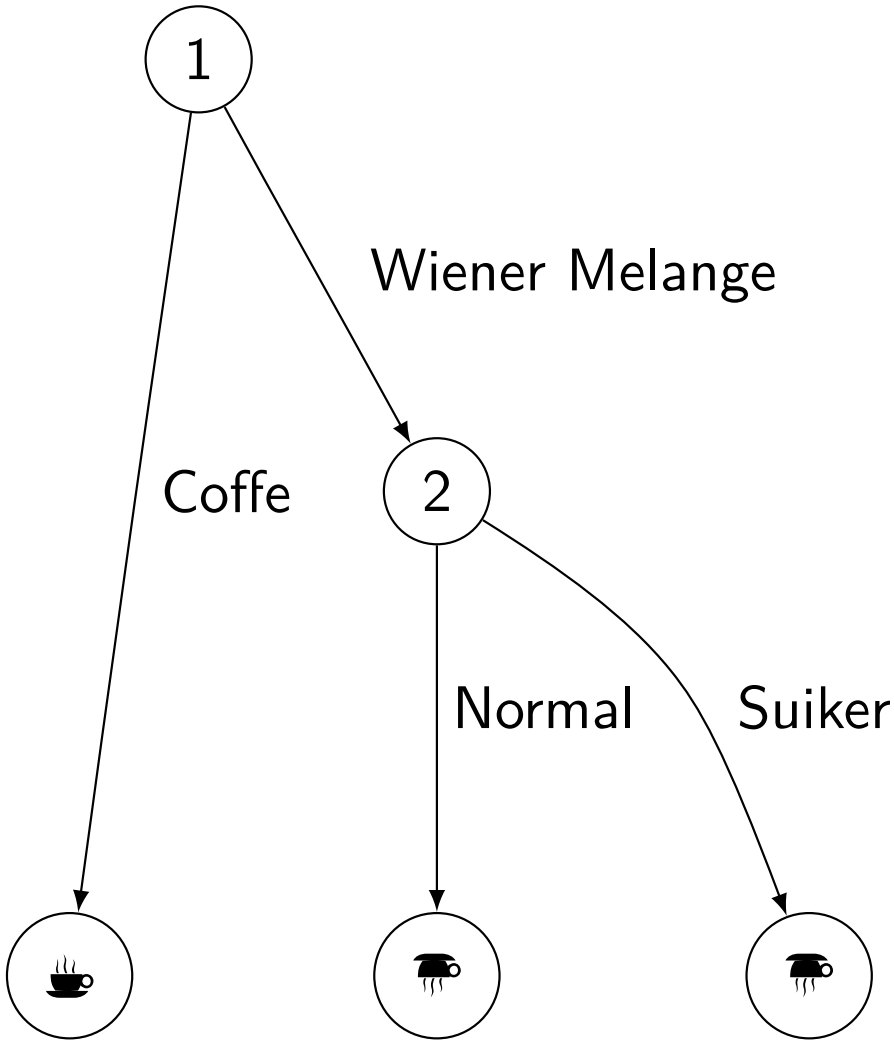
January 16th, 2015

Joint work with Johan van Benthem, Jan van Eijck and Kaile Su.

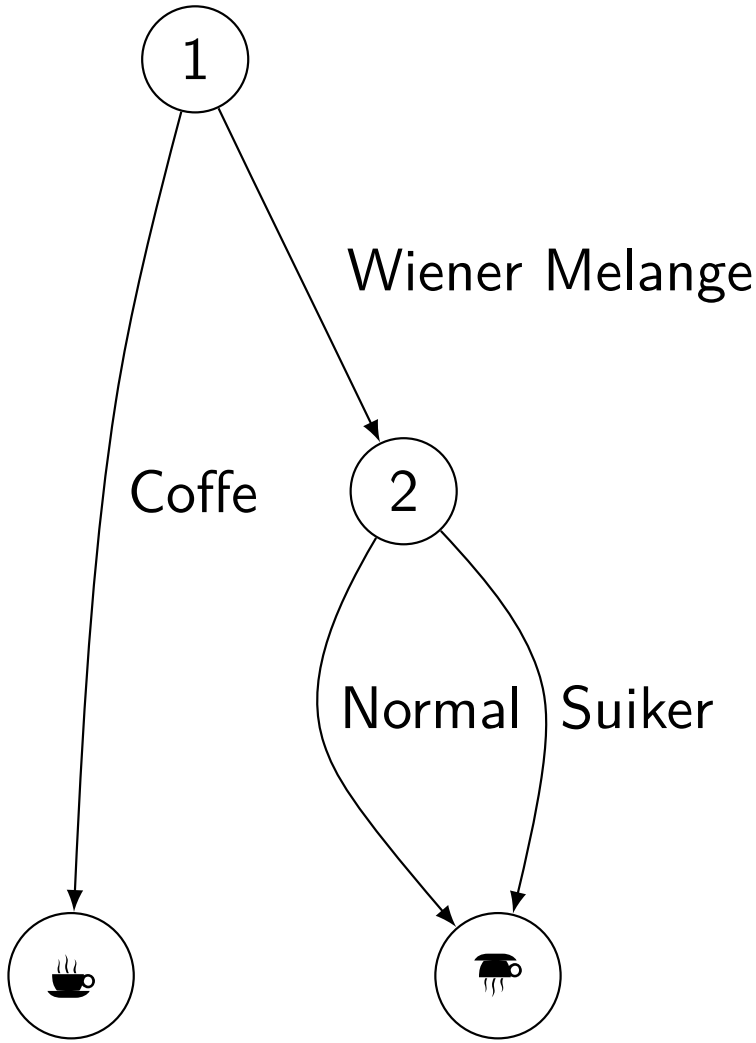
Outline

Coffee Machines

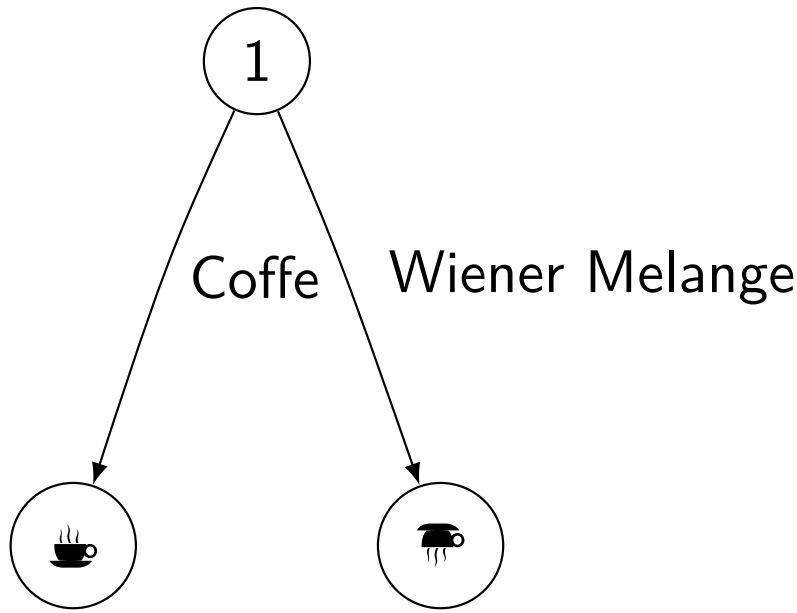
Coffee machines are annoying.



Coffee machines are still annoying.



Coffee machines do not have to be annoying.



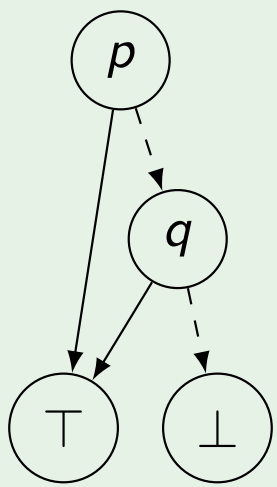
Binary Decision Diagrams

Binary Decision Diagrams

Definition

A Binary Decision Diagram [Bry86] for the variables V is a directed acyclic graph where non-terminal nodes are from V with two outgoing edges and terminal nodes are $\top$ or $\bot$.
Reduced: No redundancy, identify isomorphic subgraphs.
Ordered: Variables in a given order, maximally once.
By "BDD" we always mean a reduced and ordered BDD.

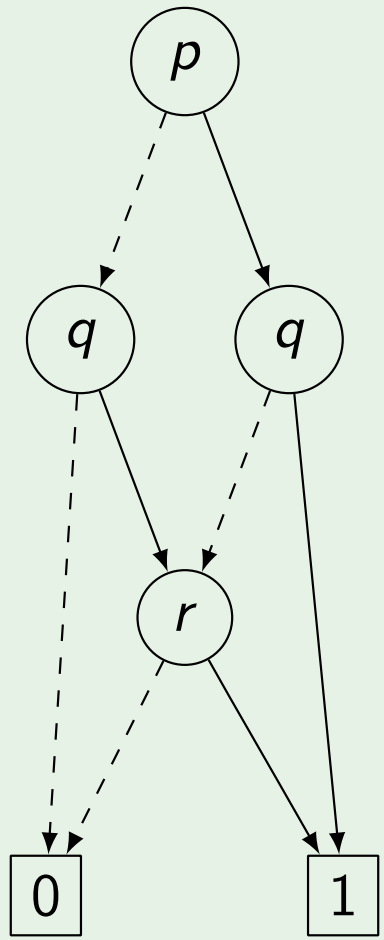
Example



Binary Decision Diagrams

Example

$$(p \wedge q) \vee (q \wedge r) \vee (p \wedge r)$$



Binary Decision Diagrams: Fun facts

Reduced and ordered BDDs provide a compact representation of boolean functions. Nice properties:

- Small: A BDD often takes less memory than a truth table.
- Canonical: All *equivalent* formulas have *identical* BDDs.
- Composable: Given the BDDs for ϕ and ψ we can easily find those for $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\forall p\phi$, ...
- Simultaneous composition: A BDD for $\forall p(\phi \to \psi)$ can be cosntructed from BDDs for ϕ and ψ "in one go".

But: Only boolean! How to use BDDs for dynamic logics?
We generalize ideas in [SSL07] and [LSSC08] to cover full DEL.

Knowledge Structures

Interlude: Our DEL for today.

Definition

The DEL language without common knowledge is:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid K_i\phi \mid [!\phi]\phi \mid [\phi]_\Delta\phi$$

where $p \in V$, $i \in I$ and $\Delta \subseteq I$. Abbreviations for $\vee$, $\rightarrow$ and $\leftrightarrow$.
The *boolean formulas* over V are:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi$$

Fact

Using Kripke Models $\mathcal{M} = (W, (R_i)_{i\in I}, \mathrm{Val})$ *we can give semantics to this language and inductively define* $\mathcal{M}, w \vDash \phi$.

Knowledge Structures

Kripke models: Which situations can you distinguish?
Knowledge structure: What facts do you observe?

Definition

A knowledge structure is a tuple $\mathcal{F} = (V, \theta, O_1, \cdots, O_n)$ where

- *Vocabulary*: V is a set of propositional variables
- *Law*: θ is a boolean formula over V
- *Observational variables*: for each agent i, $O_i \subseteq V$

An assignment $s \subseteq V$ is a *state* of $\mathcal{F}$ iff it satisfies θ.
A pair $(\mathcal{F}, s)$ is called a scenario.
The local state for i at s is $s \cap O_i$.

Example

$\mathcal{F} = (V = \{p\}, \theta = \top, O_1 = \{p\}, O_2 = \varnothing)$

DEL Semantics on Knowledge Structures

Definition

Fix $\mathcal{F} = (V, \theta, O_1, \ldots, O_n)$ and a state s of $\mathcal{F}$.

$$
\begin{array}{lcl}
(\mathcal{F}, s) \models p & \iff & p \in s \\
(\mathcal{F}, s) \models \neg\alpha & \iff & \text{not } (\mathcal{F}, s) \models \alpha \\
(\mathcal{F}, s) \models \alpha \wedge \beta & \iff & (\mathcal{F}, s) \models \alpha \text{ and } (\mathcal{F}, s) \models \beta \\
(\mathcal{F}, s) \models K_i\alpha & \iff & \text{for all states } s' \text{ of } \mathcal{F}: \\
 & & \text{if } s \cap O_i = s' \cap O_i, \text{ then } (\mathcal{F}, s') \models \alpha \\
(\mathcal{F}, s) \models [!\psi]\alpha & \iff & (\mathcal{F}, s) \models \psi \text{ implies } (\mathcal{F}^{!\psi}, s) \models \alpha \\
 & & \text{where } \|\psi\|_{\mathcal{F}} \text{ is on the next slide} \\
 & & \text{and } \mathcal{F}^{!\psi} = (V, \theta \wedge \|\psi\|_{\mathcal{F}}, O_1, \cdots, O_n) \\
(\mathcal{F}, s) \models [\psi]_\Delta\alpha & \iff & (\mathcal{F}, s) \models \psi \text{ implies } (\mathcal{F}^\Delta_\psi, s \cup p_\psi) \models \alpha \\
 & & \text{where } p_\psi \text{ is fresh, } \|\psi\|_{\mathcal{F}} \text{ on next slide,} \\
 & & \mathcal{F}^\Delta_\psi := (V \cup p_\psi, \theta \wedge (p_\psi \to \|\psi\|_{\mathcal{F}}), O'_i) \\
 & & \text{and } O'_i = O_i \cup \{p_\psi | i \in \Delta\}
\end{array}
$$

Everything is Boolean!

Definition

Fix $\mathcal{F} = (V, \theta, O_1, \cdots, O_n)$.
Translate $\alpha \in \mathcal{L}(V)$ to a boolean formula $\|\alpha\|_{\mathcal{F}}$:

$$\begin{array}{lcl} p & \mapsto & p \\ \neg\beta & \mapsto & \neg\|\beta\|_{\mathcal{F}} \\ \beta_1 \wedge \beta_2 & \mapsto & \|\beta_1\|_{\mathcal{F}} \wedge \|\beta_2\|_{\mathcal{F}} \\ K_i\beta & \mapsto & \forall(V \setminus O_i)(\theta \to \|\beta\|_{\mathcal{F}}) \\ [!\psi]\beta & \mapsto & \|\psi\|_{\mathcal{F}} \to \|\beta\|_{\mathcal{F}^{!\psi}} \\ [\psi]_\Delta\beta & \mapsto & \|\psi\|_{\mathcal{F}} \to (\|\beta\|_{\mathcal{F}^{\psi}_{\Delta}})(\frac{p_\psi}{\mathbf{true}}) \end{array}$$

Lemma

For all scenarios $(\mathcal{F}, s)$ and all formulas ϕ:

$$(\mathcal{F}, s) \models \phi \iff (\mathcal{F}, s) \models \|\phi\|_{\mathcal{F}}$$

This means we can use BDDs for all KNS and formulas ☺

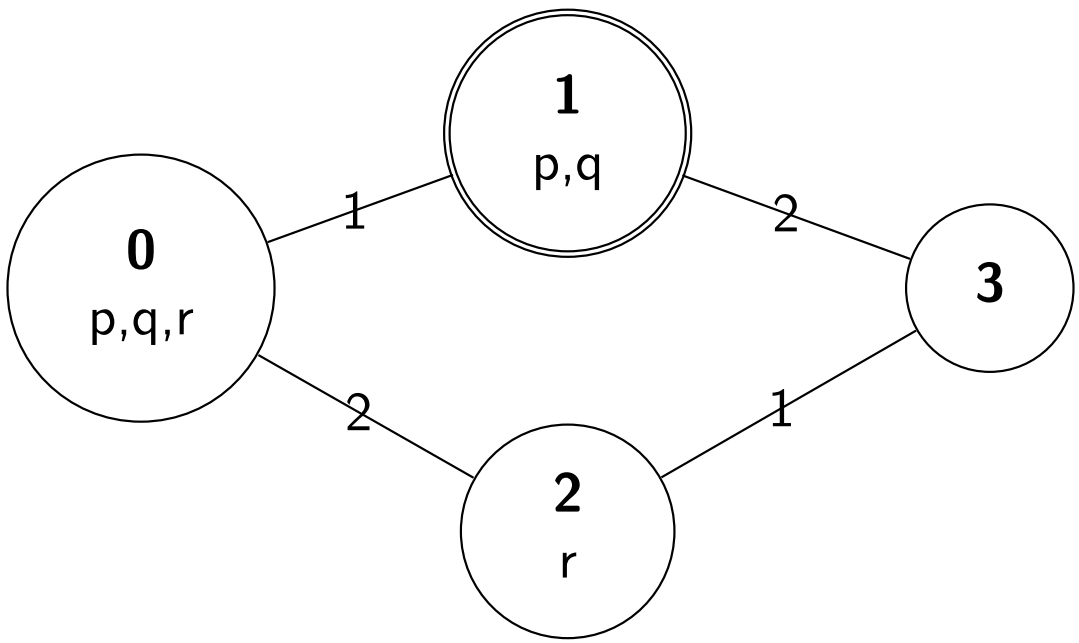
Knowledge Structures are S5 Kripke Models

Example

Consider this knowledge structure:

$$\mathcal{F} = (V = \{p, q, r\}, \theta = p \leftrightarrow q, O_1 = \{q\}, O_2 = \{r\})$$

Here we have $\mathcal{F}, \{p, q\} \models K_1 q$ and $\mathcal{F}, \{p, q\} \models \neg K_2 q$.



Knowledge Structures are S5 Kripke Models

Theorem

For any knowledge structure $\mathcal{F} = (V, \theta, O_i)$ there is an equivalent S5 Kripke model, namely:

$$\mathcal{M} = (W := \mathcal{P}(V), R_i := \{(s,t) \mid s \cap O_o = t \cap O_i\}, Val := \mathrm{id})$$

Theorem

For any S5 Kripke model, there is an equivalent knowledge structure. (Idea: Label all equivalence classes with (sets of) additional propositions to describe the observations.)

Muddy Children

Muddy Children

Father: “At least one of you has mud on their face.”
Father: “Do you know if you have mud on your face?”
Nobody reacts.
Father: “Do you know if you have mud on your face?”
Nobody reacts.
Father: “Do you know if you have mud on your face?”
All three children: “Yes, I do have.”

Muddy Children in DEL

The fact that after "Do you know if you have mud on your face?" nobody reacts means that nobody knows their own state:

$$\psi := \bigwedge_{i \in I} (\neg (K_i p_i \vee K_i \neg p_i))$$

The whole story:

$$[!(p_1 \vee p_2 \vee p_3)][!\psi][!\psi](\bigwedge_{i \in I} (K_i p_i)$$

For details and the standard analysis with Kripke semantics, see for example [DHK07, p. 93-96].

Muddy Children on Knowledge Structures

Initial Model:

$$\mathcal{F}_0 = \left(\{p_1, p_2, p_3\}, \top, \begin{array}{l} O_1 = \{p_2, p_3\} \\ O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

Announcing a formula means to add ($\wedge$) it to the law (θ) of the knowledge structure. After "At least one of you is dirty":

$$\mathcal{F}_1 = \left(\{p_1, p_2, p_3\}, p_1 \vee p_2 \vee p_3, \begin{array}{l} O_1 = \{p_2, p_3\} \\ O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

Muddy Children on Knowledge Structures

After "At least one of you is dirty.":

$$\mathcal{F}_1 = \left(\{p_1, p_2, p_3\}, p_1 \vee p_2 \vee p_3, \begin{array}{l} O_1 = \{p_2, p_3\} \\ O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

Announcing an epistemic formula means to add its boolean equivalent to the law of the knowledge structure.
After "Nobody knows their own state.":

$$\mathcal{F}_2 = \left(\{p_1, p_2, p_3\}, (p_1 \vee p_2 \vee p_3) \wedge \|\psi\|_{\mathcal{F}_1}, \begin{array}{l} O_1 = \{p_2, p_3\} \\ O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

Muddy Children on Knowledge Structures

$$
\begin{aligned}
\|K_1 p_1\|_{\mathcal{F}_1} &= \forall(V \setminus O_1)(\theta_1 \to \|p_1\|_{\mathcal{F}_1}) \\
&= \forall p_1((p_1 \vee p_2 \vee p_3) \to p_1) \\
&= ((\top \vee p_2 \vee p_3) \to \top) \wedge ((\bot \vee p_2 \vee p_3) \to \bot) \\
&= \top \wedge (\neg(p_2 \vee p_3)) \\
&= \neg(p_2 \vee p_3)
\end{aligned}
$$

$$
\begin{aligned}
\|K_1 \neg p_1\|_{\mathcal{F}_1} &= \forall(V \setminus O_1)(\theta_1 \to \|\neg p_1\|_{\mathcal{F}_1}) \\
&= \forall p_1((p_1 \vee p_2 \vee p_3) \to \neg p_1) \\
&= ((\top \vee p_2 \vee p_3) \to \neg\top) \wedge ((\bot \vee p_2 \vee p_3) \to \neg\bot) \\
&= (\top \to \bot) \wedge ((p_2 \vee p_3) \to \top) \\
&= \bot
\end{aligned}
$$

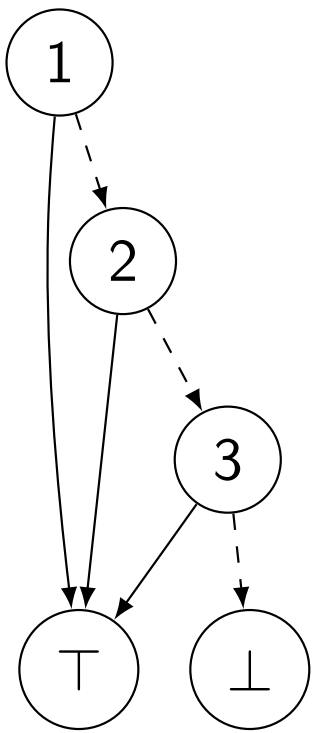
Similarly for agents 2 and 3. The announced formula then becomes

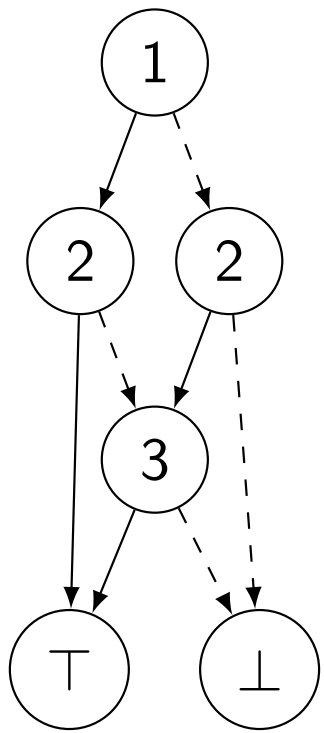
$$
\begin{aligned}
\|\psi\|_{\mathcal{F}_1} &= \|\bigwedge_{i \in I}(\neg(K_i p_i \vee K_i \neg p_i))\|_{\mathcal{F}_1} = \bigwedge_{i \in I} \|\neg(K_i p_i \vee K_i \neg p_i)\|_{\mathcal{F}_1} \\
&= \neg(\neg(p_2 \vee p_3)) \wedge \neg(\neg(p_1 \vee p_3)) \wedge \neg(\neg(p_1 \vee p_2)) \\
&= (p_2 \vee p_3) \wedge (p_1 \vee p_3) \wedge (p_1 \vee p_2)
\end{aligned}
$$

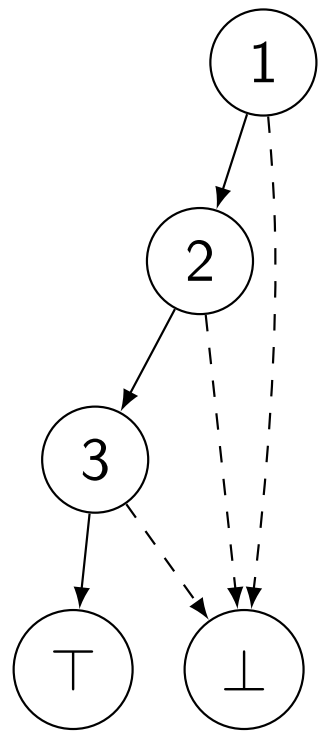
Muddy Children on Knowledge Structures

$$\mathcal{F}_k = \left(\{p_1, p_2, p_3\}, \theta_k, \begin{array}{l} O_1 = \{p_2, p_3\} \\ O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

where θ_1, θ_2 and θ_3 are given by:







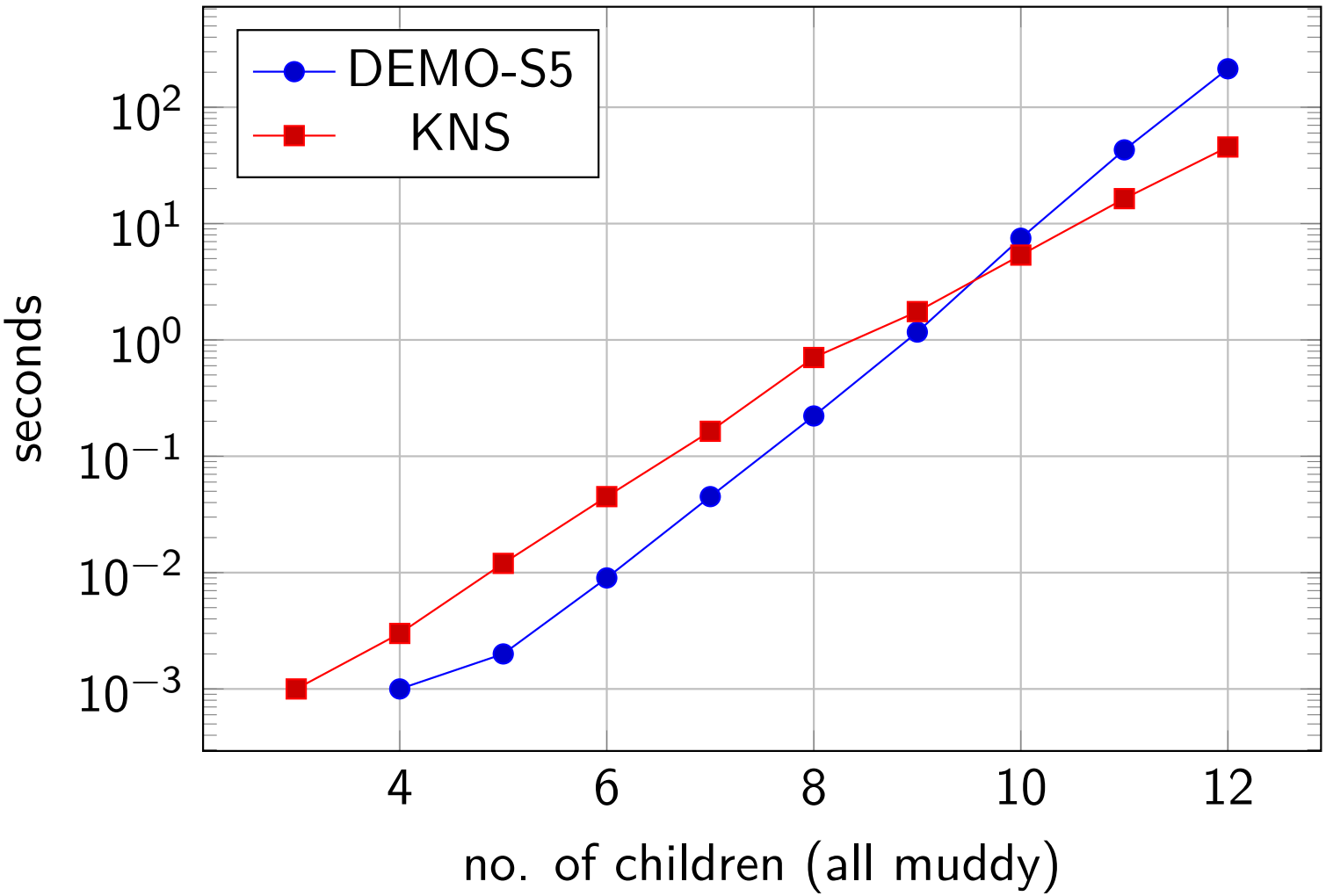
Performance

A Benchmark for Epistemic Model Checking
For n children (m of them muddy) how many announcements of ψ are needed until someone knows their own state?

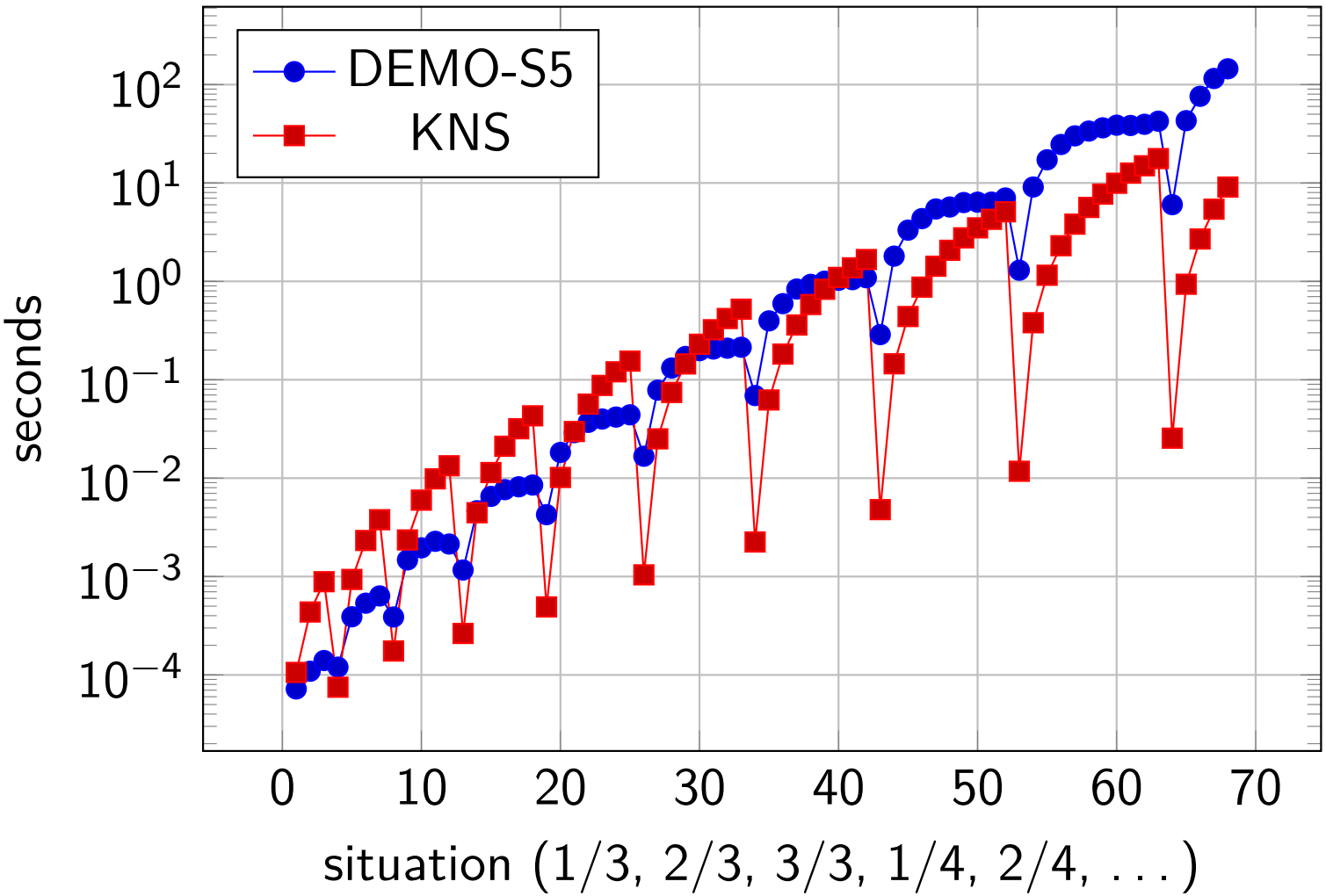
Hint: $m - 1$

We compare DEMO-S5 from [vE14] with the new BDD approach.

Performance: Overview



Performance: Details



Product Update

Group announcements

Our language also contains $[\psi]_\Delta \phi$ which says that after announcing ψ to the group Δ, ϕ holds.

On Knowledge Structures, this is true at s of

$$\mathcal{F} = (V, \theta, (O_i)_{i \in I})$$

iff $\mathcal{F}, s \vDash \psi$ implies that ϕ is true at $s \cup \{p_\psi\}$ of

$$\mathcal{F}' = (V \cup \{p_\psi\}, \theta \wedge (p_\psi \to \|\psi\|_\mathcal{F}), O'_i)_{i \in I})$$

where O'_i contains p_ψ iff $i \in \Delta$.

NB: This is *not a secret* announcement.

Now generalize this idea for any S5 product update from [BMS98].

The general case: Product Update

Similar to the product update

$$\text{Kripke Model} \otimes \text{Action Model} \to \text{Kripke Model}$$

we can define a general update mechanism

$$\text{KNS} \otimes \text{Knowledge Transformer} \to \text{KNS}$$

Definition

A *knowledge transformer* for vocabulary V is a tuple

$$\mathcal{X} = (V^+, \mu, O_1^+, \ldots, O_n^+)$$

where

1. V^+ is a set of atomic propositions such that $V \cap V^+ = \varnothing$,
2. μ is a (possibly epistemic!) formula over $V \cup V^+$ and
3. $O_i^+ \subseteq V^+$ for all agents i.

The general case: Product Update

Definition

Given $\mathcal{F} = (V, \theta, O_1, \ldots, O_n)$ and $\mathcal{X} = (V^+, \mu, O_1^+, \ldots, O_n^+)$, let

$$\mathcal{F} \otimes \mathcal{X} := (V \cup V^+, \theta \wedge ||\mu||_{\mathcal{F}}, O_1 \cup O_1^+, \ldots, O_n \cup O_n^+)$$

Theorem

For every Knowledge Transformer there is an equivalent S5 Action Model and vice versa.

Open Questions

Open Questions

- How is the performance on other puzzles and protocols? (Sum and Product, Consecutive Numbers, Russian Cards, Dining Cryptographers, . . .)
- Compare to the number triangle approach in [GS11]?
- Optimize the S5→KNS–translation by "recognizing" observational propositions before adding new ones?
- Encode non-S5 models in something similar to KNS?
- Factual change as in [BEK06]?
- Could KNS help us to find new axiomatizations or completeness proofs?

References I

[BEK06] Johan van Benthem, Jan van Eijck, and Barteld Kooi.
Logics of communication and change.
Information and computation, 204(11):1620–1662, 2006.

[BMS98] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki.
The logic of public announcements, common knowledge, and private suspicions.
In I. Bilboa, editor, *Proceedings of TARK'98*, pages 43–56, 1998.

[Bry86] Randal E. Bryant.
Graph-Based Algorithms for Boolean Function Manipulation.
IEEE Transaction on Computers, C-35(8):677–691, August 1986.

[DHK07] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi.
Dynamic epistemic logic, volume 1.
Springer Heidelberg, 2007.

References II

[GS11] Nina Gierasimczuk and Jakub Szymanik.
A note on a generalization of the Muddy Children puzzle.
In Krzysztof R. Apt, editor, *TARK*, pages 257–264. ACM, 2011.

[LSSC08] Xiangyu Luo, Kaile Su, Abdul Sattar, and Yan Chen.
Solving Sum and Product Riddle via BDD-Based Model Checking.
In *Web Intelligence/IAT Workshops*, pages 630–633. IEEE, 2008.

[SSL07] Kaile Su, Abdul Sattar, and Xiangyu Luo.
Model Checking Temporal Logics of Knowledge Via OBDDs1.
The Computer Journal, 50(4):403–420, 2007.

[vE14] Jan van Eijck.
DEMO-S5.
Technical report, CWI, 2014.

Thank you!

Image credit: *Wacken-Besucher feiern im Schlamm*, Focus Online, 05.08.2012.