Learning from the Perspective of Dynamic Epistemic Logic

Jan van Eijck
CWI & ILLC, Amsterdam

ILLC Language and Learning Workshop, March 11, 2013

Abstract

From the perspective of DEL, learning is updating an epistemic situation with new information, and thereby changing the old epistemic situation into a new one.
The most common form is learning from public announcements.
Epistemic situations can be represented as multi-agent Kripke models, and update can be implemented as a function mapping Kripke models to updated Kripke models. Using this technique, we can systematically study the effects of learning by means of epistemic model checking.
We first look at the classical case, and then discuss the extension to a probabilistic setting.

- Intro to DEL

- Intro to DEL
- Example: Three Logicians …

- Intro to DEL
- Example: Three Logicians …
- DEL Representation

- Intro to DEL
- Example: Three Logicians …
- DEL Representation
- Epistemic Model Checking Representation

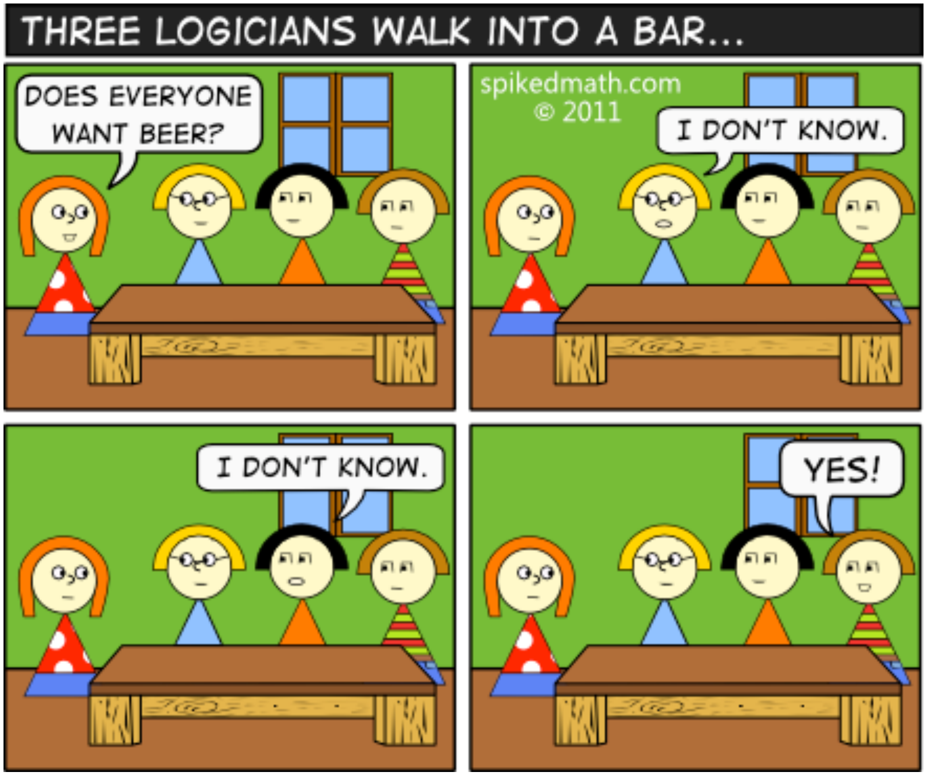
- Intro to DEL
- Example: Three Logicians . . .
- DEL Representation
- Epistemic Model Checking Representation
- Updates and Results: DEMO

- Intro to DEL
- Example: Three Logicians …
- DEL Representation
- Epistemic Model Checking Representation
- Updates and Results: DEMO
- Example: Sum and Product

- Intro to DEL
- Example: Three Logicians …
- DEL Representation
- Epistemic Model Checking Representation
- Updates and Results: DEMO
- Example: Sum and Product
- DEMO

- Intro to DEL
- Example: Three Logicians …
- DEL Representation
- Epistemic Model Checking Representation
- Updates and Results: DEMO
- Example: Sum and Product
- DEMO
- Work in Progress: Probabilistic Epistemic Model Checking

Three Logicians . . .



DEL Representation

●: “wants beer”.

○: “does not want beer”.

—: 1.

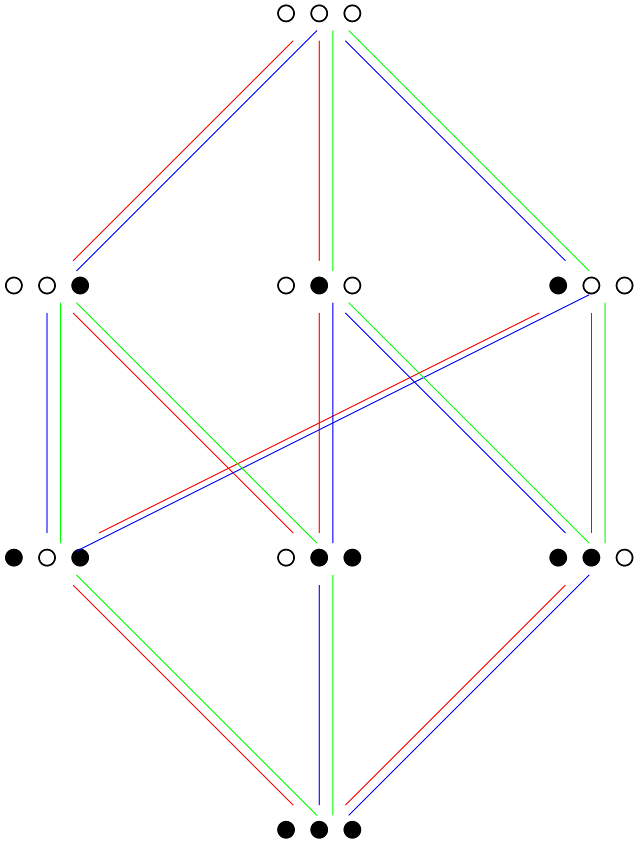
—: 2.

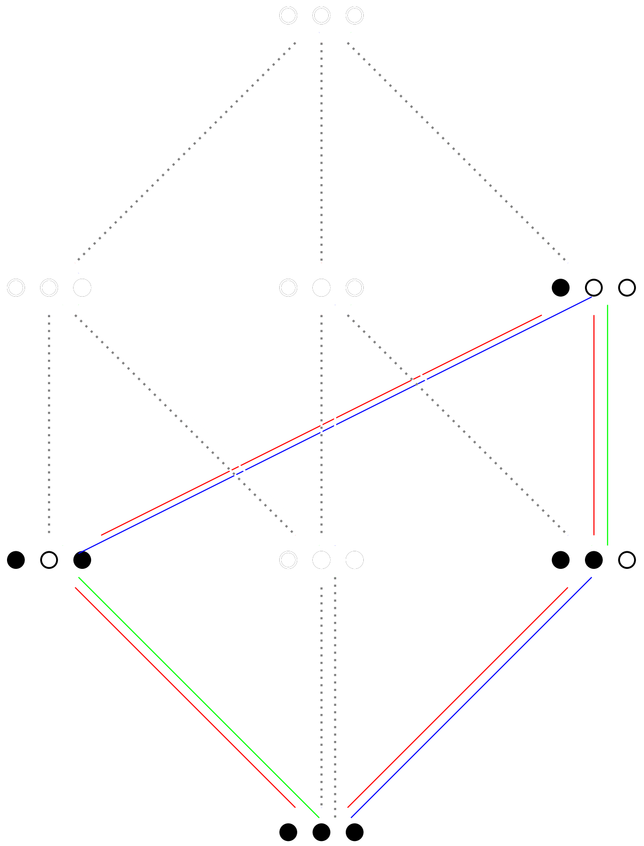
—: 3.

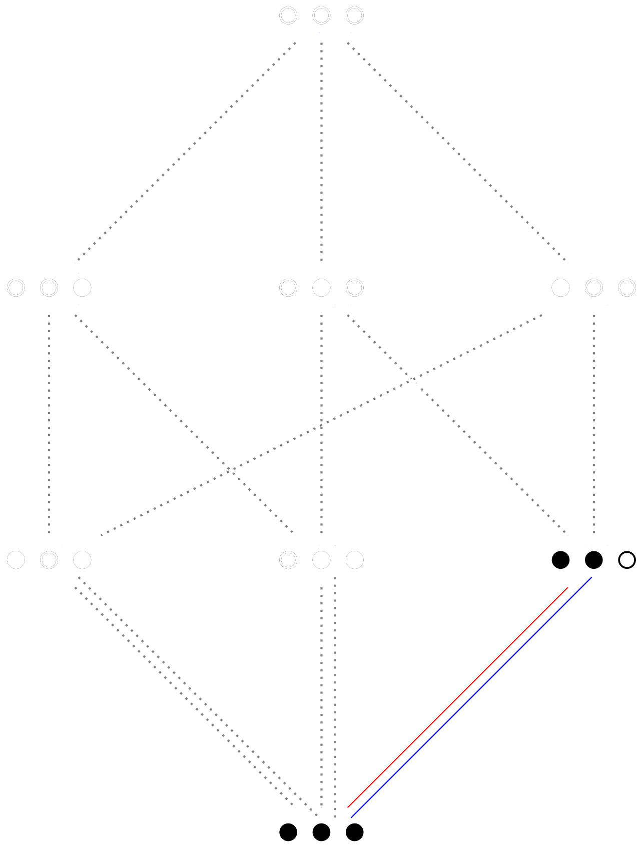
○ ○ ○: nobody wants beer

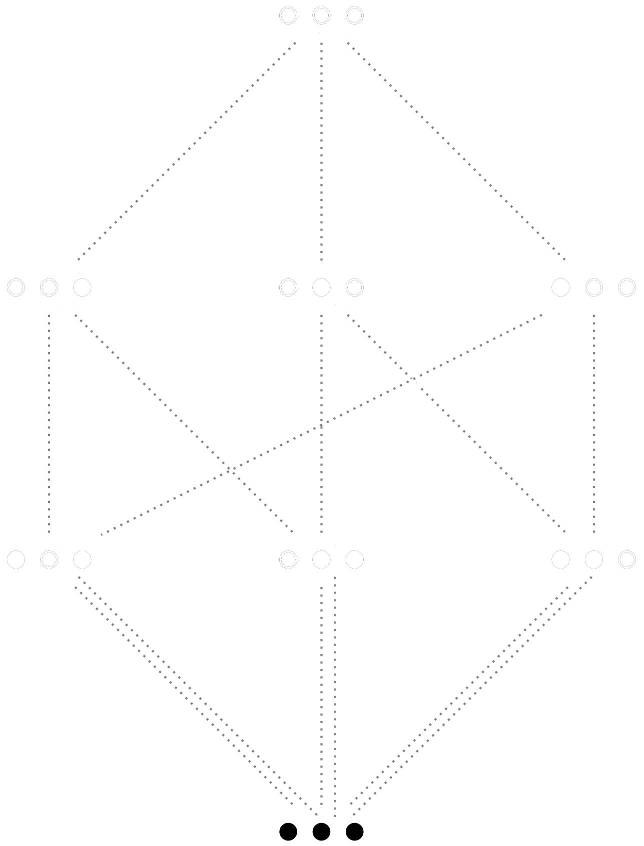
● ○ ○: 1 wants beer, 2 and 3 do not.

And so on . . .









Implementation

```
bools = [True,False]

initBar :: EpistM (Bool,Bool,Bool)
initBar = Mo states [a,b,c] [] rels [(True,True,True)]
  where
  states = [ (b1,b2,b3) | b1 <- bools,
                          b2 <- bools,
                          b3 <- bools ]
  rela = (a,[[(True,x,y)  | x <- bools, y <- bools],
             [(False,x,y) | x <- bools, y <- bools]])
  relb = (b,[[(x,True,y)  | x <- bools, y <- bools],
             [(x,False,y) | x <- bools, y <- bools]])
  relc = (c,[[(x,y,True)  | x <- bools, y <- bools],
             [(x,y,False) | x <- bools, y <- bools]])
  rels = [rela,relb,relc]
```

Initial model

```
*DEMO_S5> initBar
Mo [(True,True,True),(True,True,False),(True,False,True),
 (True,False,False),(False,True,True),(False,True,False),
 (False,False,True),(False,False,False)]
[a,b,c] []
[(a,[[(True,True,True),(True,True,False),(True,False,True),
   (True,False,False)],[(False,True,True),(False,True,False),
   (False,False,True),(False,False,False)]]),
 (b,[[(True,True,True),(True,True,False),(False,True,True),
   (False,True,False)],[(True,False,True),(True,False,False),
   (False,False,True),(False,False,False)]]),
 (c,[[(True,True,True),(True,False,True),(False,True,True),
   (False,False,True)],[(True,True,False),(True,False,False),
   (False,True,False),(False,False,False)]])]
[(True,True,True)]
```

Statements of Ignorance and Knowledge

```
allBeer :: Form (Bool,Bool,Bool)
allBeer = Info (True,True,True)

ignA, ignB, ignC :: Form  (Bool,Bool,Bool)
ignA = Conj [Ng (Kn a allBeer), Ng (Kn a (Ng allBeer))]
ignB = Conj [Ng (Kn b allBeer), Ng (Kn b (Ng allBeer))]
ignC = Conj [Ng (Kn c allBeer), Ng (Kn c (Ng allBeer))]

knowC, knowC' :: Form  (Bool,Bool,Bool)
knowC  = Kn c allBeer
knowC' = Kn c (Ng allBeer)
```

Updates and Results

```
barModel1 = upd_pa initBar ignA

barModel2 = upd_pa barModel1 ignB

barModel3 = upd_pa barModel2 knowC
```

This gives:

```
*DEMO_S5> barModel3
Mo [(True,True,True)] [a,b,c] []
[(a,[[(True,True,True)]]),
 (b,[[(True,True,True)]]),
 (c,[[(True,True,True)]])]
[(True,True,True)]
```

Sum and Product

This famous epistemic puzzle was stated by the Dutch mathematican Hans Freudenthal in a Dutch mathematics journal in 1969.

> A says to S and P: I have chosen two integers x, y such that $1 < x < y$ and $x+y \leq 100$. In a moment, I will inform S only of $s = x + y$, and P only of $p = xy$. These announcements remain private. You are required to determine the pair (x, y). He acts as said. The following conversation now takes place:
>
> 1. P says: “I do not know the pair.”
> 2. S says: “I knew you didn’t.”
> 3. P says: “I now know it.”
> 4. S says: “I now also know it.”
>
> Determine the pair (x, y).

This was solved by combinatorial means in a later issue of the journal.

There is also a version by John McCarthy (see `http://www-formal.stanford.edu/jmc/puzzles.htm`).

A model checking solution with DEMO [2] (based on a DEMO program written by Ji Ruan) was presented in [1]. The present program is an optimized version of that solution.

The list of candidate pairs:

```
pairs :: [(Int,Int)]
pairs  = [ (x,y) |  x <- [2..100], y <- [2..100],
                    x < y, x+y <= 100 ]
```

Initial S+P Model

The initial epistemic model is such that a (representing S) cannot distinguish number pairs with the same sum, and b (representing P) cannot distinguish number pairs with the same product. Instead of using a valuation, we use number pairs as worlds.

```
msnp :: EpistM (Int,Int)
msnp = (Mo pairs [a,b] [] acc pairs)
  where
  acc   = [ (a, [ [ (x1,y1) | (x1,y1) <- pairs,
                               x1+y1 == x2+y2 ] |
                                 (x2,y2) <- pairs ] ) ]
          ++
          [ (b, [ [ (x1,y1) | (x1,y1) <- pairs,
                               x1*y1 == x2*y2 ] |
                                 (x2,y2) <- pairs ] ) ]
```

The Solution

See http://homepages.cwi.nl/~jve/software/demo_s5/

```
solution = upds_pa msnp
           [k_a_statement_1e,
            statement_2e,statement_3e]
```

Demo . . .

Probabilistic Epistemic Model Checking

Example model:

References

[1] H. v. Ditmarsch, J. Ruan, and R. Verbrugge. Model checking sum and product. In S. Zhang and R. Jarvis, editors, AI 2005: Advances in Artificial Intelligence: 18th Australian Joint Conference on Artificial Intelligence, volume 3809 of Lecture Notes in Computer Science, pages 790–795. Springer-Verlag GmbH, 2005.

[2] J. v. Eijck. DEMO — a demo of epistemic modelling. In J. v. Benthem, D. Gabbay, and B. Löwe, editors, Interactive Logic — Proceedings of the 7th Augustus de Morgan Workshop, number 1 in Texts in Logic and Games, pages 305–363. Amsterdam University Press, 2007.