

Finding Cheryl's birthday with DEMO

Malvin Gattinger

April 2015

This report shows how to solve the famous riddle from Singapore with epistemic model checking. We use DEMO_S5 from http://homepages.cwi.nl/~jve/software/demo_s5 and a modified version of KRIPKEVIS from <https://w4eg.de/malvin/illc/kripkevis>.

```
16 module CHERYL where
17 import Data.List
18 import Data.Function
19 import DEMO_S5
20 import KRIPKEVIS
```

We first define the set of all possibilities:

```
25 allpos :: [(Int, String)]
26 allpos = [ (15, "May"), (16, "May"), (19, "May"), (17, "June"), (18, "June"),
27           (14, "July"), (16, "July"), (14, "August"), (15, "August"), (17, "August") ]
```

This forms the set of worlds in our initial model. Moreover, also the set of actual worlds is the full set, hence `allpos` occurs twice in the definition below. The two elements of `rels` define the epistemic relations of Albert and Bernard. Instead of listing explicitly which possibilities they can distinguish we use haskell functions to say that they confuse the same day and the same month, respectively.

```
34 initCheryl :: EpistM (Int, String)
35 initCheryl = Mo allpos [a,b] [] rels allpos where
36   rels = [ ( a, groupBy ((==) 'on' snd) allpos ),
37            ( b, groupBy ((==) 'on' fst) (sortBy (compare 'on' fst) allpos) ) ]
```

This is the initial model with all possibilities:

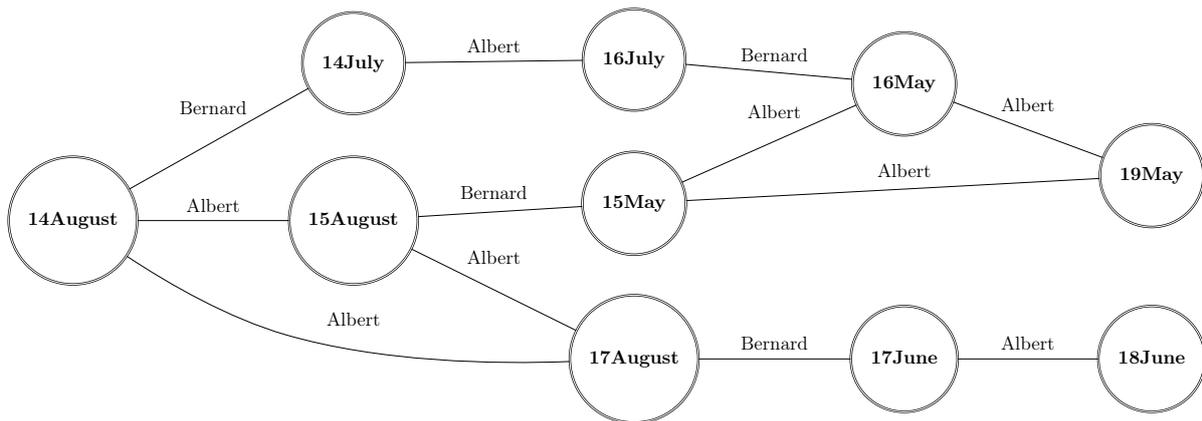


Figure 1: `initCheryl`

The formula saying that i knows Cheryl's birthday is defined as the disjunction over all statements of the form "Agent i knows that the birthday is s ":

```
45 knWhich :: Agent -> Form (Int, [Char])
46 knWhich i = Disj [ Kn i (Info s) | s <- allpos ]
```

Now we update the model three times, using the function `upd_pa` for public announcements. First with Albert: I don't know when Cheryl's birthday is and I know that Bernard does not know.

```
53 model2 = upd_pa initCheryl (Conj [Ng $ knWhich a, Kn a $ Ng (knWhich b)])
```

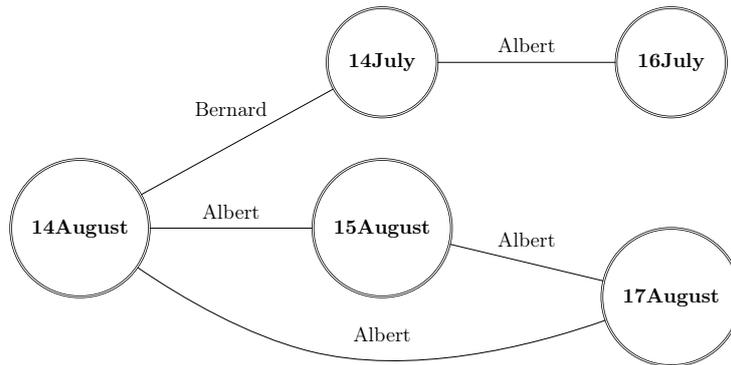


Figure 2: model2

The second announcement by Bernard: “Now I know when Cheryl's birthday is.”

```
60 model3 = upd_pa model2 (knWhich b)
```

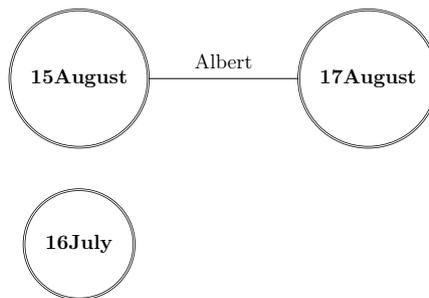


Figure 3: model3

Finally, Albert says: “Now I also know when Cheryl's birthday is.”

```
67 model4 = upd_pa model3 (knWhich a)
```



Figure 4: model4

Lastly, this helper function uses `texModel` from KRIPEVIS to generate the drawings:

```
76 myTexModel :: EpistM (Int,String) -> String -> IO String
77 myTexModel (Mo states _ _ rels pointed) fn =
78   texModel showState showAg showVal "" (VisModel states rels [(s,0)|s<-states] pointed) fn
79   where
80     showState (n,string) = (show n) ++ string
81     showVal _ = ""
82     showAg i = if i==a then "Albert" else "Bernard"
```