testkf/epi1-test2

```
-- One event.
-- _
One identifies the following strings:

        announce_H.
        announce_T.
        aly_peek_H.
        aly_peek_T.
        bob_peek_H.
        bob_peek_T.

++++++++++++++++++

-- Two events.
-- (_ ; _)
Two identifies the following strings:

        announce_H.announce_H.
        announce_H.aly_peek_H.
        announce_T.announce_T.
        aly_peek_H.announce_H.
        announce_H.bob_peek_H.
        announce_T.aly_peek_T.
        aly_peek_H.aly_peek_H.
        aly_peek_T.announce_T.

++++++++++++++++++

-- 5a. Worlds where every alternative for aly is aly_peek_H
-- out intersection at the top.
-- i.e. aly_peek_H is the sole alternative for aly.
-- It should be the unit set of aly_peek_H.
-- Without the intersection we get longer worlds in epik
-- (One & ˜aly(˜aly_peek_H))
AlyBoxAlyPeek identifies the following strings:

        aly_peek_H.

+++++++++++++++++++++++

-- 5b. Similar but with iteration of belief.
-- (One & ˜aly(˜˜aly(˜aly_peek_H)))
AlyBoxAlyPeek identifies the following strings:

        aly_peek_H.

+++++++++++++++++++++++

-- 6a. Worlds where every alternative for aly is bob_peek_H
-- i.e. bob_peek_H is the sole alternative for aly.
-- This should be null. There is no way for aly to get this
-- information in one step.
-- (One & ˜aly(˜bob_peek_H))
AlyBoxBobPeek identifies the following strings:


+++++++++++++++++++++++

-- 6b. Worlds where some alternative for aly is bob_peek_H
-- This should be bob_peek_H + bob_peek_T
-- (One & aly(bob_peek_H))
AlyBoxBobPeek identifies the following strings:

        bob_peek_H.
        bob_peek_T.
```

testkf/epi1-test2

```
++++++++++++++++++++++

-- 7. Worlds where every alternative for aly is of the form  (bob_peek_H.announce_H)
-- This lets Aly learn that it is H in the second event.
-- The result should be the unit set of bob_peek_H.announce_H.
-- (Two & ~aly(~(bob_peek_H ; announce_H)))
Example7 identifies the following strings:

        bob_peek_H.announce_H.

++++++++++++++++++++

-- 8. Similar but where the second event is any event.
-- The result should be bob_peek_H.announce_H + bob_peek_H.aly_peek_H
-- (Two & ~aly(~(bob_peek_H ; _)))
Example8 identifies the following strings:

        bob_peek_H.announce_H.
        bob_peek_H.aly_peek_H.

++++++++++++++++++++

-- 9a. Bob knows that Aly knows that Bob peeked T in the first step
-- The result should be bob_peek_H.announce_H + bob_peek_H.aly_peek_H
-- (Two & ~bob(~~aly(~(bob_peek_T ; _))))
Example9a identifies the following strings:

        bob_peek_T.announce_T.
        bob_peek_T.aly_peek_T.

++++++++++++++++++++

-- 9b. Bob knows that Aly knows that Aly peeked T in the first step
-- The result should be bob_peek_H.announce_H + bob_peek_H.aly_peek_H
-- (Two & ~bob(~~aly(~(aly_peek_T ; _))))
Example9b identifies the following strings:

        aly_peek_T.announce_T.
        aly_peek_T.aly_peek_T.
        aly_peek_T.bob_peek_T.

++++++++++++++++++++
```

testkf/epi1-test2

```
-- One event.
-- Event
T bob_peek_T T
T aly_peek_T T
T announce_T T
H bob_peek_H H
H aly_peek_H H
H announce_H H
-------------------------------------
-- Two events.
-- Cn(Event,Event)
T announce_T T bob_peek_T T
T announce_T T aly_peek_T T
T announce_T T announce_T T
T aly_peek_T T bob_peek_T T
T aly_peek_T T aly_peek_T T
T aly_peek_T T announce_T T
T bob_peek_T T bob_peek_T T
T bob_peek_T T aly_peek_T T
T bob_peek_T T announce_T T
H announce_H H bob_peek_H H
H announce_H H aly_peek_H H
H announce_H H announce_H H
H aly_peek_H H bob_peek_H H
H aly_peek_H H aly_peek_H H
H aly_peek_H H announce_H H
H bob_peek_H H bob_peek_H H
H bob_peek_H H aly_peek_H H
H bob_peek_H H announce_H H
-------------------------------------
-- 5a. Worlds where every alternative for aly is aly_peek_H
-- out intersection at the top.
-- i.e. aly_peek_H is the sole alternative for aly.
-- It should be the unit set of aly_peek_H.
-- Without the intersection we get longer worlds in epik
-- (One & Box(Ra,World(aly_peek_H)))

H aly_peek_H H
-------------------------------------
-- 5b. Similar but with iteration of belief.
-- (One & Box(Ra,Box(Ra,World(aly_peek_H))))

H aly_peek_H H
-------------------------------------
-- 6a. Worlds where every alternative for aly is bob_peek_H
-- i.e. bob_peek_H is the sole alternative for aly.
-- This should be null. There is no way for aly to get this
-- information in one step.
-- (One & Box(Ra,World(bob_peek_H)))

-------------------------------------
-- 6b. Worlds where some alternative for aly is bob_peek_H
-- This should be bob_peek_H + bob_peek_T
-- (One & Dia(Ra,World(bob_peek_H)))

T bob_peek_T T
H bob_peek_H H
-------------------------------------
-- 7. Worlds where every alternative for aly is of the form  (bob_peek_H.announce_H)
-- This lets Aly learn that it is H in the second event.
-- The result should be the unit set of bob_peek_H.announce_H.
-- (Two & Box(Ra,Cn(World(bob_peek_H),World(announce_H))))

H bob_peek_H H announce_H H
-------------------------------------
```

testkf/epi1-test2

```
-- 8. Similar but where the second event is any event.
-- The result should be bob_peek_H.announce_H + bob_peek_H.aly_peek_H
-- (Two & Box(Ra,Cn(World(bob_peek_H),Event)))

H bob_peek_H H announce_H H
H bob_peek_H H aly_peek_H H
-------------------------------------
-- 9a. Bob knows that Aly knows that Bob peeked T in the first step
-- The result should be bob_peek_H.announce_H + bob_peek_H.aly_peek_H
-- (Two & Box(Rb,Box(Ra,Cn(World(bob_peek_T),Event))))

T bob_peek_T T announce_T T
T bob_peek_T T aly_peek_T T
-------------------------------------
-- 9b. Bob knows that Aly knows that Aly peeked T in the first step
-- The result should be bob_peek_H.announce_H + bob_peek_H.aly_peek_H
-- (Two & Box(Rb,Box(Ra,Cn(World(aly_peek_T),Event))))

T aly_peek_T T announce_T T
T aly_peek_T T bob_peek_T T
-------------------------------------
```