Lecture 31: Common Knowledge

CS 181O
Spring 2016
Kim Bruce

Some slides based on those of Christina Unger

Philip Wadler

Final Projects

- Oral reports start in 10 days. 4/25, 27, 29, and May 2.
- E-mail requests by 4 p.m. today of top three choices.
  - Will fill all first requests before moving to second requests. (Should affect your strategy.!)
  - If more than two for a day then select via highest random number.

Grading of Final Projects

- Written report & program: 50%
- Oral report: 30%
- Attendance at reports and participation: 20%
  - 5% for each report day.

Both Final Reports

- Should include:
  - Project goal
  - Background papers/current state of the art
  - How you approached project
  - What you accomplished with demo
  - What you would do if more time
- Written report: LaTeX article style
  - 7 to 12 pages

Last Time

$$\frac{\alpha \lor \beta,\ \neg\beta \lor \gamma}{\alpha \lor \gamma}$$

- Resolution & Tableaux method for theorem providing.
- See me if you need to use them in your project as I have some sources for code
  - though one has some parser errors — from the author of our text!!

I/O in Haskell

- Value of type (IO a) is an action
  - that may perform some input/output
  - and deliver result of type a
- Forms language from which can call Haskell programs.

I/O

- main :: IO( )   -- "IO action"
- main = putStrLn "Hello World!"
- *where* putStrLn:: String → IO( )
- getLine :: IO String   -- "IO action" returning string

- Want echo = putStrLn getLine
  - Types don't match
  - Need >> = for IO monad!!
  - echo = do str <- getLine

    putStrLn str

```
echo = do str <- getLine
          putStrLn str
```

Connecting Actions

getLine   IO String

String   *putStrLn*   IO ( )

Glued together with >>= or
do
… ← …

More IO

```
ask :: String -> String -> IO()
ask prompt ansPrefix =
            do putStr (prompt++" ")
               response <- getLine
               putStrLn (ansPrefix ++ " " ++ response)

getInteger :: IO Integer
getInteger = do putStr "Enter an integer: "
                line <- getLine
                return (read line)
            -- converts string to Integer then to IO Integer
```

See Io.hs

Communication

- Aim of a communicative discourse is to create common knowledge between me and you.
- Model communication as change of knowledge.
  - Possible worlds thinned by additions of knowledge.

Knowledge

- Let $K_a\phi$ mean agent a knows ϕ
- $C\phi$ means ϕ is common knowledge
- What is common knowledge for a and b?
  - If we inform you that something is the case, say that there is a party on tonight, then not only do you know that there is a party on, but also we know that you know, and you know that we know that you know. And so on, ad infinitum. This is common knowledge.
  - $K_a\phi$, $K_a\phi$, $K_aK_a\phi$, $K_aK_b\phi$, $K_bK_a\phi$, ...

Muddy Children Puzzle

- A number, say n, of children are standing in a circle around their father. There are k ($1 \leq k \leq n$) children with mud on their heads. The children can see each other but they cannot see themselves. In particular they do not know if they themselves have mud on their heads. There is no communication between the children. The children all attended a course on epistemic logic and they can reason with this in a perfect way. Furthermore, they are perfectly honest and do not cheat. Now Father says aloud: 'There is at least one child with mud on its head. Will all the children who know they have mud on their heads please step forward? If no answer, father repeats question. On the kth round all k children with mud step forward.

Solving Muddy Children

- Start with two children:
  - Case 1: Only one muddy
  - Case 2: Both muddy
- What if n children with k muddy?
  - $k = 1$:
  - $k = 2$
  - General k?

Using Possible Worlds

- Let <b_1,...,b_n> represent each possible world, where b_i = true iff ith child is muddy.
  - For simplicity let n = 3, k = 2, and actual world w = <true, false, true>
  - What are accessible worlds for child 1 given knowledge?
    - <true false, true> and <false, false, true>.
  - What does child 1 think are child 3's possible worlds
    - <true false, true>, <false, false, true>, <true false, false>, <false, false, false>
  - What changes when father says at least one is muddy?
    - Lose <false, false, false>

Using Possible Worlds

- What are possible worlds for child 1?
  - <true false, true> and <false, false, true>.
- What does child 1 think are child 3's possible worlds
  - <true false, true>, <false, false, true>, <true false, false>, <false, false, false>
- After father says at least one is muddy?
  - <true false, true>, <false, false, true>, <true false, false>
- If it were <false, false, true> then child 3 knows.
  - Child 3 not stop forward.
  - Therefore child 1 knows it must be <true false, true>
  - Thus child 1 steps forward
- Same reasoning for other children

What was father's role?

- Provided common knowledge: Eliminate state
- Provided knowledge that no one knew their state.

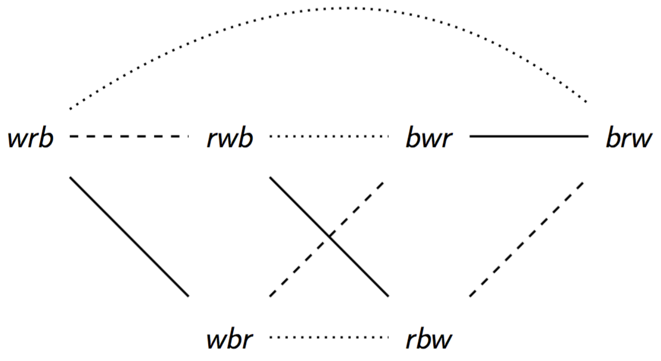
What is impact of public announcement

- Possible worlds disappear

New situation

- Three cards: red, white, blue, and three players: Alice, Bob, and Carol. Deal the three cards out to three players, face down. Each looks at her own card.

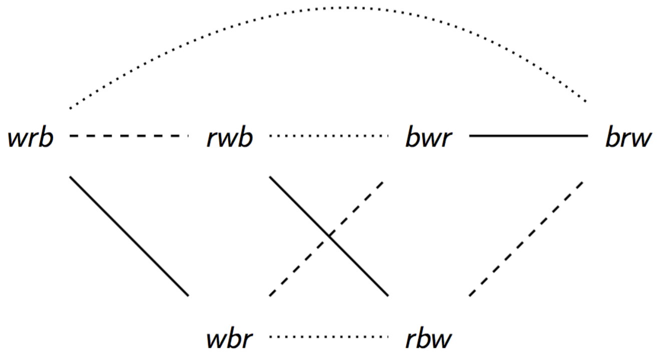
Possible Worlds



Solid = Alice
Small dotted = Bob
Big dotted = Carol

Lines represent accessibility relation for each player

Possible Worlds



What if Alice says her card is red? How does this change possible worlds?
What if Alice says her card is not white

Questions?