Lecture 4: Strengthening

Knowledge and Gossip — ESSLLI 2022

Malvin Gattinger (ILLC, Amsterdam)

2022-08-11, Galway

https://malv.in/2022/gossip/

Main reference:

Hans van Ditmarsch, Malvin Gattinger, Louwe B. Kuijer, Pere Pardo. 2019. "Strengthening Gossip Protocols using Protocol-Dependent Knowledge" *Journal of Applied Logics* 6 (1): 157-203. https://malv.in/2019/StrengtheningGossipProtocols.pdf

Motivation

When does LNS work?

Theorem (yesterday / BIMS paper)

LNS is

- strongly successful iff G is a "sun graph"
- weakly successful iff G is not a "bush" or a "double bush"

Can we improve LNS? – Strengthening Protocols

Depending on the graph, LNS can be strongly or weakly successful!

Can we make it better?

Can we improve LNS? – Strengthening Protocols

Depending on the graph, LNS can be strongly or weakly successful!

Can we make it better?

Informal Idea

Only make a call iff LNS allows it

and you know that it leads to a good situation.

Can we improve LNS? – Strengthening Protocols

Depending on the graph, LNS can be strongly or weakly successful!

Can we make it better?

Informal Idea

Only make a call iff LNS allows it

and you know that it leads to a good situation.

But how do you *know* which calls are okay?

Reminder: Protocol-dependent knowledge

Reminder: Syntax

The *language* of protocol-dependent knowledge:

$$\varphi ::= \top \mid N_i i \mid S_i i \mid C_i i \mid i = i \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i^P \varphi \mid [\pi]\varphi$$

$$\pi ::= ?\varphi \mid ii \mid \pi ; \pi \mid \pi \cup \pi \mid \pi^*$$

Definition

A *protocol* is a function P mapping any agent pair ab to a formula P_{ab} called the *protocol condition*.

Example

The *Learn New Secrets* (LNS) protocol is $LNS_{ab} := \neg S_a b$.

Reminder: Semantics

A *state* is a tuple (G, σ) where $G = (A, N, S)$ is an initial graph and σ a call sequence.

Let N^σ and S^σ be the resulting relations after executing σ.

$$
\begin{array}{lcl}
G, \sigma \models N_x y & :\Leftrightarrow & (x, y) \in N^\sigma \\
G, \sigma \models S_x y & :\Leftrightarrow & (x, y) \in S^\sigma \\
G, \sigma \models C_x y & :\Leftrightarrow & xy \in \sigma \text{ or } yx \in \sigma \\
G, \sigma \models x = y & :\Leftrightarrow & x = y \\
G, \sigma \models K_a^P \varphi & \text{iff} & G, \sigma' \models \varphi \text{ for all } (G, \sigma') \sim_a^P (G, \sigma) \\
G, \sigma \models [\pi]\varphi & \text{iff} & G, \sigma' \models \varphi \text{ for all } (G, \sigma') \in [\![\pi]\!](G, \sigma)
\end{array}
$$

$$
\begin{array}{rcl}
[\![?\varphi]\!](G, \sigma) & := & \{(G, \sigma) \mid G, \sigma \models \varphi\} \\
[\![ab]\!](G, \sigma) & := & \{(G, (\sigma; ab)) \mid G, \sigma \models N_a b\} \\
[\![\pi; \pi']\!](G, \sigma) & := & \bigcup\{[\![\pi']\!](G, \sigma') \mid (G, \sigma') \in [\![\pi]\!](G, \sigma)\} \\
[\![\pi \cup \pi']\!](G, \sigma) & := & [\![\pi]\!](G, \sigma) \cup [\![\pi']\!](G, \sigma) \\
[\![\pi^*]\!](G, \sigma) & := & \bigcup\{[\![\pi^n]\!](G, \sigma) \mid n \in \mathbb{N}\}
\end{array}
$$

Reminder: Epistemic Alternatives

The easy definition, without protocols:

Definition

For any agent a and protocol P let $\sim_a$ be the smallest relation such that:

- $(G, \epsilon) \sim_a (G, \epsilon)$;
- if $(G, \sigma) \sim_a (G, \tau)$, $N_b^\sigma = N_b^\tau$, $S_b^\sigma = S_b^\tau$,
  then $(G, \sigma; ab) \sim_a (G, \tau; ab)$;
  if $(G, \sigma) \sim_a (G, \tau)$, $N_b^\sigma = N_b^\tau$, $S_b^\sigma = S_b^\tau$,
  then $(G, \sigma; ba) \sim_a (G, \tau; ba)$;
- if $(G, \sigma) \sim_a (G, \tau)$ and $a \notin \{c, d, e, f\}$,
  then $(G, \sigma; cd) \sim_a (G, \tau; ef)$.

Note: We only do *synchronous* here.

Reminder: Protocol-dependent Epistemic Alternatives

The tricky definition, with protocols

Definition

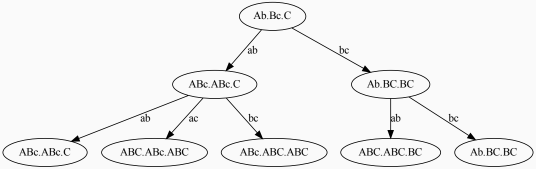
For any agent a and protocol P let $\sim_a^P$ be the smallest relation such that:

- $(G, \epsilon) \sim_a^P (G, \epsilon)$;
- if $(G, \sigma) \sim_a^P (G, \tau)$, $N_b^\sigma = N_b^\tau$, $S_b^\sigma = S_b^\tau$, and $G, \sigma \models P_{ab}$ and $G, \tau \models P_{ab}$,
  then $(G, \sigma; ab) \sim_a^P (G, \tau; ab)$;
  if $(G, \sigma) \sim_a^P (G, \tau)$, $N_b^\sigma = N_b^\tau$, $S_b^\sigma = S_b^\tau$, and $G, \sigma \models P_{ba}$ and at $G, \tau \models P_{ab}$,
  then $(G, \sigma; ba) \sim_a^P (G, \tau; ba)$;
- if $(G, \sigma) \sim_a^P (G, \tau)$ and $a \notin \{c, d, e, f\}$ such that $G, \sigma \models P_{cd}$ and $G, \tau \models P_{ef}$,
  then $(G, \sigma; cd) \sim_a^P (G, \tau; ef)$.

Note: We only do *synchronous* here.

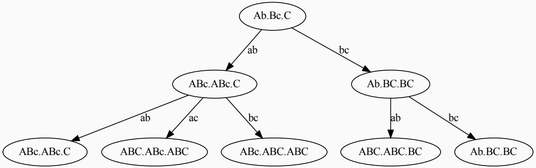
Common knowledge of a protocol prunes the execution tree!

```
GoMoChe> pdf $ treeUpTo 2 (wlog anyCall) (lineInit 3, [])
```

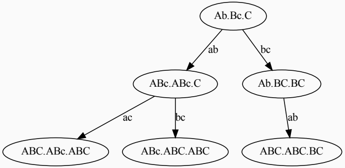


Common knowledge of a protocol prunes the execution tree!

```
GoMoChe> pdf $ treeUpTo 2 (wlog anyCall) (lineInit 3, [])
```




```
GoMoChe> pdf $ treeUpTo 2 (wlog lns) (lineInit 3, [])
```



What can our Language express?

After call ab, they know each others secret: $[ab](S_a b \wedge S_b a)$

Everyone knows all secrets: $Ex := \bigwedge_{i,j} S_i j$

After any of three calls, everyone knows all secrets: $[ab \cup bc \cup ac]Ex$

What can our Language express?

After call ab, they know each others secret: $[ab](S_a b \wedge S_b a)$

Everyone knows all secrets: $Ex := \bigwedge_{i,j} S_i j$

After any of three calls, everyone knows all secrets: $[ab \cup bc \cup ac]Ex$

Learn-New-Secrets condition: $\text{LNS}_{ab} := \neg S_a b$

LNS protocol:

$$\text{LNS} := \left(\bigcup_{a \neq b \in A} (?(N_a b \wedge \neg S_a b); ab) \right)^* ; ? \bigwedge_{a \neq b \in A} \neg (N_a b \wedge \neg S_a b)$$

LNS is strongly successful: $[\text{LNS}]Ex$

LNS is weakly successful: $\langle \text{LNS} \rangle Ex$

Improving LNS with Epistemic Logic

$\mathrm{LNS}_{ab} := \neg S_a b$

Improving LNS with Epistemic Logic

$\text{LNS}_{ab} := \neg S_a b$

Idea: Make call if LNS allows it, and you know that it leads to a good situation.

What is a good situation?

- LNS can still succeed: $\langle LNS \rangle Ex$

Improving LNS with Epistemic Logic

$\text{LNS}_{ab} := \neg S_a b$

Idea: Make call if LNS allows it, and you know that it leads to a good situation.

What is a good situation?

- LNS can still succeed: $\langle LNS \rangle Ex$

We define the **hard strengthening** of *LNS* by:

$$\text{LNS}^{\blacksquare}_{ab} := LNS_{ab} \wedge K_a^{\text{LNS}} [ab] \langle LNS \rangle Ex$$

Improving LNS with Epistemic Logic

$\mathrm{LNS}_{ab} := \neg S_a b$

Idea: Make call if LNS allows it, and you know that it leads to a good situation.

What is a good situation?

- LNS can still succeed: $\langle LNS \rangle Ex$

We define the **hard strengthening** of LNS by:

$$\mathrm{LNS}^{\blacksquare}_{ab} := LNS_{ab} \wedge K_a^{\mathrm{LNS}}[ab]\langle LNS \rangle Ex$$

Historic side note: This is actually why we found/invented K^P in the first place, to avoid self-reference.

If you still worry about Russell here, see Lecture 2 and main reference.

Four different Syntactic Strengthenings

Given protocol: P_{ab}

Hard

$$P^{\blacksquare}_{ab} := P_{ab} \wedge K^P_a[ab]\langle P\rangle Ex$$

Soft

$$P^{\blacklozenge}_{ab} := P_{ab} \wedge \hat{K}^P_a[ab]\langle P\rangle Ex$$

Hard Step-wise

$$P^{\square}_{ab} := P_{ab} \wedge K^P_a[ab](Ex \vee \bigvee_{i,j}(N_ij \wedge P_{ij}))$$

Soft Step-wise

$$P^{\lozenge}_{ab} := P_{ab} \wedge \hat{K}^P_a[ab](Ex \vee \bigvee_{i,j}(N_ij \wedge P_{ij}))$$

Semantic Strengthening: Uniform Backward Induction

Instead of syntactically defining a strengthening, we can also work semantically on the tree or set of call sequences directly!

Semantic Strengthening: Uniform Backward Induction

Instead of syntactically defining a strengthening, we can also work semantically on the tree or set of call sequences directly!

One semantic strengthening is from Game and Decision Theory:

Definition: Uniform Backward Induction/Defoliation ("hard" version)

For any set of sequences X, let $\mathrm{UBI}_P(X)$ be X without $\sigma; ab$ such that

- there is a $\sigma' \in X$ such that
  - $(G, \sigma') \sim_a^P (G, \sigma)$ and
  - $\sigma'; ab$ is terminal in X and
  - $(G, \sigma'; ab) \not\models Ex$.

Semantic Strengthening: Uniform Backward Induction

Instead of syntactically defining a strengthening, we can also work semantically on the tree or set of call sequences directly!

One semantic strengthening is from Game and Decision Theory:

Definition: Uniform Backward Induction/Defoliation ("hard" version)

For any set of sequences X, let $\text{UBI}_P(X)$ be X without $\sigma; ab$ such that

- there is a $\sigma' \in X$ such that
  - $(G, \sigma') \sim_a^P (G, \sigma)$ and
  - $\sigma'; ab$ is terminal in X and
  - $(G, \sigma'; ab) \not\models Ex$.

This is also known as "common knowledge of stable belief in rationality" (Baltag, Smets, and Zvesper 2009) or "common belief in future rationality" (Perea 2014).

$\square = \text{UBI}$

Theorem

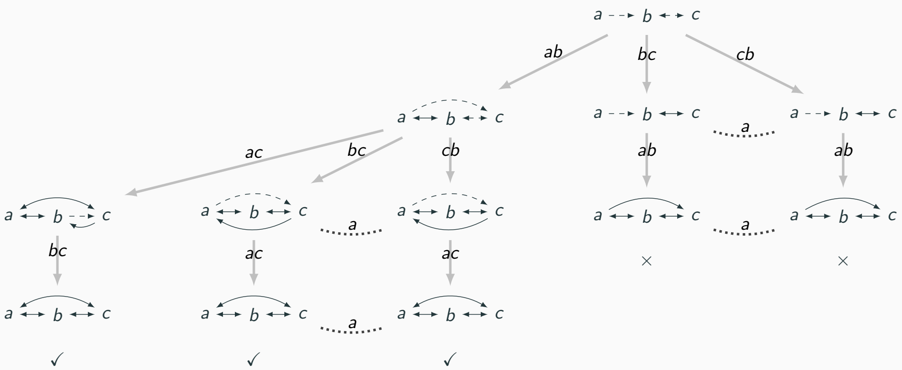
Step-wise Strengthening is the same as Uniform Backward Induction:

$$P^{\square}(G) = \text{UBI}_P(P(G))$$

Good News

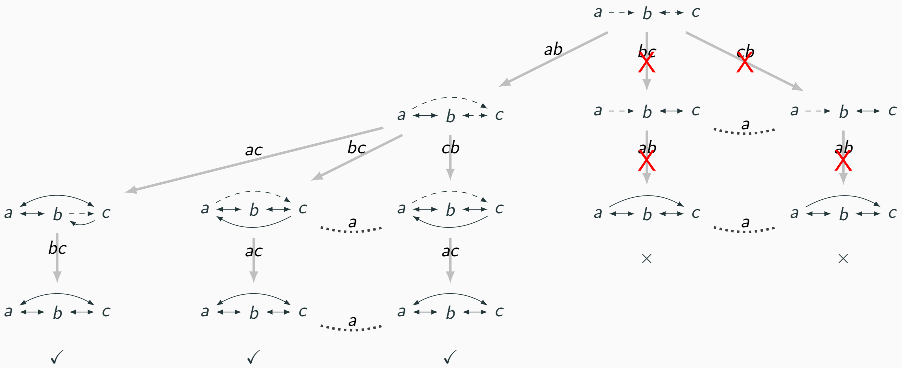
Example of Strongly Successful Strengthening

With LNS:



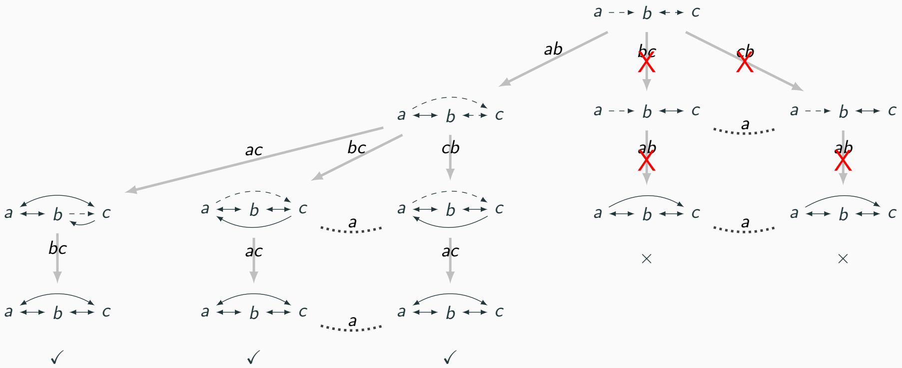
Example of Strongly Successful Strengthening II

With hard strengthening of LNS:



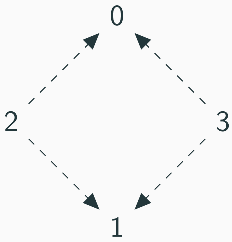
Example of Strongly Successful Strengthening II

With hard strengthening of LNS:



Note: The strengthening "repairs" LNS for this example, but not in general!

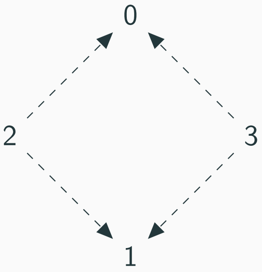
The Diamond Example I



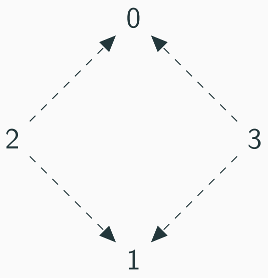
All LNS-sequences up to the decision point:

20; 01	×	21; 10	×	30; 01	×	31; 10	×
20; 21	×	21; 20	×	30; 20; 01	✓	31; 20	✓
20; 30; 01	✓	21; 30	✓	30; 20; 21	✓	31; 21; 10	✓
20; 30; 21	×	21; 31; 10	✓	30; 20; 31	×	31; 21; 20	✓
20; 30; 31	✓	21; 31; 20	×	30; 21	✓	31; 21; 30	×
20; 31	✓	21; 31; 30	✓	30; 31	×	31; 30	×

The Diamond Example II

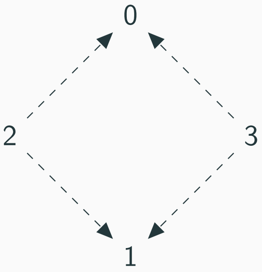


The Diamond Example II



Protocol	successful	unsuccessful
LNS	48	44
$LNS^{\blacksquare}$	8	8
$LNS^{\blacksquare 2}$	0	4
$LNS^{\blacksquare 3}$	0	0
$LNS^{\blacklozenge}$	48	8
$LNS^{\blacklozenge 2}$	48	8
$LNS^{\blacklozenge 3}$	48	8
$LNS^{\square}$	24	36
$LNS^{\square 2}$	8	16
$LNS^{\square 3}$	8	4
$LNS^{\square 4}$	0	4
$LNS^{\square 5}$	0	0
$LNS^{\lozenge}$	48	36
$LNS^{\lozenge 2}$	48	32
$LNS^{\lozenge 3}$	48	32

The Diamond Example II



But there is another LNS strengthening which is strongly successful here!
$\Rightarrow$ Exercise!

Protocol	successful	unsuccessful
LNS	48	44
LNS$^{\blacksquare}$	8	8
LNS$^{\blacksquare 2}$	0	4
LNS$^{\blacksquare 3}$	0	0
LNS$^{\blacklozenge}$	48	8
LNS$^{\blacklozenge 2}$	48	8
LNS$^{\blacklozenge 3}$	48	8
LNS$^{\square}$	24	36
LNS$^{\square 2}$	8	16
LNS$^{\square 3}$	8	4
LNS$^{\square 4}$	0	4
LNS$^{\square 5}$	0	0
LNS$^{\lozenge}$	48	36
LNS$^{\lozenge 2}$	48	32
LNS$^{\lozenge 3}$	48	32

Bad News

The Question

Is there a *perfect strengthening* of LNS?

Formally, is there a protocol which strengthens LNS to become strongly successful on all graphs where the original LNS is weakly successful?

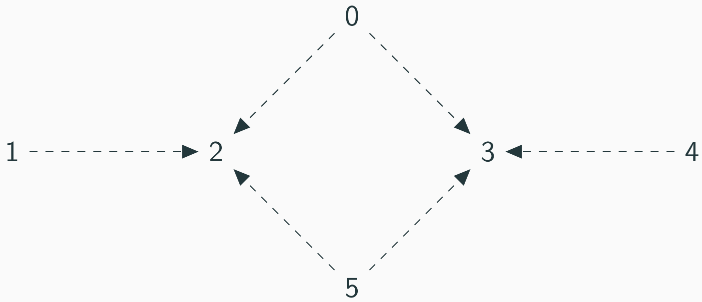
The Question

Is there a *perfect strengthening* of LNS?

Formally, is there a protocol which strengthens LNS to become strongly successful on all graphs where the original LNS is weakly successful?

Hint: No.

The Diamond with Hands aka Candy



Claim

LNS is weakly successful on this graph, but there is no epistemic symmetric protocol that is a strengthening of LNS and that is strongly successful on this graph.

Proof by Exhaustive Search

- LNS is weakly successful here:
  - 02; 12; 53; 43; 13; 03; 23; 52; 42 is successful
  - 02; 12; 53; 43; 13; 03; 52; 42 is unsuccessful

Proof by Exhaustive Search

- LNS is weakly successful here:
  - 02; 12; 53; 43; 13; 03; 23; 52; 42 is successful
  - 02; 12; 53; 43; 13; 03; 52; 42 is unsuccessful

- There are 9468 LNS-sequences for the given graph. 

Proof by Exhaustive Search

- LNS is weakly successful here:
  - 02; 12; 53; 43; 13; 03; 23; 52; 42 is successful
  - 02; 12; 53; 43; 13; 03; 52; 42 is unsuccessful

- There are 9468 LNS-sequences for the given graph. 

- How to check all of them?

Proof by Exhaustive Search

- LNS is weakly successful here:
  - 02; 12; 53; 43; 13; 03; 23; 52; 42 is successful
  - 02; 12; 53; 43; 13; 03; 52; 42 is unsuccessful
- There are 9468 LNS-sequences for the given graph. 
- How to check all of them? Using GoMoChe, obviously ;-)

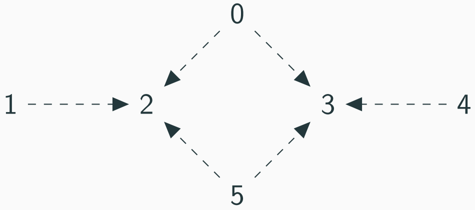
Proof by Exhaustive Search

- LNS is weakly successful here:
  - 02; 12; 53; 43; 13; 03; 23; 52; 42 is successful
  - 02; 12; 53; 43; 13; 03; 52; 42 is unsuccessful
- There are 9468 LNS-sequences for the given graph.
- How to check all of them? Using GoMoChe, obviously ;-)

We use a combination of model checking and “manual” proof by case distinction . . .

Proof Idea

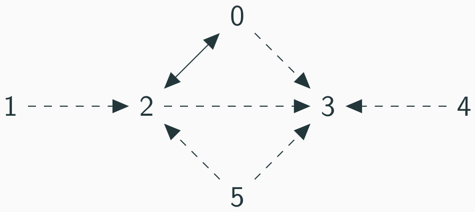
Suppose there is a perfect strengthening of LNS on this graph.



What could be a successful sequence of calls allowed by that protocol?

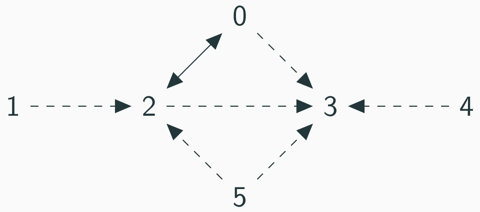
- 0, 1, 4 and 5 do not have incoming arrows $\Rightarrow$ they will never be called.
- If 1 calls 2 first, then 1 never becomes an expert, same for 4 and 3.
- Hence, w.l.o.g. the first call is 02

Proof Idea II



- After 02, can we continue with 12?
- First call could have been 03 which *looks the same* to agent 1.
- But 03; 12 is losing, since then 1 cannot become an expert

Proof Idea II



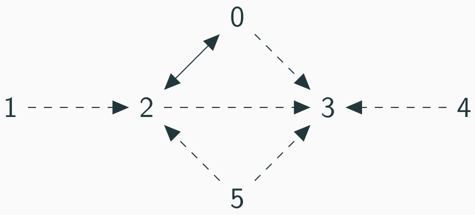
- After 02, can we continue with 12?
- First call could have been 03 which *looks the same* to agent 1.
- But 03; 12 is losing, since then 1 cannot become an expert

(We now use that the protocol is symmetric and epistemic.)

- If 03; 12 is not allowed, also 02; 12 must be forbidden.

⇒ we cannot continue with 12

Proof Idea III



More formally, suppose our new protocol condition is P_{ab}:

$(G, 02) \sim_1 (G, 03)$ implies that $(G, 02) \models P_{12}$ iff $(G, 03) \models P_{12}$

But we must have $(G, 03) \not\models P_{12}$.

Now continue with a lot more case distinctions like this ...

An Impossibility Result

Theorem

There is *no* epistemic protocol which strengthens LNS to become strongly successful on all graphs where the original LNS is weakly successful.

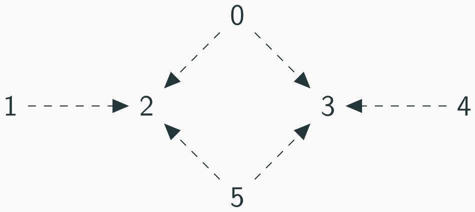
An Impossibility Result

Theorem

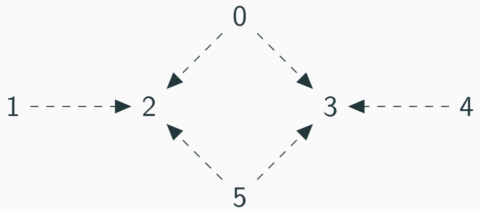
There is *no* epistemic protocol which strengthens LNS to become strongly successful on all graphs where the original LNS is weakly successful.

Note about generality: This theorem is *not* syntax/language dependent. It applies to all (semantic) strengthenings of LNS, even those not in our language.

So what happens if we do it anyway?



So what happens if we do it anyway?



If we apply hard strengthening to this graph, only the first 02 call is allowed. Afterwards we have an empty protocol.

Strengthening in GoMoChe

Strengthening in GoMoChe I

```
GoMoChe> lns
(\(z,y) -> Neg (S z y))
```

Strengthening in GoMoChe I

```
GoMoChe> lns
(\(z,y) -> Neg (S z y))

GoMoChe> strengHard lns
(\(v,u) -> Conj
  [ Neg (S v u)
  , K v (\(z,y) -> Neg (S z y))
    (Box (Call v u)
         (Dia
           (Seq [ Star (CupAg (\y -> CupAg (\z -> Cup
                    [ Seq [ Test (Neg (Same y z))
                          , Seq [Test (Conj [N y z,Neg (S y z)]),Call y z] ]
                    , Seq [ Test (Neg (Neg (Same y z)))
                          , Test Bot ] ])))
                , Test (ForallAg (\y -> ForallAg (\z -> Disj
                    [ Same y z
                    , Disj [Neg (N y z),Neg (Neg (S y z)) ] ]))) ] )
           (ForallAg (\y -> ForallAg (\z -> S y z))) ) ] )
```

Strengthening in GoMoChe II

```
type Strengthening = Protocol -> Protocol

strengHard, strengSoft, strengStepHard, strengStepSoft :: Strengthening
strengHard     p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengSoft     p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengStepHard p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
strengStepSoft p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
```

Strengthening in GoMoChe II

```haskell
type Strengthening = Protocol -> Protocol

strengHard, strengSoft, strengStepHard, strengStepSoft :: Strengthening
strengHard     p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengSoft     p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengStepHard p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
strengStepSoft p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
```

Another strengthening, relevant for tomorrow:

```haskell
super :: Protocol -> Protocol
super proto (x, y) = Conj [ Neg (superExpert x cmo) , proto (x,y) ]
```

Strengthening in GoMoChe II

```haskell
type Strengthening = Protocol -> Protocol

strengHard, strengSoft, strengStepHard, strengStepSoft :: Strengthening
strengHard     p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengSoft     p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengStepHard p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
strengStepSoft p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
```

Another strengthening, relevant for tomorrow:

```haskell
super :: Protocol -> Protocol
super proto (x, y) = Conj [ Neg (superExpert x cmo) , proto (x,y) ]
```

See `src/Gossip/Strengthening.hs`,

Strengthening in GoMoChe II

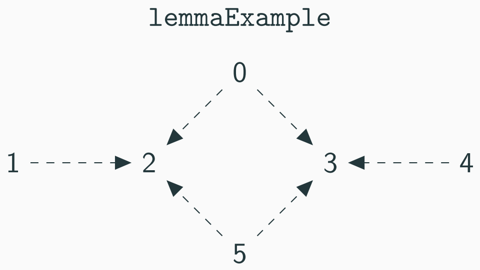
```
type Strengthening = Protocol -> Protocol

strengHard, strengSoft, strengStepHard, strengStepSoft :: Strengthening
strengHard     p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengSoft     p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]
strengStepHard p (a,b) = Conj [p (a,b) , K    a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
strengStepSoft p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Disj [allExperts, protoCanGoOn p])]
```

Another strengthening, relevant for tomorrow:

```
super :: Protocol -> Protocol
super proto (x, y) = Conj [ Neg (superExpert x cmo) , proto (x,y) ]
```

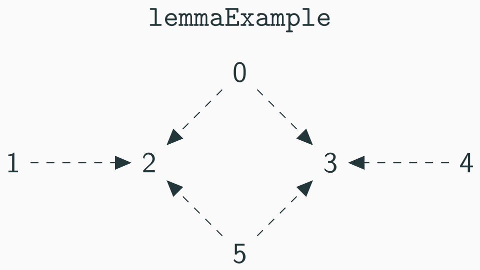
See `src/Gossip/Strengthening.hs`, in particular `diamondProto` and `diamondProtoOld` for a protocol that *is* strongly successful on the diamond example 😺
(and thus not a strengthening of LNS).

Diamond with Hands in GoMoChe




```
GoMoChe> isWeaklySucc localLns lemmaExample
True
```

Diamond with Hands in GoMoChe




```
GoMoChe> isWeaklySucc localLns lemmaExample
True
```

Hard strengthening of LNS is empty after 02 (this takes a while to compute!):

```
GoMoChe> tree (strengHard lns) (lemmaExample,[(0,2)]) == Node (lemmaExample,[(0,2)]) []
True
```

Diamond with Hands in GoMoChe II

```
GoMoChe> showTreeUpToDecision (tree lns (lemmaExample, []))
023-12-2-3-34-235 I6
     (0,2): 023-12-023-3-34-235 02-1-02-3-4-5
          (0,3): ϟ 186
          (1,2): 023-0123-0123-3-34-235 02-012-012-3-4-5
               (0,3): ϟ 76
               (1,3): ϟ 76
               (2,3): ϟ 48
               (4,3): ϟ 96
               (5,2): ϟ 120
               (5,3): 023-0123-0123-235-34-235 02-012-012-35-4-35
                    (0,3): ϟ 18
                    (1,3): ϟ 18
                    (2,3): ϟ 24
                    (3,2): ϟ 24
                    (4,3): 023-0123-0123-2345-2345-235 02-012-012-345-345-35
                         (0,3): ϟ 14
                         (1,3): 023-012345-0123-012345-2345-235 02-012345-012-012345-345-35
                              (0,3): 012345-012345-0123-012345-2345-235 012345-012345-012-012345-345-35
                                   (2,3): ☺ 2
                                   (4,2): ☺ 1
                                   (5,2): ϟ 1
                              (2,3): ☺ 6
                              (4,2): ☺ 2
                              ...
```

Summary

Summary

- We can improve gossip protocols by semantic or syntactic *strengthenings*.
- There is no “perfect” strengthening of LNS.

Summary

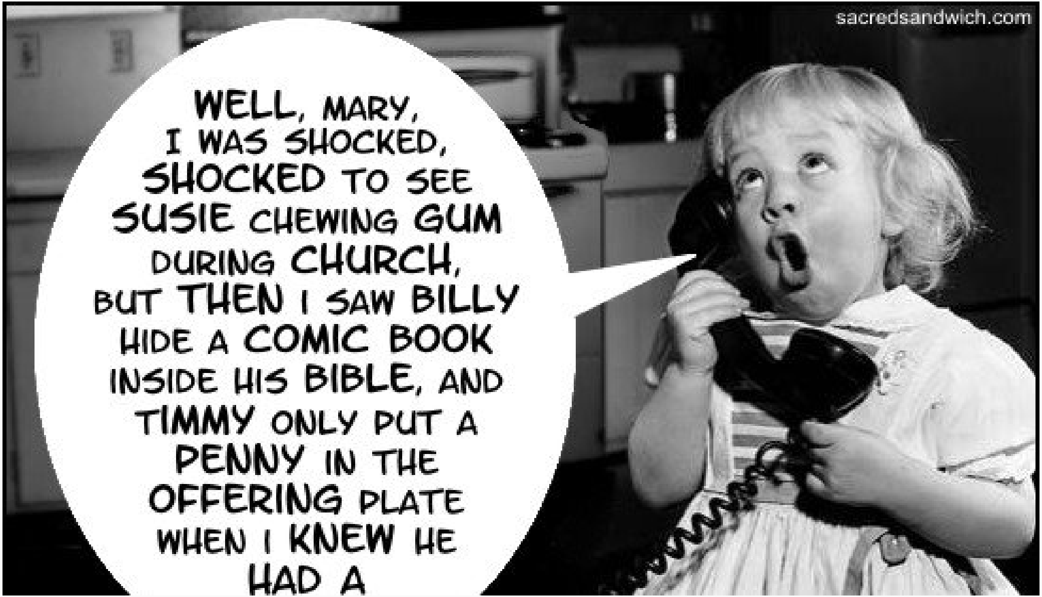
- We can improve gossip protocols by semantic or syntactic *strengthenings*.
- There is no "perfect" strengthening of LNS.

Open Questions

- *How* good are step-wise strengthenings? (They are easier to compute!)
- Is there an incomparable but "LNS-like" protocol that beats LNS on many/most/all graphs?
- Is there a complete axiomatization for K^P (in general, not just for gossip!)?
- When are self-referential strengthenings well-defined?

  $P^*_{ab} := P_{ab} \wedge K_a^{P^*}[ab]\langle P \rangle Ex$

  $P^*_{ab} := P_{ab} \wedge K_a^{P^*}[ab]\langle P^* \rangle Ex$


sacredsandwich.com
WELL, MARY, I WAS SHOCKED, SHOCKED TO SEE SUSIE CHEWING GUM DURING CHURCH, BUT THEN I SAW BILLY HIDE A COMIC BOOK INSIDE HIS BIBLE, AND TIMMY ONLY PUT A PENNY IN THE OFFERING PLATE WHEN I KNEW HE HAD A

References

Main reference:

Hans van Ditmarsch, Malvin Gattinger, Louwe B. Kuijer, Pere Pardo. 2019. "Strengthening Gossip Protocols using Protocol-Dependent Knowledge" *Journal of Applied Logics* 6 (1): 157-203. https://malv.in/2019/StrengtheningGossipProtocols.pdf

Other references:

Baltag, Alexandru, Sonja Smets, and Jonathan Alexander Zvesper. 2009. "Keep 'Hoping' for Rationality: A Solution to the Backward Induction Paradox." *Synthese* 169 (2): 301–33. https://doi.org/10.1007/s11229-009-9559-z.

Perea, Andrés. 2014. "Belief in the Opponents' Future Rationality." *Games and Economic Behavior* 83 (Supplement C): 231–54. https://doi.org/10.1016/j.geb.2013.11.008.