

Lecture 4: Strengthening

Knowledge and Gossip — ESSLLI 2025

Malvin Gatteringer (ILLC, Amsterdam)

2025-07-31, Bochum

<https://malv.in/2025/esslli-gossip>

Motivation

Protocol-dependent knowledge

Good News

Bad News

Strengthening in GoMoChe

The Logic of K^P

Summary

Motivation

When does LNS work?

Theorem (yesterday)

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?

Protocol-dependent knowledge

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 $LNS_{ab} := \neg S_a b$.

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 .

$$G, \sigma \models N_x y \quad :\Leftrightarrow \quad (x, y) \in N^\sigma$$

$$G, \sigma \models S_x y \quad :\Leftrightarrow \quad (x, y) \in S^\sigma$$

$$G, \sigma \models C_x y \quad :\Leftrightarrow \quad xy \in \sigma \text{ or } yx \in \sigma$$

$$G, \sigma \models x = y \quad :\Leftrightarrow \quad x = y$$

$$G, \sigma \models K_a^P \varphi \quad \text{iff} \quad G, \sigma' \models \varphi \text{ for all } (G, \sigma') \sim_a^P (G, \sigma)$$

$$G, \sigma \models [\pi] \varphi \quad \text{iff} \quad G, \sigma' \models \varphi \text{ for all } (G, \sigma') \in \llbracket \pi \rrbracket (G, \sigma)$$

$$\llbracket ?\varphi \rrbracket (G, \sigma) \quad := \quad \{(G, \sigma) \mid G, \sigma \models \varphi\}$$

$$\llbracket ab \rrbracket (G, \sigma) \quad := \quad \{(G, (\sigma; ab)) \mid G, \sigma \models N_a b\}$$

$$\llbracket \pi; \pi' \rrbracket (G, \sigma) \quad := \quad \bigcup \{ \llbracket \pi' \rrbracket (G, \sigma') \mid (G, \sigma') \in \llbracket \pi \rrbracket (G, \sigma) \}$$

$$\llbracket \pi \cup \pi' \rrbracket (G, \sigma) \quad := \quad \llbracket \pi \rrbracket (G, \sigma) \cup \llbracket \pi' \rrbracket (G, \sigma)$$

$$\llbracket \pi^* \rrbracket (G, \sigma) \quad := \quad \bigcup \{ \llbracket \pi^n \rrbracket (G, \sigma) \mid n \in \mathbb{N} \}$$

Epistemic Alternatives (standard)

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.

Protocol-dependent Epistemic Alternatives

The tricky definition, with protocols

Definition

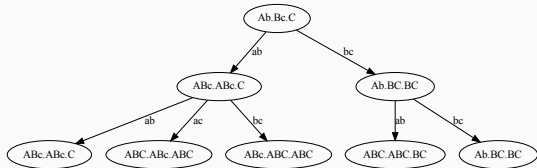
For any agent a and protocol P let \sim_a^P be the smallest relation such that:

- $(G, \epsilon) \sim_a^P (G, \epsilon)$;
- 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.

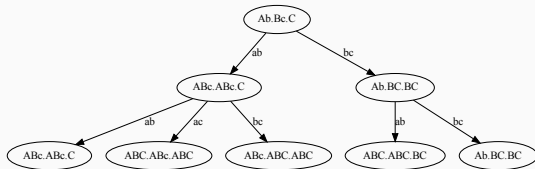
Common knowledge of a protocol prunes the execution tree!

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

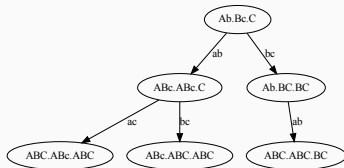


Common knowledge of a protocol prunes the execution tree!

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



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



Avoiding Russel's Protocol

Protocol(condition)s may not refer to themselves!

That is, we do *not* allow this:

$$P_{ab} := \dots K_a^P \dots$$

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: $Ex := \bigwedge_{i,j} S_{ij}$

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: $[ab](S_ab \wedge S_ba)$

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

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

Learn-New-Secrets condition: $LNS_{ab} := \neg S_ab$

LNS protocol:

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

LNS is strongly successful: $[LNS]Ex$

LNS is weakly successful: $\langle LNS \rangle Ex$

Improving LNS with Epistemic Logic

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

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: $\langle \text{LNS} \rangle Ex$

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: $\langle \text{LNS} \rangle Ex$

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

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

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: $\langle \text{LNS} \rangle Ex$

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

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

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 the main reference.

Four different Syntactic Strengthenings

Given protocol: P_{ab}

Hard

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

Soft

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

Hard Step-wise

$$P_{ab}^{\square} := P_{ab} \wedge K_a^P[ab] (Ex \vee \bigvee_{i,j} (N_{ij} \wedge P_{ij}))$$

Soft Step-wise

$$P_{ab}^{\diamond} := P_{ab} \wedge \hat{K}_a^P[ab] (Ex \vee \bigvee_{i,j} (N_{ij} \wedge P_{ij}))$$

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 $UBI_P(X)$ 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
 - $(G, \sigma'; ab) \not\models E_X$.

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 $UBI_P(X)$ 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).

Theorem

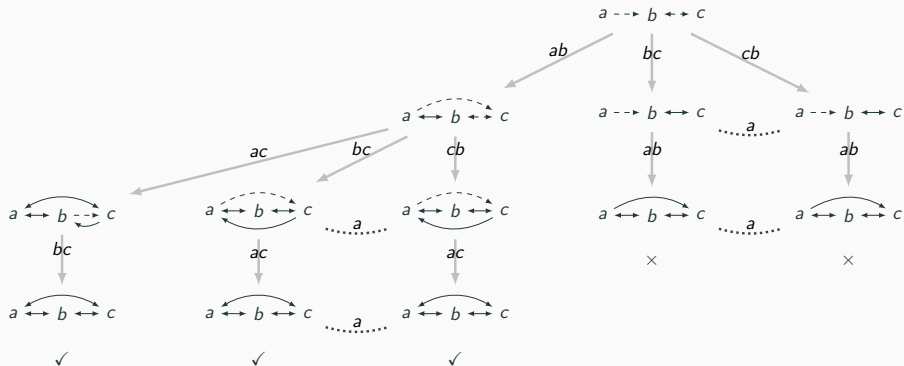
Step-wise Strengthening is the same as Uniform Backward Induction:

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

Good News

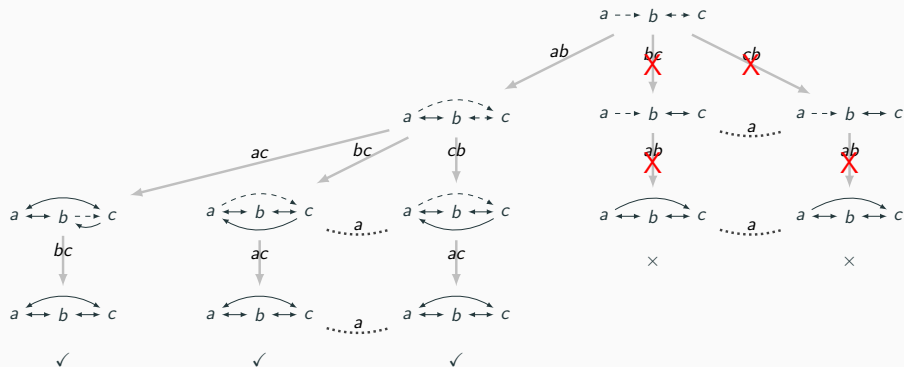
Example of Strongly Successful Strengthening

With LNS:



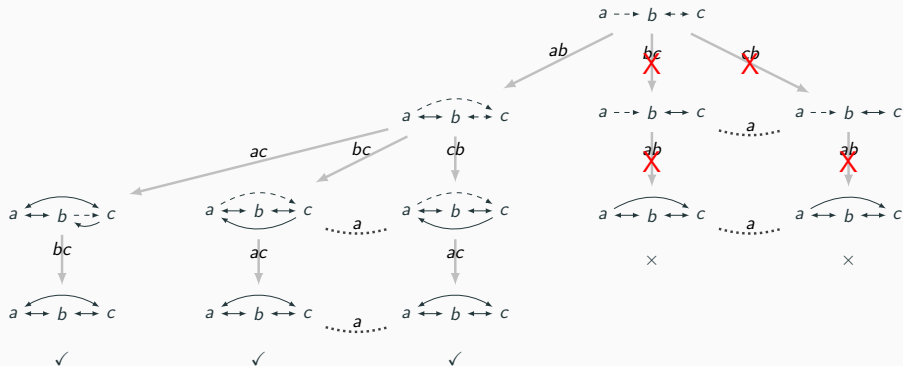
Example of Strongly Successful Strengthening II

With hard strengthening of LNS:



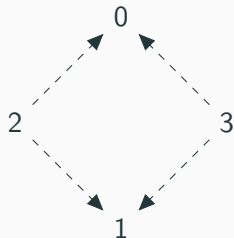
Example of Strongly Successful Strengthening II

With hard strengthening of LNS:



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

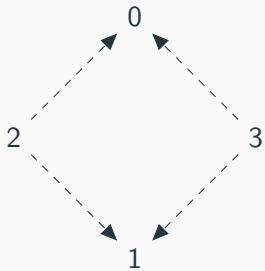
The Diamond Example I



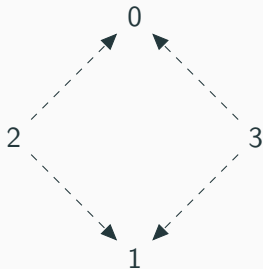
All LNS-sequences up to the decision point:

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

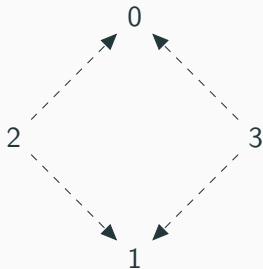


The Diamond Example II



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 \diamond	48	36
LNS \diamond^2	48	32
LNS \diamond^3	48	32

The Diamond Example II



Bonus exercise: but there is another LNS strengthening which is strongly successful here!

Protocol	successful	unsuccessful
LNS	48	44
LNS■	8	8
LNS■ ²	0	4
LNS■ ³	0	0
LNS◆	48	8
LNS◆ ²	48	8
LNS◆ ³	48	8
LNS□	24	36
LNS□ ²	8	16
LNS□ ³	8	4
LNS□ ⁴	0	4
LNS□ ⁵	0	0
LNS◇	48	36
LNS◇ ²	48	32
LNS◇ 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

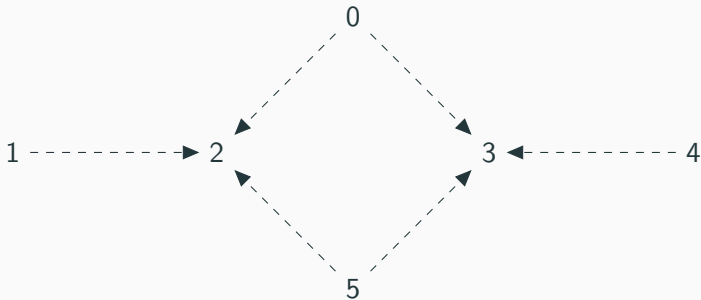
Is there a *perfect strengthening* of LNS?

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

(This example was found by Louwe Kuijer.)



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 ;-)

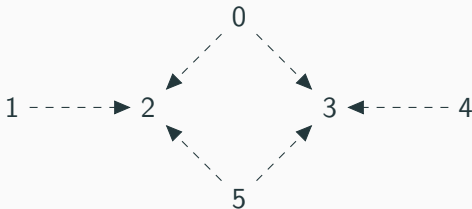
Proof by Exhaustive Search

- 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

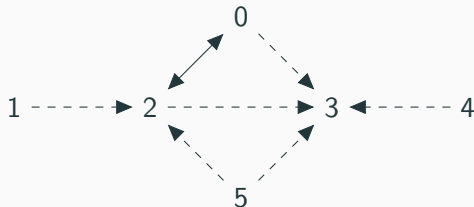
Suppose there is a perfect strengthening of LNS on this graph.



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

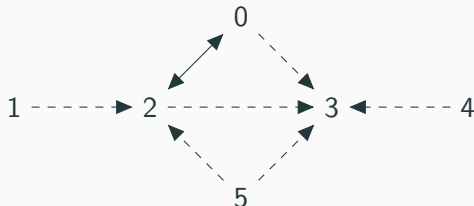
- 0, 1, 4 and 5 do not have incoming arrows \Rightarrow they will never be called.
- If 1 calls 2 first, then 1 never becomes an expert, same for 4 and 3.
- Hence, w.l.o.g. the first call is 02

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



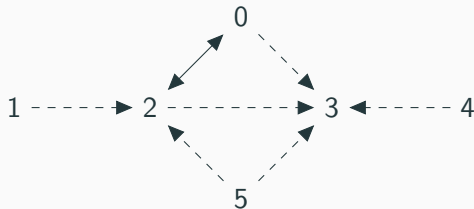
- After 02, can we continue with 12?
- First call could have been 03 which *looks the same* to agent 1.
- But 03; 12 is losing, since then 1 cannot become an expert

(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 $(G, 03) \not\models P_{12}$ to make P strongly successful.

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.

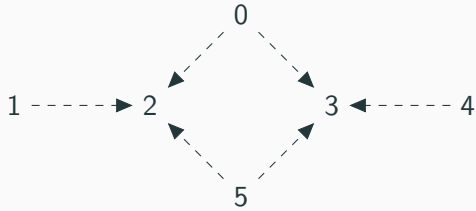
An Impossibility Result

Theorem

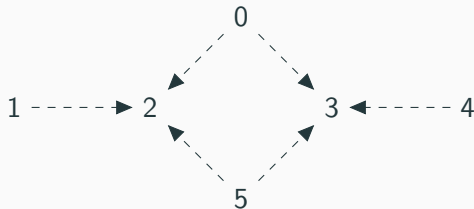
There is *no* epistemic protocol which strengthens LNS to become strongly successful on all graphs where the original LNS is weakly successful.

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

```
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) ]
```


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`,

Strengthening in GoMoChe II

```
type Strengthening = Protocol -> Protocol
```

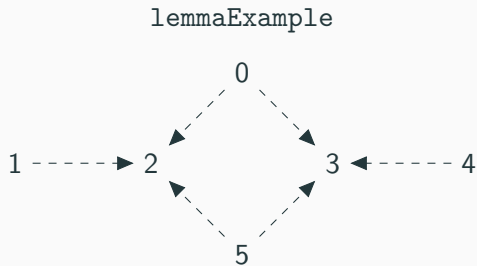
```
strengHard, strengSoft, strengStepHard, strengStepSoft :: Strengthening  
strengHard      p (a,b) = Conj [p (a,b) , K      a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]  
strengSoft      p (a,b) = Conj [p (a,b) , HatK a p $ Box (Call a b) (Dia (protoTerm p) allExperts)]  
strengStepHard  p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

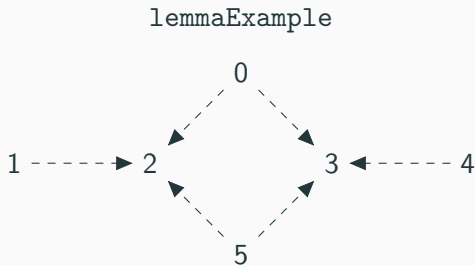
Diamond with Hands in GoMoChe



```
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): ♢ 186
```

```
(1,2): 023-0123-0123-3-34-235 02-012-012-3-4-5
```

```
(0,3): ♢ 76
```

```
(1,3): ♢ 76
```

```
(2,3): ♢ 48
```

```
(4,3): ♢ 96
```

```
(5,2): ♢ 120
```

```
(5,3): 023-0123-0123-235-34-235 02-012-012-35-4-35
```

```
(0,3): ♢ 18
```

```
(1,3): ♢ 18
```

```
(2,3): ♢ 24
```

```
(3,2): ♢ 24
```

```
(4,3): 023-0123-0123-2345-2345-235 02-012-012-345-345-35
```

```
(0,3): ♢ 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): ♢ 1
```

```
(2,3): ⊕ 6
```

```
(4,2): ⊕ 2
```

```
...
```

The Logic of K^P

The following ideas and results are from

- Wouter J. Smit: *Axiomatising Protocol-Dependent Knowledge in Gossip*

MSc thesis, Amsterdam 2024.

<https://eprints.illc.uva.nl/id/eprint/2330/>

and an upcoming DaLí 2025 paper based on this thesis.

If we allow arbitrary protocols P (including non-symmetric and non-epistemic), then K^P can express:

- that some call happened: $K_i^\perp \perp$
- that at least this many calls happened (“counting formulas”)
- that a specific call sequence happened
- ...

If we allow arbitrary protocols P (including non-symmetric and non-epistemic), then K^P can express:

- that some call happened: $K_i^\perp \perp$
- that at least this many calls happened (“counting formulas”)
- that a specific call sequence happened
- ...

Hence K_i^P is *much more expressive* than K_i .

(And it motivates a different notion of bisimulation.)

Question: What is **the logic** of the K_i^P modality?

- How does it compare to K in standard epistemic logic?

Axiomatization of K^P

Question: What is **the logic** of the K_i^P modality?

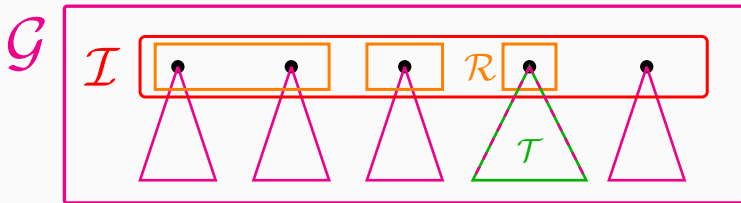
- How does it compare to K in standard epistemic logic?
- It is **S5**, but only *until a protocol is violated*.
- It interacts with the $[ab]$ call modality.

But what other principles / axioms do we need for completeness?

Model Classes

Which class of gossip models do we actually want to axiomatize?

- \mathcal{G} — all gossip models
- \mathcal{I} — all initial models
- \mathcal{R} — the root model
- \mathcal{T} — the tree model (including all its states)



Static Axioms

The following system is sound and complete for the **call-free** language on the *root* model \mathcal{R} .

Prop	propositional tautologies	K	$K_a^P(\varphi \rightarrow \psi) \rightarrow (K_a^P\varphi \rightarrow K_a^P\psi)$
MP	$\vdash \varphi, \vdash \varphi \rightarrow \psi$ imply $\vdash \psi$	T	$K_a^P\varphi \rightarrow \varphi$
Sub	$\vdash \varphi \leftrightarrow \psi$ implies $\vdash \chi \leftrightarrow \chi[\varphi/\psi]$	4	$K_a^P\varphi \rightarrow K_a^PK_a^P\varphi$
Own	S_aa	5	$\neg K_a^P\varphi \rightarrow K_a^P\neg K_a^P\varphi$
Only	O_aa	Nec	$\vdash \varphi$ implies $\vdash K_a^P\varphi$
PFi	$S_ab \rightarrow K_a^PS_ab$		
NPi	$\neg S_ab \rightarrow K_a^P\neg S_ab$	PI	$K^P\varphi \rightarrow K^Q\varphi$

Static Axioms

The following system is sound and complete for the **call-free** language on the *root* model \mathcal{R} .

Prop	propositional tautologies	K	$K_a^P(\varphi \rightarrow \psi) \rightarrow (K_a^P\varphi \rightarrow K_a^P\psi)$
MP	$\vdash \varphi, \vdash \varphi \rightarrow \psi$ imply $\vdash \psi$	T	$K_a^P\varphi \rightarrow \varphi$
Sub	$\vdash \varphi \leftrightarrow \psi$ implies $\vdash \chi \leftrightarrow \chi[\varphi/\psi]$	4	$K_a^P\varphi \rightarrow K_a^PK_a^P\varphi$
Own	S_aa	5	$\neg K_a^P\varphi \rightarrow K_a^P\neg K_a^P\varphi$
Only	O_aa	Nec	$\vdash \varphi$ implies $\vdash K_a^P\varphi$
PFi	$S_ab \rightarrow K_a^PS_ab$		
NPi	$\neg S_ab \rightarrow K_a^P\neg S_ab$	PI	$K^P\varphi \rightarrow K^Q\varphi$

Leaving out **Only** gets us a system complete for \mathcal{I} , the class of all initial models.

Call Reductions

To axiomatize the language with calls we use the following reduction axioms, valid on \mathcal{I} .

Call Basics		Call Effects	
Con	$[ab](\varphi \wedge \psi) \leftrightarrow ([ab]\varphi \wedge [ab]\psi)$	Eff	$[ab]S_c d \leftrightarrow (S_a d \vee S_b d) \quad c \in \{a, b\}$
Fnc	$[ab]\neg\varphi \leftrightarrow \neg[ab]\varphi$	Ext	$[ab]S_c d \leftrightarrow S_c d \quad c \notin \{a, b\}$

Call Reductions

To axiomatize the language with calls we use the following reduction axioms, valid on \mathcal{I} .

Call Basics		Call Effects	
Con	$[ab](\varphi \wedge \psi) \leftrightarrow ([ab]\varphi \wedge [ab]\psi)$	Eff	$[ab]S_c d \leftrightarrow (S_a d \vee S_b d) \quad c \in \{a, b\}$
Fnc	$[ab]\neg\varphi \leftrightarrow \neg[ab]\varphi$	Ext	$[ab]S_c d \leftrightarrow S_c d \quad c \notin \{a, b\}$

Calls and Protocol-Dependent Knowledge

Obs₁	$[ab]K_a^P \varphi \leftrightarrow (P_{ab} \rightarrow \bigvee_{R \subseteq \mathbb{S}} (O_b R \wedge K_a^P (P_{ab} \rightarrow (O_b R \rightarrow [ab]\varphi))))$	$a \in \{a, b\}$
Obs₂	$[ab]K_b^P \varphi \leftrightarrow (P_{ab} \rightarrow \bigvee_{R \subseteq \mathbb{S}} (O_a R \wedge K_b^P (P_{ab} \rightarrow (O_a R \rightarrow [ab]\varphi))))$	$b \in \{a, b\}$
Pri	$[ab]K_c^P \varphi \leftrightarrow (P_{ab} \rightarrow \bigwedge_{d, e \neq a} K_c^P (P_{de} \rightarrow [de]\varphi))$	$c \notin \{a, b\}$

Putting it all together

In standard PAL and DEL axiomatizations we just combine static and dynamic axioms.

Putting it all together

In standard PAL and DEL axiomatizations we just combine static and dynamic axioms.

BUT here we cannot do this: the axiom $\mathbf{T} (K_i^P \varphi \rightarrow \varphi)$ is only valid at initial states.

Example: if $\sigma \not\models P_{ab}$ then $\sigma.ab \models K_i^P \perp$ but still $\sigma.ab \not\models \perp$.



Putting it all together

In standard PAL and DEL axiomatizations we just combine static and dynamic axioms.

BUT here we cannot do this: the axiom \mathbf{T} ($K_i^P \varphi \rightarrow \varphi$) is only valid at initial states.

Example: if $\sigma \not\models P_{ab}$ then $\sigma.ab \models K_i^P \perp$ but still $\sigma.ab \not\models \perp$.



Instead, we decide validity of φ for \mathcal{G} (or \mathcal{T}) as follows:

- Prove that we only need to consider sequences σ up to a certain finite length.
- Rewrite all formulas $[\sigma]\varphi$ to call-free equivalents with the reduction axioms.
- Check whether all those formulas are provable in *lsystem* (or *Rsystem*).

Putting it all together

In standard PAL and DEL axiomatizations we just combine static and dynamic axioms.

BUT here we cannot do this: the axiom $\mathbf{T} (K_i^P \varphi \rightarrow \varphi)$ is only valid at initial states.

Example: if $\sigma \not\models P_{ab}$ then $\sigma.ab \models K_i^P \perp$ but still $\sigma.ab \not\models \perp$.



Instead, we decide validity of φ for \mathcal{G} (or \mathcal{T}) as follows:

- Prove that we only need to consider sequences σ up to a certain finite length.
- Rewrite all formulas $[\sigma]\varphi$ to call-free equivalents with the reduction axioms.
- Check whether all those formulas are provable in *lsystem* (or *Rsystem*).

See Smit (2024) for details.

Summary

Summary

- We can *strengthen* gossip protocols using epistemic logic.
- There is no “perfect” strengthening of LNS.
- All four logic(s) of K^P are decidable.

Summary

- We can *strengthen* gossip protocols using epistemic logic.
- There is no “perfect” strengthening of LNS.
- All four logic(s) of K^P are decidable.

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 of proto-dep Knowledge?~~ (General, not just gossip?)
- When are self-referential strengthenings well-defined?

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

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

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

- Hans van Ditmarsch, Malvin Gattinger, Louwe B. Kuiper, Pere Pardo. 2019. “Strengthening Gossip Protocols using Protocol-Dependent Knowledge” *Journal of Applied Logics* 6 (1): 157-203. <https://malv.in/2019/StrengtheningGossipProtocols.pdf>
- Hans van Ditmarsch, Malvin Gattinger, Wouter J. Smit. 2025. “Completeness and Decidability of Protocol-Dependent Knowledge in Gossip” *DaLí Workshop*. To appear.

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>.