Lecture 1, part 2: ElmGossip

Gossip and Knowledge — ESSLLI 2025

Malvin Gattinger (ILLC, Amsterdam)

2025-07-28, Bochum

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

Dynamic Gossip Graphs “by hand”

Given a gossip graph, we can “make a call” by drawing a new graph.

Or we can *add* edges to the same drawing.

Dynamic Gossip Graphs “by hand”

Given a gossip graph, we can “make a call” by drawing a new graph.

Or we can *add* edges to the same drawing.

But then, what if we want to go back / make a different call?

Dynamic Gossip Graphs “by hand”

Given a gossip graph, we can “make a call” by drawing a new graph.

Or we can *add* edges to the same drawing.

But then, what if we want to go back / make a different call?

What if we want to check many different call sequences?

Dynamic Gossip Graphs “by hand”

Given a gossip graph, we can “make a call” by drawing a new graph.

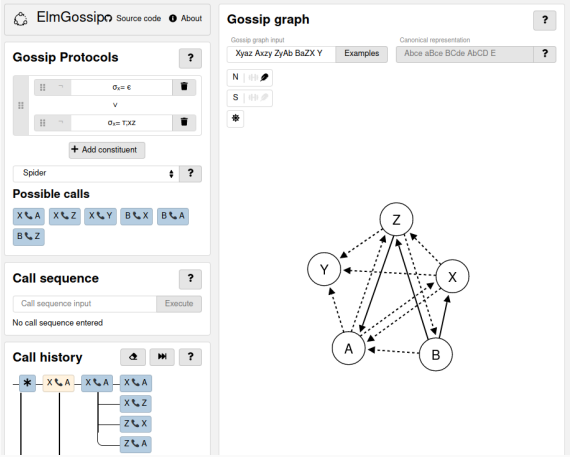
Or we can *add* edges to the same drawing.

But then, what if we want to go back / make a different call?

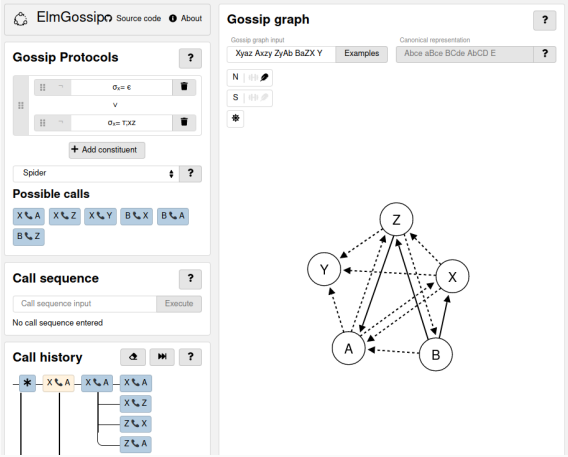
What if we want to check many different call sequences?

This quickly becomes tedious. Hence, let’s automate!

ElmGossip


ElmGossip
Source code
About
Gossip Protocols
Add constituent
Spider
Possible calls
X A
X Z
X Y
B X
B A
B Z
Call sequence
Call sequence input
Execute
No call sequence entered
Call history
Gossip graph
Gossip graph input
Xyaz Axzy ZyAb BaZX Y
Examples
Canonical representation
Abce aBce BCde AbCD E
N
S
Z
Y
X
A
B

ElmGossip

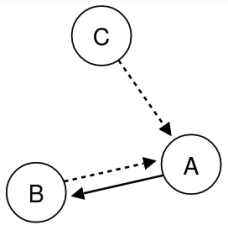


Ramon Meffert: *Tools for Gossip* (2021), Bachelor thesis AI, University of Groningen.

Code: https://github.com/RamonMeffert/elm-gossip

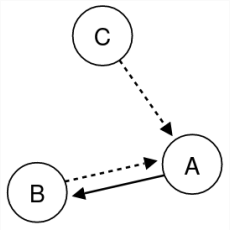
Try it: https://r3n.nl/elm-gossip/

Short notation for gossip graphs



`AB aB aC`

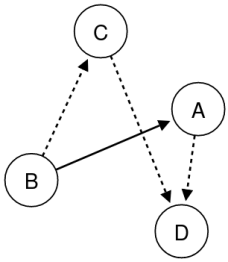
Short notation for gossip graphs



`AB aB aC`

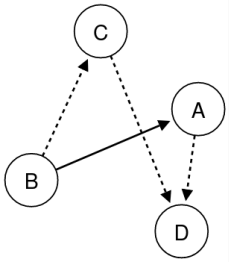
- A graph of n agents is described by n words separated by spaces.
- Knowing the **number** of agent a is denoted by `a`
- Knowing the **secret** of agent a is denoted by `A`

Examples

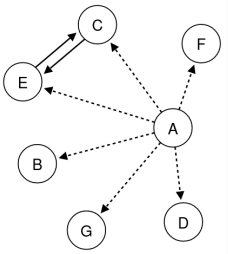


`Ad ABc Cd D`

Examples



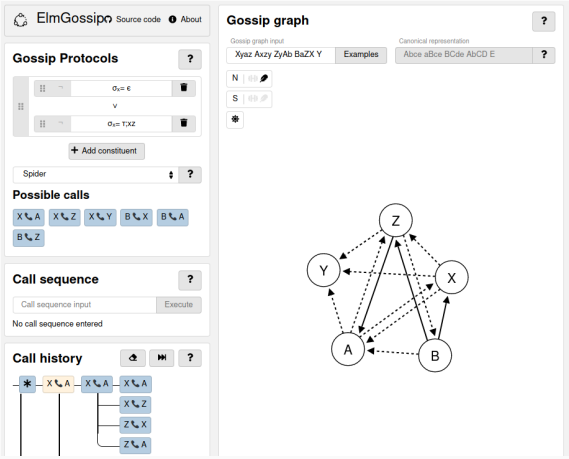
`Ad ABc Cd D`



`Abcdefg B CE D CE F G`

Making calls

Click on a possible call to change the graph!



Protocols

In ElmGossip the following protocols are predefined:

Protocol	Calling condition
Any	$\top$
Call Once	$xy \notin \sigma_x \wedge yx \notin \sigma_x$
Lean New Secrets	$\neg S^{\sigma} xy$
Spider	$\sigma_x = \epsilon \vee \sigma_x = \tau; xz$
Token	$\sigma_x = \epsilon \vee \sigma_x = \tau; zx$
Weak Call Once	$xy \notin \sigma_x$

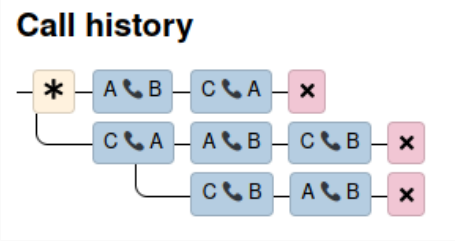
Protocols

In ElmGossip the following protocols are predefined:

Protocol	Calling condition
Any	$\top$
Call Once	$xy \notin \sigma_x \wedge yx \notin \sigma_x$
Lean New Secrets	$\neg S^{\sigma} xy$
Spider	$\sigma_x = \epsilon \vee \sigma_x = \tau; xz$
Token	$\sigma_x = \epsilon \vee \sigma_x = \tau; zx$
Weak Call Once	$xy \notin \sigma_x$

And you can define your own custom protocols!

Execution Trees



Comparing Protocols

Definition

We say that protocol A is *stronger* than protocol B iff the condition of A implies the condition of B. Hence, a *weaker* protocol can allow *more* calls!

Comparing Protocols

Definition

We say that protocol A is *stronger* than protocol B iff the condition of A implies the condition of B. Hence, a *weaker* protocol can allow *more* calls!

Lemma

- LNS is stronger than CO.
- CO is stronger than weak CO.

Comparing Protocols

Definition

We say that protocol A is *stronger* than protocol B iff the condition of A implies the condition of B. Hence, a *weaker* protocol can allow *more* calls!

Lemma

- LNS is stronger than CO.
- CO is stronger than weak CO.
- All LNS sequences are also CO sequences. (But not vice versa $\rightarrow$ exercise!)

Define your own protocol!

You can also define your own protocols in ElmGossip!

Example:

$$\sigma^x = \epsilon \ \vee \ xy \in \sigma^x$$

What does this say? 

What ElmGossip does not cover

Hans also talked about the higher-order effects of gossip calls and K_i.

What would be a protocol condition that we **cannot** define in ElmGossip? 

What ElmGossip does not cover

Hans also talked about the higher-order effects of gossip calls and K_i.

What would be a protocol condition that we **cannot** define in ElmGossip? 

Example: 

$$PIG_{xy} := \hat{K}_x \; \exists z \; \neg(Sxz \leftrightarrow Syz)$$

Why can we not check such a protocol in ElmGossip?

What ElmGossip does not cover

Hans also talked about the higher-order effects of gossip calls and K_i.

What would be a protocol condition that we **cannot** define in ElmGossip? 

Example: 

$$PIG_{xy} := \hat{K}_x \exists z \neg(Sxz \leftrightarrow Syz)$$

Why can we not check such a protocol in ElmGossip?

$\Rightarrow$ Tomorrow we will see a more general model checker for more general protocols.

Bonus: How does it work?

ElmGossip is written in the functional programming language *Elm*. Example piece of code:

```
containing : CallSequence -> AgentId -> CallSequence
containing sequence agent =
    case sequence of
        [] ->
            []
        call :: calls ->
            if includes call agent then
                call :: containing calls agent
            else
                containing calls agent
```

Links: https://github.com/RamonMeffert/elm-gossip · https://guide.elm-lang.org/

Exercises

1. With pen an paper, draw an initial total gossip graph for 4 agents, and execute the call sequence *ab*; *bc*; *cd*; *da*. Draw a new graph after each call, such that you have five graphs with four calls between them.
2. Open ElmGossip and check your drawings by executing the same call sequence there, step by step. Note that you first need to enter `Abcd aBcd abCd abcD` for the initial graph.
3. Is the sequence *ab*; *bc*; *cd*; *da* allowed according to the ANY, the LNS and the CMO protocols?
4. Find a call sequence that is permitted for CMO but not for LNS.
5. Find a (non-total) gossip graph where CMO is weakly successful, but LNS is unsuccessful.
6. Define a protocol which allows more calls than LNS, but fewer than CMO.
7. Consider the "Spider" protocol. Why does its condition use $\sigma^{\times} = \epsilon$ and not $\sigma{=}\epsilon$?