Lecture 1, part 2: ElmGossip

Knowledge and Gossip — ESSLLI 2022

Malvin Gattinger (ILLC, Amsterdam)

2022-08-08, Galway

https://malv.in/2022/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.

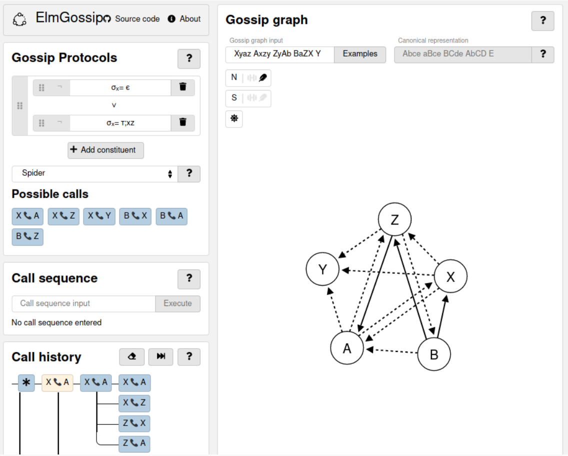
Or we can *add* edges to the same drawing.

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



Gossip Protocols

$\neg$ $\sigma_x = \epsilon$

v

$\neg$ $\sigma_x = \top;xz$

Spider

Possible calls

X☎A X☎Z X☎Y B☎X B☎A B☎Z

Call sequence

Call sequence input

No call sequence entered

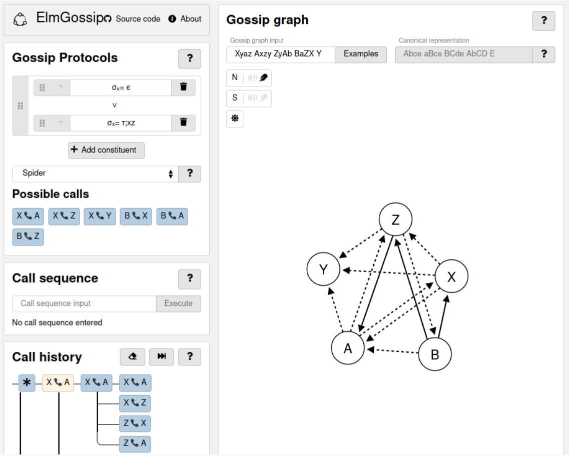
Call history

Gossip graph

Gossip graph input: Xyaz Axzy ZyAb BaZX Y

Canonical representation: Abce aBce BCde AbCD E

ElmGossip



Gossip Protocols

$\sigma_x = \epsilon$

∨

$\sigma_x = \top;XZ$

Spider

Possible calls

X☎A X☎Z X☎Y B☎X B☎A B☎Z

Call sequence

Call sequence input

No call sequence entered

Call history

* X☎A X☎A X☎A X☎Z Z☎X Z☎A

Gossip graph

Gossip graph input: Xyaz Axzy ZyAb BaZX Y

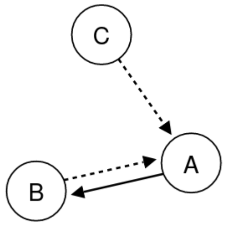
Canonical representation: Abce aBce BCde AbCD E

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

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

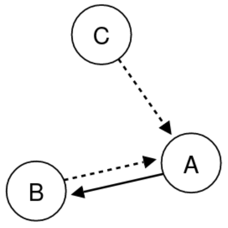
Try it: https://r3n.nl/elm-gossip/

Short notation for gossip graphs



```
AB aB aC
```

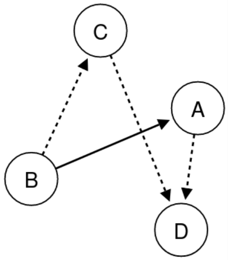
Short notation for gossip graphs



`AB aB aC`

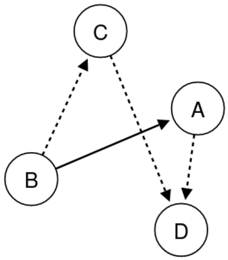
- A graph of *n* agents is described by *n* words separated by spaces.
- Knowing the **number** of agent *a* is denoted by `a`
- Knowing the **secret** of agent *a* is denoted by `A`

Examples

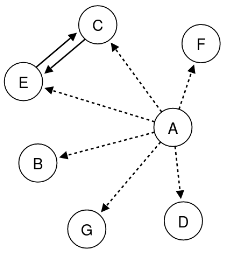


`Ad ABc Cd D`

Examples



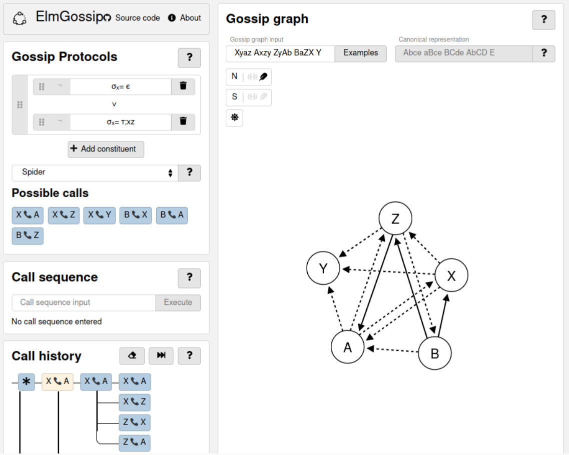
`Ad ABc Cd D`



`Abcdefg B CE D CE F G`

Making calls

Click on a possible call to change the graph!



Gossip Protocols

$\neg$ $\sigma_x = \epsilon$

$\vee$

$\neg$ $\sigma_x = \top;xz$

Spider

Possible calls

X☎A X☎Z X☎Y B☎X B☎A B☎Z

Call sequence

Call sequence input: (empty)

No call sequence entered

Call history

∗ — X☎A — X☎A — X☎A

X☎Z

Z☎X

Z☎A

Gossip graph

Gossip graph input: Xyaz Axzy ZyAb BaZX Y

Canonical representation: Abce aBce BCde AbCD E

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	$xy \notin \sigma_x$

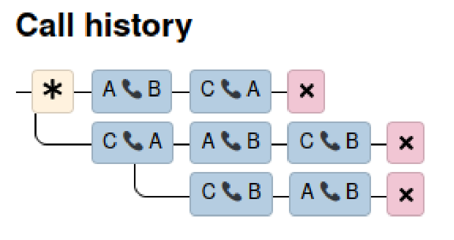
Protocols

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 → exercise!)

Define your own protocol!

You can also define your own protocols in ElmGossip!

Example:

$$\sigma^{x} = \epsilon \ \lor \ xy \in \sigma^{x}$$

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?

⇒ 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

See course website!

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