

Tools for Gossip

Bachelor's Project

R.A. Meffert

Supervisor: Dr. B.R.M. Gattinger

University of Groningen

Bachelor's Symposium, January 25, 2021

About this bachelor's project

- The telephone problem¹



Figure 1: Two agents merging their secrets

¹e.g. Tijdeman, 1971.

About this bachelor's project

- ▶ The telephone problem¹
- ▶ Dynamic gossip²



Figure 1: Two agents merging their secrets

¹e.g. Tijdeman, 1971.

²Van Ditmarsch et al., 2018.

About this bachelor's project

- ▶ The telephone problem¹
- ▶ Dynamic gossip²
- ▶ Goal: easy-to-use educational tool



Figure 1: Two agents merging their secrets

¹e.g. Tijdeman, 1971.

²Van Ditmarsch et al., 2018.

Why dynamic gossip?



Blockchain



Social Media

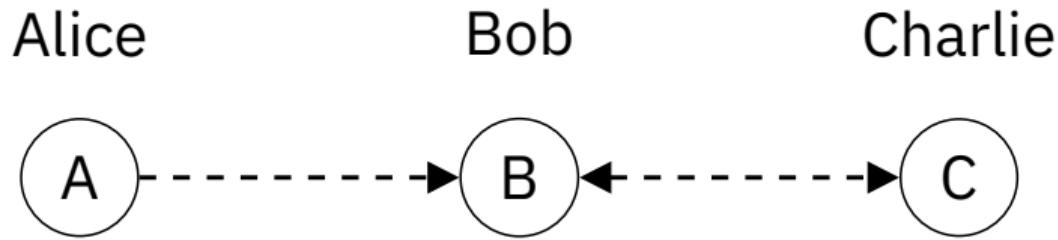


Genome analysis



Distributed databases

Example

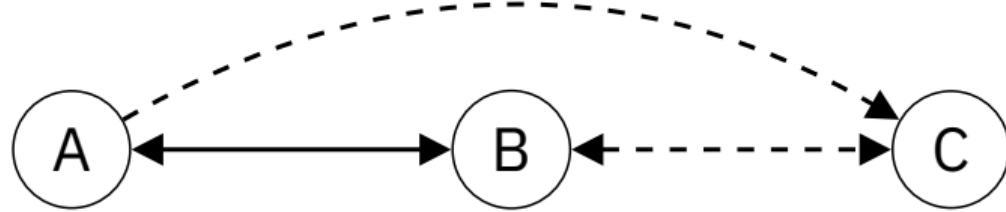


$$N = \{(a, b), (b, c), (c, b)\}$$

$$S = I_A$$

Example

After call **ab**

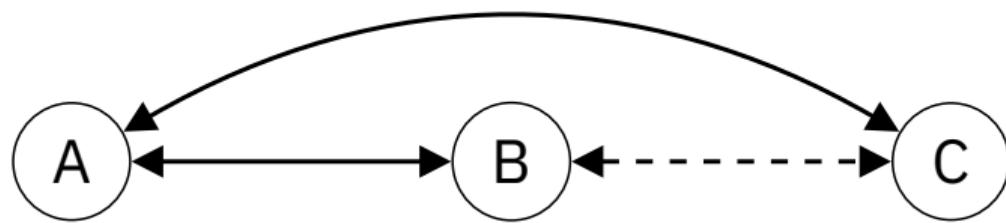


$$N = I_A \cup \{(a, b), (a, c), (b, a), (b, c), (c, b)\}$$

$$S = I_A \cup \{(a, b), (b, a)\}$$

Example

After call **ac**

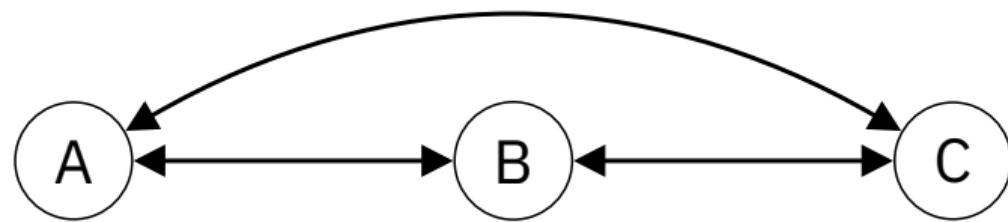


$$N = A^2$$

$$S = I_A \cup \{(a, b), (a, c), (b, a), (c, a)\}$$

Example

After call **bc**



$$N = A^2$$

$$S = A^2$$

The tool

Tools for Gossip Bachelor's project · R.A. Meffert (r.a.meffert@student.rug.nl) · Supervisor: Dr. B.R.M. Gattinger

Gossip graph

Gossip graph input: ABC AbC abC

Canonical representation: Abc aBc abC

Examples: N | S | *

```

graph TD
    A((A)) --> B((B))
    B((B)) --> C((C))
    C((C)) -.-> A((A))
    C((C)) -.-> B((B))
  
```

Protocol condition

T

+ Add constituent

Any

$\pi(x, y)$ (T) LaTeX

Possible calls: A ↔ B, A ↔ C, B ↔ A, B ↔ C, C ↔ A, C ↔ B

Call sequence

Call sequence input: Execute

No call sequence entered

Call history

```

graph LR
    * --- AB[AB]
    AB --- AC[AC]
    AC --- BA[B↔A]
    BA --- BC[BC]
  
```

Generate LaTeX file | Copy GraphViz DOT code

✉ 8

Graph visualisation

Tools for Gossip Bachelor's project · R.A. Meffert (r.a.meffert@student.rug.nl) · Supervisor: Dr. B.R.M. Göttinger

Gossip graph

Gossip graph input: ABC AbC abC

Canonical representation: Abc aBc abC

Examples: N | S | *

Protocol condition: T

+ Add constituent

Any

$\pi(x, y)$ (T) LaTeX

Possible calls: A ↗ B, A ↗ C, B ↗ A, B ↗ C, C ↗ A, C ↗ B

Call sequence: Call sequence input Execute

No call sequence entered

Call history: * - A ↗ B - A ↗ C
 |
 B ↗ A - B ↗ C

Generate LaTeX file Copy GraphViz DOT code

8

Gossip protocols

Tools for Gossip

Bachelor's project · R.A. Meffert (r.a.meffert@student.rug.nl) · Supervisor: Dr. B.R.M. Göttinger

Gossip graph

Gossip graph input: ABC ABc abC

Canonical representation: Abc aBc abC

Operations: N (New), S (Save), * (Copy)

```
graph TD; A((A)) --> B((B)); A((A)) -.-> C((C)); B((B)) -.-> A((A)); B((B)) --> C((C)); C((C)) -.-> A((A)); C((C)) --> B((B))
```

Generate LaTeX file | Copy GraphViz DOT code

Protocol condition

Protocol condition: T

+ Add constituent

Any

$\pi(x, y)$ (T) LaTeX

Possible calls

A ↗ B A ↗ C B ↗ A B ↗ C C ↗ A C ↗ B

Call sequence

Call sequence input: Execute

No call sequence entered

Call history

* — A ↗ B — A ↗ C
 |
 B ↗ A — B ↗ C

8

Call sequence validation & execution

Tools for Gossip Bachelor's project · R.A. Meffert (r.a.meffert@student.rug.nl) · Supervisor: Dr. B.R.M. Göttinger

Gossip graph

Gossip graph input: ABC ABc abC

Canonical representation: Abc aBc abC

Protocol condition

Possible calls: A ↗ B, A ↗ C, B ↗ A, B ↗ C, C ↗ A, C ↗ B

Call sequence

Call history

Generate LaTeX file | Copy GraphViz DOT code

Protocol condition

Any $\pi(x, y)$ (T) LaTeX

Possible calls

A ↗ B, A ↗ C, B ↗ A, B ↗ C, C ↗ A, C ↗ B

Call sequence

No call sequence entered

Call history

* — A ↗ B — A ↗ C
 |
 B ↗ A — B ↗ C

8

Execution tree

Tools for Gossip

Bachelor's project · R.A. Meffert (r.a.meffert@student.rug.nl) · Supervisor: Dr. B.R.M. Gattinger

Gossip graph

Gossip graph input: ABC AbC abC

Canonical representation: Abc aBc abC

Examples: N | S | *

Possible calls: A → B, A → C, B → A, B → C, C → A, C → B

Call sequence input: No call sequence entered

Call history: * → A → B → A → C → B → A → B → C

Protocol condition

Protocol condition: Any (T) LaTeX

Possible calls: A → B, A → C, B → A, B → C, C → A, C → B

Call history

Call history: * → A → B → A → C → B → A → B → C

Implementation

- ▶ Language: Elm³
 - ▶ Open source, functional web language



³Czaplicki and Chong, 2013.

Implementation

- ▶ Language: Elm³
 - ▶ Open source, functional web language
- ▶ Pure functions
 - ▶ Easier to translate mathematical functions into code



³Czaplicki and Chong, 2013.

Implementation

- ▶ Language: Elm³
 - ▶ Open source, functional web language
- ▶ Pure functions
 - ▶ Easier to translate mathematical functions into code
- ▶ Compiled to Javascript
 - ▶ Free cross-platform compatibility



³Czaplicki and Chong, 2013.

Implementation

- ▶ Language: Elm³
 - ▶ Open source, functional web language
- ▶ Pure functions
 - ▶ Easier to translate mathematical functions into code
- ▶ Compiled to Javascript
 - ▶ Free cross-platform compatibility
- ▶ Static type checking
 - ▶ Zero runtime exceptions



³Czaplicki and Chong, 2013.

Survey

- ▶ Short exploratory survey



Survey

- ▶ Short exploratory survey
- ▶ 12 respondents



Survey

- ▶ Short exploratory survey
- ▶ 12 respondents
- ▶ General impression: positive



Survey

- ▶ Short exploratory survey
- ▶ 12 respondents
- ▶ General impression: positive
- ▶ Useful feedback



Survey results



What do you think of the tool's visual design?



Do you think the error messages shown when providing incorrect input are useful?



How much time does it take you to understand what each part of the interface is for?

100%

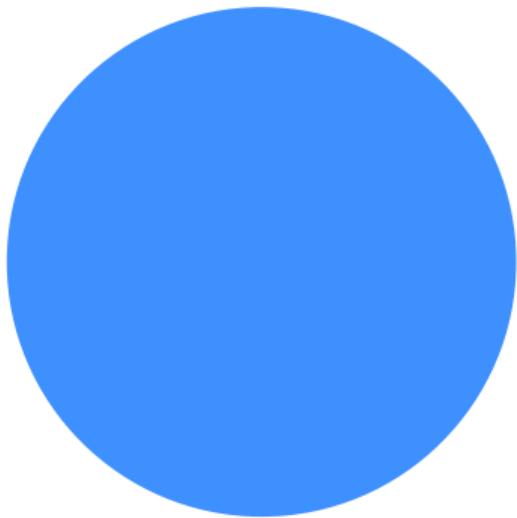
50%

0

50%

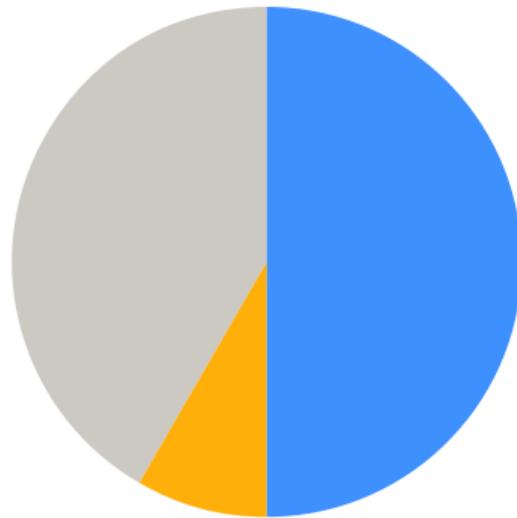
100%

Survey results



■ Yes ■ Maybe ■ No

Do you think this tool could be useful in a
study setting?



■ Yes ■ Maybe ■ No

Do you think this tool could be useful in a
research setting?

Further research and extensions

- Unreliable gossip⁴



⁴Van den Berg and Gattinger, 2020.

⁵Herzig and Maffre, 2017.

⁶Cooper et al., 2019.

Further research and extensions

- ▶ Unreliable gossip⁴
- ▶ Higher level knowledge⁵



⁴Van den Berg and Gattinger, 2020.

⁵Herzig and Maffre, 2017.

⁶Cooper et al., 2019.

Further research and extensions

- ▶ Unreliable gossip⁴
- ▶ Higher level knowledge⁵
- ▶ Temporal gossip⁶



⁴Van den Berg and Gattinger, 2020.

⁵Herzig and Maffre, 2017.

⁶Cooper et al., 2019.

Further research and extensions

- ▶ Unreliable gossip⁴
- ▶ Higher level knowledge⁵
- ▶ Temporal gossip⁶
- ▶ And more!



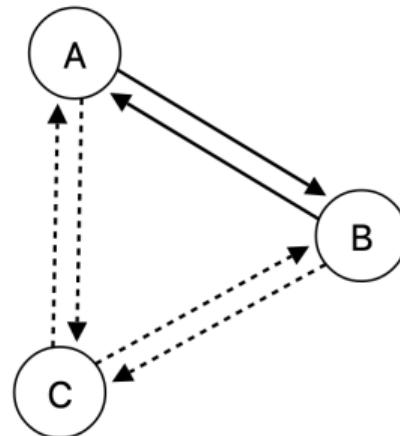
⁴Van den Berg and Gattinger, 2020.

⁵Herzig and Maffre, 2017.

⁶Cooper et al., 2019.

Summary

- ▶ Tool for exploring dynamic gossip
 - ▶ Graph visualisation
 - ▶ Gossip protocols
 - ▶ Call (sequence) evaluation & execution
 - ▶ Execution tree
- ▶ Positively evaluated
- ▶ Open source + free license (GPLv3)



Quick links

	RESULT	SOURCE CODE
TOOL	r3n.nl/bsc/gossip	r3n.nl/bsc/src/gossip
THESIS	r3n.nl/bsc/thesis	r3n.nl/bsc/src/thesis
SLIDES	r3n.nl/bsc/slides	r3n.nl/bsc/src/slides

Thesis will be available after Februari 1st, slides after today

References I

- van den Berg, L. & Gatteringer, M. (2020). Dealing with unreliable agents in dynamic gossip [Series Title: Lecture Notes in Computer Science]. In M. A. Martins & I. Sedlár (Eds.), *Dynamic logic. new trends and applications* (pp. 51–67). Series Title: Lecture Notes in Computer Science. Springer International Publishing. https://doi.org/10.1007/978-3-030-52021-2_4
- Cooper, M. C., Herzig, A., Maris, F. & Vianey, J. (2019). Temporal epistemic gossip problems. In M. Slavkovik (Ed.), *Multi-agent systems* (pp. 1–14). Springer International Publishing. https://doi.org/10.1007/978-3-030-21202-6_1
- Czaplicki, E. & Chong, S. (2013). Asynchronous functional reactive programming for GUIs. *SIGPLAN Not.*, 48(6), 411–422. <https://doi.org/10.1145/2535838.2535860>
- van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R. & Schwarzenbacher, F. (2018). Dynamic gossip. *Bulletin of the Iranian Mathematical Society*, 45(3), 701–728. <https://doi.org/10.1007/s41980-018-0050-2>
- Herzig, A. & Maffre, F. (2017). How to share knowledge by gossiping. *AI Communications*, 30(1), 1–17. <https://doi.org/10.31263/osf.io/94qxh>

References II

Tijdeman, R. (1971). On a telephone problem. *Nieuw Archief voor Wiskunde*, 3(19), 188–192.