Sous la co-tutelle de :
CNRS
ÉCOLE DES PONTS PARISTECH
UNIVERSITÉ GUSTAVE EIFFEL

2024-11-14

Formalization of real query languages and application to theory

 Claire David

 Nadime Francis

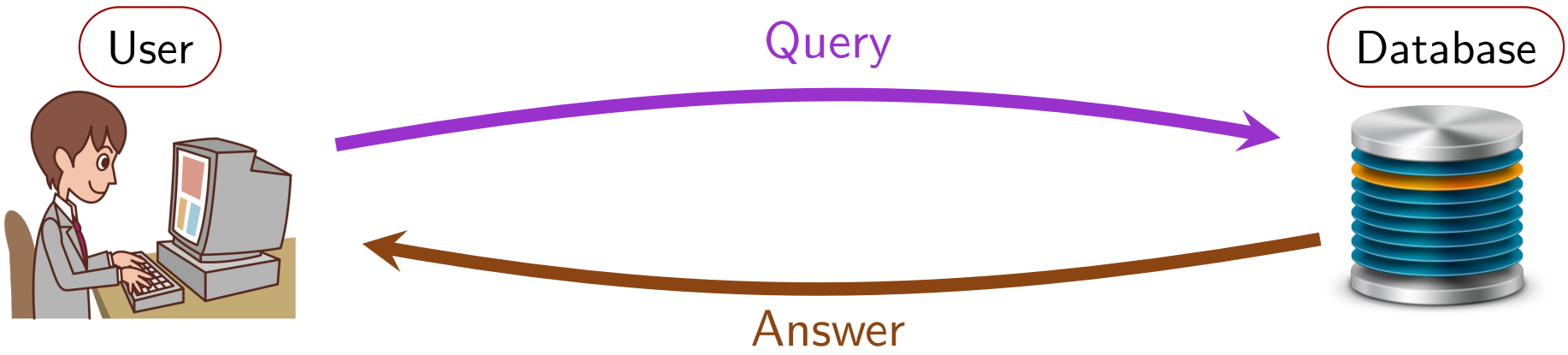
 Victor Marsault

 Antoine Meyer

 Liat Peterfreund

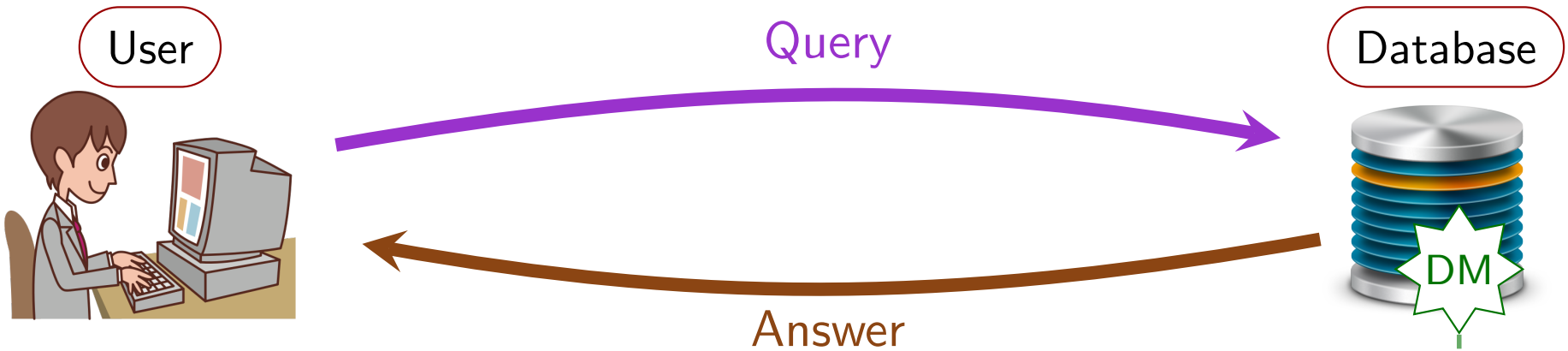
and many other coauthors from academia and industry

Setting



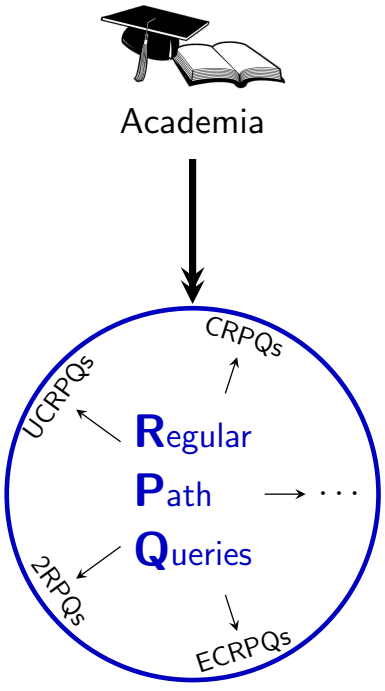
Setting

- Query language
  - "*What can the user write?*"



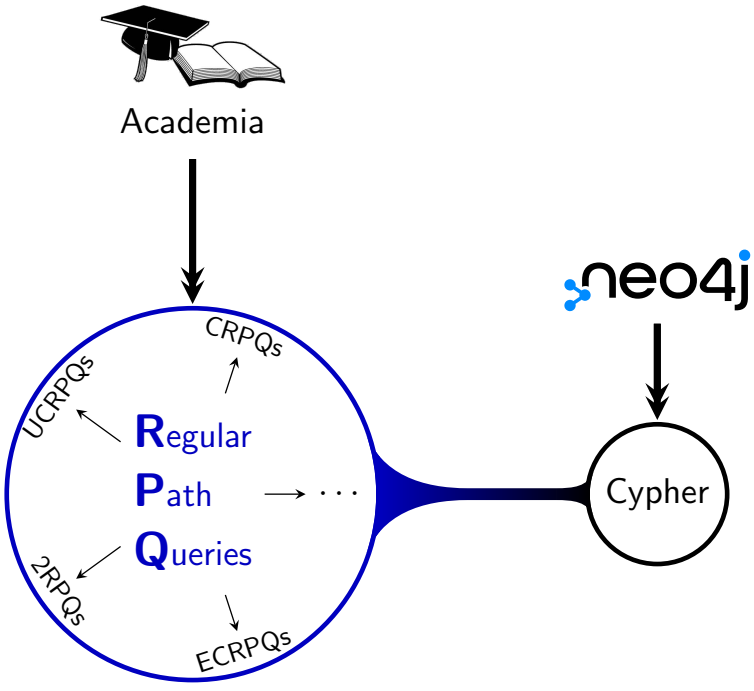
- Semantics of query
  - "*What data is extracted?*"
- DM = Data model
  - "*How the data is structured*"

Graph query languages: from RPQs to GQL



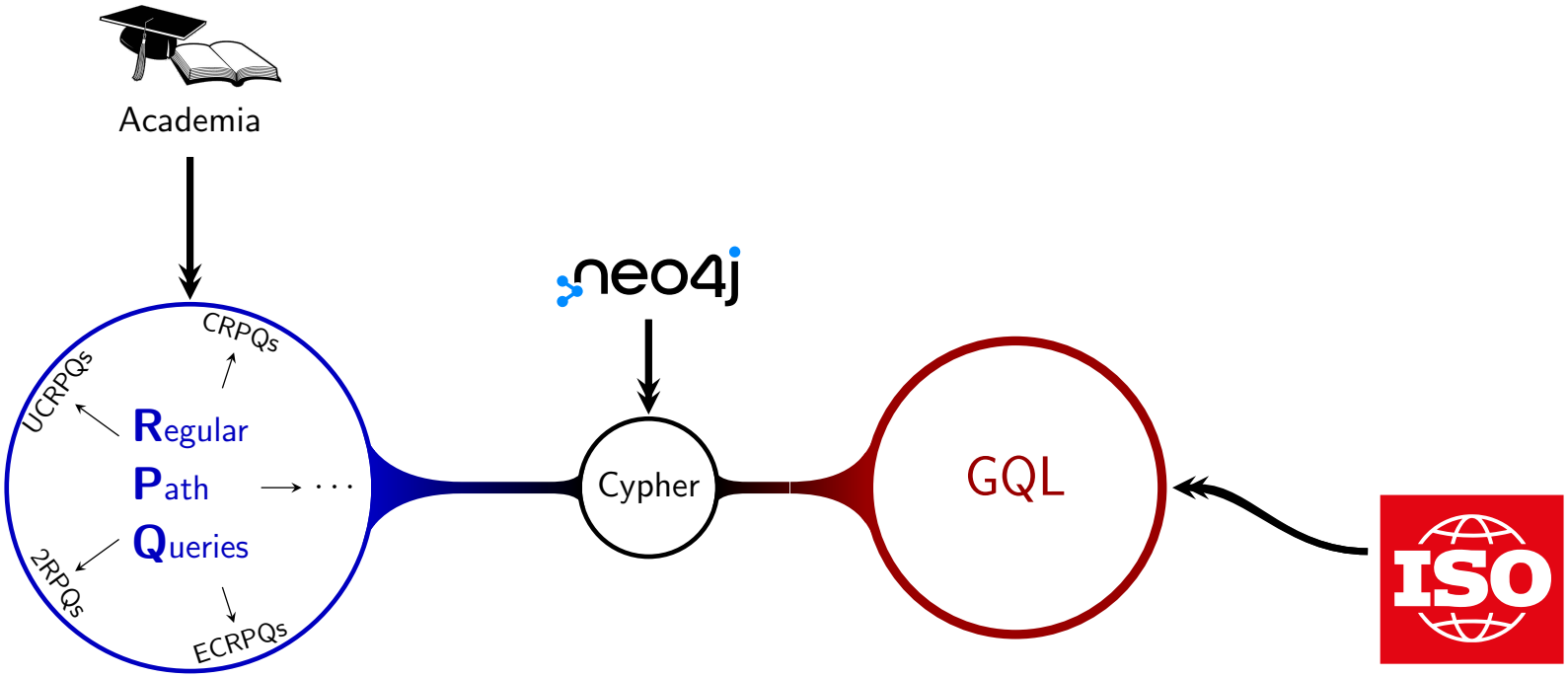
- RPQs (1990's) pattern matching with regular expressions

Graph query languages: from RPQs to GQL



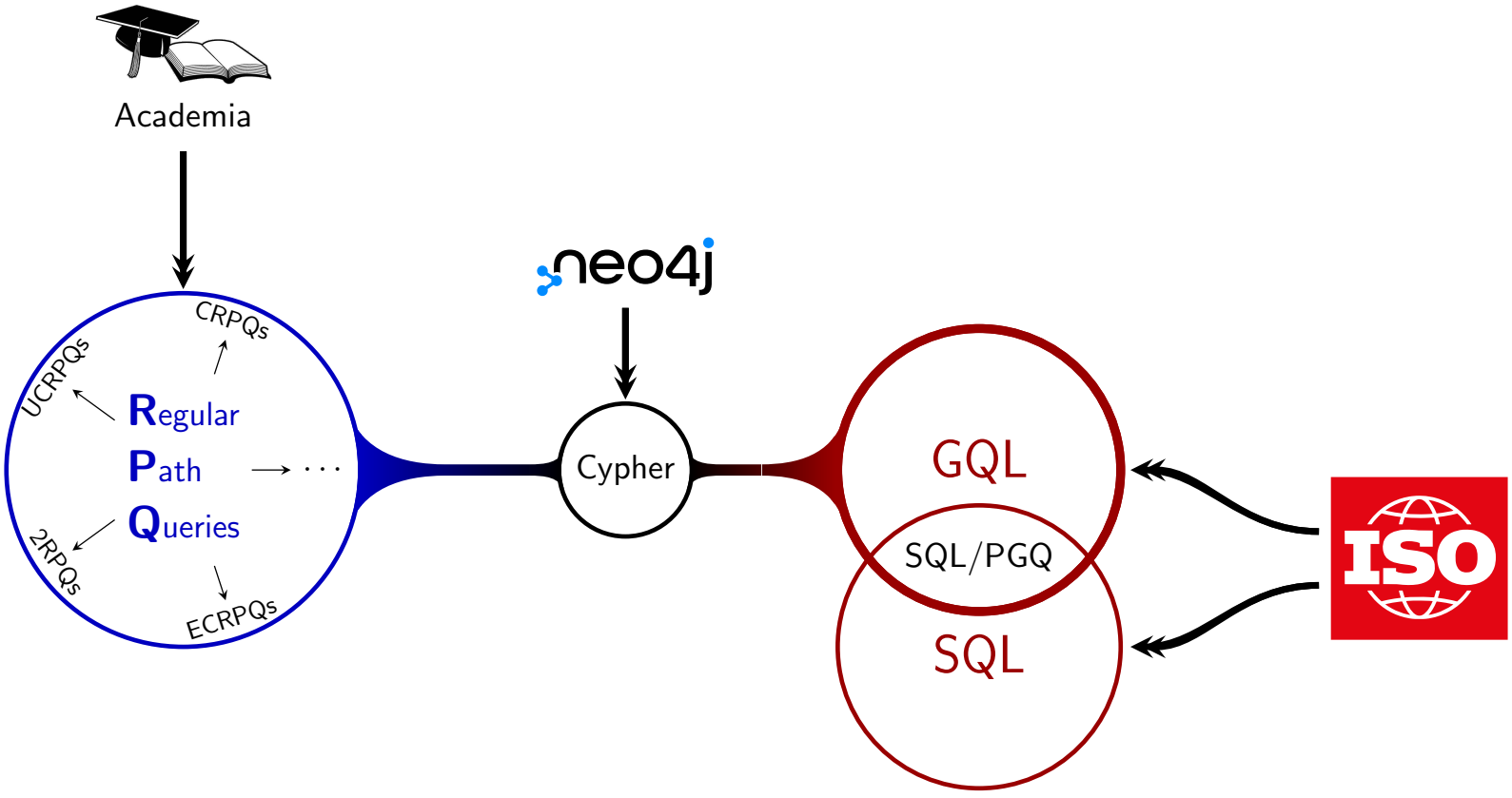
- RPQs (1990's) pattern matching with regular expressions
- Cypher (2011) query language for a new data model (property graphs)

Graph query languages: from RPQs to GQL



- RPQs (1990's) pattern matching with regular expressions
- Cypher (2011) query language for a new data model (property graphs)
- GQL (2024) standard query language for property graphs

Graph query languages: from RPQs to GQL



- RPQs (1990's) pattern matching with regular expressions
- Cypher (2011) query language for a new data model (property graphs)
- GQL (2024) standard query language for property graphs
- SQL/PGQ (2023) support for querying property graphs from SQL

General flow of our work

1 Formalize

- Collaboration with private actors
- Access to database engine and/or documentation
- Produce formal semantics of query languages

General flow of our work

1. **Formalize**
   - Collaboration with private actors
   - Access to database engine and/or documentation
   - Produce formal semantics of query languages

2. **Observe** what is done in practice
   - "Strange" decisions
   - Discrepancies with theoretical models
   - Goals and needs

General flow of our work

1. **Formalize**
   - Collaboration with private actors
   - Access to database engine and/or documentation
   - Produce formal semantics of query languages

2. **Observe** what is done in practice
   - "Strange" decisions
   - Discrepancies with theoretical models
   - Goals and needs

3. **Apply to theory**
   - Propose theoretical models closer to reality
   - Adapt algorithms
   - Propose features in line with practice

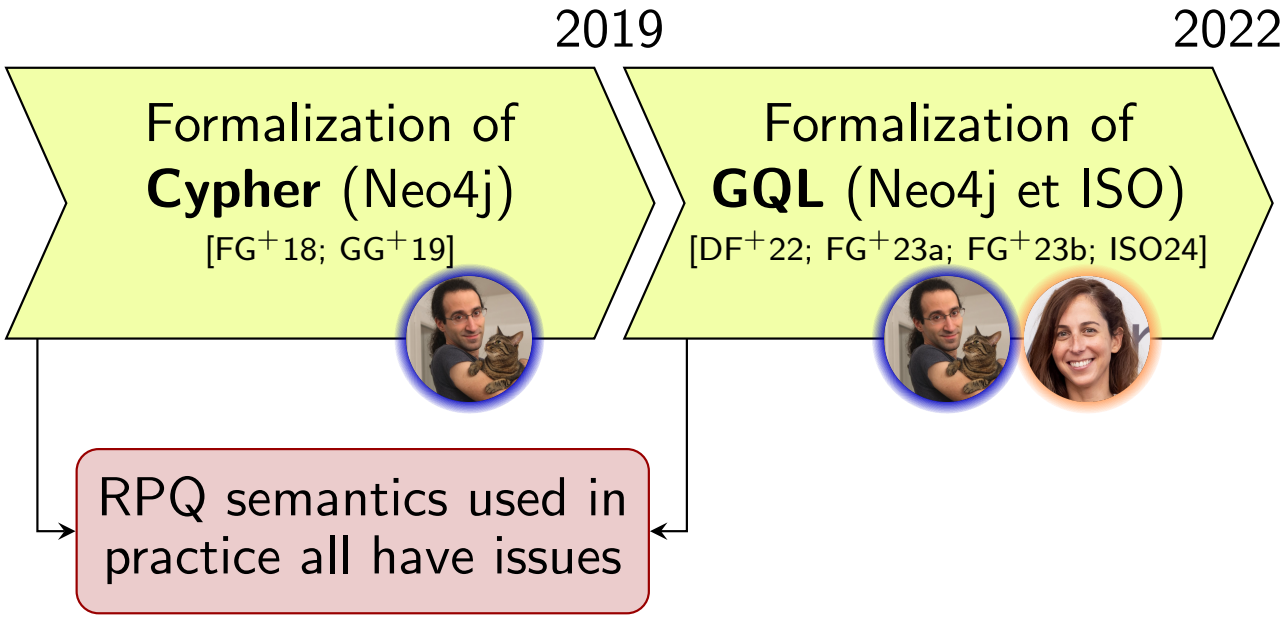
2019

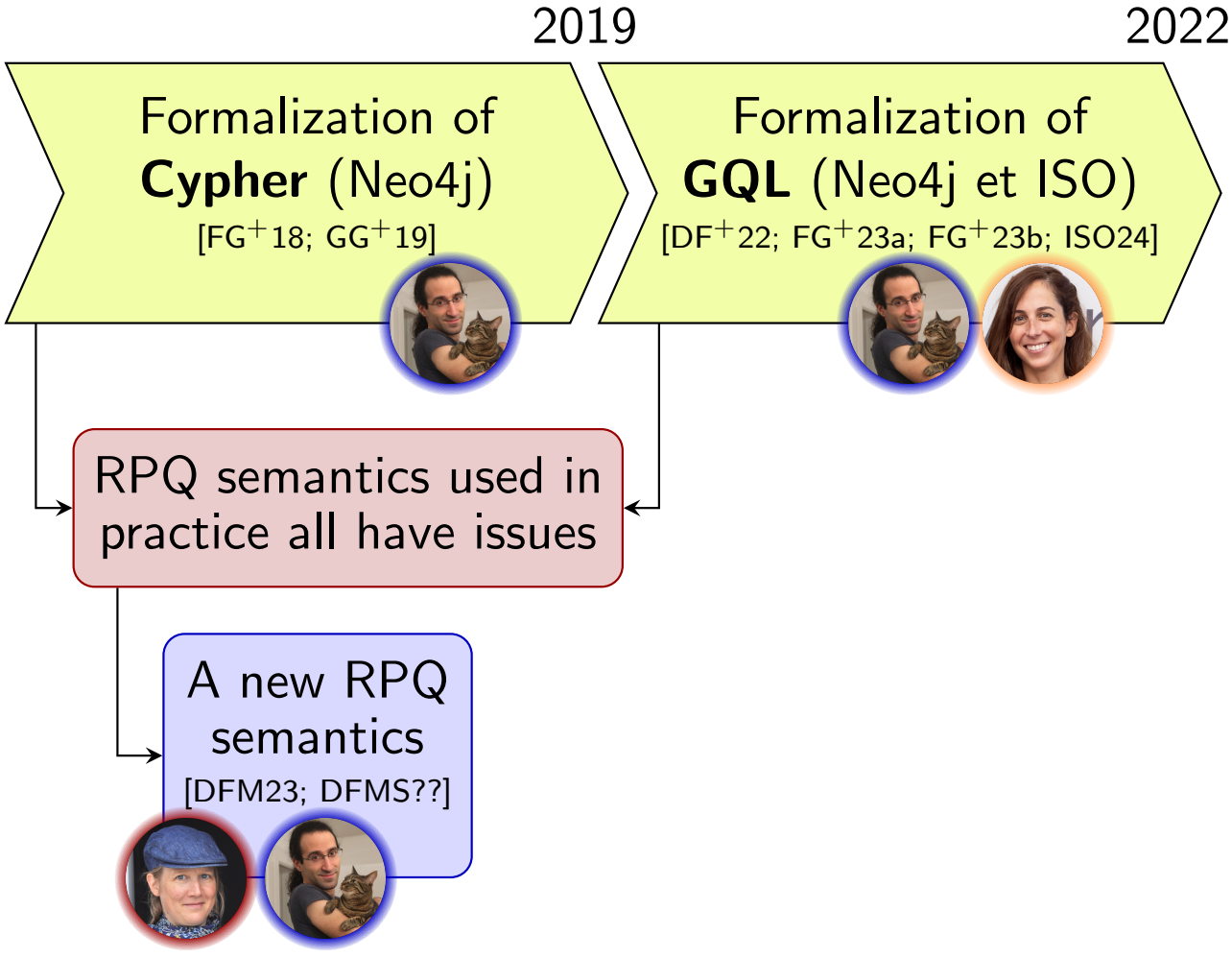
Formalization of
Cypher (Neo4j)
[FG+18; GG+19]

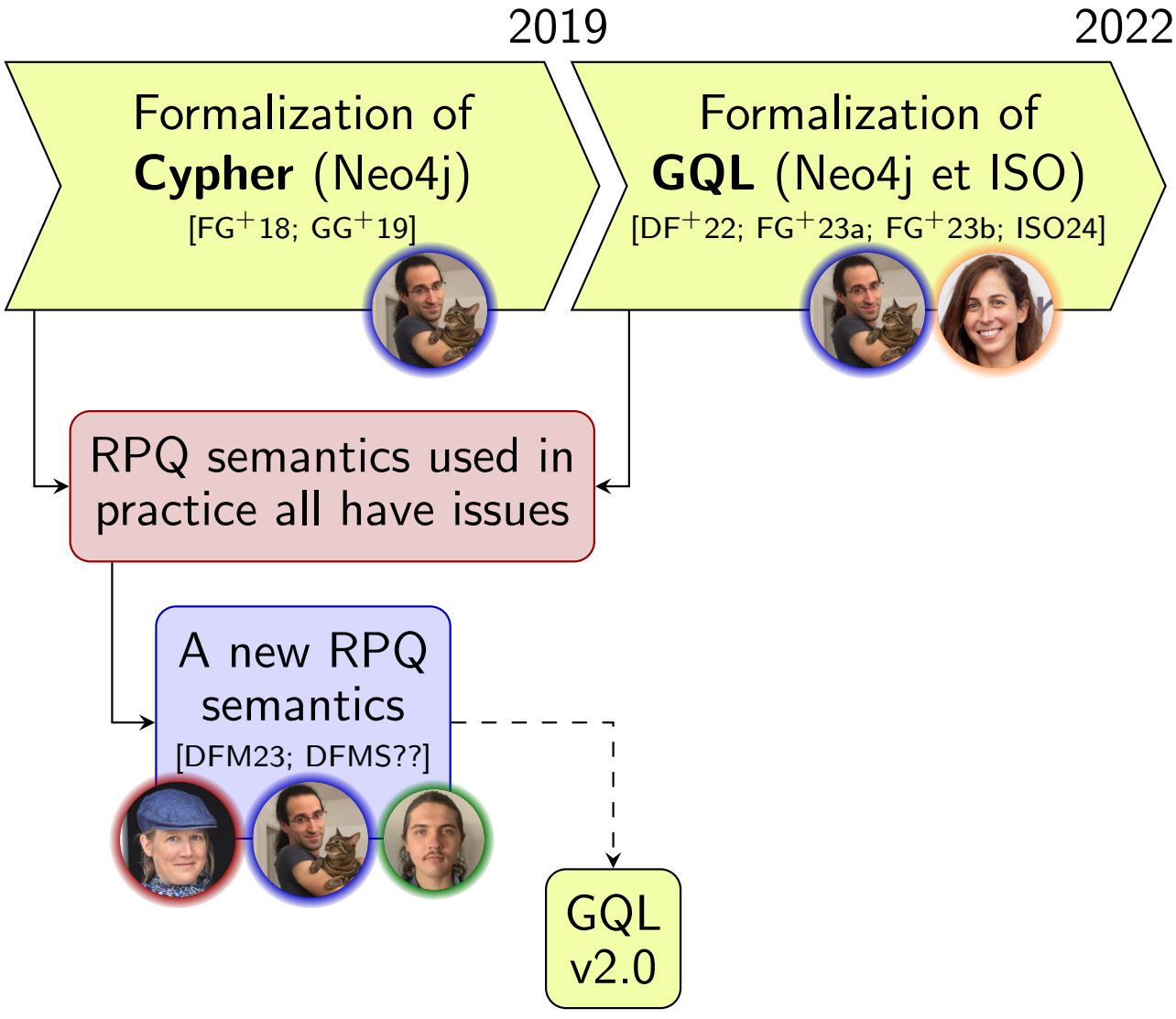
2019 2022

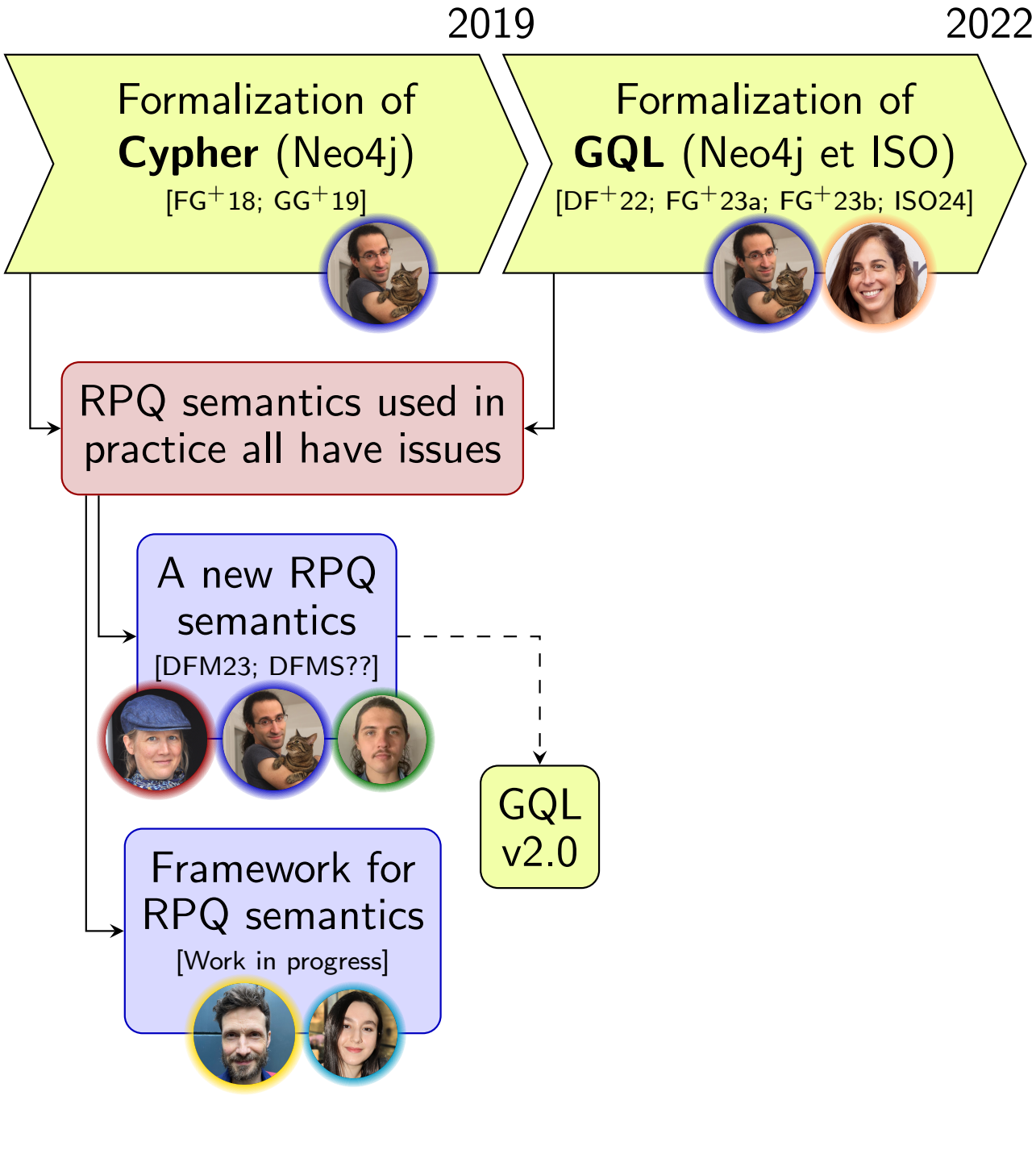
Formalization of
Cypher (Neo4j)
[FG+18; GG+19]

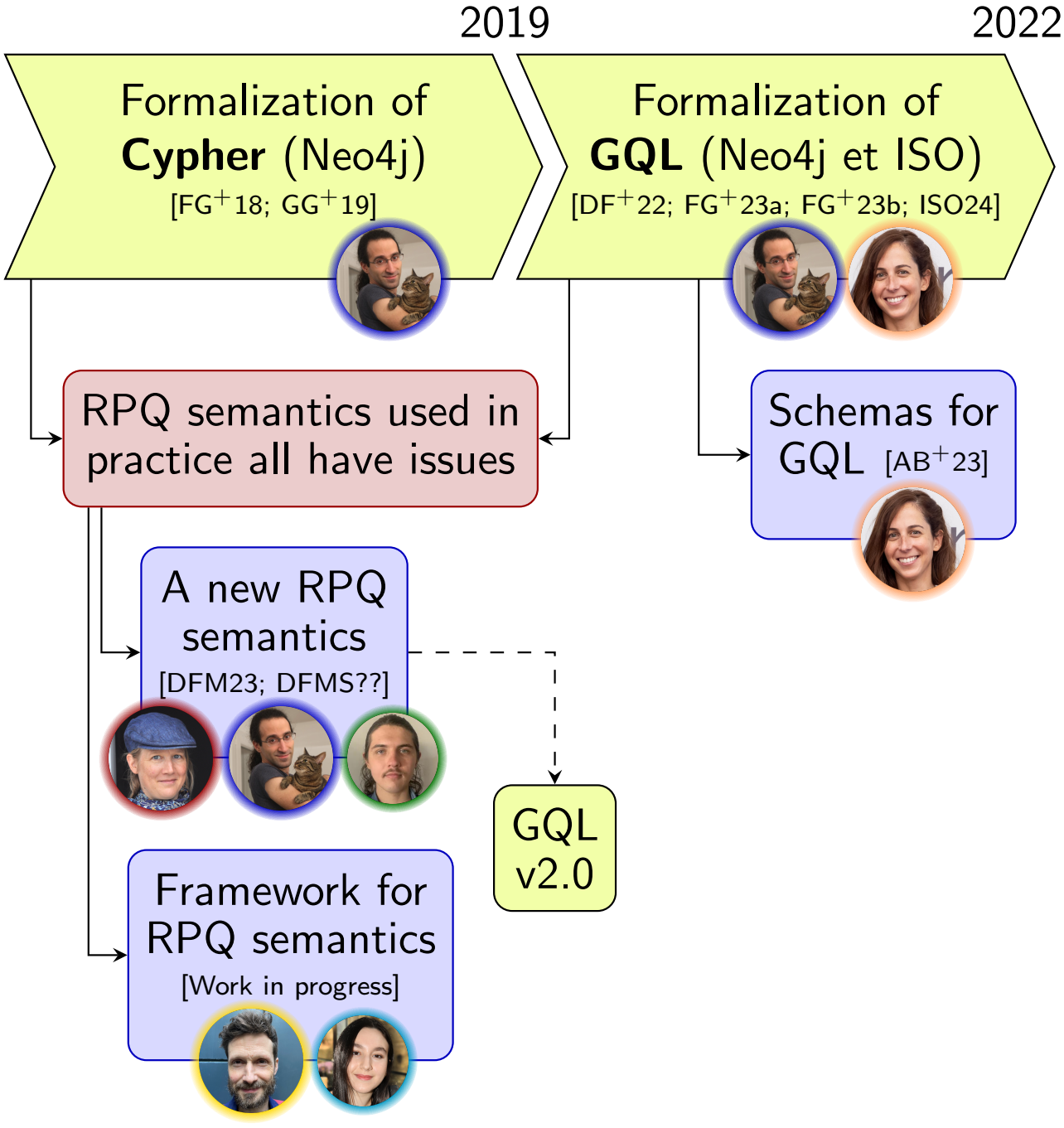
Formalization of
GQL (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]

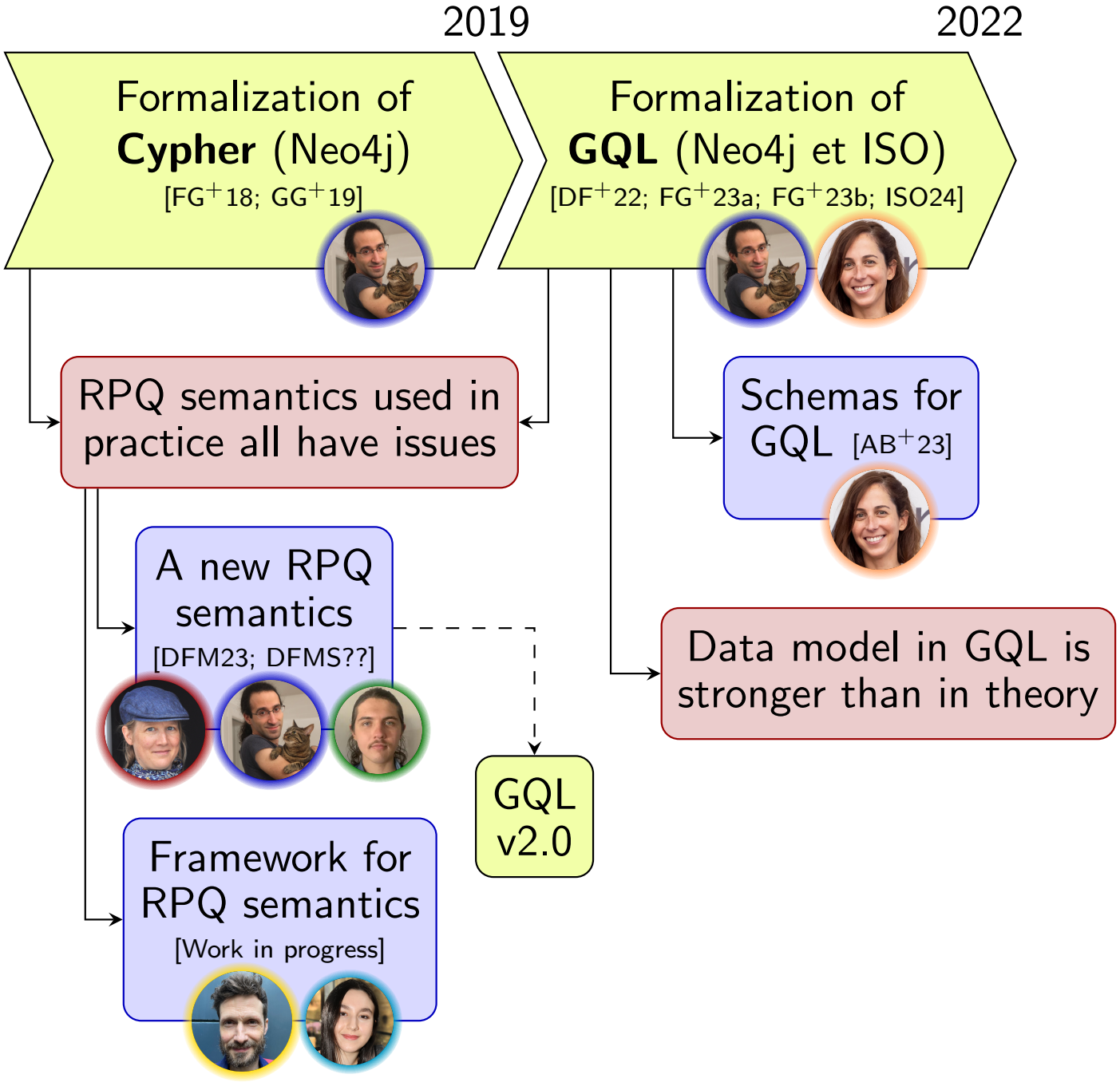

2019
2022
Formalization of
Cypher (Neo4j)
[FG+18; GG+19]
Formalization of
GQL (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in
practice all have issues

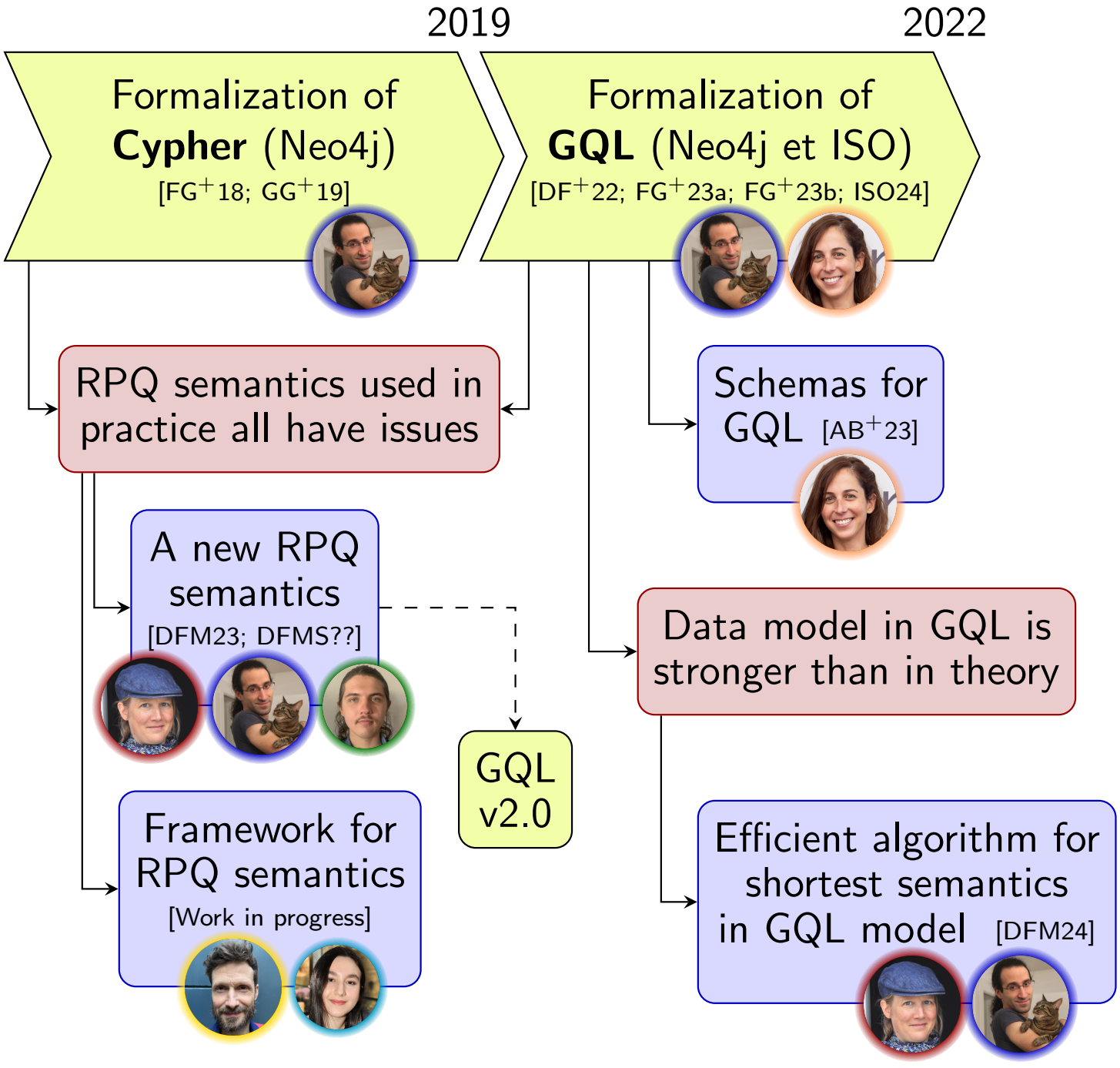

2019
2022
Formalization of
Cypher (Neo4j)
[FG+18; GG+19]
Formalization of
GQL (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
A new RPQ semantics
[DFM23; DFMS??]


2019
2022
Formalization of
Cypher (Neo4j)
[FG+18; GG+19]
Formalization of
GQL (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
A new RPQ semantics
[DFM23; DFMS??]
GQL v2.0


2019
2022
Formalization of
Cypher (Neo4j)
[FG+18; GG+19]
Formalization of
GQL (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
A new RPQ semantics
[DFM23; DFMS??]
GQL v2.0
Framework for RPQ semantics
[Work in progress]


2019
2022
Formalization of **Cypher** (Neo4j)
[FG+18; GG+19]
Formalization of **GQL** (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
Schemas for GQL [AB+23]
A new RPQ semantics
[DFM23; DFMS??]
GQL v2.0
Framework for RPQ semantics
[Work in progress]


2019
2022
Formalization of **Cypher** (Neo4j)
[FG+18; GG+19]
Formalization of **GQL** (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
Schemas for GQL [AB+23]
A new RPQ semantics
[DFM23; DFMS??]
Data model in GQL is stronger than in theory
GQL v2.0
Framework for RPQ semantics
[Work in progress]


2019
2022
Formalization of **Cypher** (Neo4j) [FG+18; GG+19]
Formalization of **GQL** (Neo4j et ISO) [DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
Schemas for GQL [AB+23]
A new RPQ semantics [DFM23; DFMS??]
Data model in GQL is stronger than in theory
GQL v2.0
Framework for RPQ semantics [Work in progress]
Efficient algorithm for shortest semantics in GQL model [DFM24]

Focus on [DFM23]:
Run-based semantics for RPQs

Setting of [DFM23]

User

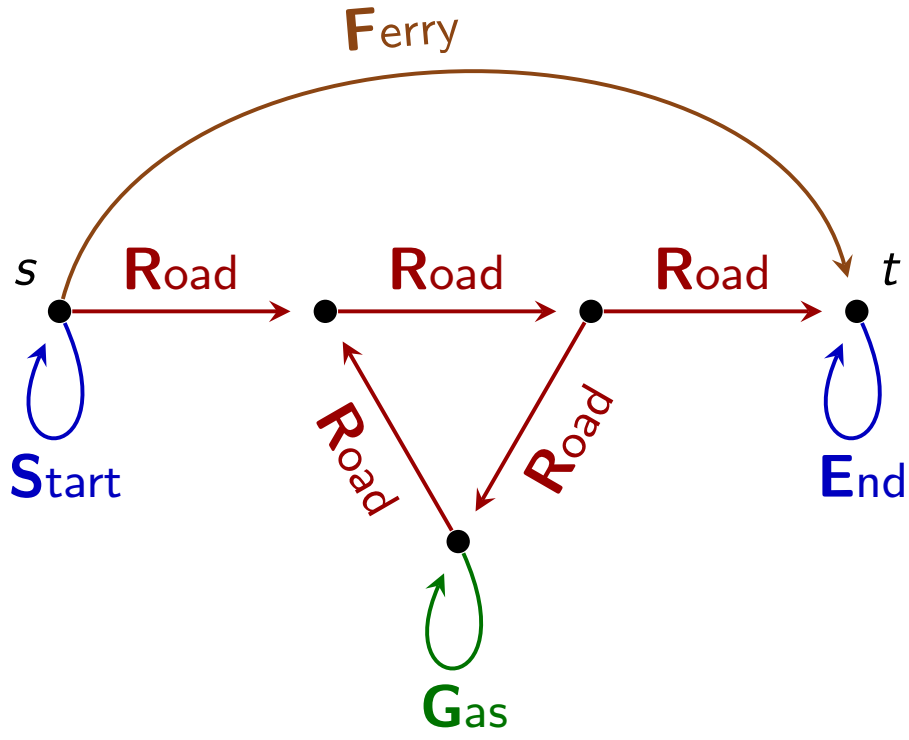
Query is an RPQ = Regular Expression

Answer

Graph Database

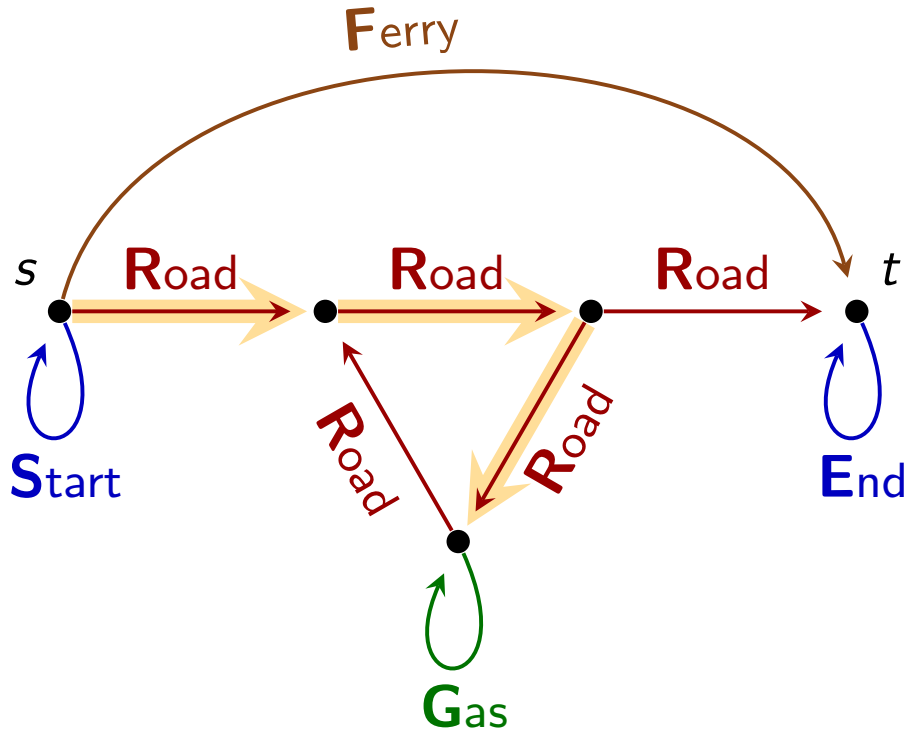
Our database is a labeled graph

- Finite label alphabet: $\Sigma = \{\mathbf{S}, \mathbf{R}, \mathbf{F}, \mathbf{G}, \mathbf{E}\}$
- Vertices
- Edges labelled over Σ



Our database is a labeled graph

- Finite label alphabet: $\Sigma = \{\mathbf{S}, \mathbf{R}, \mathbf{F}, \mathbf{G}, \mathbf{E}\}$
- Vertices
- Edges labelled over Σ

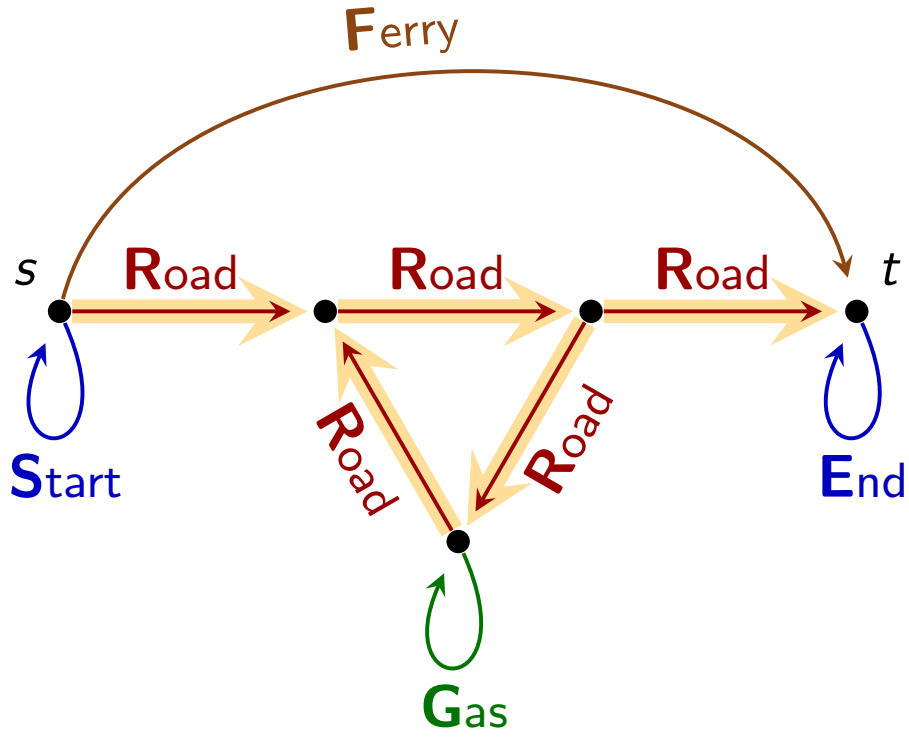


Terminogy: Walk

- Consistent sequence of edges
- Repetitions are allowed

Our database is a labeled graph

- Finite label alphabet: $\Sigma = \{\mathbf{S}, \mathbf{R}, \mathbf{F}, \mathbf{G}, \mathbf{E}\}$
- Vertices
- Edges labelled over Σ



Terminogy: Walk

- Consistent sequence of edges
- Repetitions are allowed

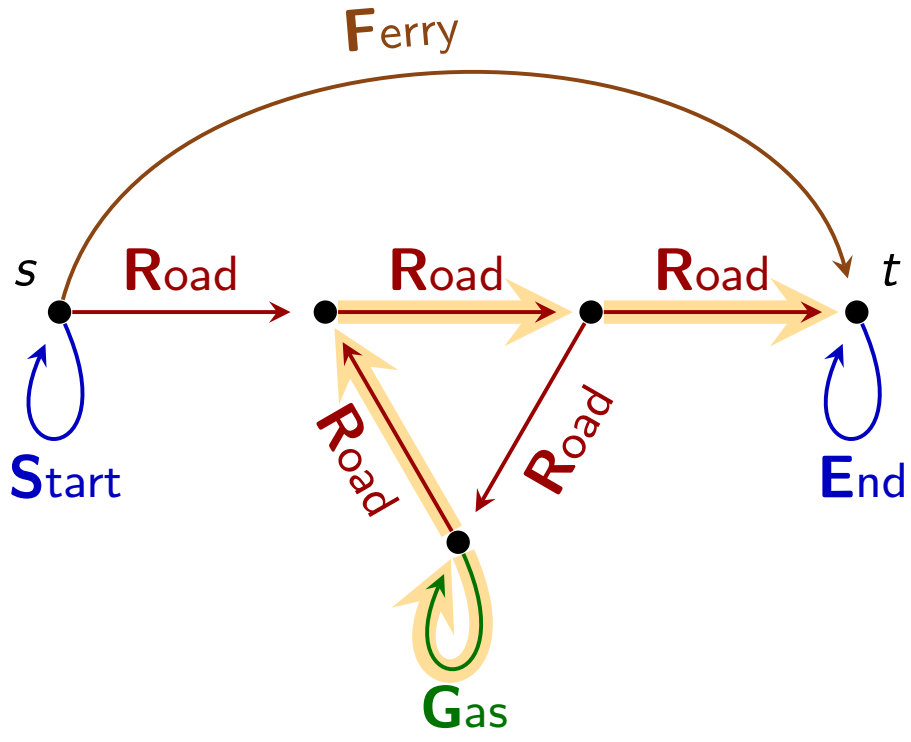
RPQ : pattern matching of walks

$Q ::=$	**A**	Atoms
	QQ	Concatenation
	$Q + Q$	Disjunction
	Q^*	Kleene star

where **A** is a label in the graph.

Ex: **GR***

"One **G** *then any number of* **R***"*



Definition

A walk matches Q
$\iff$ its label matches Q.

Ex: *Orange walk is labelled* **GRRR**
hence matches **GR***

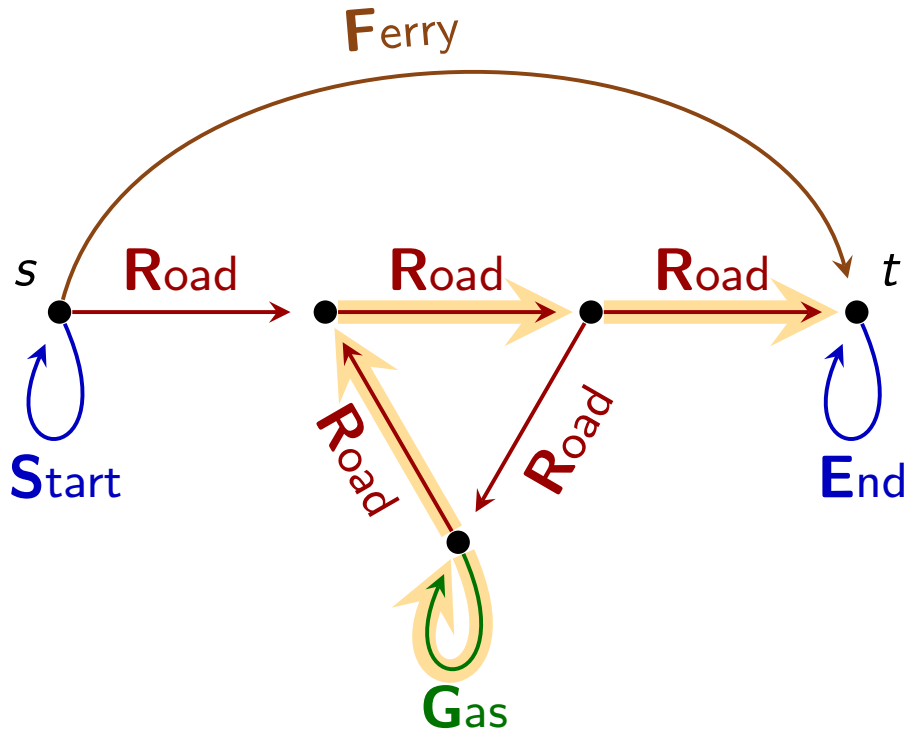
RPQ : pattern matching of walks

$Q ::=$	**A**	Atoms
	QQ	Concatenation
	$Q + Q$	Disjunction
	Q^*	Kleene star

where **A** is a label in the graph.

Ex: **GR***

"One **G** *then any number of* **R***"*



Definition

A walk matches Q
$\iff$ its label matches Q.

Ex: *Orange walk is labelled* **GRRR** *hence matches* **GR***

⚠ Infinitely many matches ⚠

RPQ Semantics

How to ensure finiteness ?

RPQ Semantics

How to ensure finiteness ?

Endpoint semantics

- Main semantics in theory
- Return endpoints of matches

RPQ Semantics

How to ensure finiteness ?

Endpoint semantics

- Main semantics in theory
- Return endpoints of matches

Trail semantics (Cypher, GQL)

- Most used in practice
- Repeating edges is forbidden

RPQ Semantics

How to ensure finiteness ?

Endpoint semantics

- Main semantics in theory
- Return endpoints of matches

Trail semantics (Cypher, GQL)

- Most used in practice
- Repeating edges is forbidden

Shortest semantics (GQL, etc)

- Second most used in practice
- Return the match with the least number of edges

RPQ Semantics

How to ensure finiteness ?

Endpoint semantics

- Main semantics in theory
- Return endpoints of matches

Trail semantics (Cypher, GQL)

- Most used in practice
- Repeating edges is forbidden

Shortest semantics (GQL, etc)

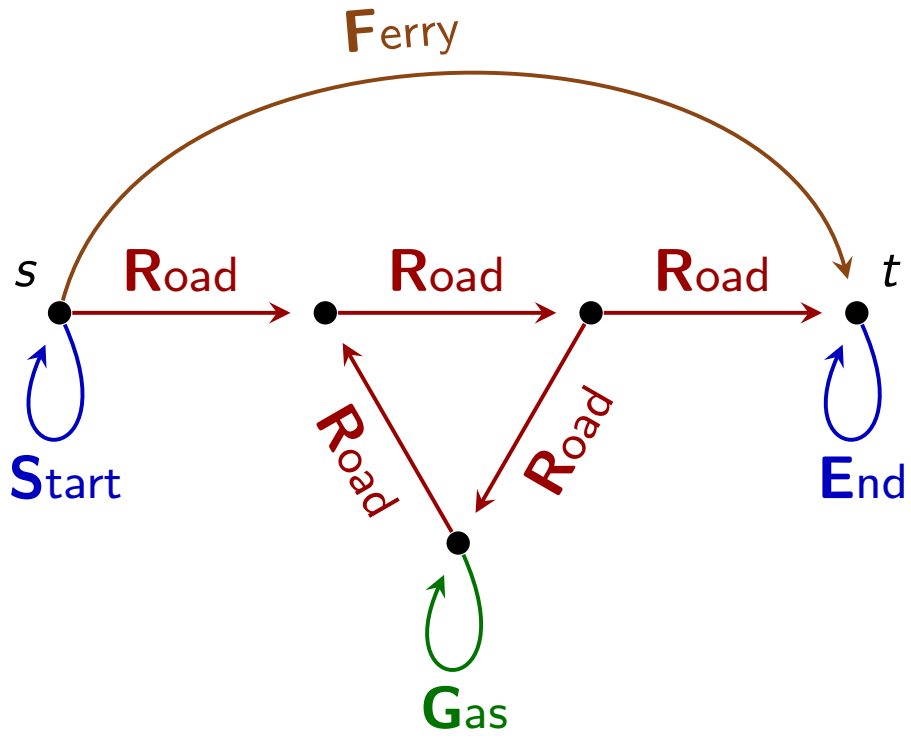
- Second most used in practice
- Return the match with the least number of edges

Binding-trail semantics

- Our proposal
- Edges may be repeated when using different atoms

Finding a way from s to t

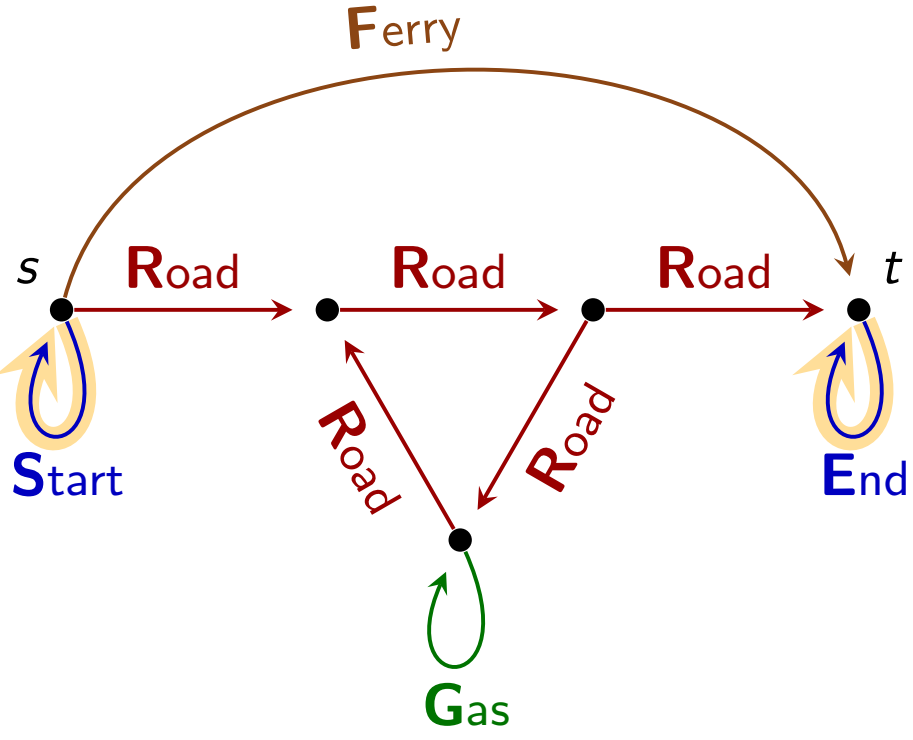
$$Q_1 = \quad \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$



Finding a way from s to t

$$Q_1 = \quad \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

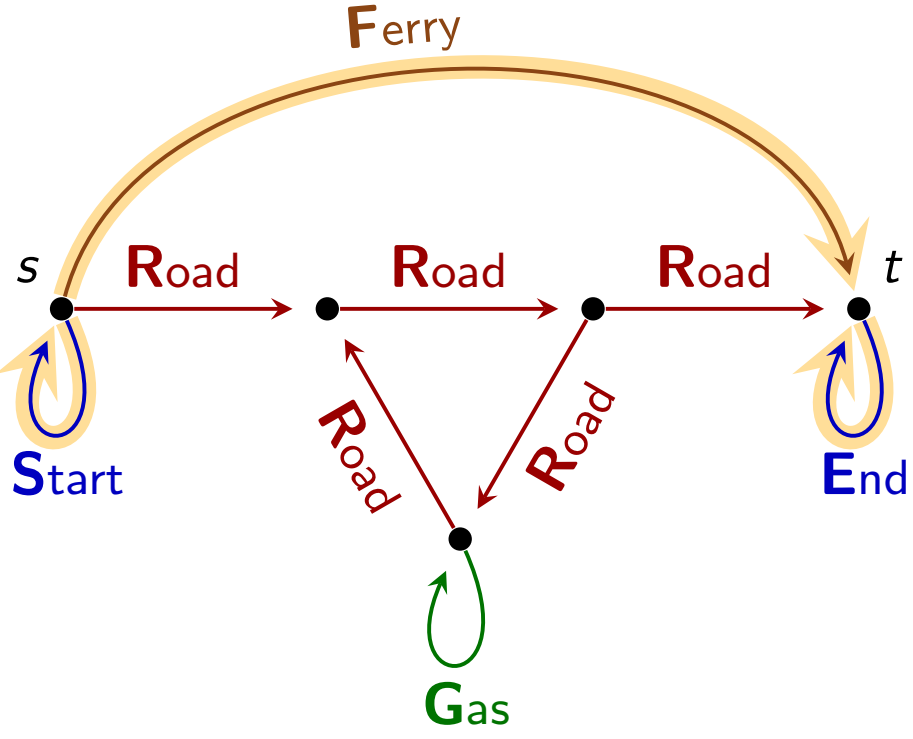


Finding a way from s to t

$$Q_1 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry

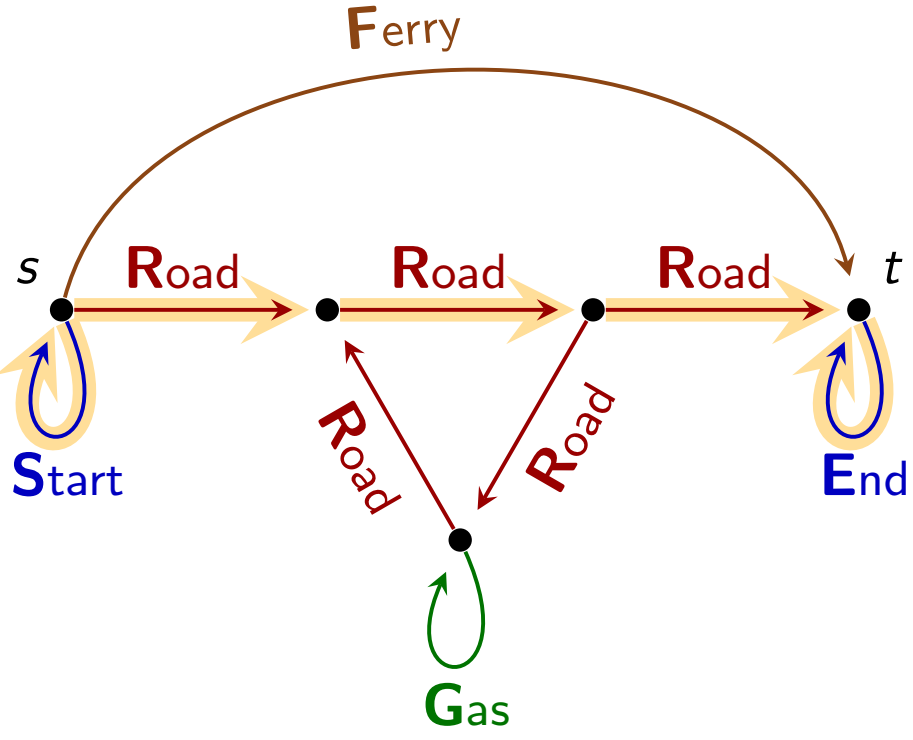


Finding a way from s to t

$$Q_1 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry
- The straight road

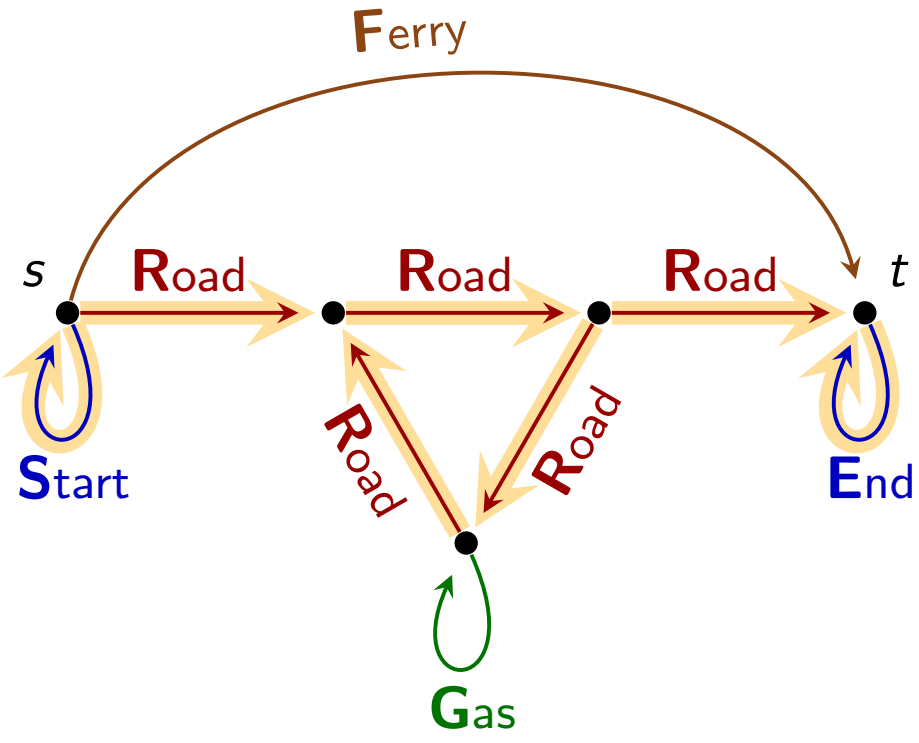


Finding a way from s to t

$$Q_1 = \quad \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry
- The straight road
- Walks with some circuit laps



Finding a way from s to t

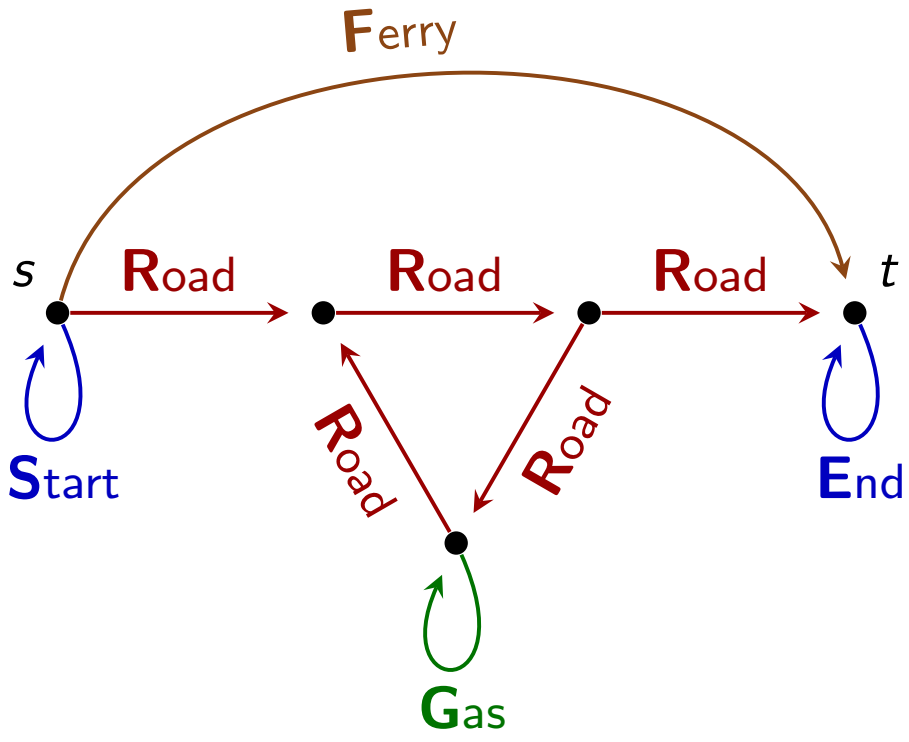
$$Q_1 = \quad \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry
- The straight road
- Walks with some circuit laps

Endpoint returns (s, t).

⚠ Very little information



Finding a way from s to t

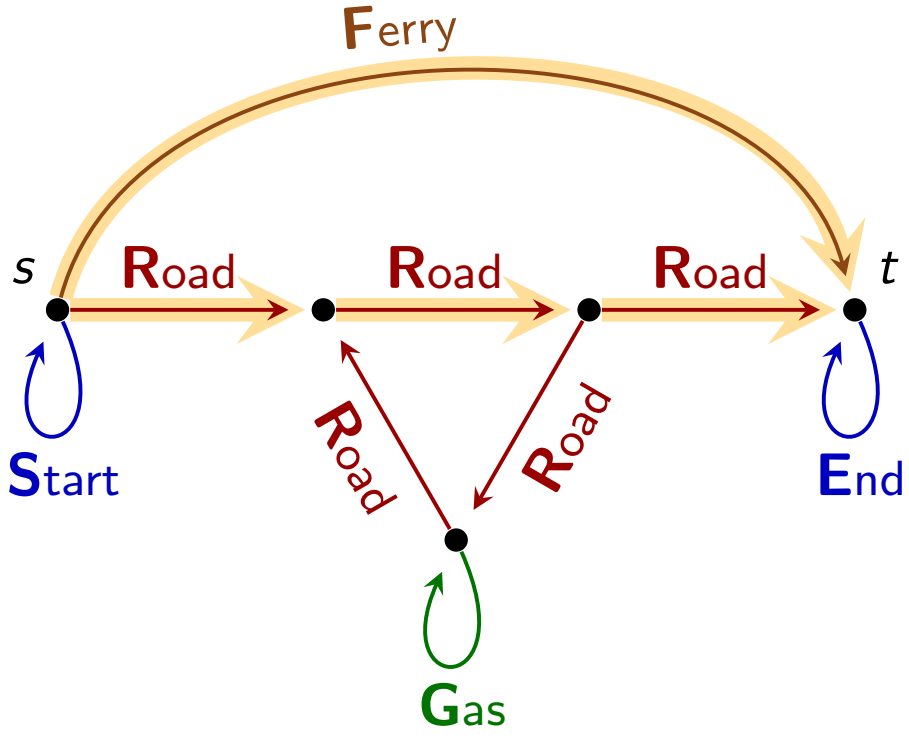
$$Q_1 = \quad \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry
- The straight road
- Walks with some circuit laps

Endpoint returns (s, t).
⚠ Very little information

Trail returns the ferry and the straight road.

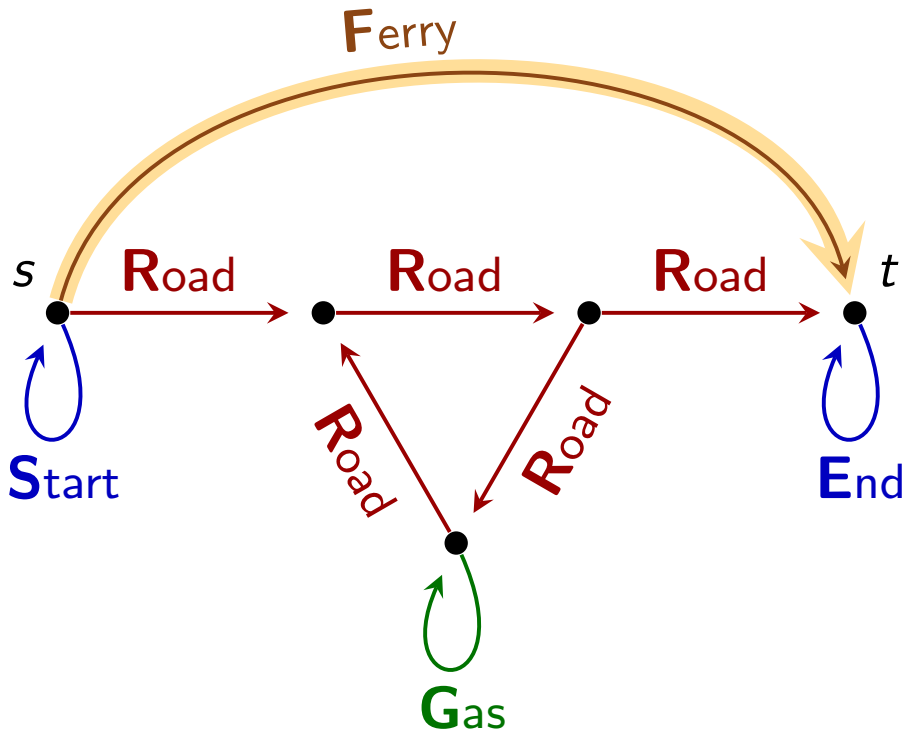


Finding a way from s to t

$$Q_1 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry
- The straight road
- Walks with some circuit laps



Endpoint returns (s, t).
⚠ Very little information

Trail returns the ferry and the straight road.

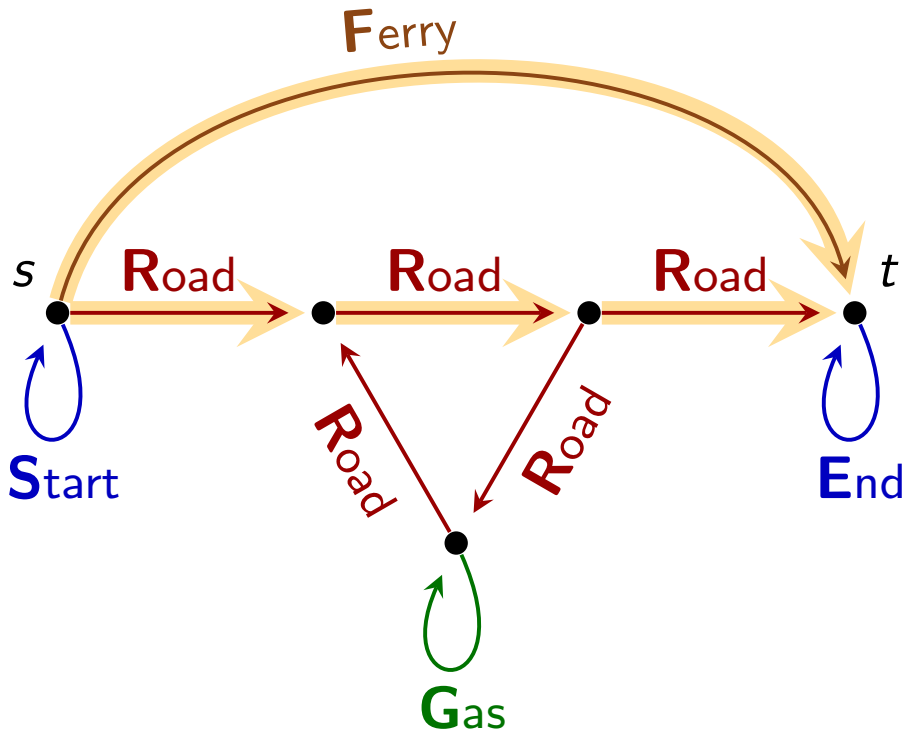
Shortest returns the ferry.
⚠ Metrics is arbitrary

Finding a way from s to t

$$Q_1 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_1?

- The ferry
- The straight road
- Walks with some circuit laps



Endpoint returns (s, t).
⚠ Very little information

Trail returns the ferry and the straight road.

Shortest returns the ferry.
⚠ Metrics is arbitrary

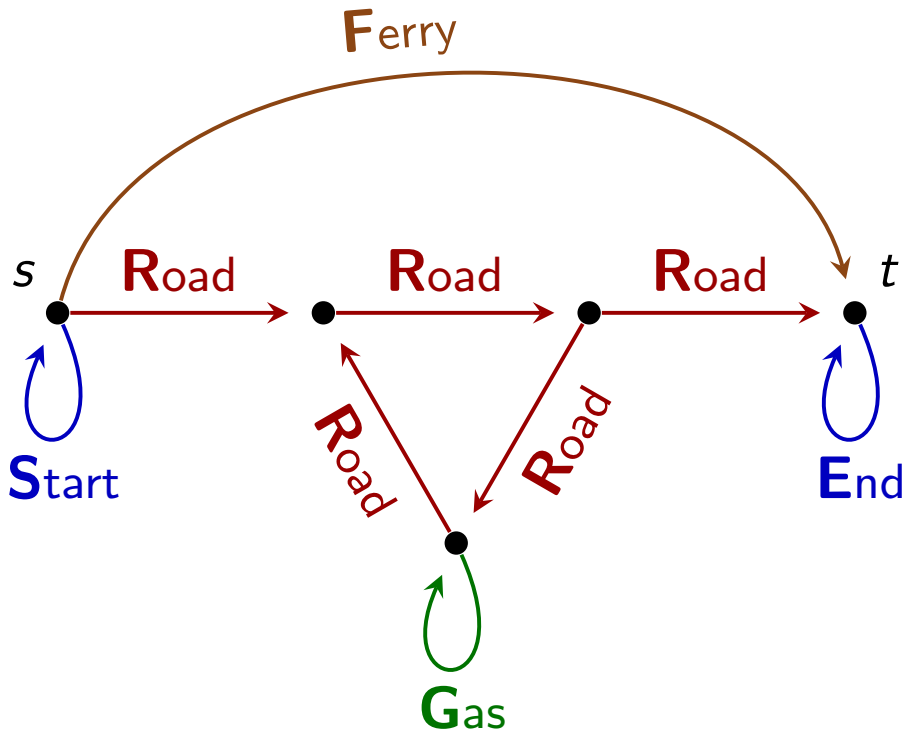
Binding-trail returns the ferry and the straight road

Finding a way from s to t with mandatory stop

$$Q_2 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{G}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_2?

- Walks with ≥ 1 circuit laps



Finding a way from s to t with mandatory stop

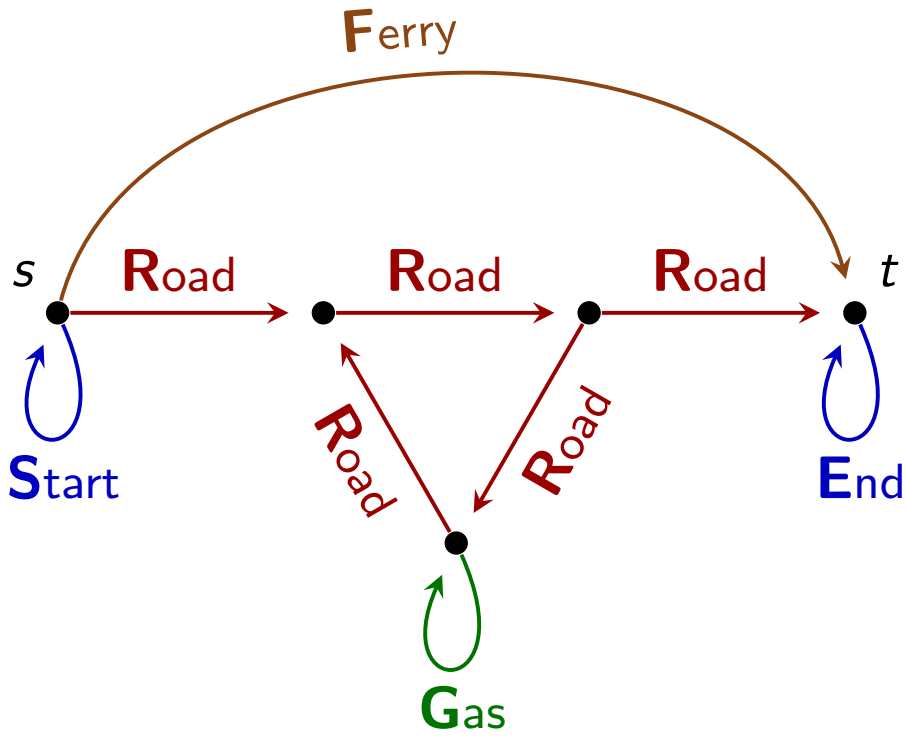
$$Q_2 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{G}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_2?

- Walks with ≥ 1 circuit laps

Endpoint returns (s, t).

⚠ Very little information

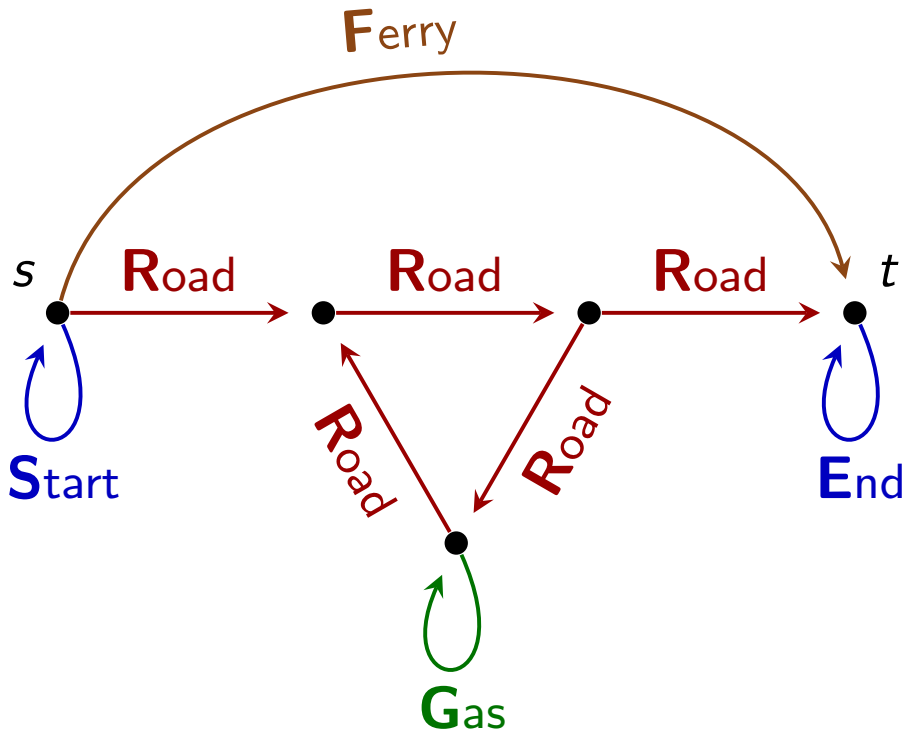


Finding a way from s to t with mandatory stop

$$Q_2 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{G}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_2?

- Walks with ≥ 1 circuit laps



Endpoint returns (s, t).

⚠ Very little information

Trail returns nothing.

⚠ Not very satisfying

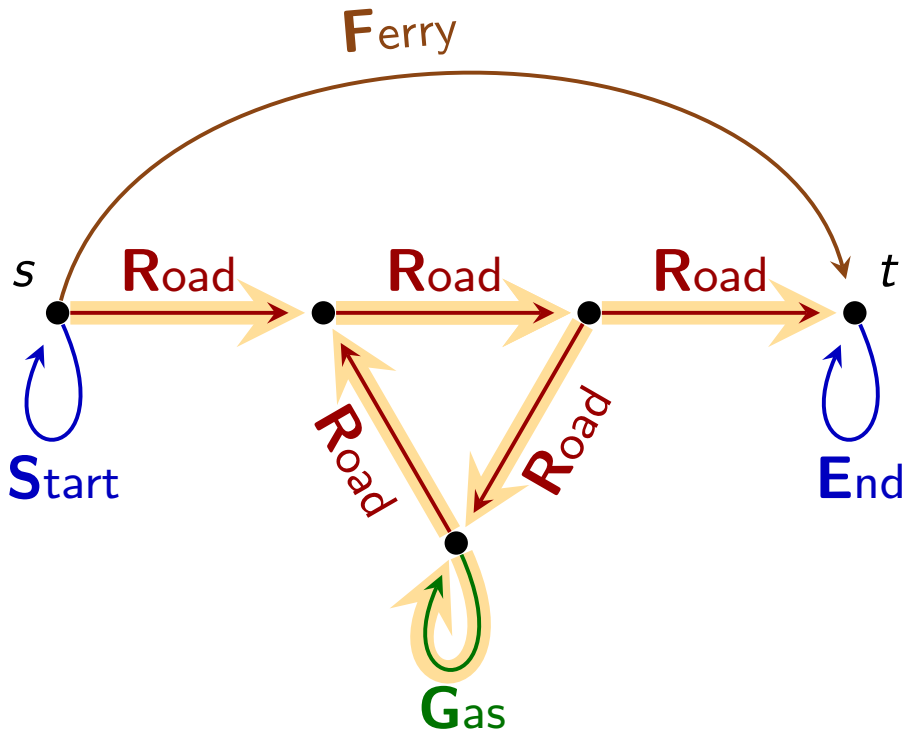
⚠ Untractable

Finding a way from s to t with mandatory stop

$$Q_2 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{G}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_2?

- Walks with ≥ 1 circuit laps



Endpoint returns (s, t).

⚠ Very little information

Trail returns nothing.

⚠ Not very satisfying

⚠ Untractable

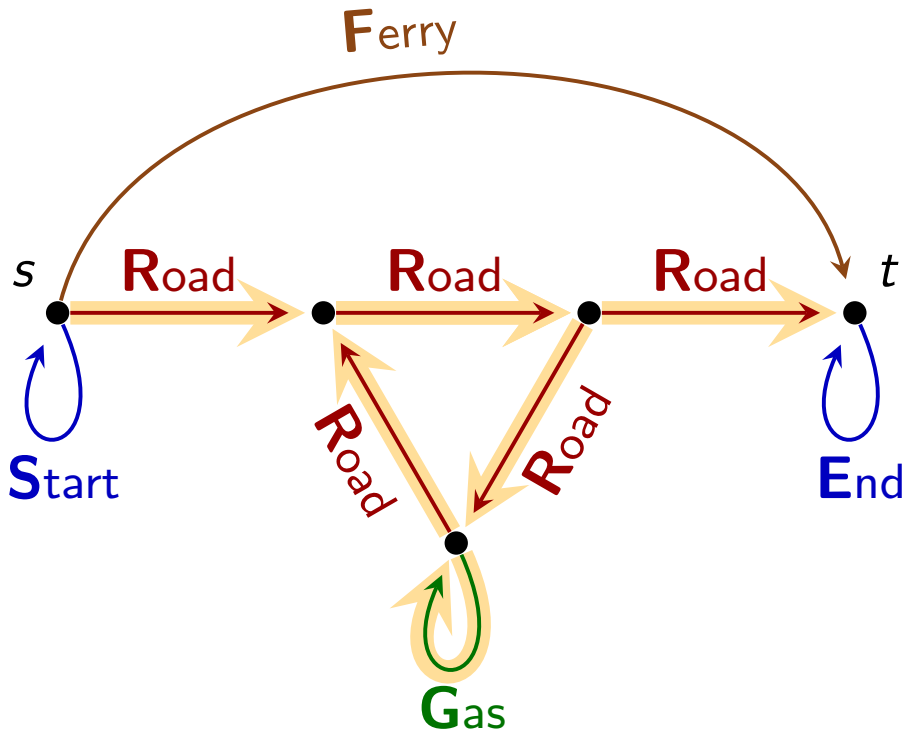
Shortest returns the walk with 1 circuit lap.

Finding a way from s to t with mandatory stop

$$Q_2 = \mathbf{S}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{G}\ (\mathbf{R}+\mathbf{F})^*\ \mathbf{E}$$

Which walks match Q_2?

- Walks with ≥ 1 circuit laps



Endpoint returns (s, t).

⚠ Very little information

Trail returns nothing.

⚠ Not very satisfying

⚠ Untractable

Shortest returns the walk with 1 circuit lap.

Binding-trail returns the walk with 1 circuit lap.

Recap of [DFM23]

User

Query is an RPQ = Regular Expression

Graph Database

Possibly infinite answer

Question: How to ensure finiteness?

In theory

Endpoint semantics

- Little information is returned

Our proposal

Binding-trail semantics

- A good compromise?

In practice

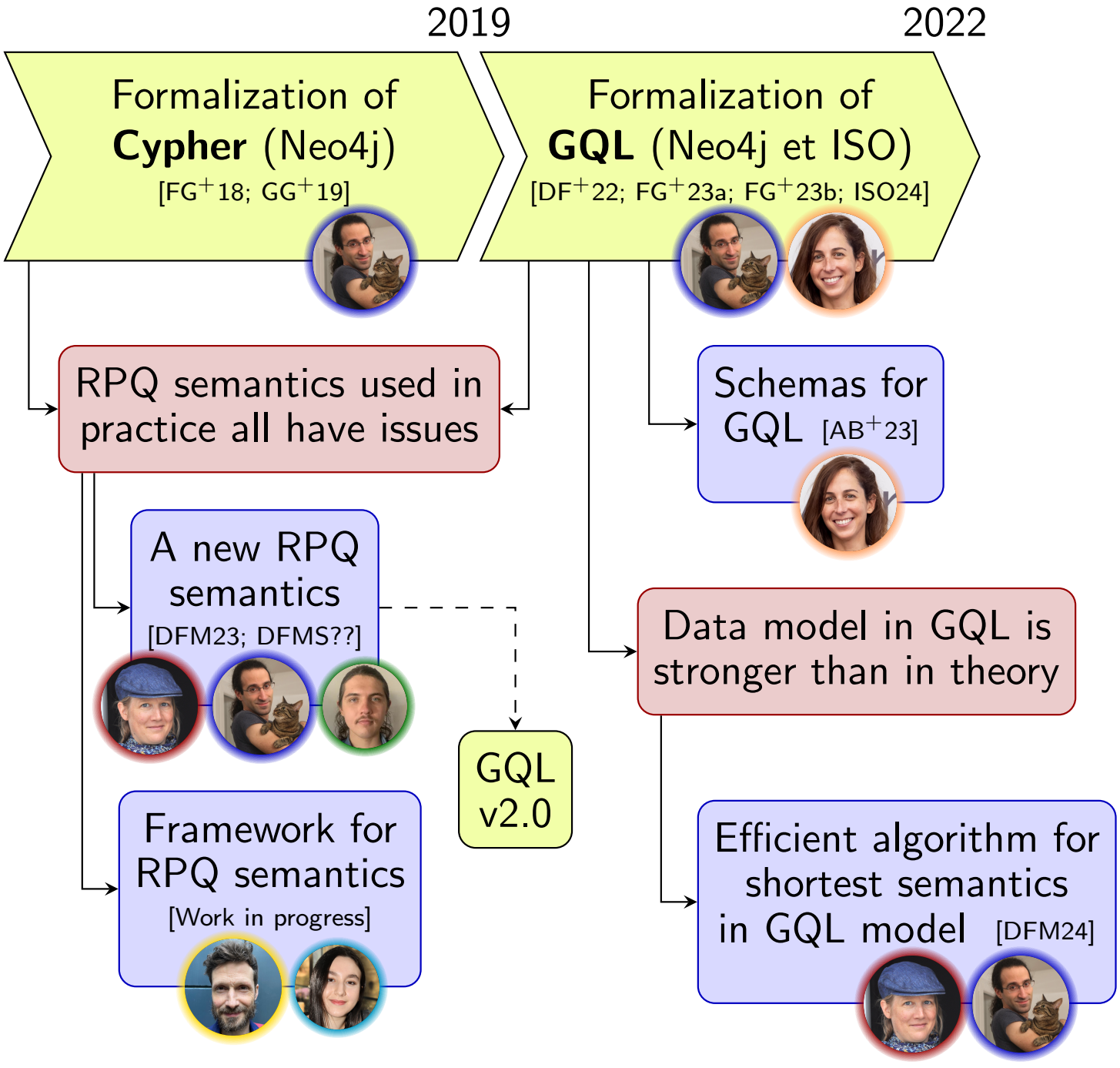
Trail semantics

- Untractable

Shortest sem.

- Arbitrary

Ongoing/Future work


2019
2022
Formalization of **Cypher** (Neo4j) [FG+18; GG+19]
Formalization of **GQL** (Neo4j et ISO) [DF+22; FG+23a; FG+23b; ISO24]
RPQ semantics used in practice all have issues
Schemas for GQL [AB+23]
A new RPQ semantics [DFM23; DFMS??]
Data model in GQL is stronger than in theory
GQL v2.0
Framework for RPQ semantics [Work in progress]
Efficient algorithm for shortest semantics in GQL model [DFM24]

2019 2022

Formalization of **Cypher** (Neo4j)
[FG+18; GG+19]

Formalization of **GQL** (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]

Formalization/Design of **Rel** (RelationalAi)
[Work in progress]

RPQ semantics used in practice all have issues

Schemas for GQL [AB+23]

A new RPQ semantics
[DFM23; DFMS??]

GQL v2.0

Data model in GQL is stronger than in theory

Framework for RPQ semantics
[Work in progress]

Efficient algorithm for shortest semantics in GQL model [DFM24]

2019

2022

Formalization of **Cypher** (Neo4j)
[FG+18; GG+19]

Formalization of **GQL** (Neo4j et ISO)
[DF+22; FG+23a; FG+23b; ISO24]

Formalization/Design of **Rel** (RelationalAi)
[Work in progress]

RPQ semantics used in practice all have issues

Schemas for GQL [AB+23]

Rel manipulates infinite objects

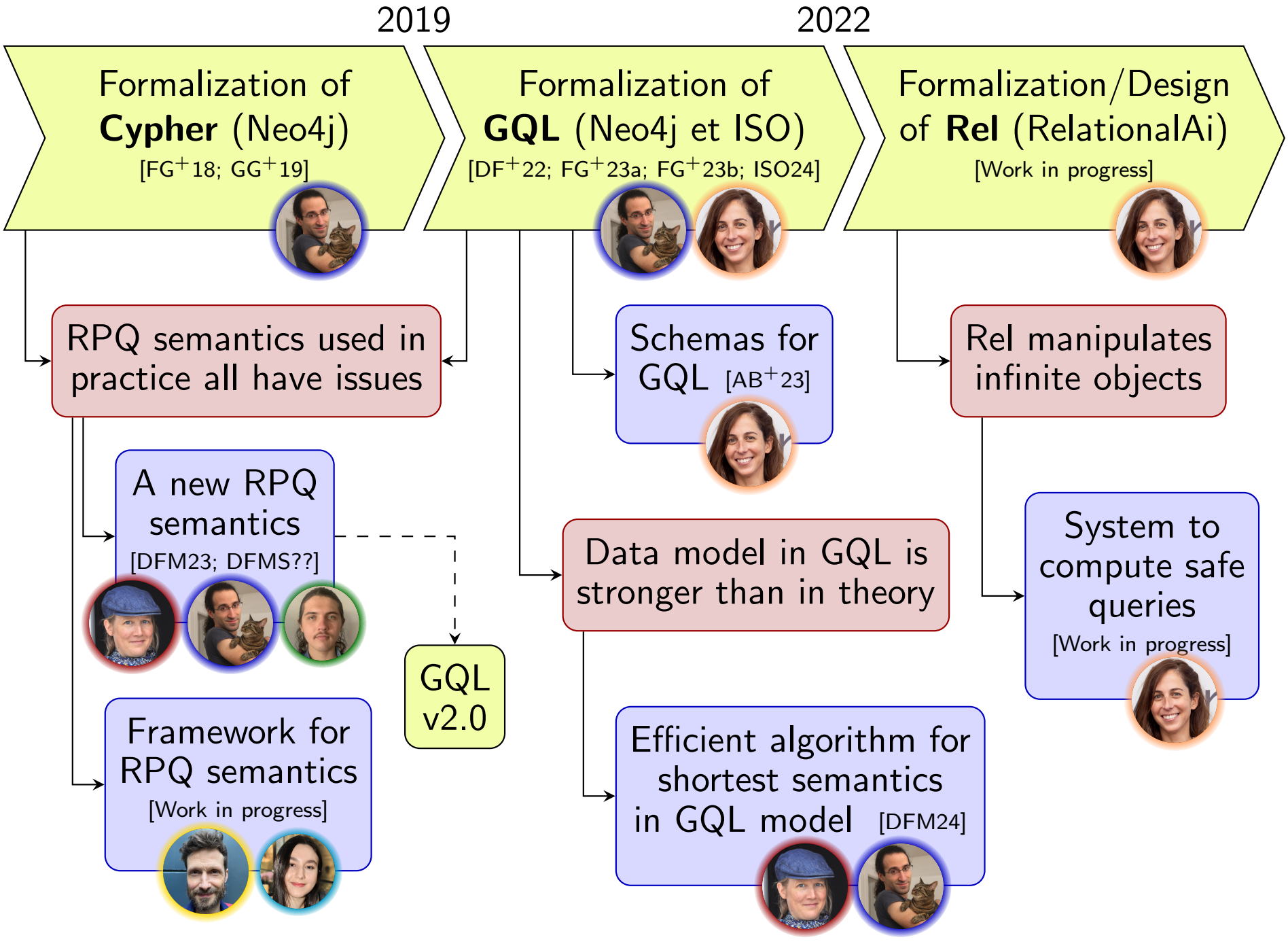
A new RPQ semantics
[DFM23; DFMS??]

GQL v2.0

Data model in GQL is stronger than in theory

Framework for RPQ semantics
[Work in progress]

Efficient algorithm for shortest semantics in GQL model [DFM24]


2019
2022
Formalization of **Cypher** (Neo4j) [FG+18; GG+19]
Formalization of **GQL** (Neo4j et ISO) [DF+22; FG+23a; FG+23b; ISO24]
Formalization/Design of **Rel** (RelationalAi) [Work in progress]
RPQ semantics used in practice all have issues
Schemas for GQL [AB+23]
Rel manipulates infinite objects
A new RPQ semantics [DFM23; DFMS??]
GQL v2.0
Data model in GQL is stronger than in theory
System to compute safe queries [Work in progress]
Framework for RPQ semantics [Work in progress]
Efficient algorithm for shortest semantics in GQL model [DFM24]

Bibliography I

[AB+23] R. Angles, A. Bonifati, S. Dumbrava, G. Fletcher, A. Green, J. Hidders, B. Li, L. Libkin, V. Marsault, W. Martens, F. Murlak, S. Plantikow, O. Savković, M. Schmidt, J. Sequeda, S. Staworko, D. Tomaszuk, H. Voigt, D. Vrgoč, M. Wu, and D. Živković. "PG-Schema: Schemas for Property Graphs". In: *Proceedings of the ACM on Management of Data (SIGMOD)*. Vol. 1. 2. Industrial Track. ACM, June 2023.

[DFM23] C. David, N. Francis, and V. Marsault. "Run-Based Semantics for RPQs". In: *Principles of Knowledge Representation and Reasoning (KR'23)*. Aug. 2023, pp. 178–187.

[DFM24] C. David, N. Francis, and V. Marsault. "Distinct Shortest Walk Enumeration for RPQs". In: *PODS*. Vol. 2. 2. ACM, 2024. URL: https://doi.org/10.1145/3651601.

[DF+22] A. Deutsch, N. Francis, A. Green, K. Hare, B. Li, L. Libkin, T. Lindaaker, V. Marsault, W. Martens, J. Michels, F. Murlak, S. Plantikow, P. Selmer, H. Voigt, O. van Rest, D. Vrgoč, M. Wu, and F. Zemke. "Graph Pattern Matching in GQL and SQL/PGQ". In: *SIGMOD'22*. 2022. URL: https://doi.org/10.1145/3514221.3526057.

Bibliography II

[FG+23a] N. Francis, A. Gheerbrant, P. Guagliardo, L. Libkin, V. Marsault, W. Martens, F. Murlak, L. Peterfreund, A. Rogova, and D. Vrgoč. "A Researcher's Digest of GQL". In: *ICDT*. Invited talk. 2023. URL: https://doi.org/10.4230/LIPIcs.ICDT.2023.1.

[FG+23b] N. Francis, A. Gheerbrant, P. Guagliardo, L. Libkin, V. Marsault, W. Martens, F. Murlak, L. Peterfreund, A. Rogova, and D. Vrgoč. "GPC: A Pattern Calculus for Property Graphs". In: *PODS'23*. 2023. URL: https://doi.org/10.1145/3584372.3588662.

[FG+18] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor. "Cypher: An Evolving Query Language for Property Graphs". In: *SIGMOD'18*. ACM, 2018. URL: http://dx.doi.org/10.1145/3183713.3190657.

[ISO24] *GQL*. Standard ISO/IEC CD 39075. International Organization for Standardization (ISO), 2024. URL: https://www.iso.org/standard/76120.html.

Bibliography III

[GG+19] A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Schuster, P. Selmer, and H. Voigt. "Updating Graph Databases with Cypher". In: *VLDB*. 2019. URL: https://doi.org/10.14778/3352063.3352139.

Example Queries in Cypher and GQL

Q_1 in Cypher

```
MATCH ({id:"Paris"}) -[:Road|Ferry*]-> ({id:"Lyon"})
```

Q_2 in GQL

```
MATCH(s WHERE s.id="Paris")
     -[IS Road|Ferry]->* (m WHERE m.hasGas=true)
     -[IS Road|Ferry]->* (t WHERE t.id="Lyon")
```