Run-based semantics for RPQs

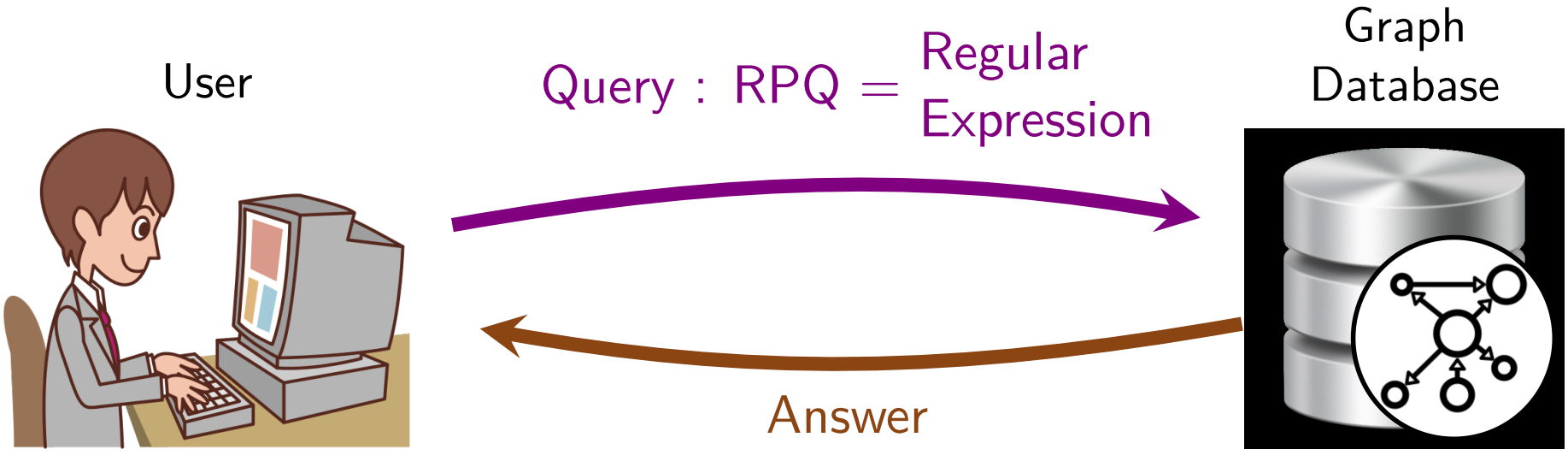
Victor MARSAULT*
joint work with Claire DAVID* and Nadime FRANCIS*

* Université Gustave-Eiffel, CNRS, LIGM

Verigraph Meeting, 2024-04-30, Grenoble, France

(mostly Claire slides used for KR'23)

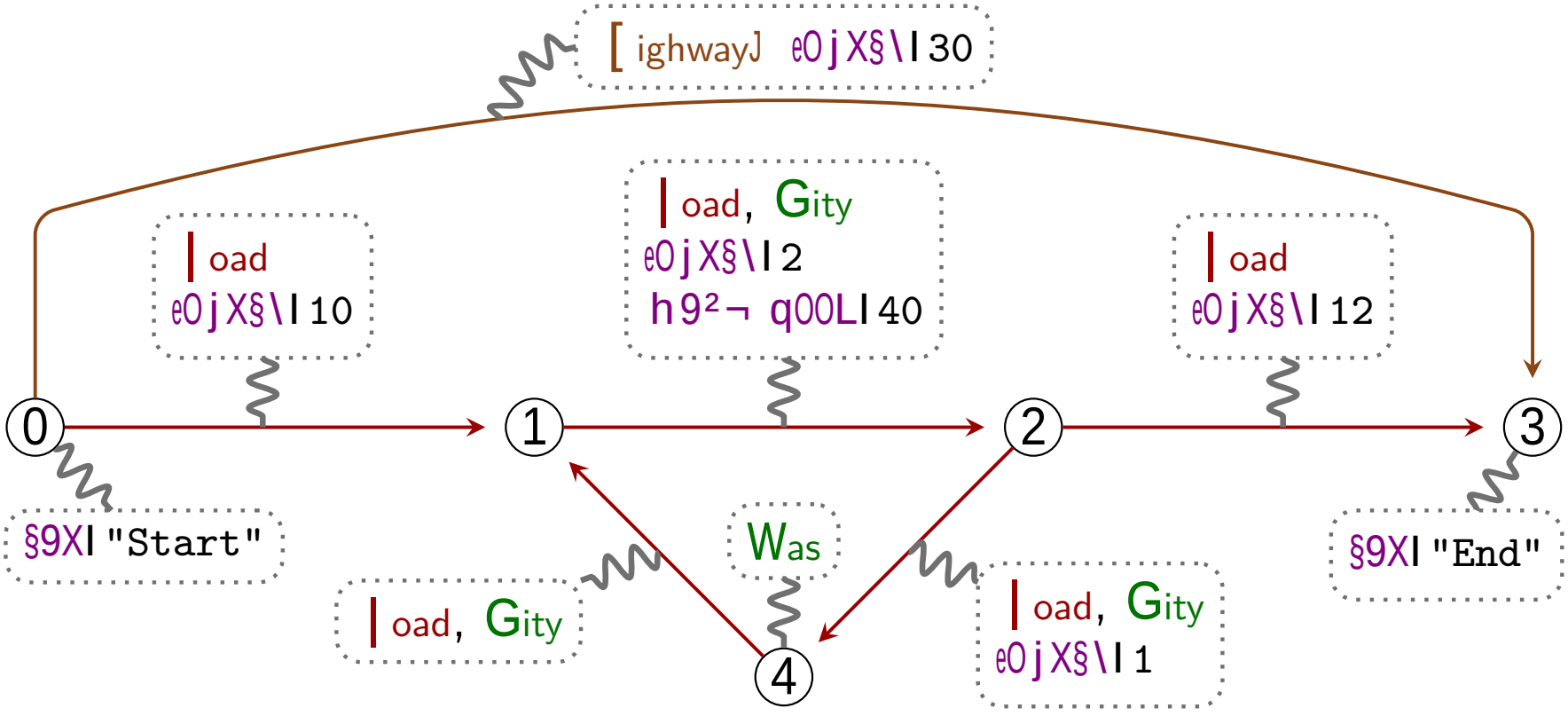
How to extract efficiently meaningful answers ?



What is a good semantics for RPQ's over graph DB's ?

- Meaningful answers
- Good complexity

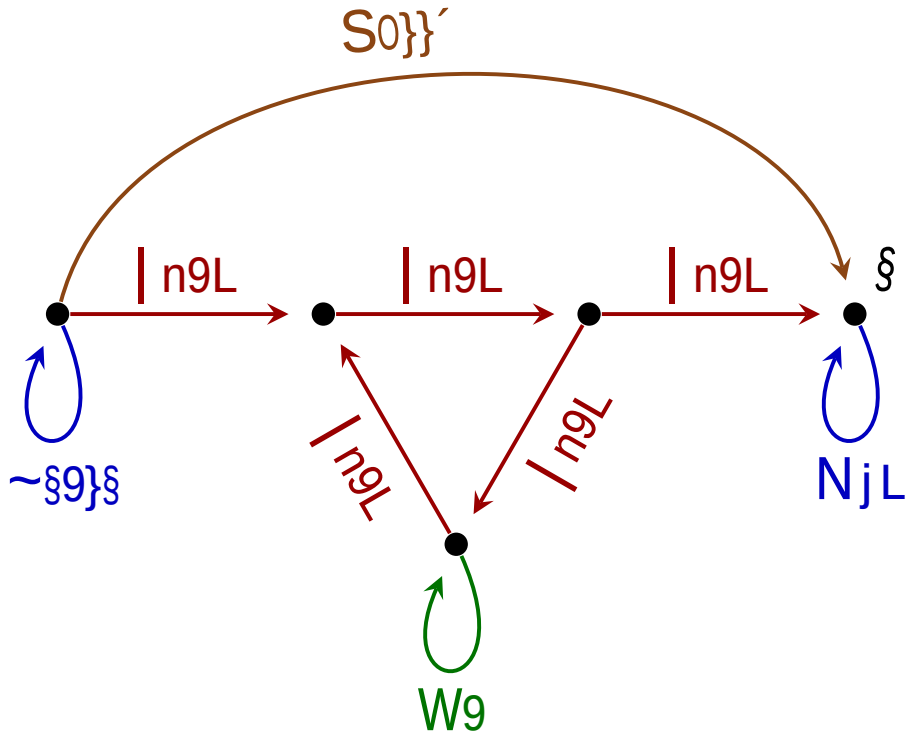
Graph DB - In practice Property graphs



RDF is another data model used in practice...

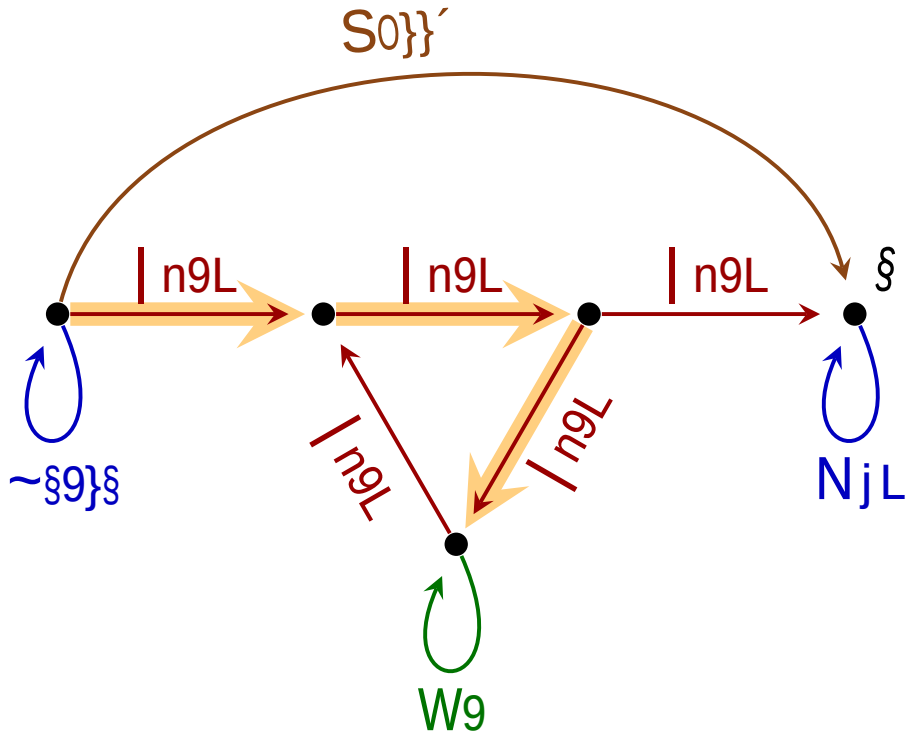
Graph DB - Our abstraction

- Finite label alphabet:
  $= f\sim;\ |\ ;\ S;\ W;\ Ng$
- Vertices
- Edges labelled over



Graph DB - Our abstraction

- Finite label alphabet:
  $= f\sim;\ |\ ; S; W; Ng$
- Vertices
- Edges labelled over

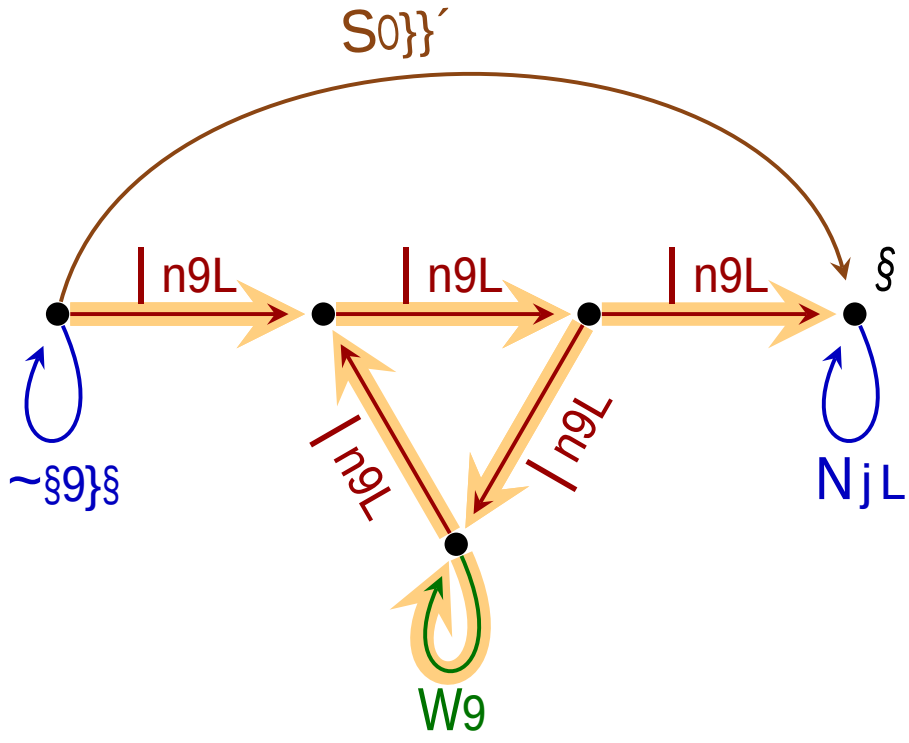


Terminogy: Walk

- Consistent sequence of edges
- Repetitions are allowed

Graph DB - Our abstraction

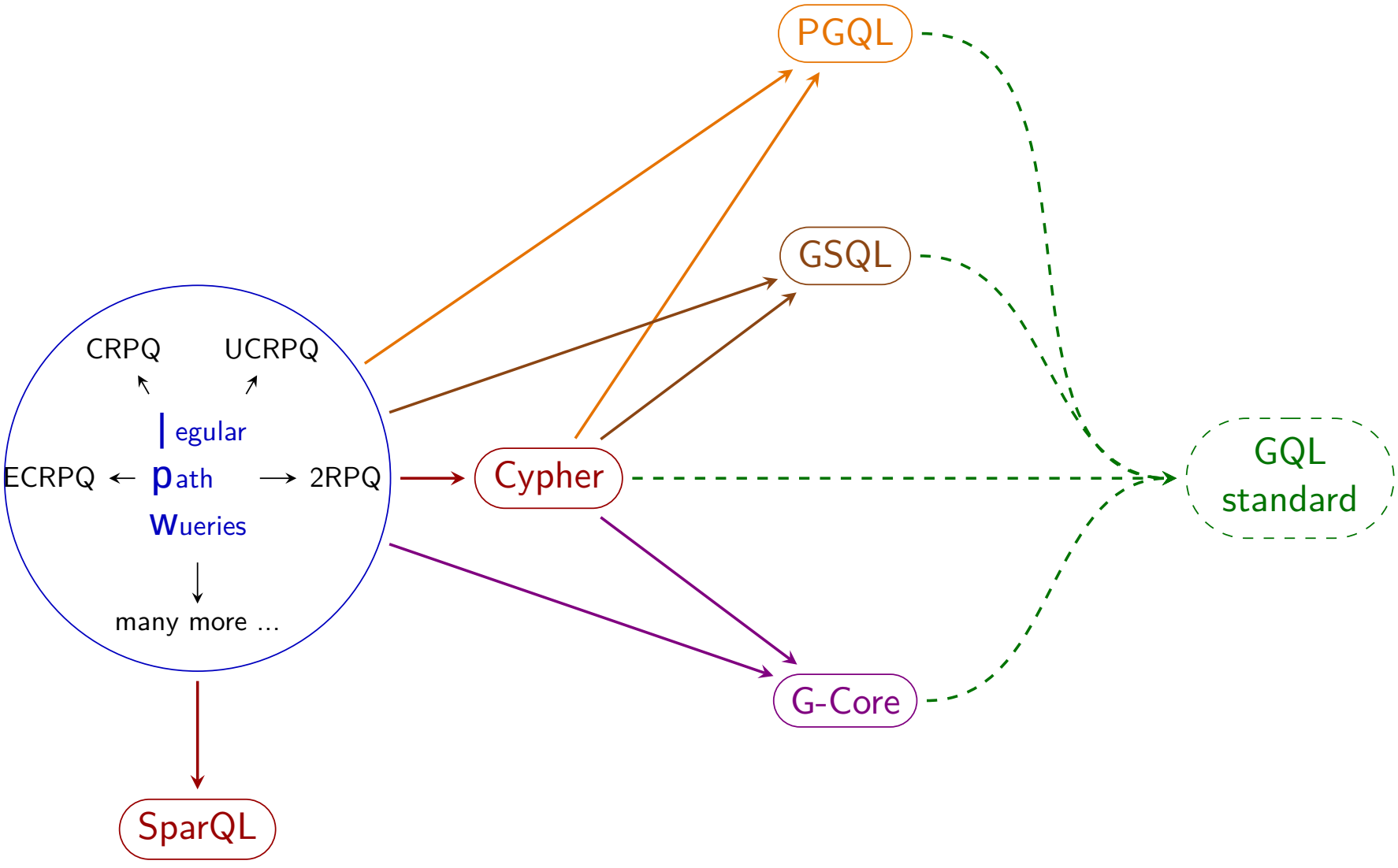
- Finite label alphabet:
  $= f\sim;\ |\ ; S; W; Ng$
- Vertices
- Edges labelled over



Terminogy: Walk

- Consistent sequence of edges
- Repetitions are allowed

RPQs - Core of most query languages for graph DB



RPQ - in theory

$w ::=$	$\mathbf{A}$	Atoms
	ww	Concatenation
	$w + w$	Disjunction
	w^*	Kleene star

where $\mathbf{A}$ is a label in the graph.

Label of a walk

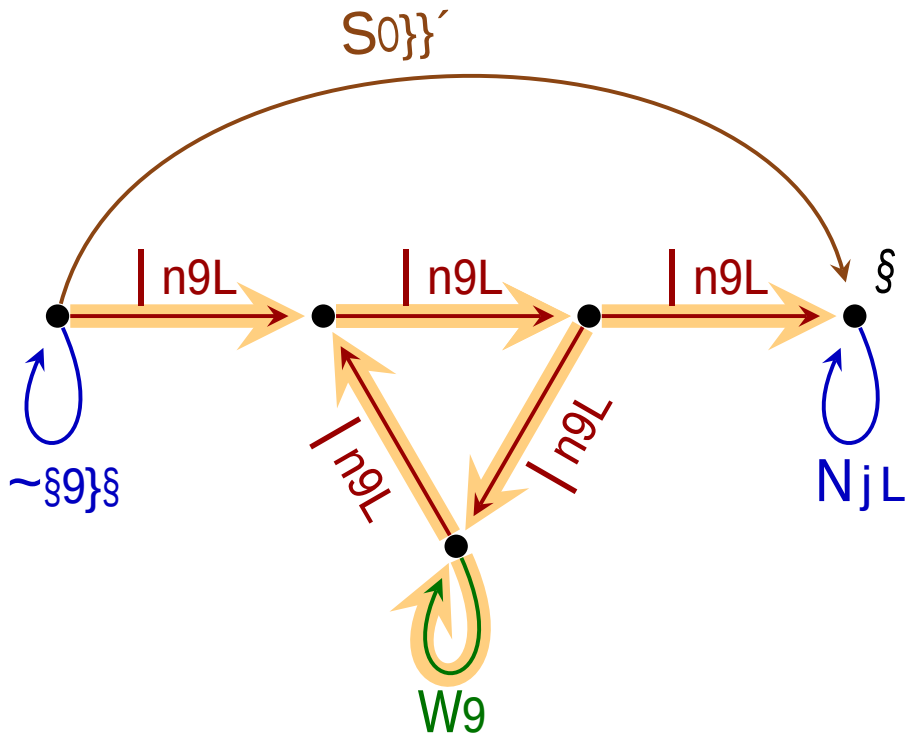
Contenation of labels of edges

Ex : | | |W| | |

Definition: match for w

A walk ° such that the label of ° matches w.

Ex: match for |*W|*

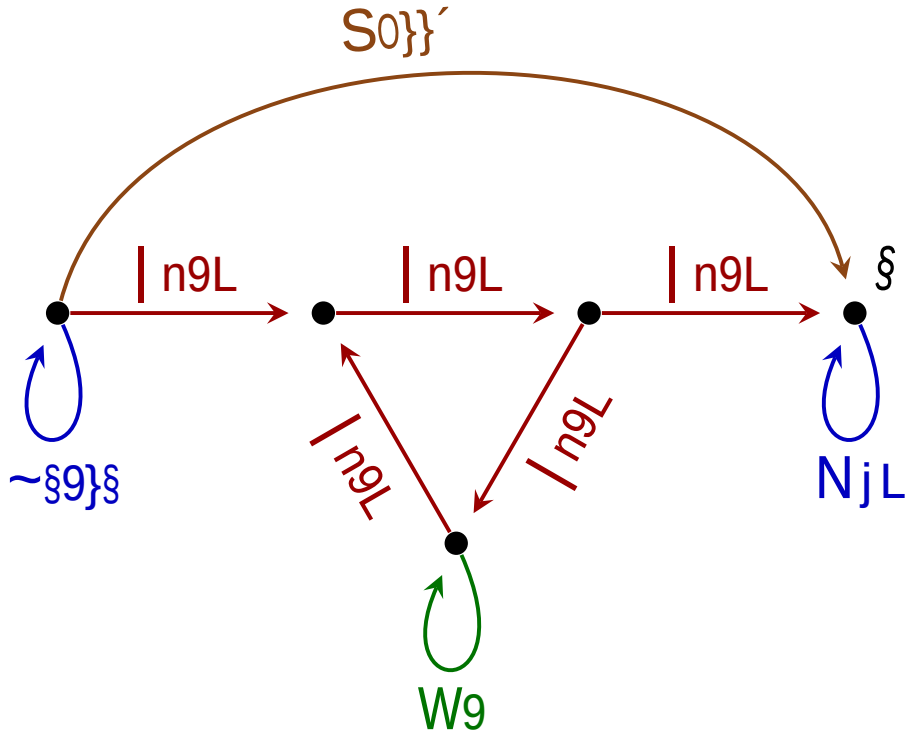


Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| + S)* N

Which walks match w_1?



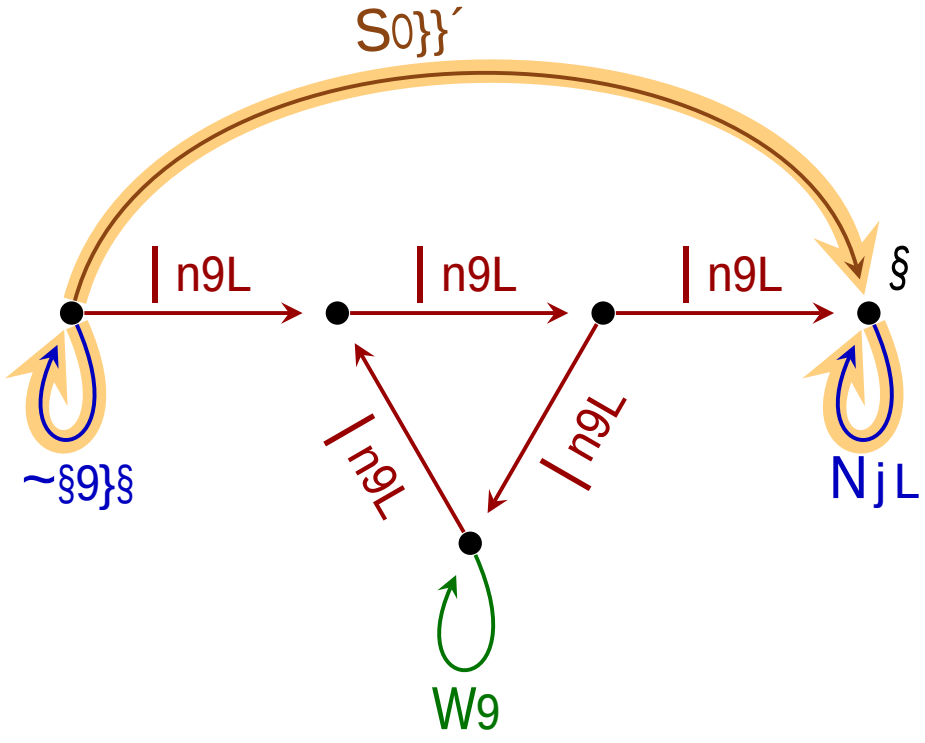
Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| + S)* N

Which walks match w_1?

- The ferry



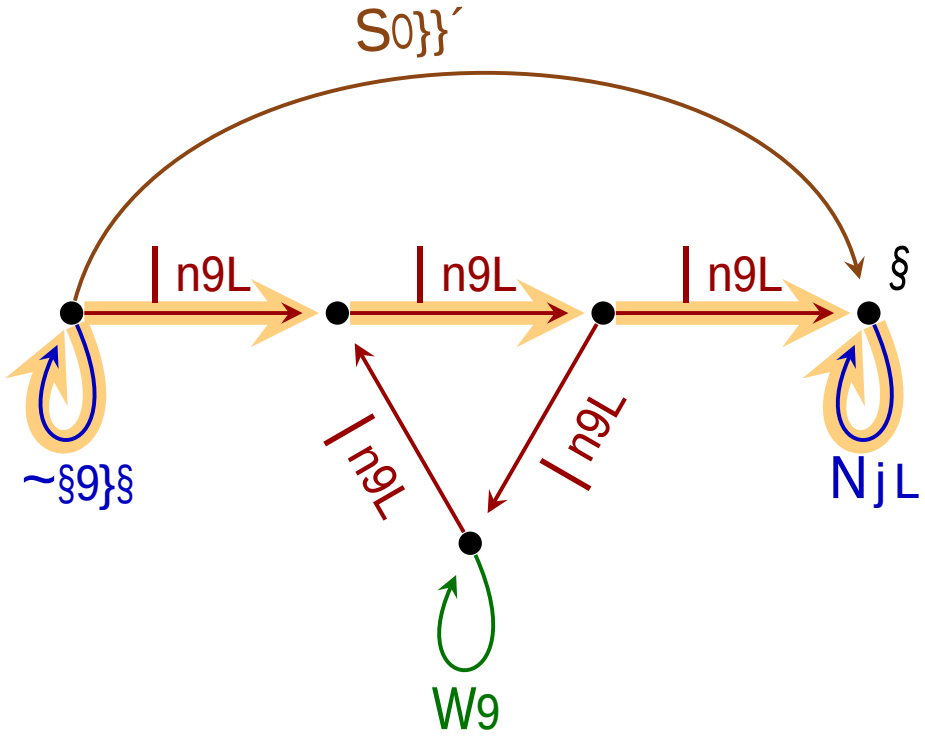
Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| + S)* N

Which walks match w_1?

- The ferry
- The straight road



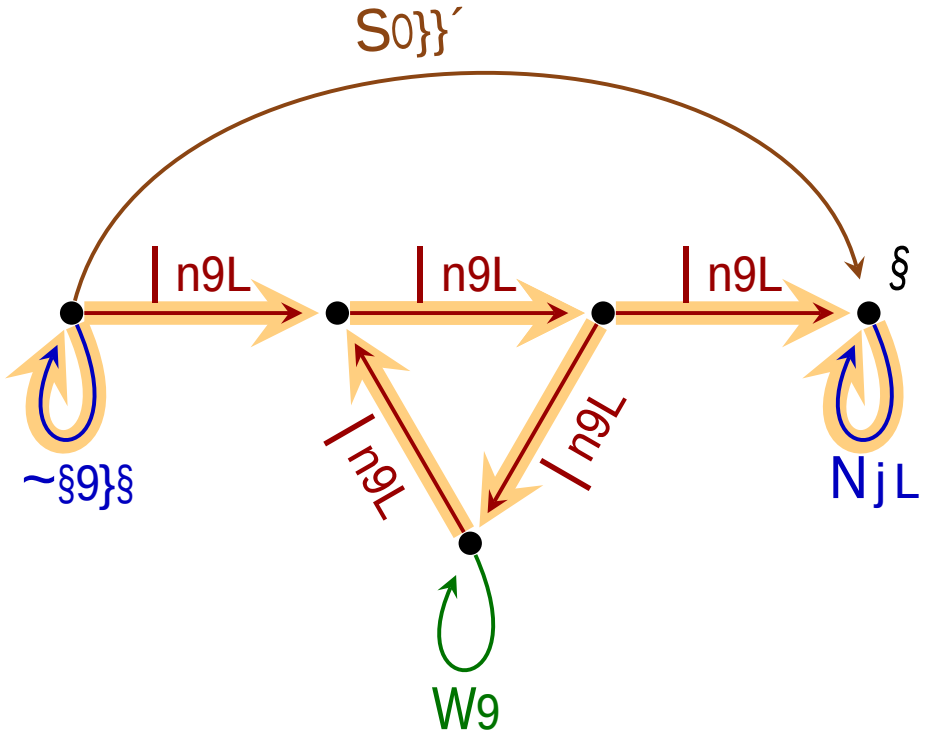
Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| + S)* N

Which walks match w_1?

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



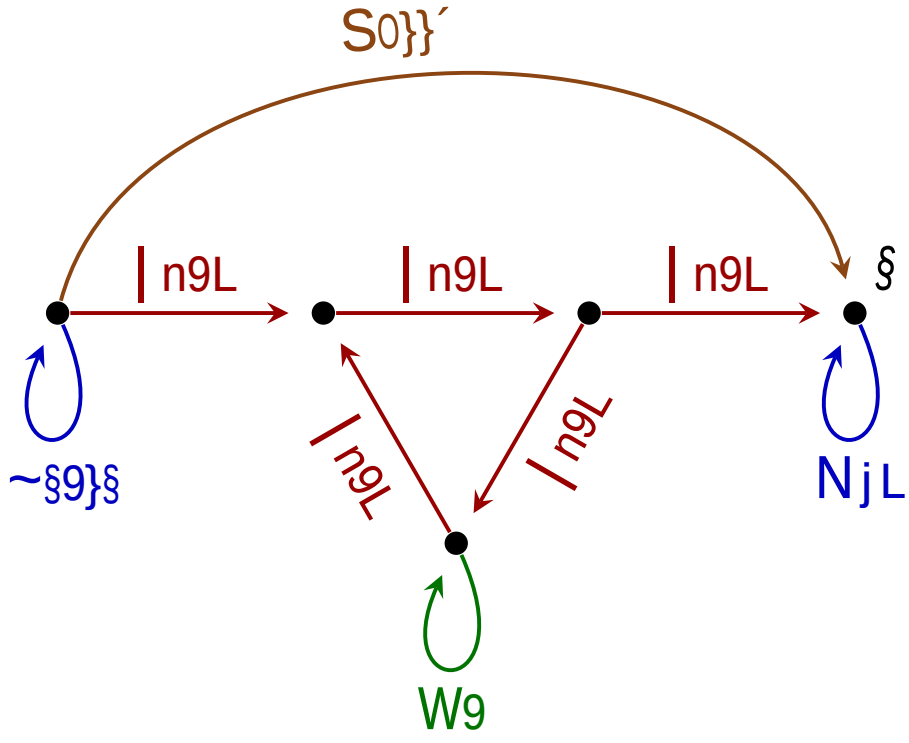
Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| + S)* N

Which walks match w_1?

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



ā̈uu °_§\ h9jL9§n}´ X9 §nqȲ

w_2 = ~ (| + S)* W (| + S)* N

Which walks match w_2?

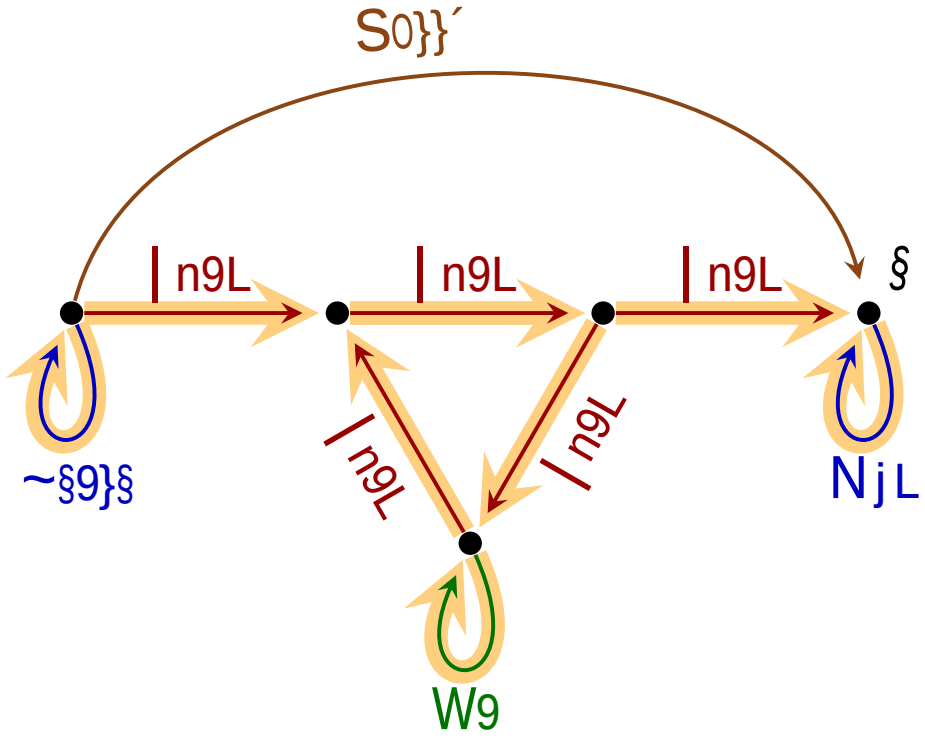
Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| + S)* N

Which walks match w_1?

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



ā̈uu °_§\ h9jL9§n}´ X9 §nqȲ

w_2 = ~ (| + S)* W (| + S)* N

Which walks match w_2?

- Walks with some circuit laps

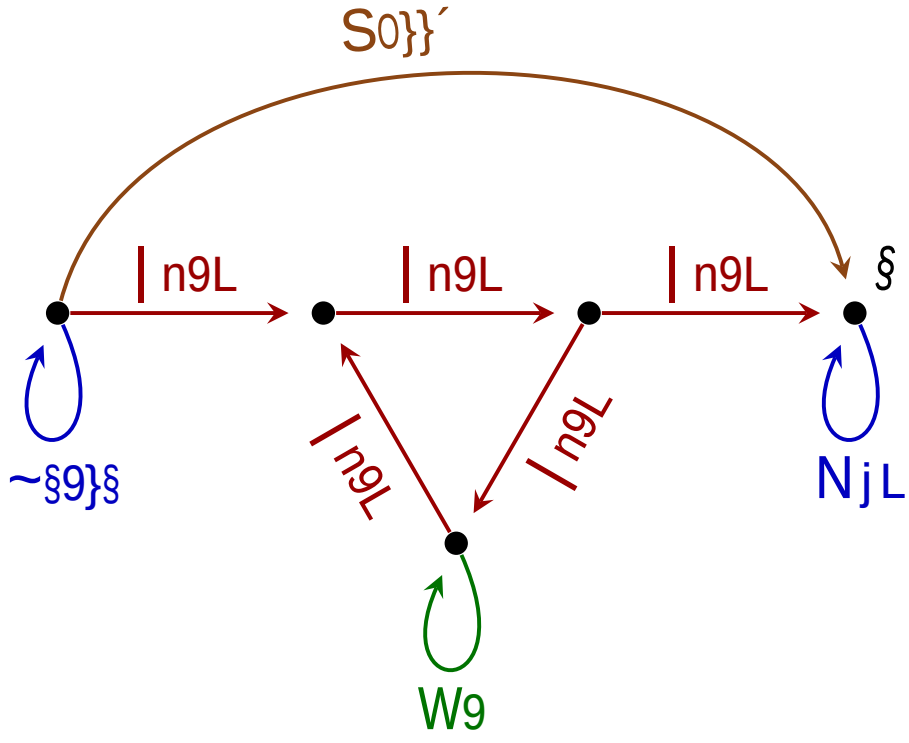
Our two running RPQs

ōS_jL 9 °9´ T}nh §n §Ȳ

w_1 = ~ (| +S)* N

Which walks match w_1?

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



ā̈uu °_§\ h9jL9§n}´ X9 §nqȲ

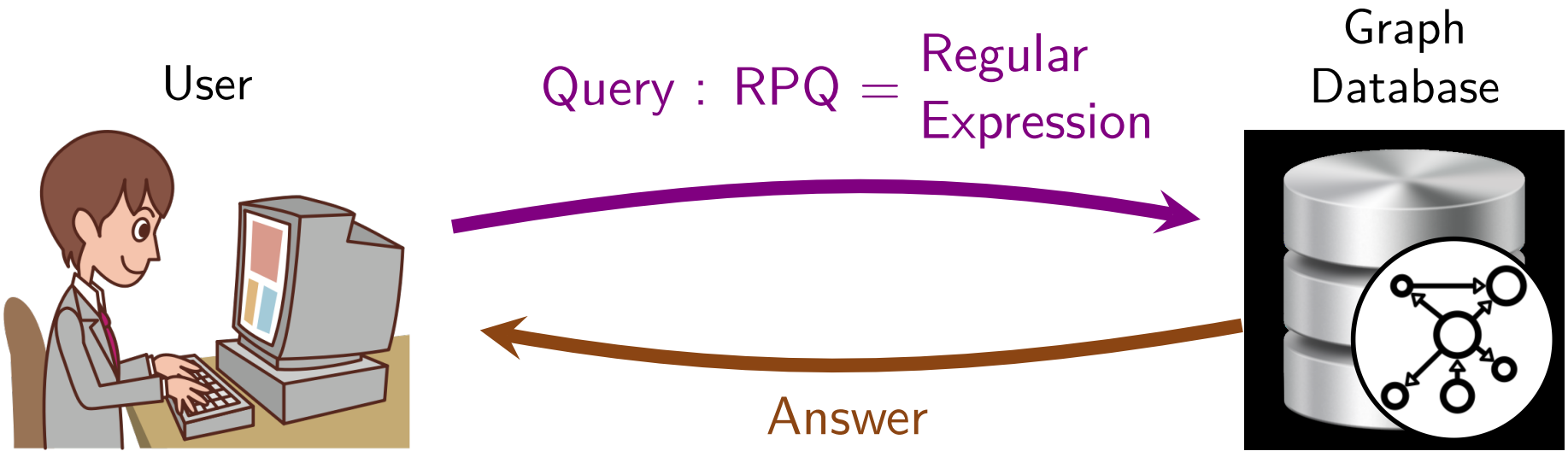
w_2 = ~ (| +S)* W (| +S)* N

Which walks match w_2?

- Walks with some circuit laps

) Infinitely many matches

How to extract efficiently meaningful answers ?



What is a good semantics for RPQ's over graph DB's ?

- Meaningful answer
- Good complexity

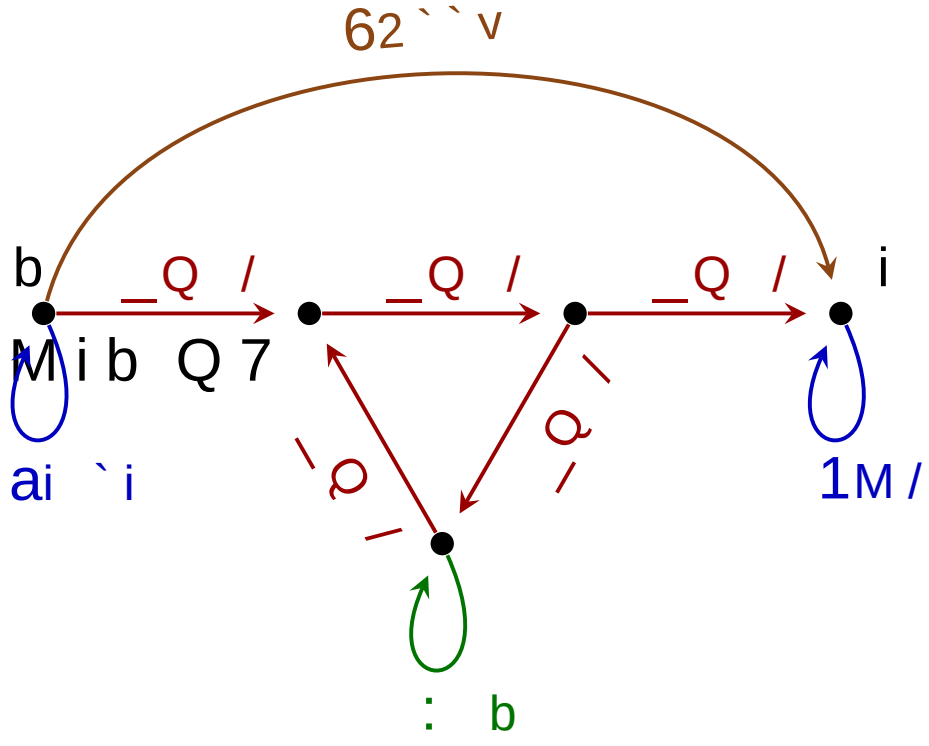
⚠ Infinitely many matches but users expect a finite answer ⚠

>QKQKQ`T?BbK b2K MiB+b

J BMi?2Q`2iB+ H b2K MiB+b- aS _ZG U_.6V

.2}MBiBQM

- _2im`Mb i?2 2M/TQBMib Q7 K i+?BM; r HFb



$Z_1 =$ a U_ Y6) 1

$Z_2 =$ a U_ Y6) : U_ Y6) 1

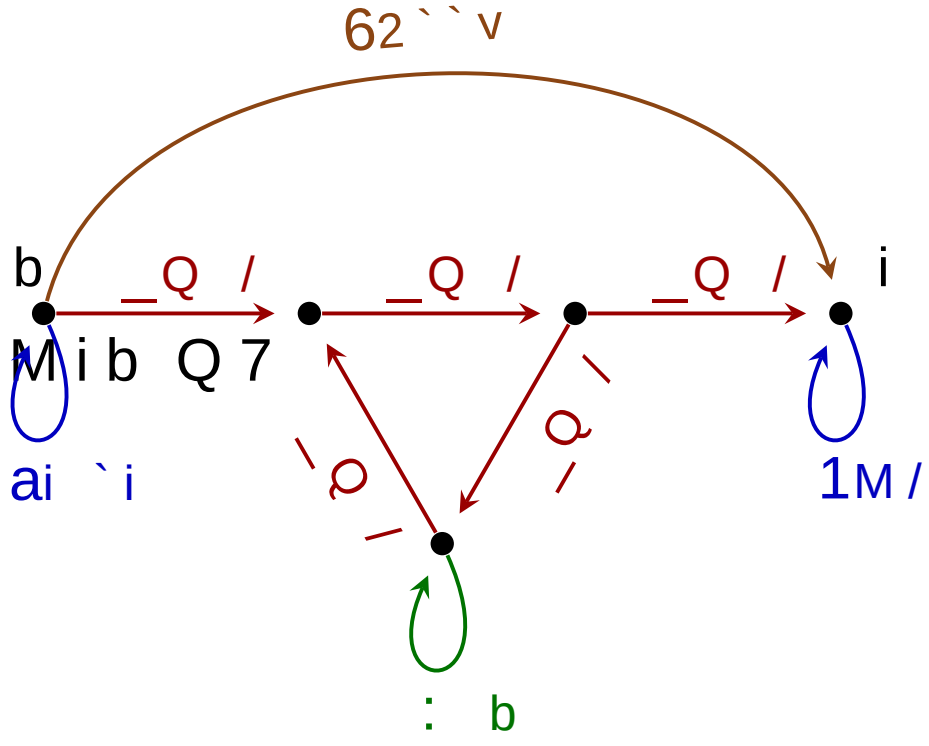
- "Qi? `2im`M Q(M2 T B`,

>QKQKQ`T?BbK b2K MiB+b

J BMi?2Q`2iB+ H b2K MiB+b- aS _ZG U_.6V

.2}MBiBQM

- _2im`Mb i?2 2M/TQBMib Q7 K i+?BM; r HFb



$Z_1 =$ a U_ Y6) 1

$Z_2 =$ a U_ Y6) : U_ Y6) 1

- "Qi? `2im`M Q(M2 T B`

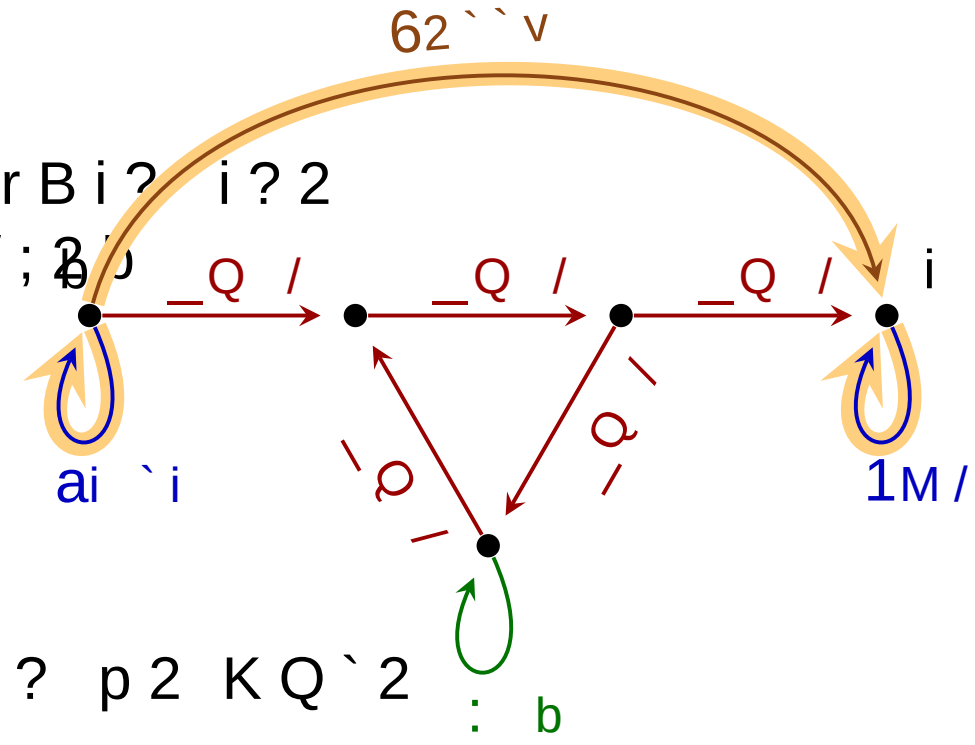
- q2HH ;`QmM/2/ i?2Q
- 1{+B2Mi H;Q`Bi?Kb
- _2im`Mb HBiiH2 BM7

a?Q`i2bi@r HF b2K MiB+b

S:ZG UP` +H2V- :aZG UhB;2`:` T?V- :@*Q`2 (M;H2b 2i

.2}MBiBQM

- _2im`Mb i?2 r HF rBi? i?2 H2 bi MmK#2` Q7 2/;2b



Z_1 = a U_ Y6) 1

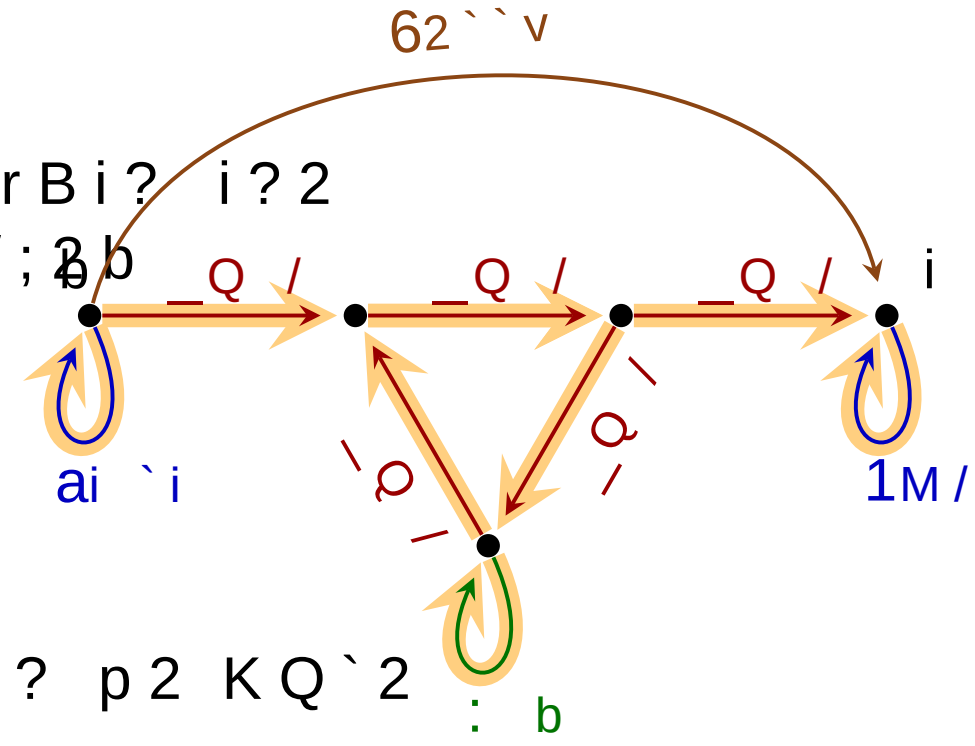
- _2im`Mb i?2 72``v
- q HFb mbBM;`Q /b ? p2 KQ`2 2/;2b

a?Q`i2bi@r HF b2K MiB+b

S:ZG UP` +H2V- :aZG UhB;2`:` T?V- :@*Q`2 (M;H2b 2i

.2}MBiBQM

- _2im`Mb i?2 r HF rBi? i?2 H2 bi MmK#2` Q7 2/;2b



Z_1 = a U_Y6) 1

- _2im`Mb i?2 72``v
- q HFb mbBM;`Q /b ? p2 KQ`2 2/;2b

Z_2 = a U_Y6) : U_Y6) 1

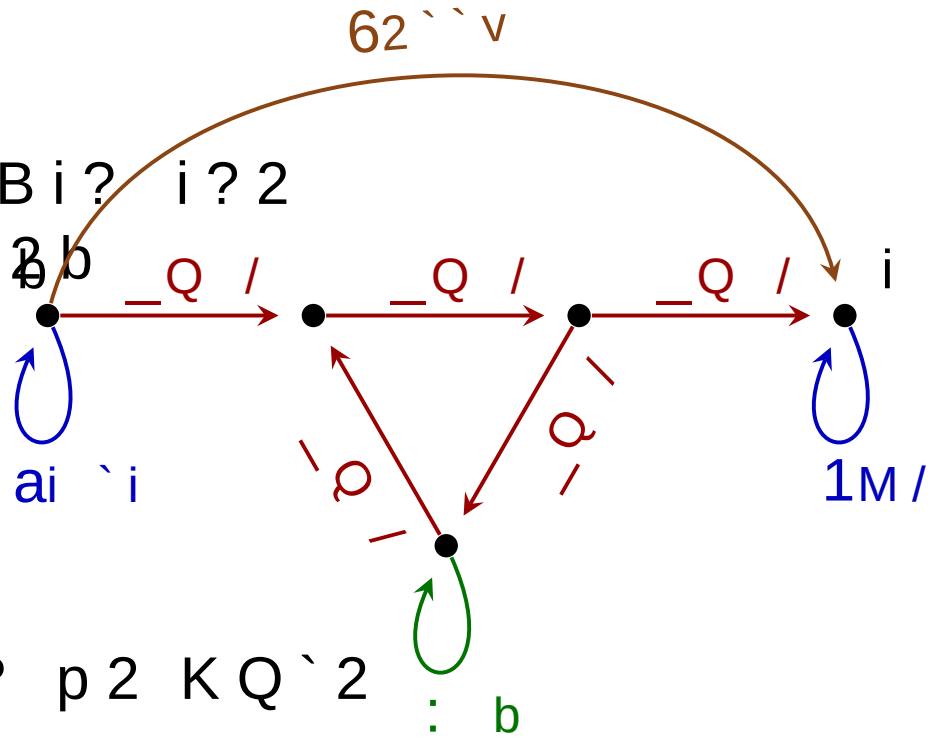
- _2im`Mb i?2 r HF rBi? QM2 +B`+mBi H T

a?Q`i2bi@r HF b2K MiB+b

S:ZG UP` +H2V- :aZG UhB;2`:` T?V- :@*Q`2 (M;H2b 2i

.2}MBiBQM

- _2im`Mb i?2 r HF rBi? i?2 H2 bi MmK#2` Q7 2/;2b



Z_1 = a U_ Y6) 1

- _2im`Mb i?2 72``v
- q HFb mbBM;`Q /b ? p2 KQ`2 2/;2b

Z_2 = a U_ Y6) : U_ Y6) 1

- _2im`Mb i?2 r HF rBi +B`+mBi H T

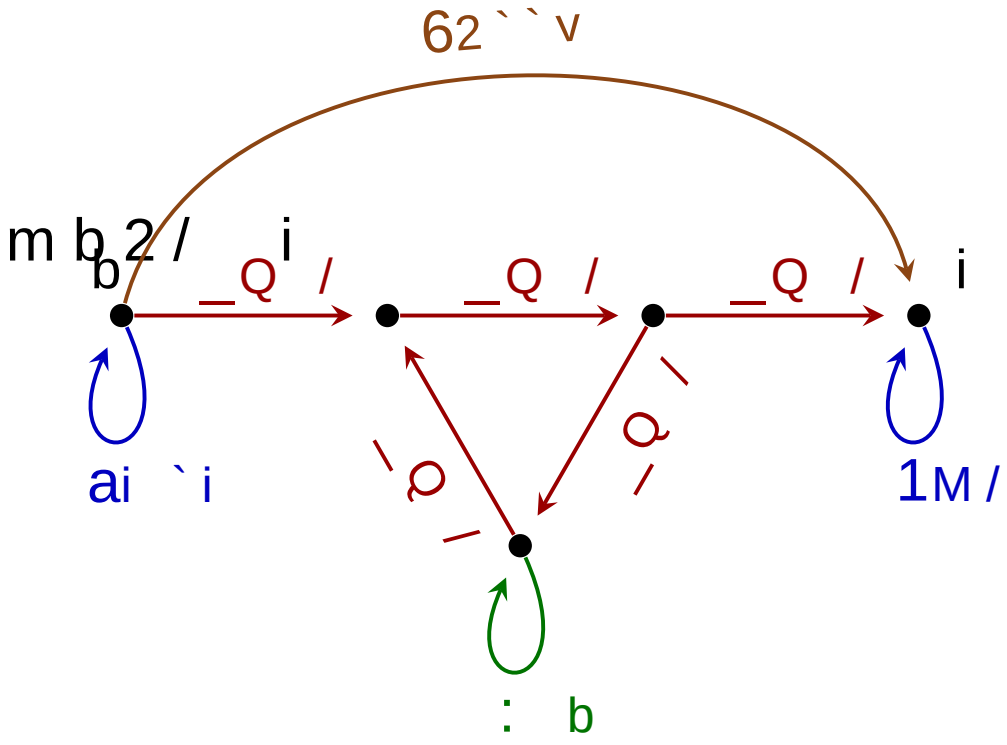
- 1{+B2Mi H;Q`Bi?Kb
- `#Bi` `v +?QB+2 Q7
- LQ p2`iB+ H TQbi@T

h` BH b2K MiB+b

*vT?2` UL2Q9DV- :ZG

.2}MBiBQM

- _2im`Mb r HFb
- 1 +? 2/;2 + M #2 mb2/ i KQbi QM+2



h` BH b2K MiB+b

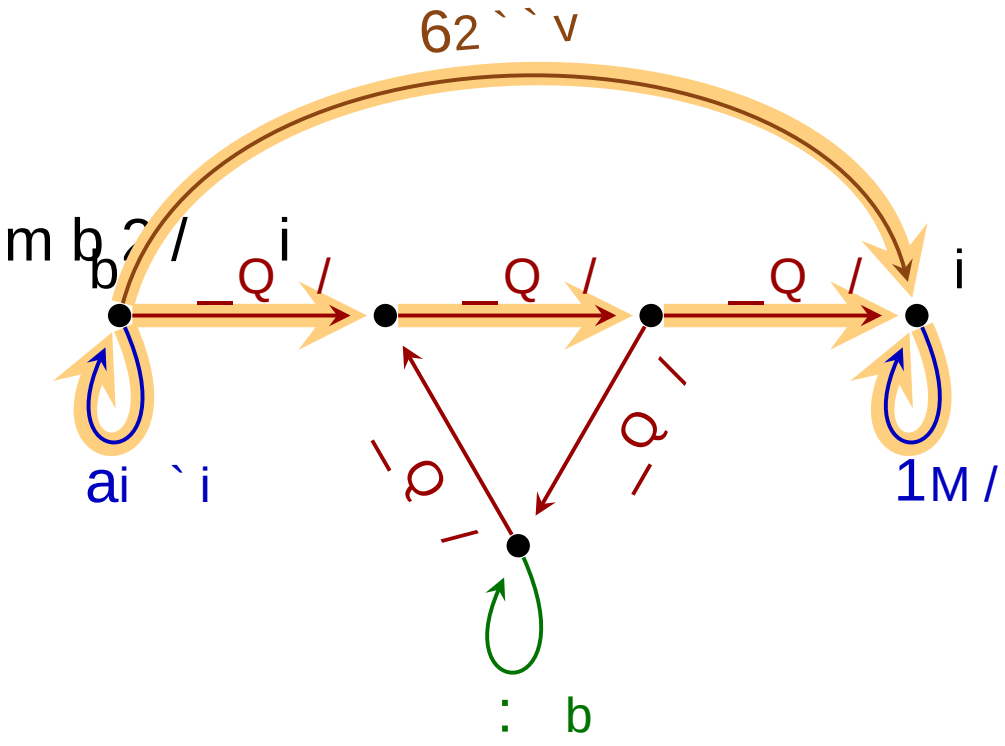
*vT?2` UL2Q9DV- :ZG

.2}MBiBQM

- _2im`Mb r HFb
- 1 +? 2/;2 + M #2 mb?/ i KQbi QM+2

Z_1 = a U_ Y6) 1

- Z_1 `2im`Mb
  - i?2 72``v
  - i?2 bi` B;?i `Q /



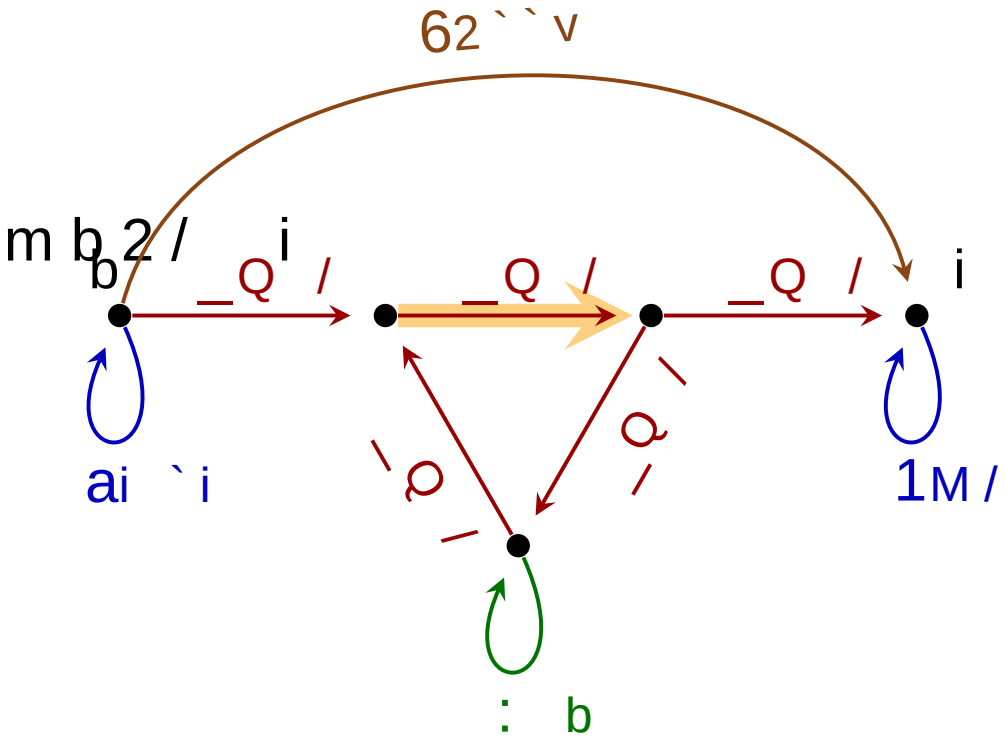
h` BH b2K MiB+b

*vT?2` UL2Q9DV- :ZG

.2}MBiBQM

- _2im`Mb r HFb
- 1 +? 2/;2 + M #2 mb2/ i KQbi QM+2

Z_1 = a U_ Y6) 1



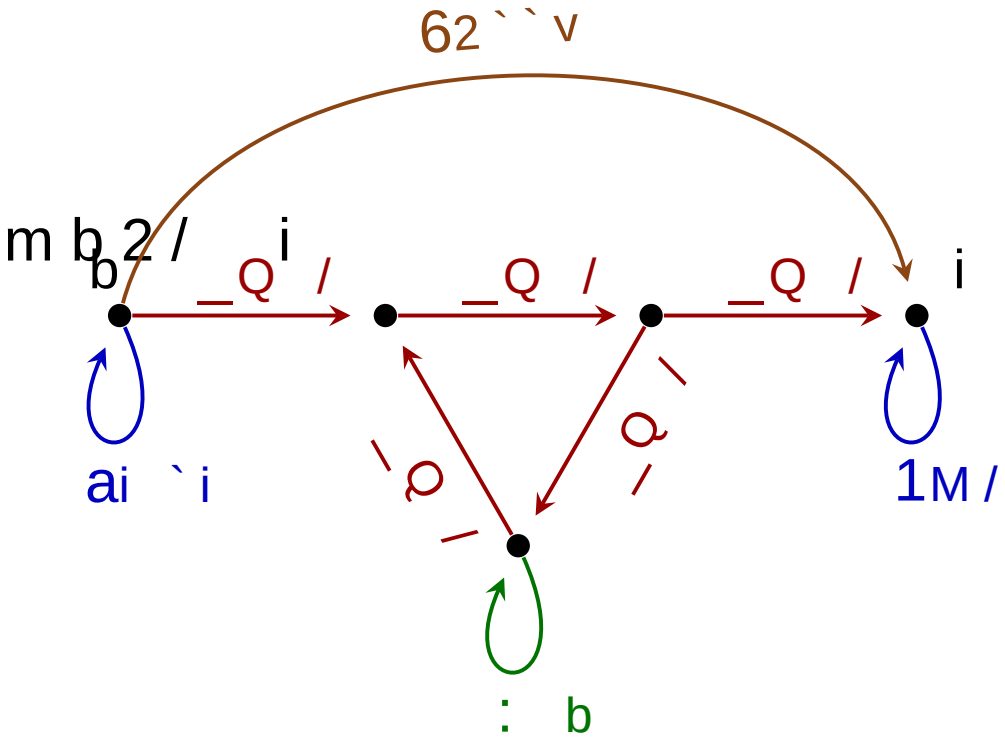
- Z_1 `2im`Mb
  - i?2 72``v
  - i?2 bi` B;?i `Q /
- q HFb rBi? +B`+mBi H Tb ) `2T2 i i?2 KB//H2 2/;2

h` BH b2K MiB+b

*vT?2` UL2Q9DV- :ZG

.2}MBiBQM

- _2im`Mb r HFb
- 1 +? 2/;2 + M #2 mb2/ i KQbi QM+2



Z_1 = a U_ Y6) 1

- Z_1 `2im`Mb
  - i?2 72``v
  - i?2 bi` B;?i `Q /
- q HFb rBi? +B`+mBi H Tb ) `2T2 i i?2 KB//H2 2/;2

Z_2 = a U_ Y6) : U_ Y6) 1

- Z_2 `2im`Mb MQ `2bmHib 55

h` BH b2K MiB+b

*vT?2` UL2Q9DV- :ZG

.2}MBiBQM

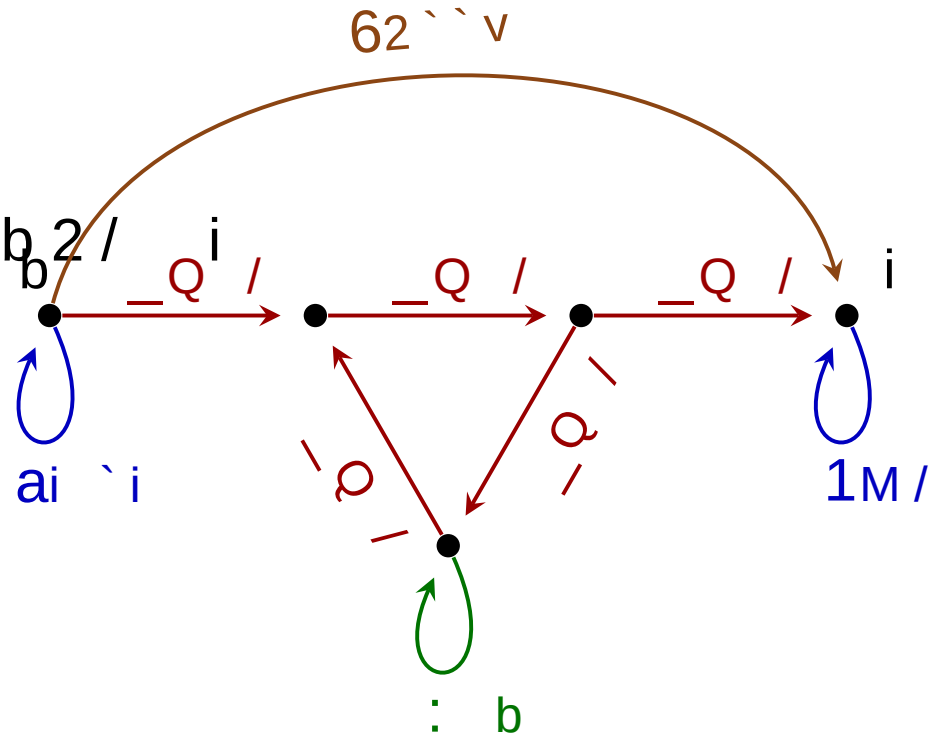
- _2im`Mb r HFb
- 1 +? 2/;2 + M #2 mb2/ i KQbi QM+2

Z_1 = a U_ Y6) 1

- Z_1 `2im`Mb
  - i?2 72``v
  - i?2 bi` B;?i `Q /
- q HFb rBi? +B`+mBi
) `2T2 i i?2 KB//H2 2

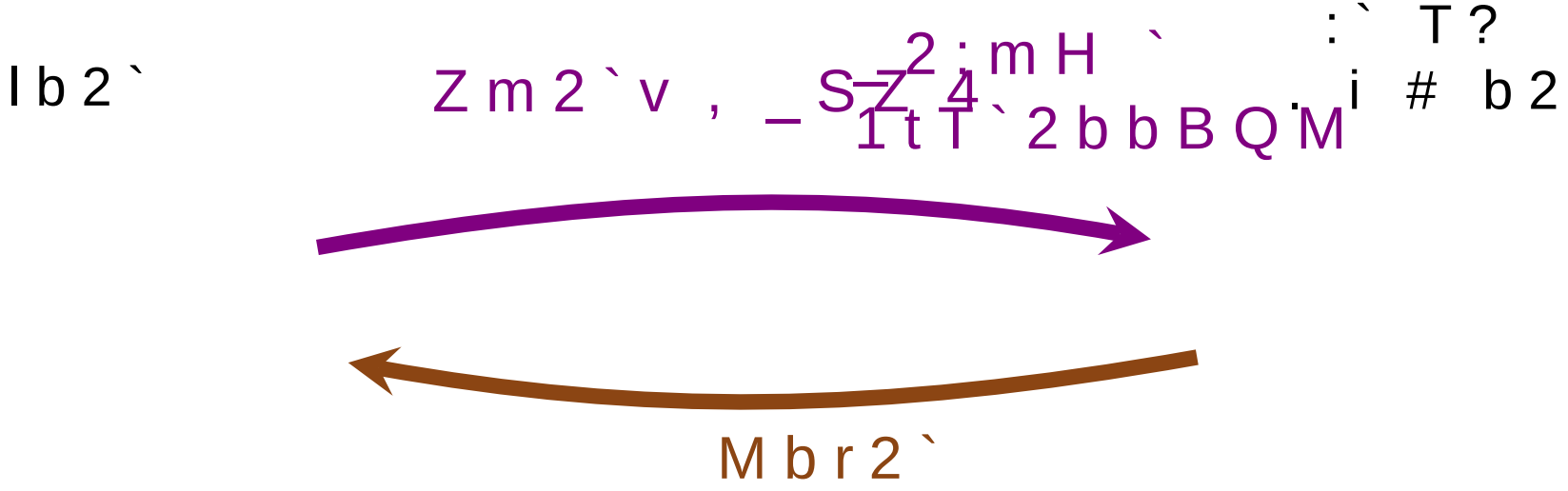
Z_2 = a U_ Y6) : U_ Y6) 1

- Z_2 `2im`Mb MQ `2bm



- 1M #H2b TQbi@T`Q+
- S`Q#H2Kb `2 mMi` +
- .Bb+ `/b K2 MBM;7m Ki+?BM; r HFb

q? i B b ;Q Q/ b 2K M i B + b 7Q ` _ S Z



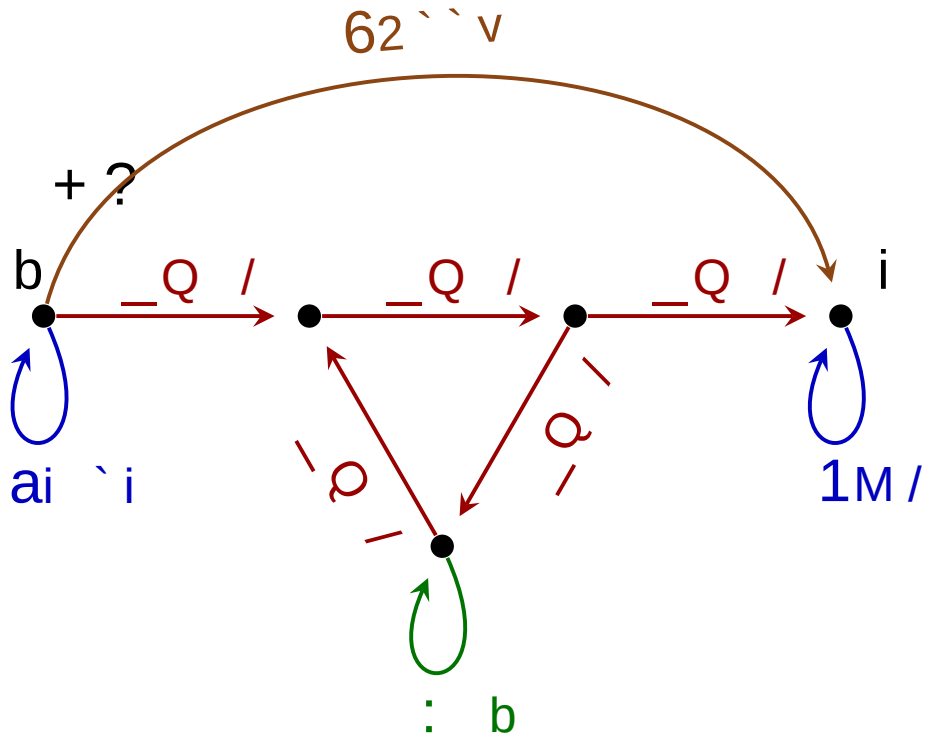
A M } M B i 2 H v K M v K i + ? 2 b # m i i ? 2 m b 2 ` 2 t T

- > Q K Q K Q ` T ? B b K i 2 ` b Q m i K Q b i B M 7 Q ` K i B Q M
- a ? Q ` i 2 b i ! @ r H F + Q p 2 ` ; 2 ô Q 7 K i + ? B M ; r H F b
- h ` B H S ` Q # H 2 K b ` 2 + Q K T m i i B Q M H H v ? ` /
  J v / B b + ` / K 2 M B M ; 7 m H K i + ? B M ; r H F

L Q b Q H m i B Q M B b + H 2 ` H v b m T 2 ` B

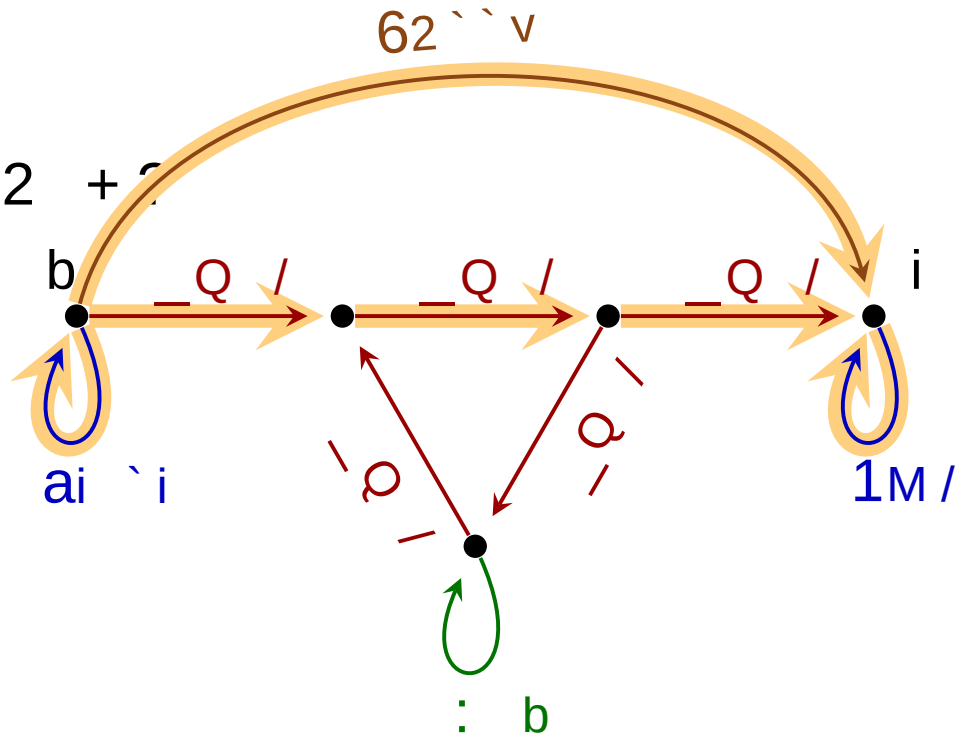
. 2 } M B i B Q M

- _ 2 i m ` M b r H F b
- 1 + ? 2 / ; 2 + M K i + ? 2 + ? i Q K Q 7 i K Q b i Q M + 2



. 2 } M B i B Q M

- _ 2 i m ` M b r H F b
- 1 + ? 2 / ; 2 + M K i + ? 2 +
  i Q K Q 7 i K Q b i Q M + 2

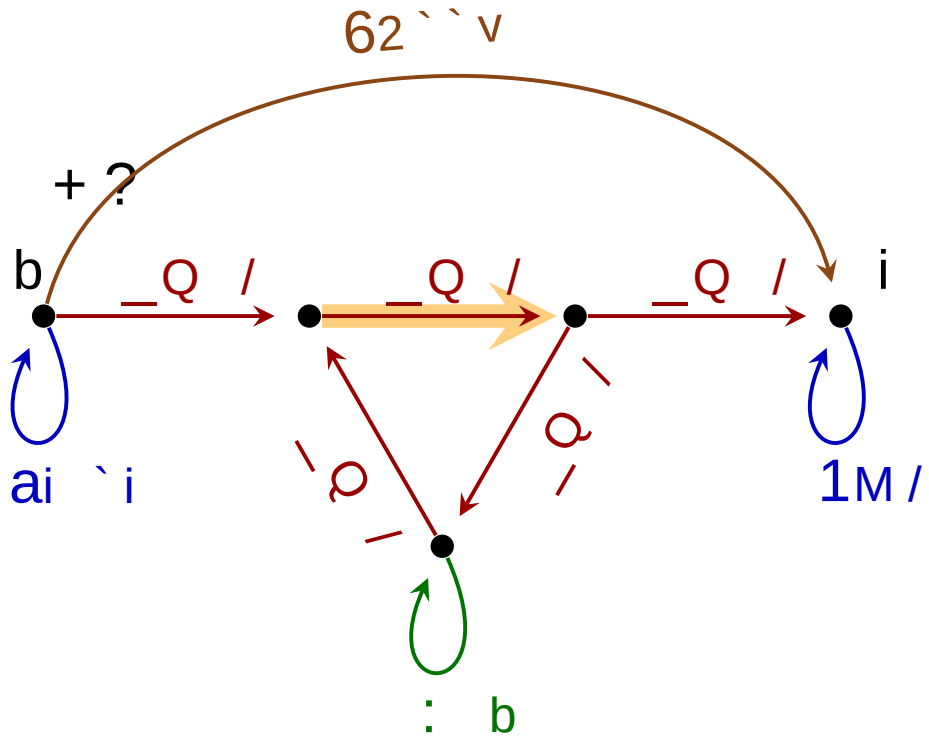


Z_1 = a U _ Y 6) 1

- _ 2 i m ` M b
  - i ? 2 7 2 ` ` v
  - i ? 2 b i ` B ; ? i ` Q /

. 2 } M B i B Q M

- _ 2 i m ` M b r H F b
- 1 + ? 2 / ; 2 + M K i + ? 2 + ? i Q K Q 7 i K Q b i Q M + 2

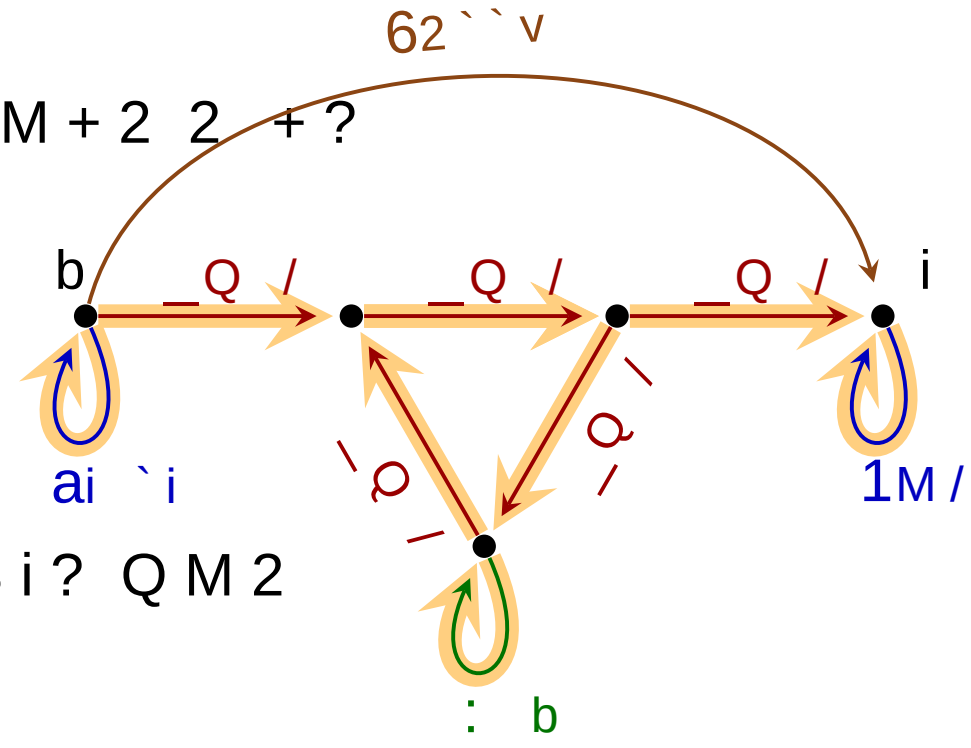


Z1 = a U _ Y 6) 1

- _ 2 i m ` M b
  - i ? 2 7 2 ` ` v
  - i ? 2 b i ` B ; ? i ` Q /
- A M r H F b r B i ? + B ` + m B i H T b 4 ) i ? 2 K B / / H 2 2 / ; 2 ` 2 m b 2 b

. 2 } M B i B Q M

- _ 2 i m ` M b r H F b
- 1 + ? 2 / ; 2 K v m b 2 Q M + 2 2 + ? i Q K B M Z

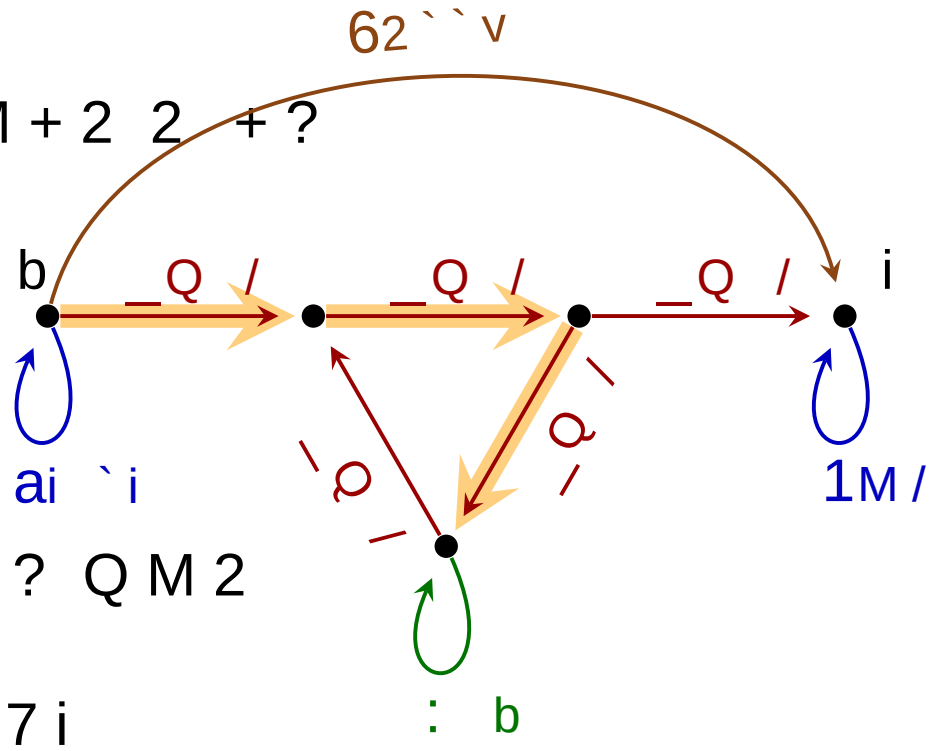


$Z_2 =$ a U _ Y 6) : U _ Y 6) 1

- _ 2 i m ` M b i ? 2 r H F r B i ? Q M 2 + B ` + m B i H T

. 2 } M B i B Q M

- _ 2 i m ` M b r H F b
- 1 + ? 2 / ; 2 K v m b 2 Q M + 2 2 + ? i Q K B M Z

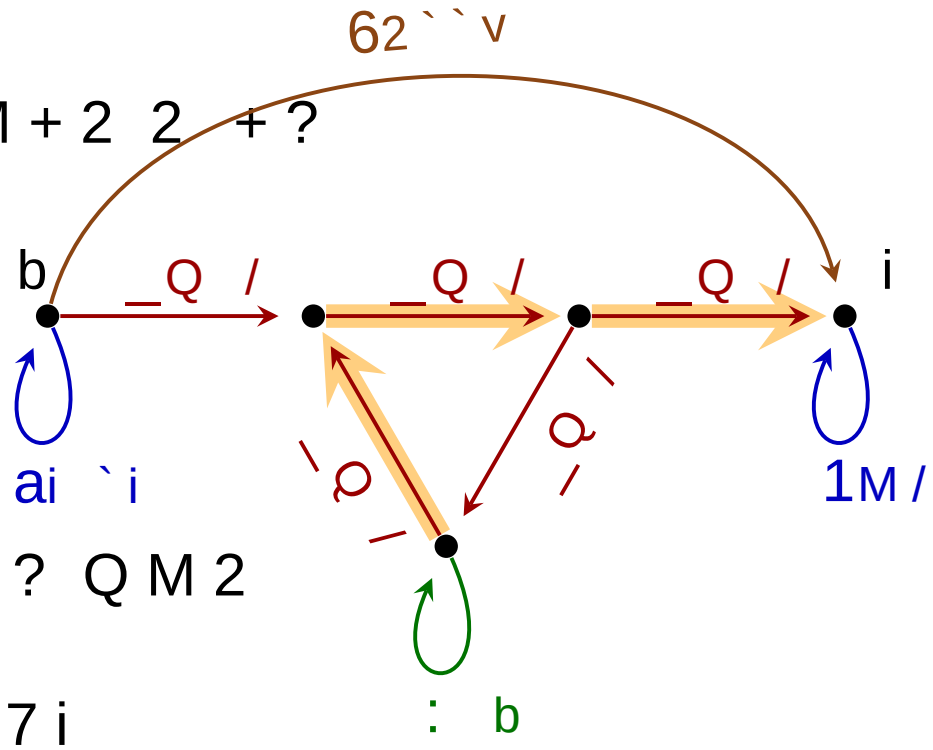


Z2 = a U _ Y 6) : U _ Y 6) 1

- _ 2 i m ` M b i ? 2 r H F r B i ? Q M 2 + B ` + m B i H T
  - " 2 7 Q : ` ! 2 m b 2 i ? 2 _ H 2 7 i

. 2 } M B i B Q M

- _ 2 i m ` M b r H F b
- 1 + ? 2 / ; 2 K v m b 2 Q M + 2 2 + ? i Q K B M Z



$Z_2 =$ a U _ Y 6) : U _ Y 6) 1

- _ 2 i m ` M b i ? 2 r H F r B i ? Q M 2 + B ` + m B i H T
  - " 2 7 Q : ` ! 2 m b 2 i ? 2 _ H 2 7 i
  - 7 i 2 : ` ! m b 2 i ? 2 ` B ; ? i

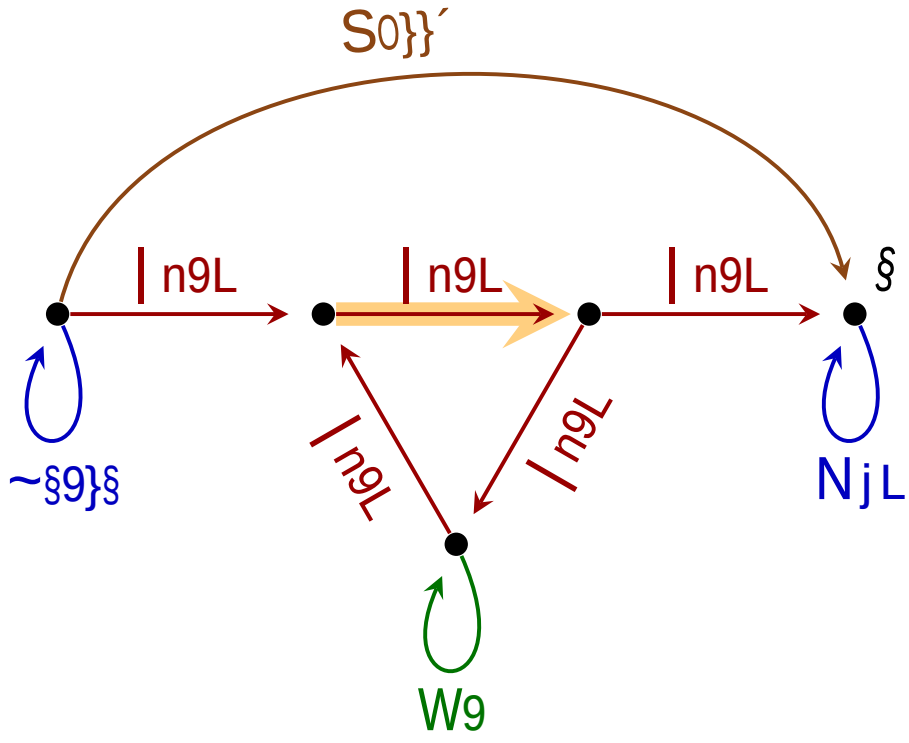
Our proposal: binding-trail semantics (2)

Definition

- Returns walks
- Each edge may use once each atom in Q

w_2 = ~ (| +S)* W (| +S)* N

- Returns the walk with one circuit lap
  - Before W ! use the left |
  - After W ! use the right |
- In walks with 2+ circuit laps
  Q) the middle edge reuses the left | or the right |



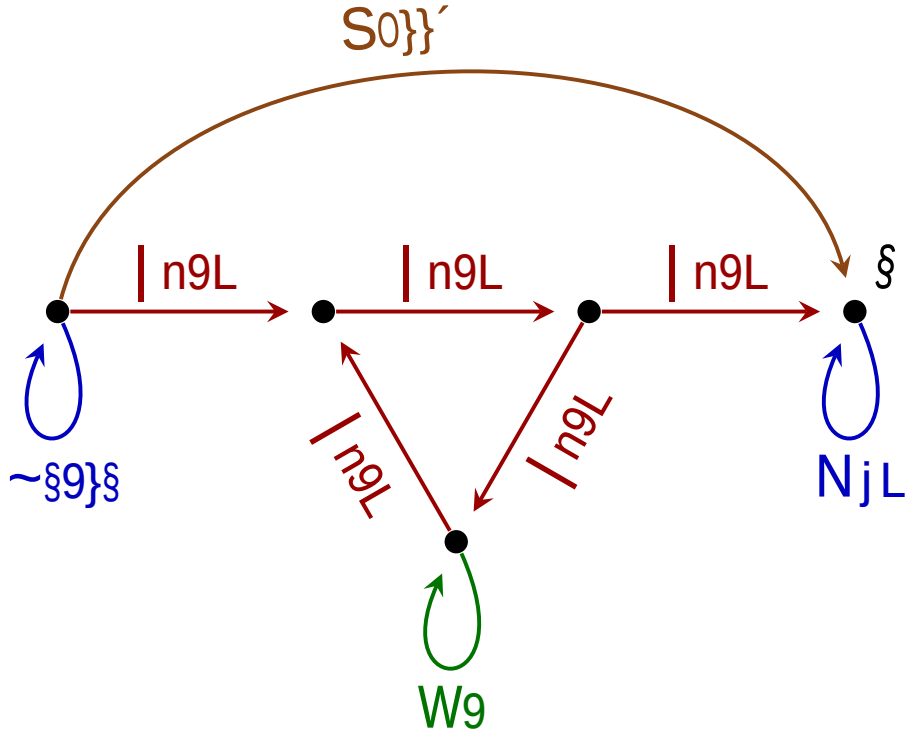
Binding-trail semantics provides “better coverage”

$w_1 =$ ~ (| +S)* N

- The ferry
- The straight road

$w_2 =$ ~ (| +S)* W (| +S)* N

- The walk with one circuit lap



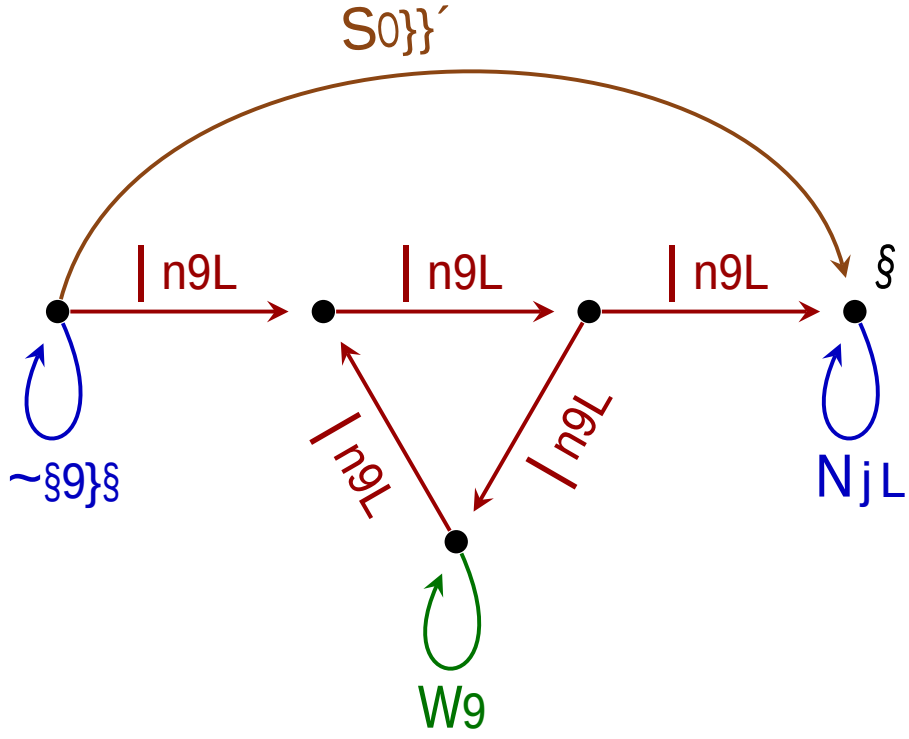
Binding-trail semantics provides “better coverage”

w_1 = ~ (| +S)* N

- The ferry
- The straight road

w_2 = ~ (| +S)* W (| +S)* N

- The walk with one circuit lap



Lemma

θ match $\circ$ of w

Q) some subwalk $\circ$ returned

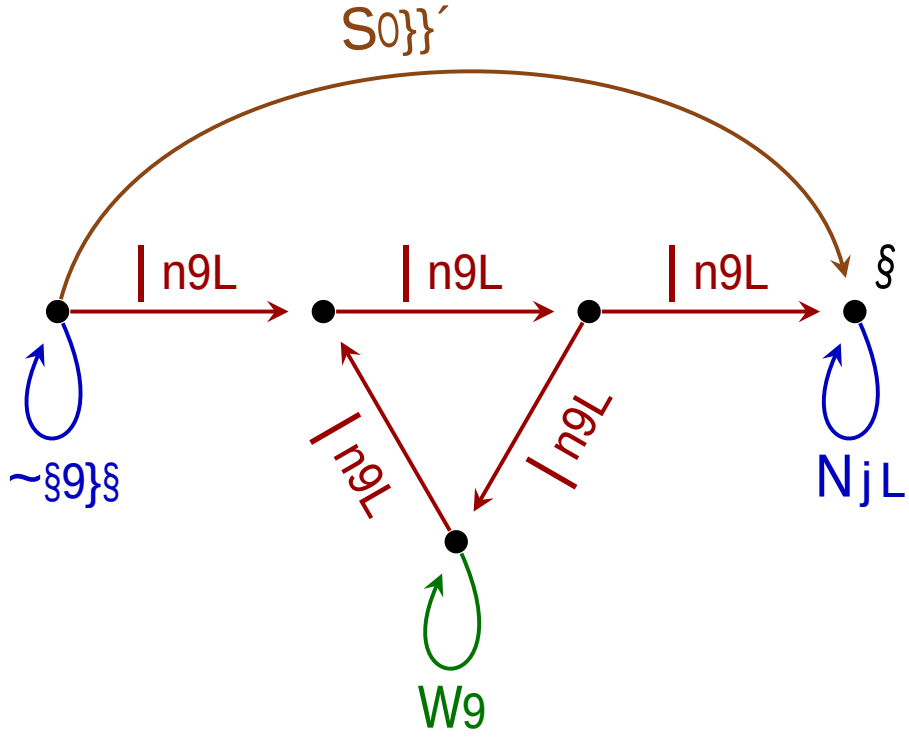
Binding-trail semantics provides “better coverage”

w_1 = ~ (| +S)* N

- The ferry
- The straight road

w_2 = ~ (| +S)* W (| +S)* N

- The walk with one circuit lap



Lemma

δ match $\circ$ of w

Q) some subwalk $\circ$ returned

Open questions

- Define “coverage”
- Define “good” coverage

Retro-compatibilities

Binding-trail is compatible with Homomorphism

Homomorphism semantics returns (; §)
( ) Binding-trail semantics returns some walk ° from to §.

Q) If users need endpoints only, use algorithm for homomorphism

Binding-trail is compatible with Shortest-walk

Shortest matching walks Binding-trail matching walks.

Q) If users need only one walk, use algorithm for shortest-walk

Binding-trail is compatible with Trail

Matching trails Binding-trail matching walks.

Computational properties of binding-trail semantics

Tractable problems

1. Nhq§_j0 is NL-complete.
2. Nj«h0}9§_jX the @9X of answers is Poly-delay.

Untractable problems

3. Gn«j§_jX the number of matched walk is l p complete.
4. g0h@0} _q of a given walk is NP-complete.

About 3 : Counting is l p complete for any reasonable semantics.
About 4 : Mostly a theoretical concern.

Open problem

5. Nj«h0}9§_jX the 0§ of answers.

The 🦣 in the room

¦\0 n«§q«§ L0q0jL nj §\0 ´j§9² nT §\0 x«0}´

| *

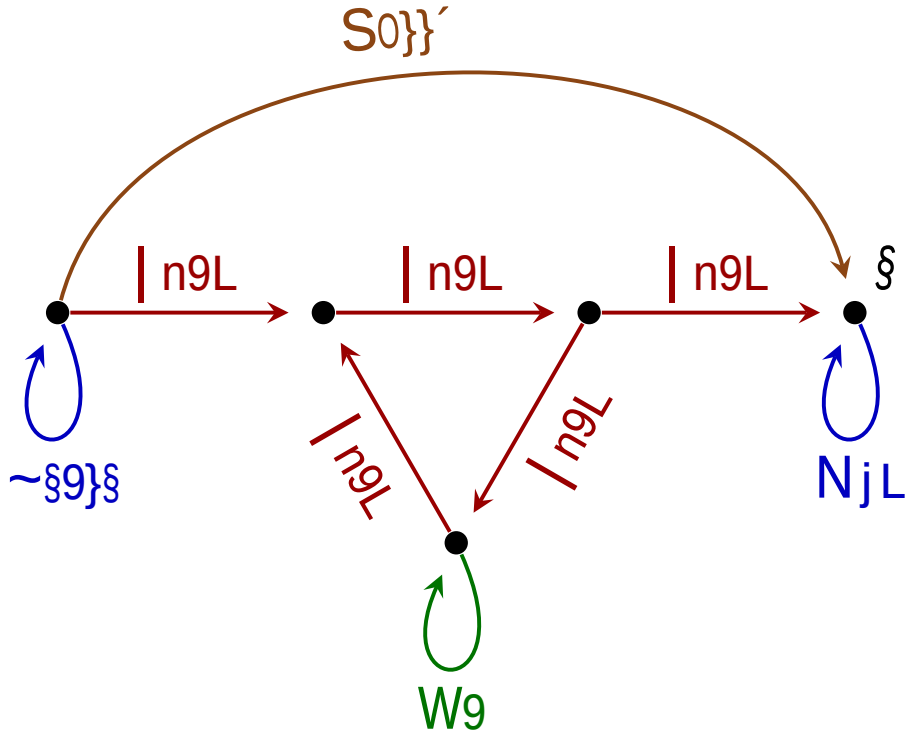
- allows no lap in the circuit

| * | *

- allows 1 lap in the circuit

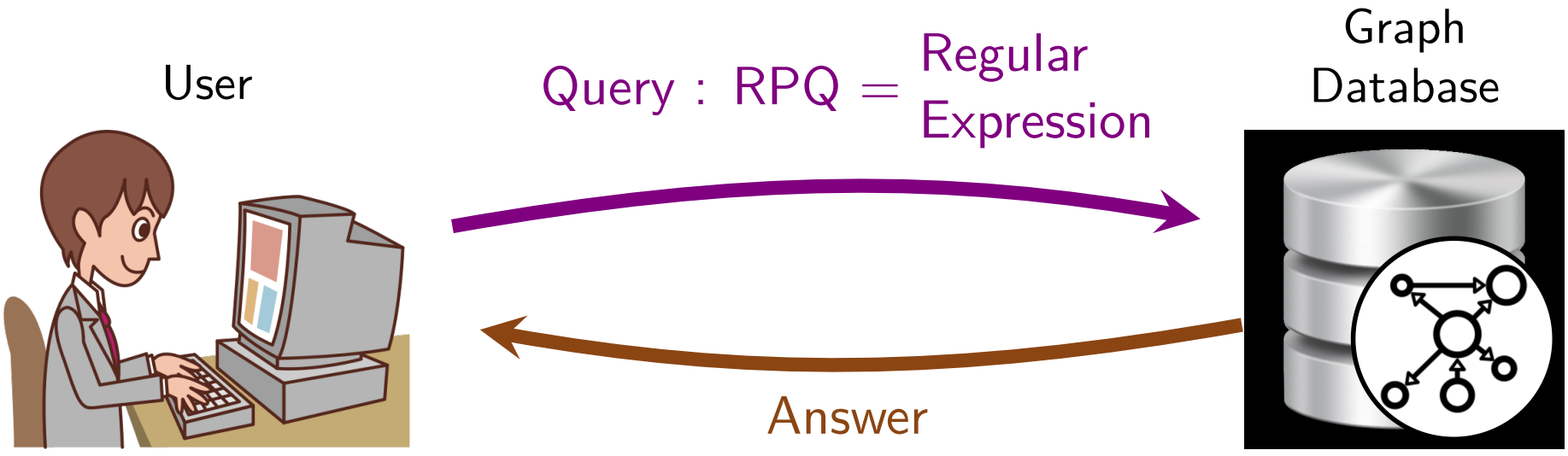
(| + |)*

- allows 1 lap in the circuit
- In general, ~~6~~ **| * | ***



Unusual from theoretical point of view 🤯
The user has finer control on the output ☺️
This kind of syntax quirks exists in practice 😵‍💫

Run-based semantics, a reasonable compromise?



- Coverage of matches
- Tractable emptiness and enumeration
- Syntax-dependent

Perspectives

- Deduplicated enumeration
- Containment
- How to deal with data values?
- Get binding-trail into GQL 🤩

How to decide if a semantics is good?

- Theoreticians mostly worry about complexity 👩‍🎓
- But complexity is not the only relevant criterion
  Why? Trail semantics is the most popular in industry 👷

Ideas for a comparison framework 🏗️

- Coverage (horizontal aggregation)
- Monotonicity (distributed databases)
- Compatibility with operators (result predictability)
- Kinds of definition (selector/restrictor in GQL)

Promising semantics 🤞

- Undominated semantics (minimal for the subwalk order)
- Shortest coverage semantics

Appendix

Comparison of semantics

Semantics	Shortest-walk	Trail	Run-based
Existence	■ Tractable	■ Untractable	■ Tractable
Enumeration	■ Tractable	■ Untractable	■ Tractable
Distinct Enum	■ Tractable	■ Untractable	Open
Counting	■ Meaningless	■ Untractable	■ Untractable
Walk Memb.	■ Tractable	■ Tractable	■ Untractable
Coverage	■ None	■ No guarantee	■ "Subwalk" guarantee

Tools and technics

Simple run

- Query given as an automaton
- Outputs simple walks in the product $K \quad A_Q$
- Good formal setting for theory

Binding trail

- Query given as a RegExp
- Outputs matching walks that do not reuse edge on a same atom
- Closer to practice

Theorem

The two semantics are computationally equivalent.

Key idea for $=)$

From an automaton A, we build a regular expression N such that runs of A are encoded into runs of the Glushkov automaton of N.