

Run-based semantics for RPQs

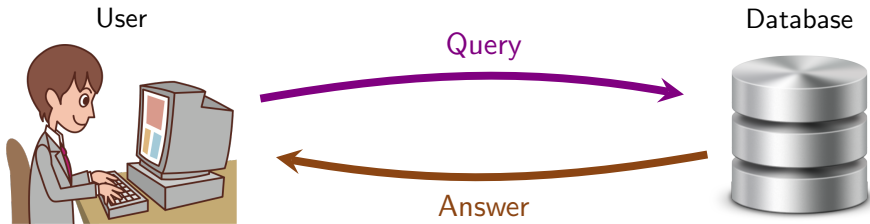
Victor MARSAULT*

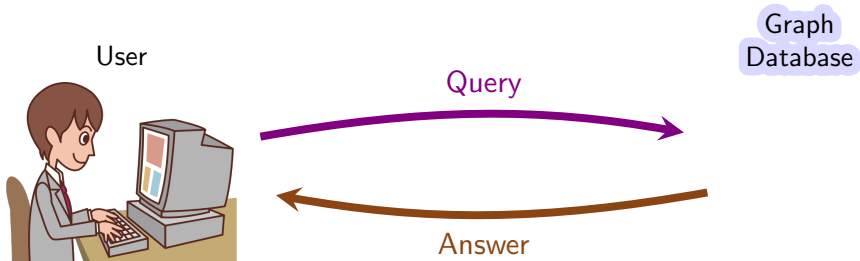
joint work with Claire DAVID* and Nadime FRANCIS*

* Université Gustave-Eiffel, CNRS, LIGM

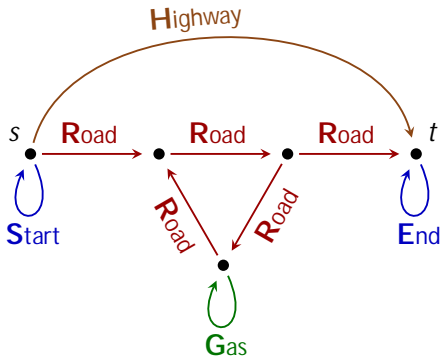
Highlights'22

2022-06-29

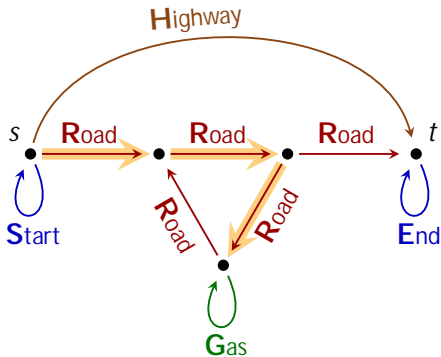




- Finite label alphabet:
 $\Sigma = fS;R;H;G;Eg$
- Vertices
- Edges labelled over Σ



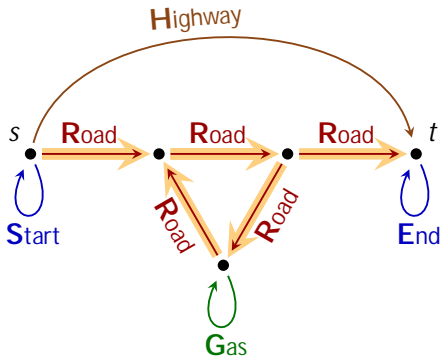
- Finite label alphabet:
 $\Sigma = fS;R;H;G;Eg$
- Vertices
- Edges labelled over Σ



Terminology: Walk

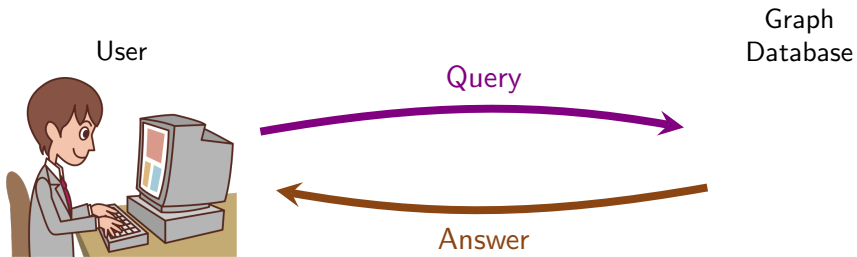
- a.k.a. Path
- Consistent sequence of edges

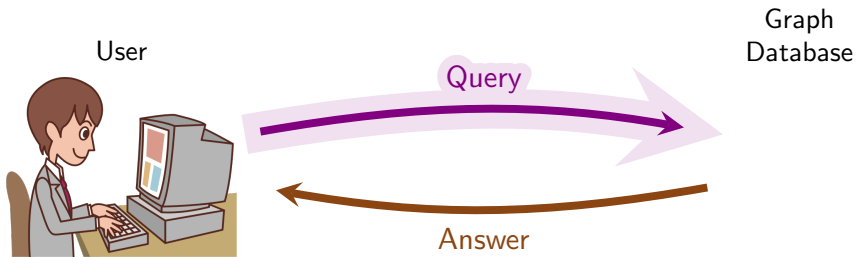
- Finite label alphabet:
 $\Sigma = fS;R;H;G;Eg$
- Vertices
- Edges labelled over Σ



Terminology: Walk

- a.k.a. Path
- Consistent sequence of edges





User



Query : RPQ = Regular Expression



Graph Database

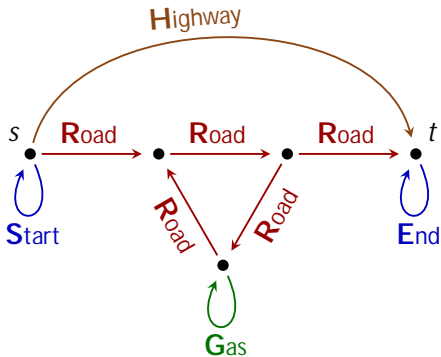


Answer

"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

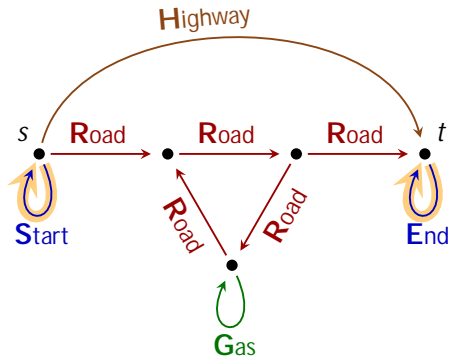
Which walks match Q_1 ?



"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

Which walks match Q_1 ?

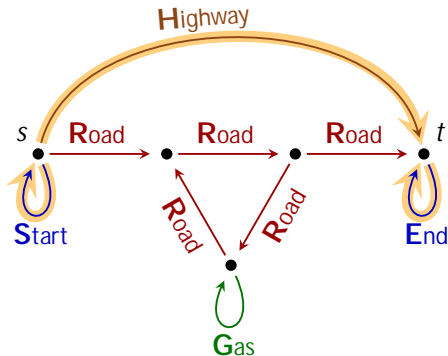


"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

Which walks match Q_1 ?

- The highway

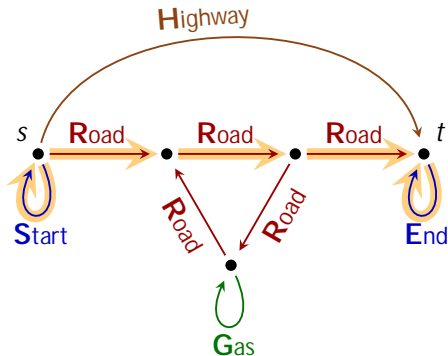


"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

Which walks match Q_1 ?

- The highway
- The straight road

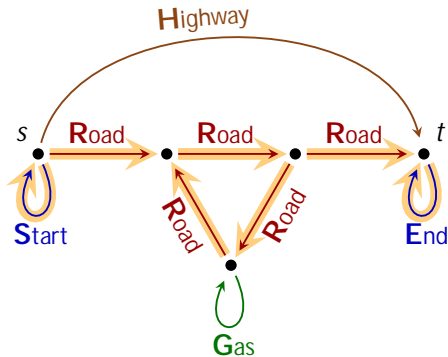


"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

Which walks match Q_1 ?

- The highway
- The straight road
- Road with laps in the circuit



"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

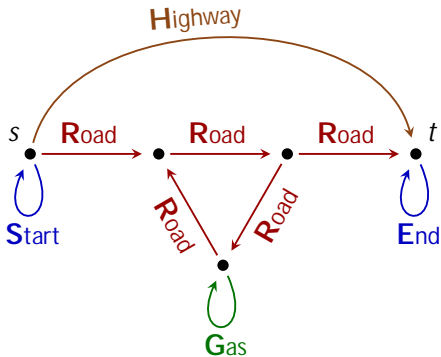
Which walks match Q_1 ?

- The highway
- The straight road
- Road with laps in the circuit

"...with mandatory gas stop"

$$Q_2 = S (R+H)^* G (R+H)^* E$$

Which walks match Q_2 ?



"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

Which walks match Q_1 ?

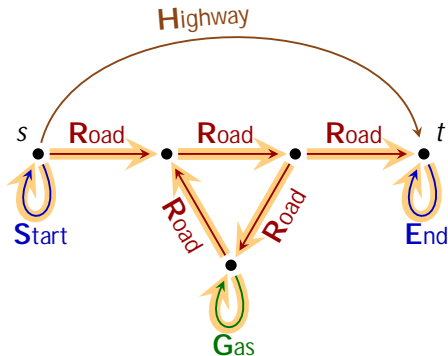
- The highway
- The straight road
- Road with laps in the circuit

"...with mandatory gas stop"

$$Q_2 = S (R+H)^* G (R+H)^* E$$

Which walks match Q_2 ?

- Road with laps in the circuit



"Find a way from s to t"

$$Q_1 = S (R+H)^* E$$

Which walks match Q_1 ?

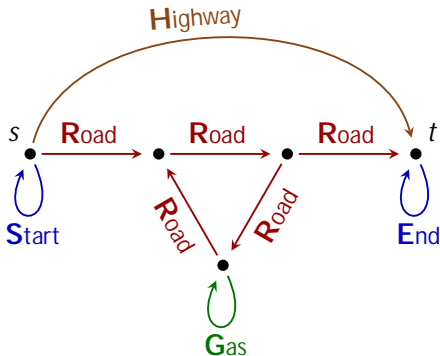
- The highway
- The straight road
- Road with laps in the circuit

"...with mandatory gas stop"

$$Q_2 = S (R+H)^* G (R+H)^* E$$

Which walks match Q_2 ?

- Road with laps in the circuit



) Infinitely many matches

User



Query : RPQ = Regular Expression



Graph Database

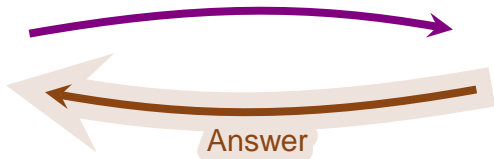


Answer

User

Query : RPQ = Regular
Expression

Graph
Database



User

Query : RPQ = Regular Expression

Graph Database



Answer

Several way to ensure niteness



User

Query : RPQ = Regular Expression

Graph Database



Answer

Several way to ensure niteness

In theory

Homomorphism semantics

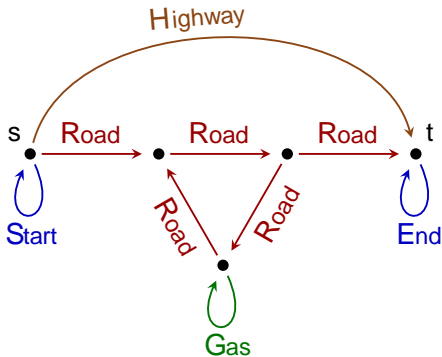
In practice

Trail semantics

... among others

Definition

- Returns the endpoints of matches



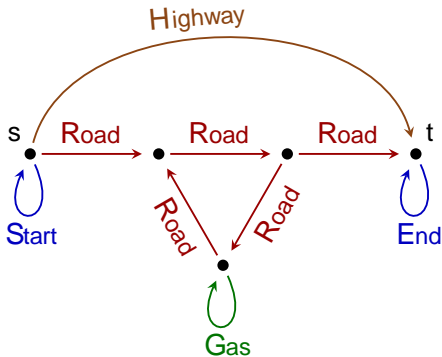
Definition

- Returns the endpoints of matches

$$Q_1 = S (R+H) E$$

$$Q_2 = S (R+H) G (R+H) E$$

- Q_1 and Q_2 return $f(s; t)g$



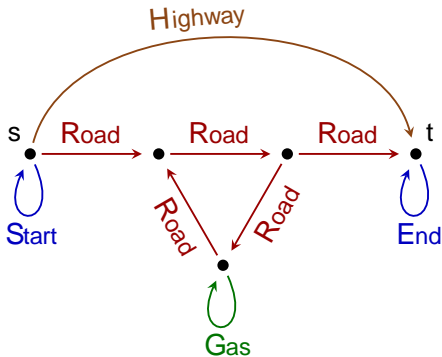
Definition

- Returns the endpoints of matches

$Q_1 = S (R+H) E$

$Q_2 = S (R+H) G (R+H) E$

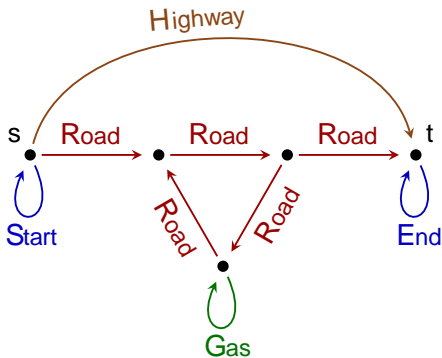
- Q_1 and Q_2 return $f(s; t)g$



- Return partial information (Boolean for Q_1 and Q_2)
- Cannot count

Definition

- Return walks
- Each edge may be used once

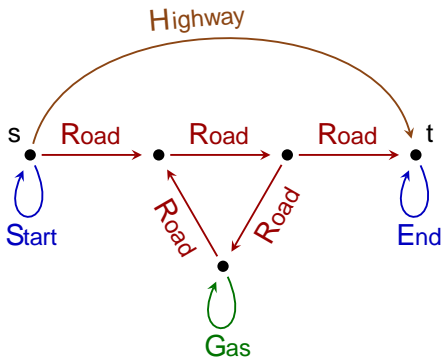


Definition

- Return walks
- Each edge may be used once

$$Q_1 = S(R+H)E$$

- Q_1 returns 2 walks
 - the highway
 - the straight road

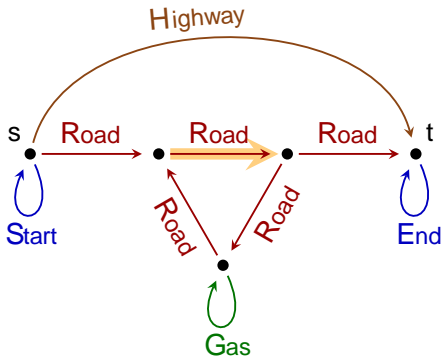


Definition

- Return walks
- Each edge may be used once

$$Q_1 = S(R+H)E$$

- Q_1 returns 2 walks
 - the highway
 - the straight road
- Any road with circuit laps
) repeated edges



Definition

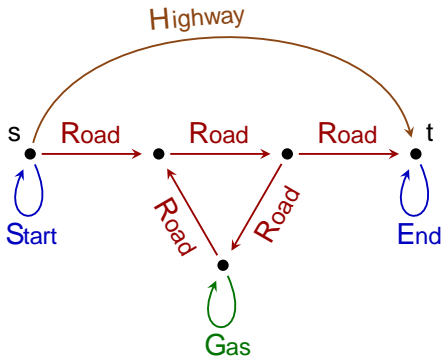
- Return walks
- Each edge may be used once

$$Q_1 = S(R+H)E$$

- Q_1 returns 2 walks
 - the highway
 - the straight road
- Any road with circuit laps
) repeated edges

$$Q_2 = S(R+H)G(R+H)E$$

- Q_2 returns no results



Definition

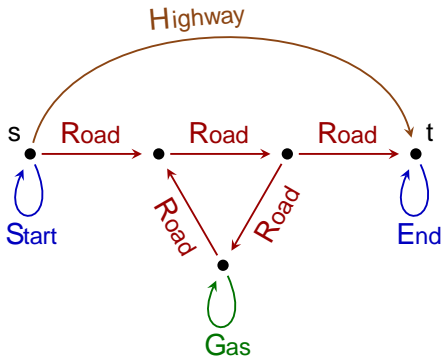
- Return walks
- Each edge may be used once

$$Q_1 = S(R+H)E$$

- Q_1 returns 2 walks
 - the highway
 - the straight road
- Any road with circuit laps
) repeated edges

$$Q_2 = S(R+H)G(R+H)E$$

- Q_2 returns no results



- Problems are untractable

User

Query : RPQ = Regular
Expression

Graph
Database



Answer

Several way to ensure niteness

In theory

Homomorphism semantics

- Return endpoints
- Tractable
- No counting

In practice

Trail semantics

- Returns walks
- Untractable
- Counting

... among others

User

Query : RPQ = Regular Expression

Graph Database



Answer

Several way to ensure niteness

In theory

Homomorphism semantics

- Return endpoints
- Tractable
- No counting

Our proposal

Run-based semantics

In practice

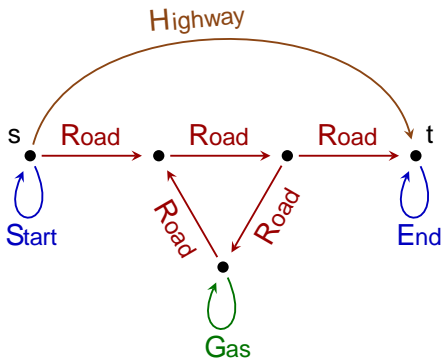
Trail semantics

- Returns walks
- Untractable
- Counting

... among others

Definition

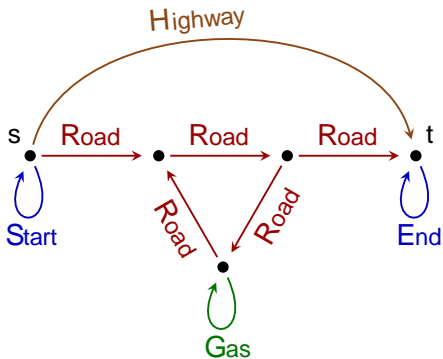
- Returns walks
- Each edge may use once each atom in Q



Definition

- Returns walks
- Each edge may be used once each atom in Q

$$Q_2 = S (R+H) G (R+H) E$$

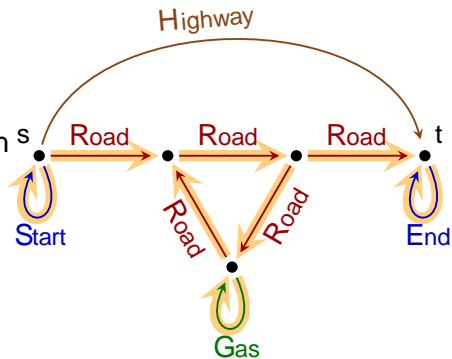


Definition

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

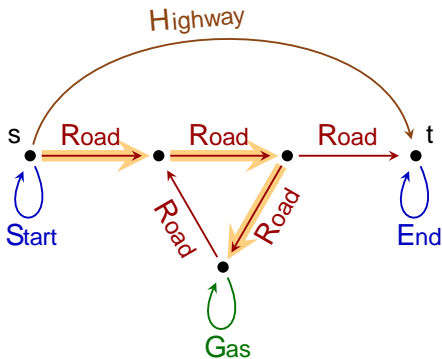
$$Q_2 = S (R+H) G (R+H) E$$

- Returns the 1-lap road only



Definition

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

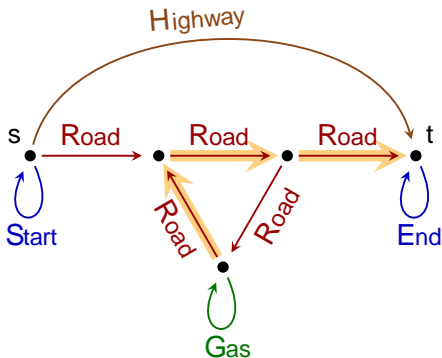


$$Q_2 = S (R+H) G (R+H) E$$

- Returns the 1-lap road only
 - Before G ! use the left R

Definition

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

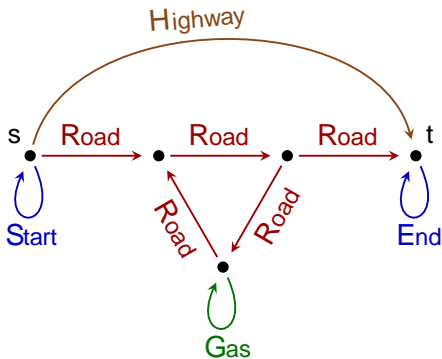


$$Q_2 = S (R+H) G (R+H) E$$

- Returns the 1-lap road only
 - Before $G!$ use the left R
 - After $G!$ use the right R

Definition

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



$$Q_2 = \text{S (R+H) G (R+H) E}$$

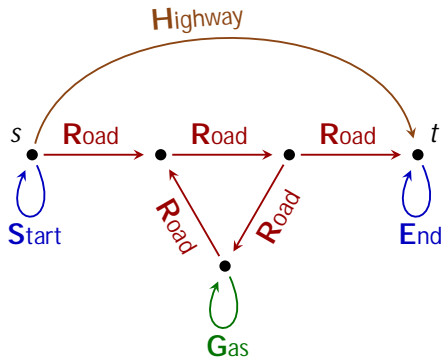
- Returns the 1-lap road only
 - Before **G** ! use the left **R**
 - After **G** ! use the right **R**
- > 1 circuit lap) some edge use the same atom twice

Definition

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

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

- Returns the 1-lap road only
 - Before \mathbf{G} / use the left \mathbf{R}
 - After \mathbf{G} / use the right \mathbf{R}
- > 1 circuit lap) some edge use the same atom twice



Property

W.r.t. emptiness:

Homomorphism semantics

, Binding-trail semantics

User



Query : RPQ = Regular Expression

Graph Database



Answer

Several way to ensure finiteness

In theory

Homomorphism semantics

- Return endpoints
- Tractable
- No counting

Our proposal

Run-based semantics

- Returns walks
- Emptiness tractable
- Enumeration tractable
- Counting untractable

In practice

Trail semantics

- Returns walks
- Untractable
- Counting

... among others