

Fauré: a Partial Approach to Network Analysis

Fangping Lan, Bin Gui, and Anduo Wang
Temple University

network analysis — significant progress

- [NSDI 20] Tiramisu: Fast Multilayer Network Verification.
- [SIGCOMM'02] Route oscillations in I-BGP route reflection.
- [HotNets'20] Solver-Aided Multi-Party Configuration.
- [NSDI'15] General Approach to Network Configuration Analysis.
- [SIGCOMM'16] Fast Control Plane Analysis Using an Abstract Representation.
- [TON'02] The Stable Paths Problem and Interdomain Routing.
- [SIGCOMM'19] Validating Datacenters at Scale.
- [CoNEXT 20] AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks.
- [NSDI 13] Real Time Network Policy Checking Using Header Space Analysis
- [HotSDN 12] VeriFlow: Verifying Network-Wide Invariants in Real Time
- [NSDI 15] Checking Beliefs in Dynamic Networks.
- [POPL 16] Scaling Network Verification Using Symmetry and Surgery
- [NSDI 20] Plankton: Scalable network configuration verification through model checking
- [IEEE Networks 05] Modeling the routing of an autonomous system with C-BGP.
- [INFOCOM 18] Polynomial-Time What-If Analysis for Prefix-Manipulating MPLS Networks
- [SIGCOMM 19] Safely and Automatically Updating In-Network ACL Configurations with Intent Language.
- [INFOCOM 05] On static reachability analysis of IP networks
- [SIGCOMM 20] Accuracy, Scalability, Coverage: A Practical Configuration Verifier on a Global WAN
- [HotNets 20] Incremental Network Configuration Verification
- [NSDI 20] APKeep: Realtime Verification for Real Networks

with

... ...

network analysis — significant progress

[NSDI 20] Tiramisu: Fast Multilayer Network Verification.

[SIGCOMM'02] Route oscillations in I-BGP route reflection.

[HotNets'20] Solver-Aided Multi-Party Configuration.

[NSDI'15] General Approach to Network Configuration Analysis.

[SIGCOMM'16] Fast Control Plane Analysis Using an Abstract Representation.

[TON'02] The Stable Paths Problem and Interdomain Routing.

[SIGCOMM'19] Validating Datacenters at Scale.

[CoNEXT 20] AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks.

[NSDI 13] Real Time Network Policy Checking Using Header Space Analysis

[HotSDN 12] VeriFlow: Verifying Network-Wide Invariants in Real Time

[NSDI 15] Checking Beliefs in Dynamic Networks.

[POPL 16] Scaling Network Verification Using Symmetry and Surgery

[NSDI 20] Plankton: Scalable network configuration verification through model checking

[IEEE Networks 05] Modeling the routing of an autonomous system with C-BGP.

[INFOCOM 18] Polynomial-Time What-If Analysis for Prefix-Manipulating MPLS Networks

[SIGCOMM 19] Safely and Automatically Updating In-Network ACL Configurations with Intent Language.

[INFOCOM 05] On static reachability analysis of IP networks

[SIGCOMM 20] Accuracy, Scalability, Coverage: A Practical Configuration Verifier on a Global WAN

[HotNets 20] Incremental Network Configuration Verification

[NSDI 20] APKeep: Realtime Verification for Real Networks

....

with

The screenshot shows a Microsoft Research website page titled "Network Verification". The page features a large background image of a server rack in a datacenter. At the top, there's a navigation bar with links to Microsoft, Research, Our research, Programs & events, Blogs & podcasts, and About. Below the navigation, the title "Network Verification" is prominently displayed. Underneath the title, there are two tabs: "Overview" (which is underlined) and "Publications". A descriptive paragraph explains the goal of reliable, efficient network operation without user notice, mentioning tools for datacenter networks.

network analysis — significant progress

[NSDI 20] Tiramisu: Fast Multilayer Network Verification.

[SIGCOMM'02] Route oscillations in I-BGP route reflection.

[HotNets'20] Solver-Aided Multi-Party Configuration.

[NSDI'15] General Approach to Network Configuration Analysis.

[SIGCOMM'16] Fast Control Plane Analysis Using an Abstract Representation.

[TON'02] The Stable Paths Problem and Interdomain Routing.

[SIGCOMM'19] Validating Datacenters at Scale.

[CoNEXT 20] AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks.

[NSDI 13] Real Time Network Policy Checking Using Header Space Analysis

[HotSDN 12] VeriFlow: Verifying Network-Wide Invariants in Real Time

[NSDI 15] Checking Beliefs in Dynamic Networks.

[POPL 16] Scaling Network Verification Using Symmetry and Surgery

[NSDI 20] Plankton: Scalable network configuration verification through model checking

[IEEE Networks 05] Modeling the routing of an autonomous system with C-BGP.

[INFOCOM 18] Polynomial-Time What-If Analysis for Prefix-Manipulating MPLS Networks

[SIGCOMM 19] Safely and Automatically Updating In-Network ACL Configurations with Intent Language.

[INFOCOM 05] On static reachability analysis of IP networks

[SIGCOMM 20] Accuracy, Scalability, Coverage: A Practical Configuration Verifier on a Global WAN

[HotNets 20] Incremental Network Configuration Verification

[NSDI 20] APKeep: Realtime Verification for Real Networks

....

with

The screenshot shows a Microsoft Research page titled "Network Verification". The page features a background image of server racks in a datacenter. The title "Network Verification" is prominently displayed in the center. Below the title, there is a section for the "Half-Day Tutorial: Network Verification" scheduled for "Monday 17th August, Afternoon Session". It lists presenters: George Varghese and Nikolaj Bjorner from Microsoft Research. The "Tutorial location" is mentioned as being in the "SIGCOMM 2015" conference area. A blue link "Conference click to expand contents" is visible, and a navigation menu includes "Home" and "Program".

network analysis — significant progress

[NSDI 20] Tiramisu: Fast Multilayer Network Verification.

[SIGCOMM'02] Route oscillations in I-BGP route reflection.

[HotNets'20] Solver-Aided Multi-Party Configuration.

[NSDI'15] General Approach to Network Configuration Analysis.

[SIGCOMM'16] Fast Control Plane Analysis Using an Abstract Representation.

[TON'02] The Stable Paths Problem and Interdomain Routing.

[SIGCOMM'19] Validating Datacenters at Scale.

[CoNEXT 20] AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks.

[NSDI 13] Real Time Network Policy Checking Using Header Space Analysis

[HotSDN 12] VeriFlow: Verifying Network-Wide Invariants in Real Time

[NSDI 15] Checking Beliefs in Dynamic Networks.

[POPL 16] Scaling Network Verification Using Symmetry and Surgery

[NSDI 20] Plankton: Scalable network configuration verification through model checking

[IEEE Networks 05] Modeling the routing of an autonomous system with C-BGP.

[INFOCOM 18] Polynomial-Time What-If Analysis for Prefix-Manipulating MPLS Networks

[SIGCOMM 19] Safely and Automatically Updating In-Network ACL Configurations with Intent Language.

[INFOCOM 05] On static reachability analysis of IP networks

[SIGCOMM 20] Accuracy, Scalability, Coverage: A Practical Configuration Verifier on a Global WAN

[HotNets 20] Incremental Network Configuration Verification

[NSDI 20] APKeep: Realtime Verification for Real Networks

with

05 // 12 // 17

NETWORK VERIFICATION WITH VERIFLOW

Author
PETER WELCHER
Architect, Operations Technical Advisor

Did my Networking Field Day (#NFD13) [blog](#) about [Forward Networks](#) (@FwdNetworks) catch your eye? Do you need to verify that changes won't break your network? Or that there are no loopholes in your security?

Forward Networks defined a somewhat new network and security tool category with its announcement at #NFD13. Specifically, "Network Assurance" — making sure your network and changes will meet your specifications of correct behavior. See the [blog](#) I wrote for more about Forward Networks.

Network Verification

SIGCOMM 2015

- Home
- Program

Conference click to expand contents

Half-Day Tutorial: Network Verification

Monday 17th August, Afternoon Session

Presenters

George Varghese, Microsoft Research
Nikolaj Bjorner, Microsoft Research

Tutorial location

network analysis — significant progress

[NSDI 20] Tiramisu: Fast Multilayer Network Verification.

[SIGCOMM'02] Route oscillations in I-BGP route reflection.

[HotNets'20] Solver-Aided Multi-Party Configuration.

[NSDI'15] General Approach to Network Configuration Analysis.

[SIGCOMM'16] Fast Control Plane Analysis Using an Abstract Representation.

[TON'02] The Stable Paths Problem and Interdomain Routing.

[SIGCOMM'19] Validating Datacenters at Scale.

[CoNEXT 20] AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks.

[NSDI 13] Real Time Network Policy Checking Using Header Space Analysis

[HotSDN 12] VeriFlow: Verifying Network-Wide Invariants in Real Time

[NSDI 15] Checking Beliefs in Dynamic Networks.

[POPL 16] Scaling Network Verification Using Symmetry and Surgery

[NSDI 20] Plankton: Scalable network configuration verification through model checking

[IEEE Networks 05] Modeling the routing of an autonomous system with C-BGP.

[INFOCOM 18] Polynomial-Time What-If Analysis for Prefix-Manipulating MPLS Networks

[SIGCOMM 19] Safely and Automatically Updating In-Network ACL Configurations with Intent Language.

[INFOCOM 05] On static reachability analysis of IP networks

[SIGCOMM 20] Accuracy, Scalability, Coverage: A Practical Configuration Verifier on a Global WAN

[HotNets 20] Incremental Network Configuration Verification

[NSDI 20] APKeep: Realtime Verification for Real Networks

... ...

with

The collage includes:

- A Microsoft blog post titled "NETWORK VERIFICATION WITH VERIFLOW" by Peter Welcher, dated 05/12/17. It discusses Forward Networks and their announcement at NFD13.
- A "Network Verification" section from a SIGCOMM 2015 page.
- A "Half-Day Tutorial: Network Verification" section for Monday 17th August, Afternoon Session, with a "Presenters" link.
- An "ACM SIGCOMM 2021 TUTORIAL: Introduction to Network Verification" page.
- A "Tutorial Program" section with a link to the "Go to Tutorial Slack channel".
- A search bar labeled "Filter items...".
- A red banner at the bottom indicating the session is from 1:00 pm - 2:15 pm.
- A sidebar titled "RELATED TOPICS" with links to various network-related blogs.

network analysis

*enterprise,
private WANs,
inter-domain...*

network

query

*reachability,
multi-path consistency,
convergence...*

analyzer

answer

*guarantee,
bug, ...*

network analysis

*enterprise,
private WANs,
inter-domain...*

network

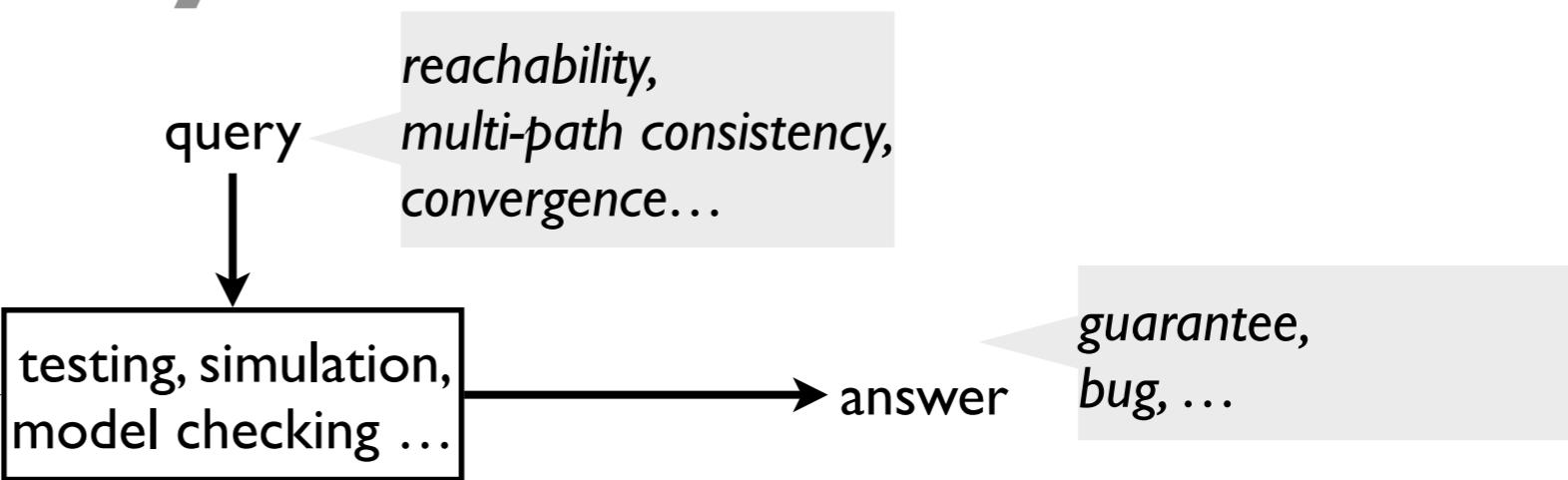
query

*reachability,
multi-path consistency,
convergence...*

testing, simulation,
model checking ...

answer

*guarantee,
bug, ...*



complete network analysis

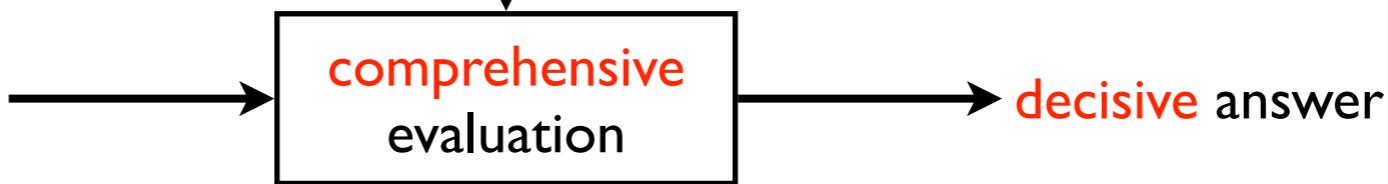
classical formal analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer



challenges with complete analysis

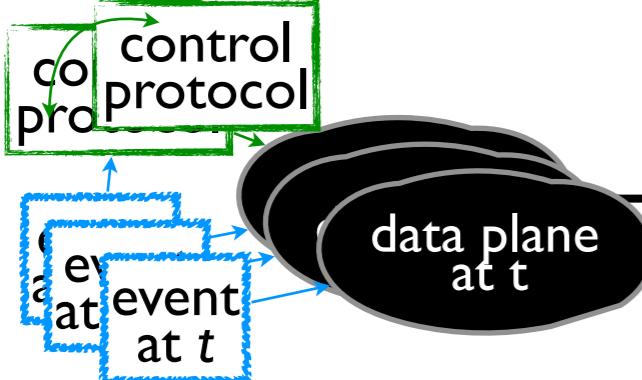
classical formal analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer



query

comprehensive
evaluation

decisive answer

challenges with complete analysis

classical formal analysis

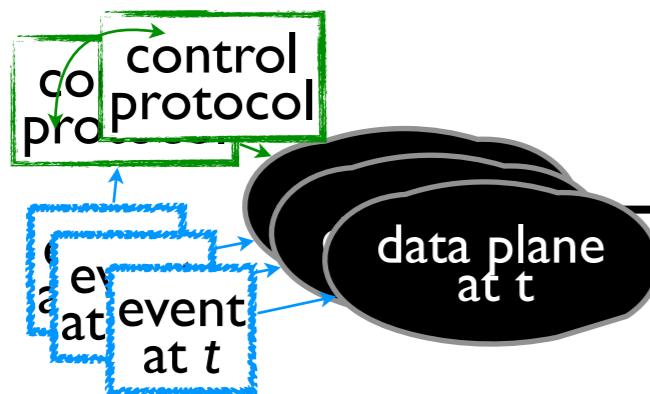
definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

uncertain environment



query

repeated analysis?

decisive answer

challenges with complete analysis

classical formal analysis

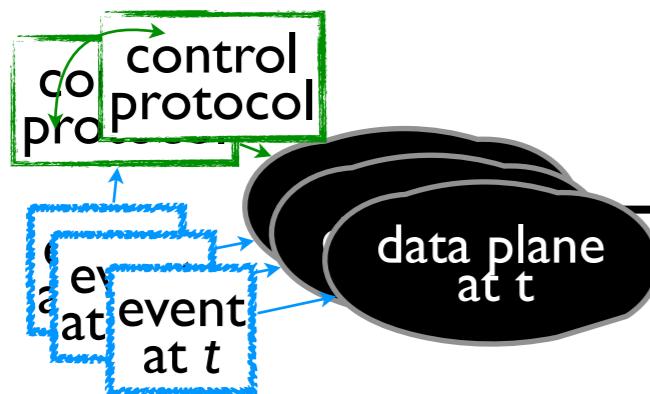
definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

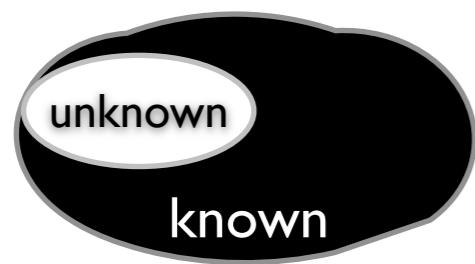
uncertain environment



query

repeated analysis?

decisive answer



query

comprehensive
evaluation

decisive answer

challenges with complete analysis

classical formal analysis

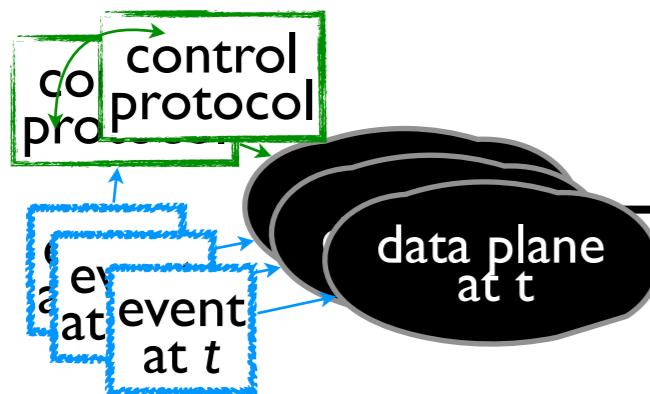
definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

uncertain environment

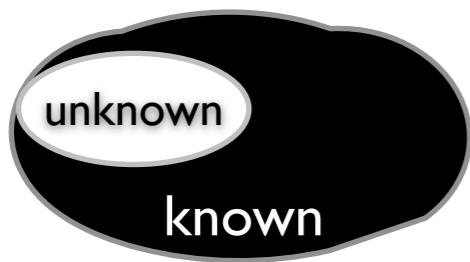


query

repeated analysis?

decisive answer

unknown information



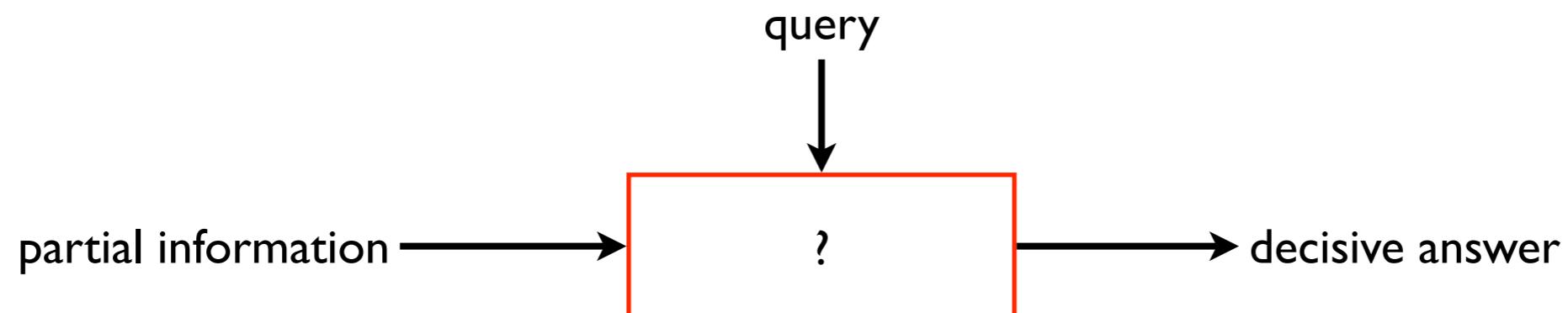
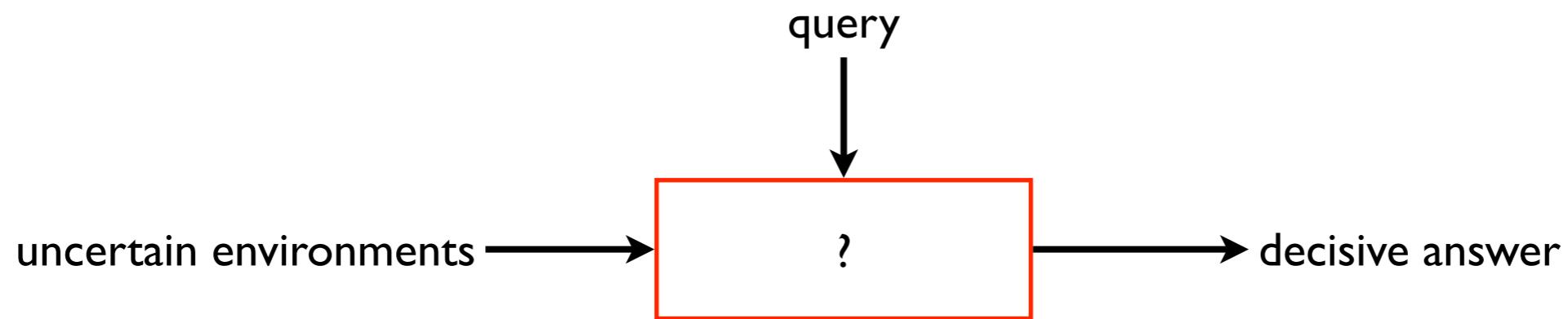
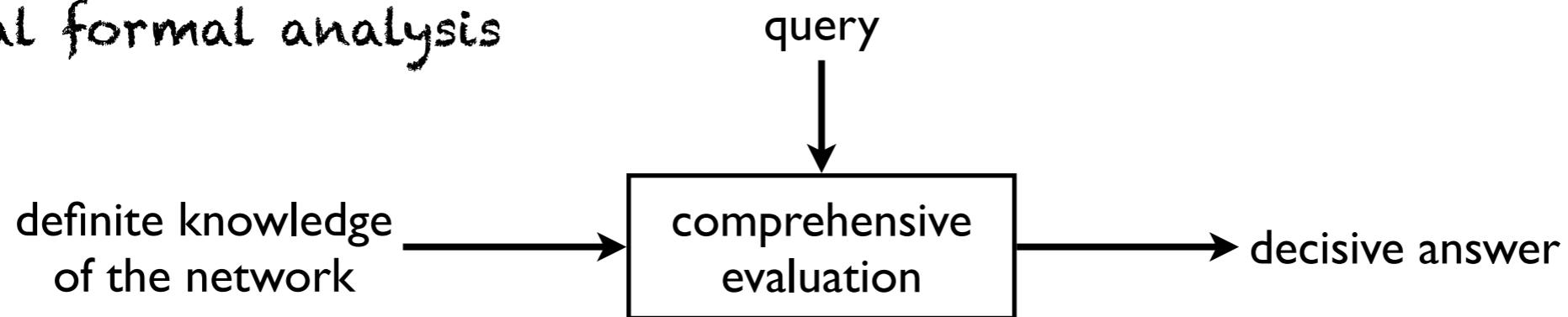
query

stop working
entirely?

decisive answer

challenges with complete analysis

classical formal analysis



a partial approach

classical formal analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

Loss-less modeling

uncertain environments

query

partial model

uncorrupted answer

partial information

query

?

decisive answer

a partial approach

classical formal analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

Loss-less modeling

uncertain environments

query

partial model

uncorrupted answer

relative-complete verification

partial information

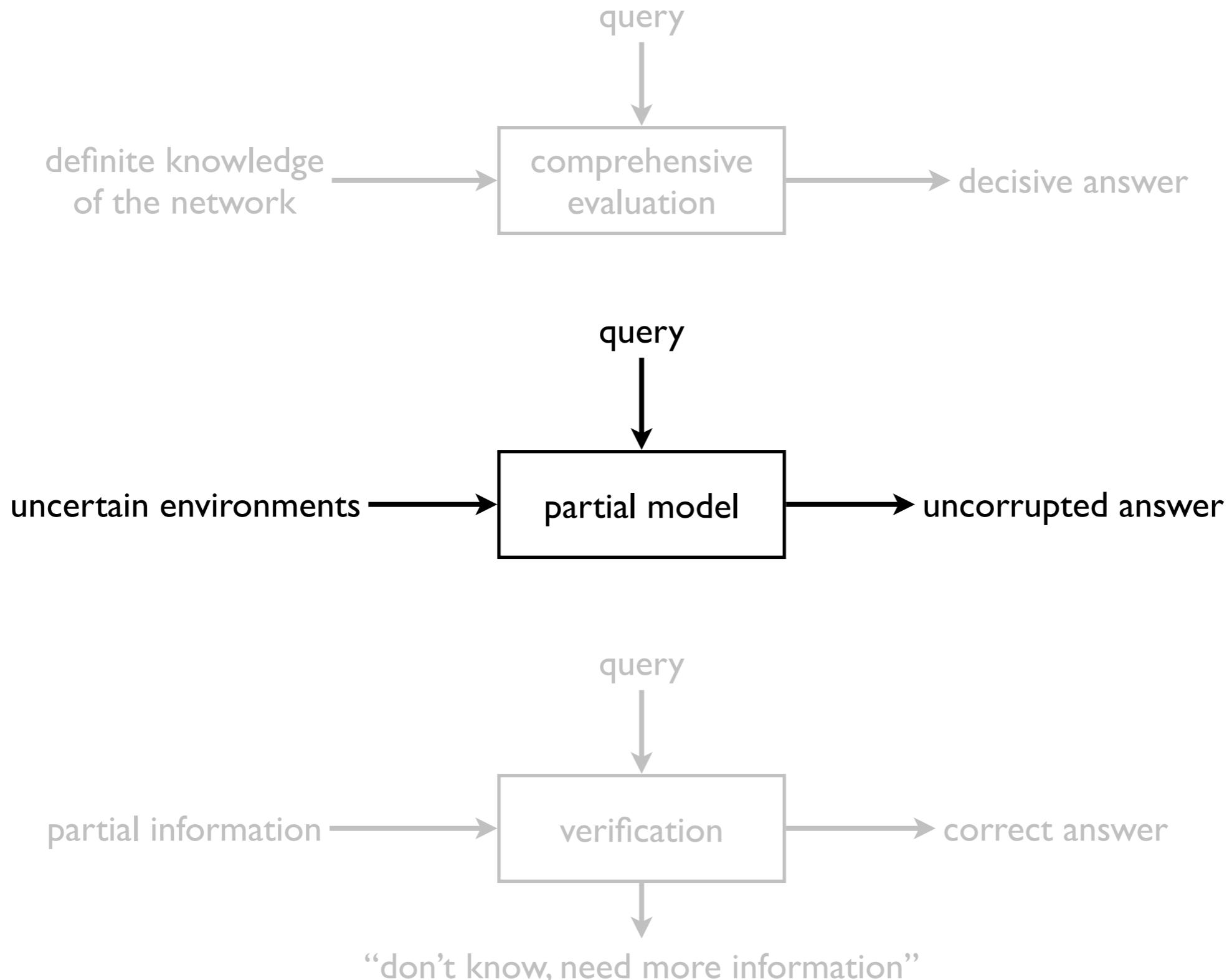
query

verification

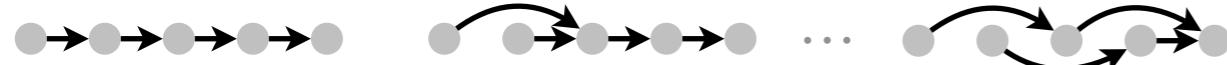
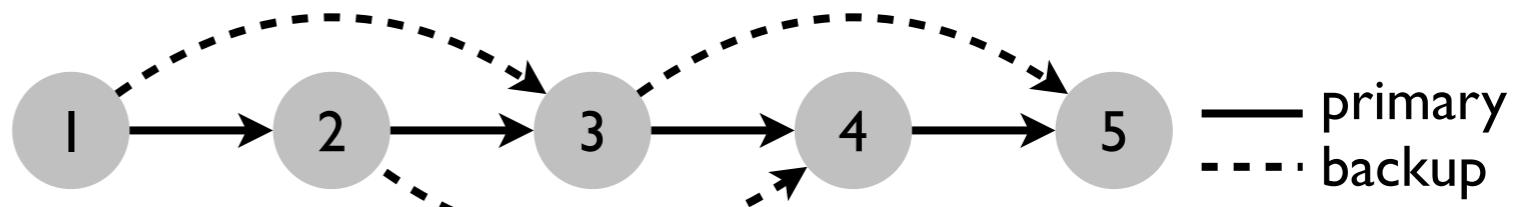
correct answer

“don’t know, need more information”

loss-less modeling



modeling



{

F	node	node
	1	2
	2	3
	3	4
	4	5

{

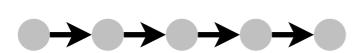
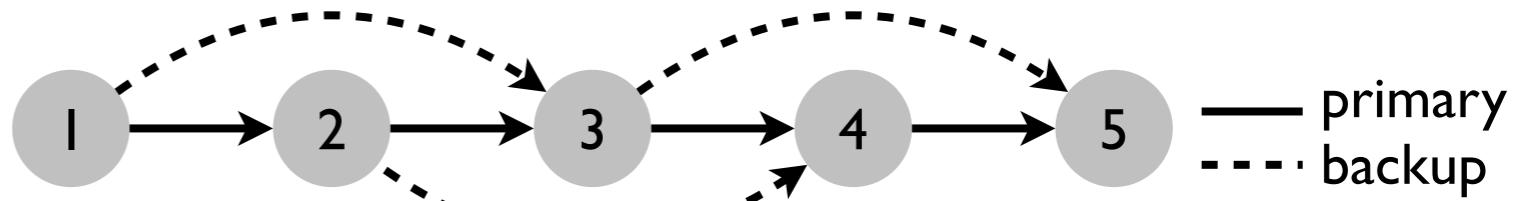
F	node	node
	1	3
	2	3
	3	4
	4	5

{

F	node	node
	1	3
	2	4
	3	5
	4	1

forwarding
tables

modeling



F	node	node
	1	2
	2	3
	3	4
	4	5

F	node	node
	1	3
	2	3
...		
	3	4
	4	5

F	node	node
	1	3
	2	4
...		
	3	5
	4	5

forwarding
tables



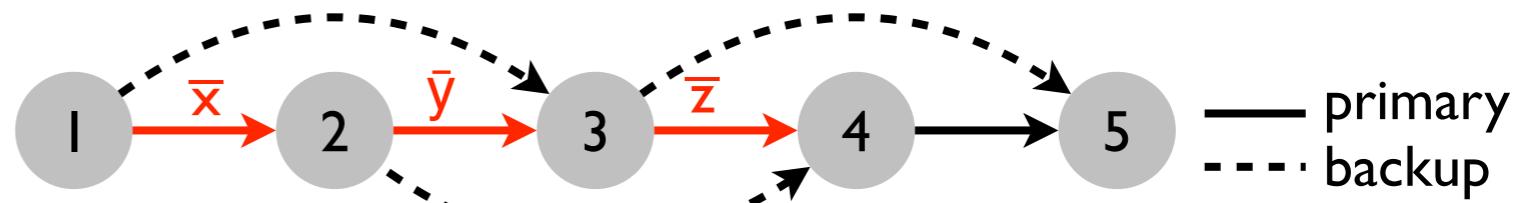
(reachability)
query

R	src	dest
	1	2
	1	3
	1	4
	1	5
	2	3
	2	4
...		

R	src	dest
	1	3
	1	4
...		
	1	5
	2	3
	2	4
...		

R	src	dest
	1	3
	1	5
...		
	2	4
...		

modeling



F	node	node
	1	2
	2	3
	3	4
	4	5

F	node	node
	1	3
	2	3
...		
	3	4
	4	5

F	node	node
	1	3
	2	4
...		
	3	5
	4	5

c-table:

F	node	node
	1	2
	1	3
	2	3
	2	4
	3	4
	3	5
	4	5

a tuple can occur only when the condition is satisfied

Rep

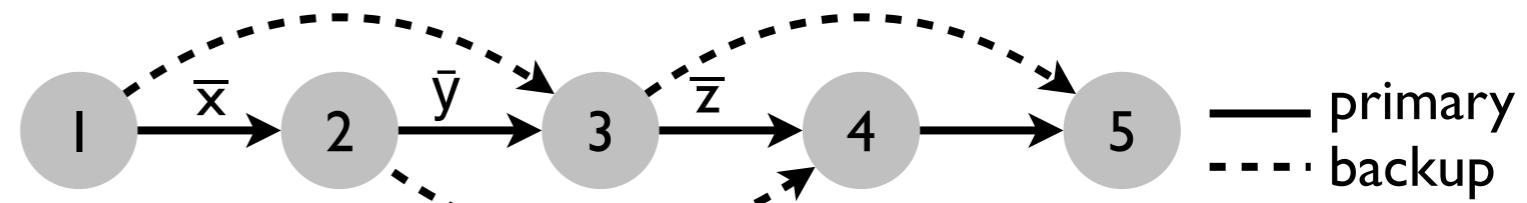
I: normal
0: failed

R	src	dest
	1	2
	1	3
	1	4
	1	5
	2	3
	2	4
...		

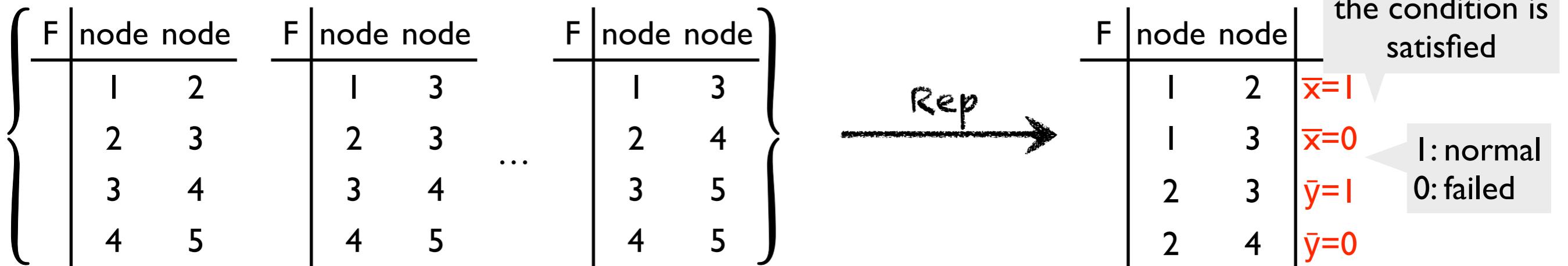
R	src	dest
	1	3
	1	4
...		
	1	5
	2	3
	2	4

R	src	dest
	1	3
	1	5
	2	4
...		

modeling



a tuple can occur only when the condition is satisfied



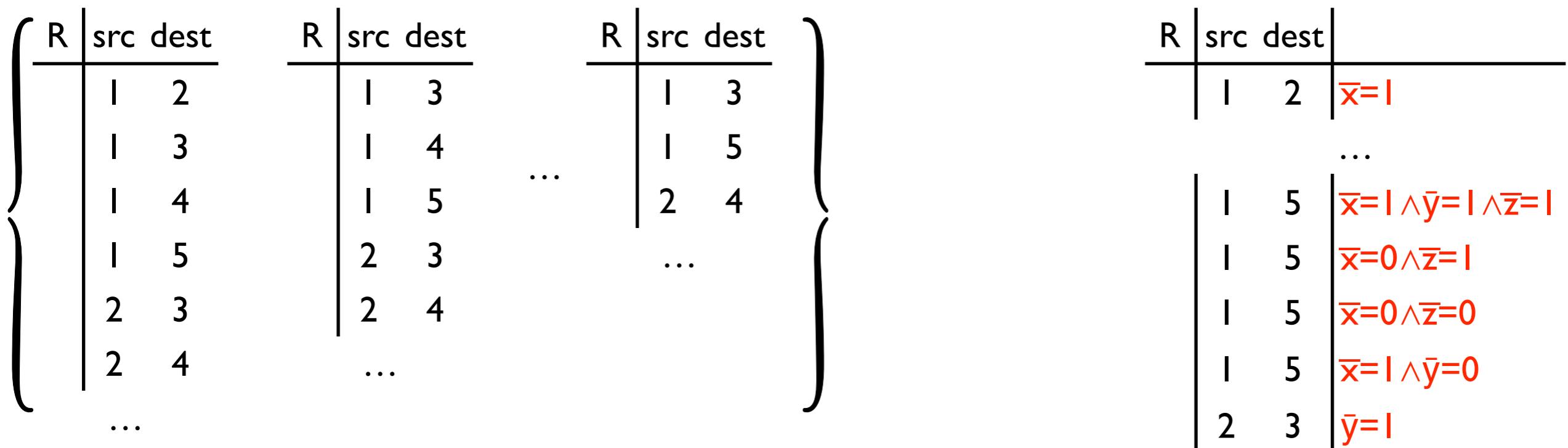
Rep

卷之三

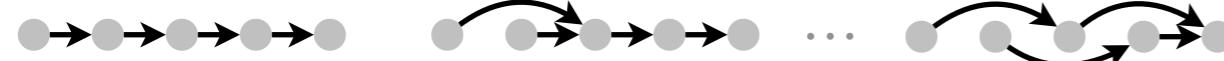
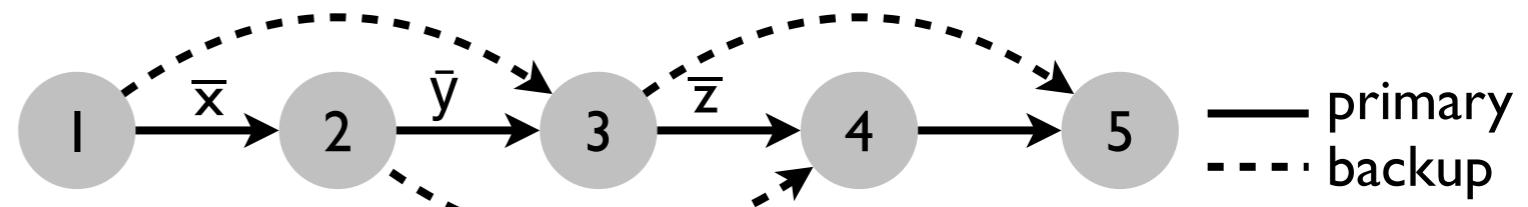
↓
query

A vertical black arrow pointing downwards, indicating the direction of the following text.

...
↓ query



modeling



F	node	node									
	1	2		1	3		1	3		1	2
	2	3		2	3		2	4		2	3
	3	4		3	4		3	5		2	3
	4	5		4	5		4	5		2	4

Rep

...
query
query

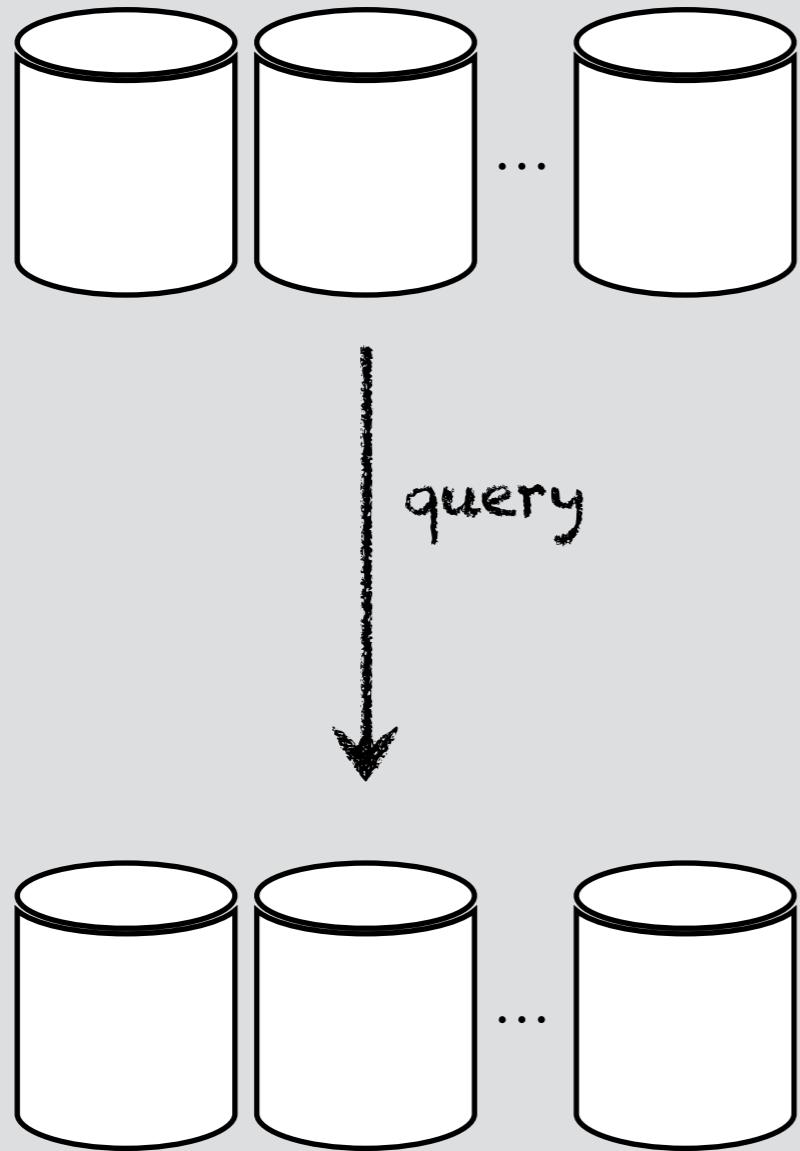
R	src	dest									
	1	2		1	3		1	3		1	2
	1	3		1	4		1	5		1	3
	1	4		1	5		2	4		1	5
	1	5		2	3		...			1	5
	2	3		2	4		...			1	5
	2	4								1	5
...										2	3

Rep

...

loss-less modeling

all definite instances (regular tables)

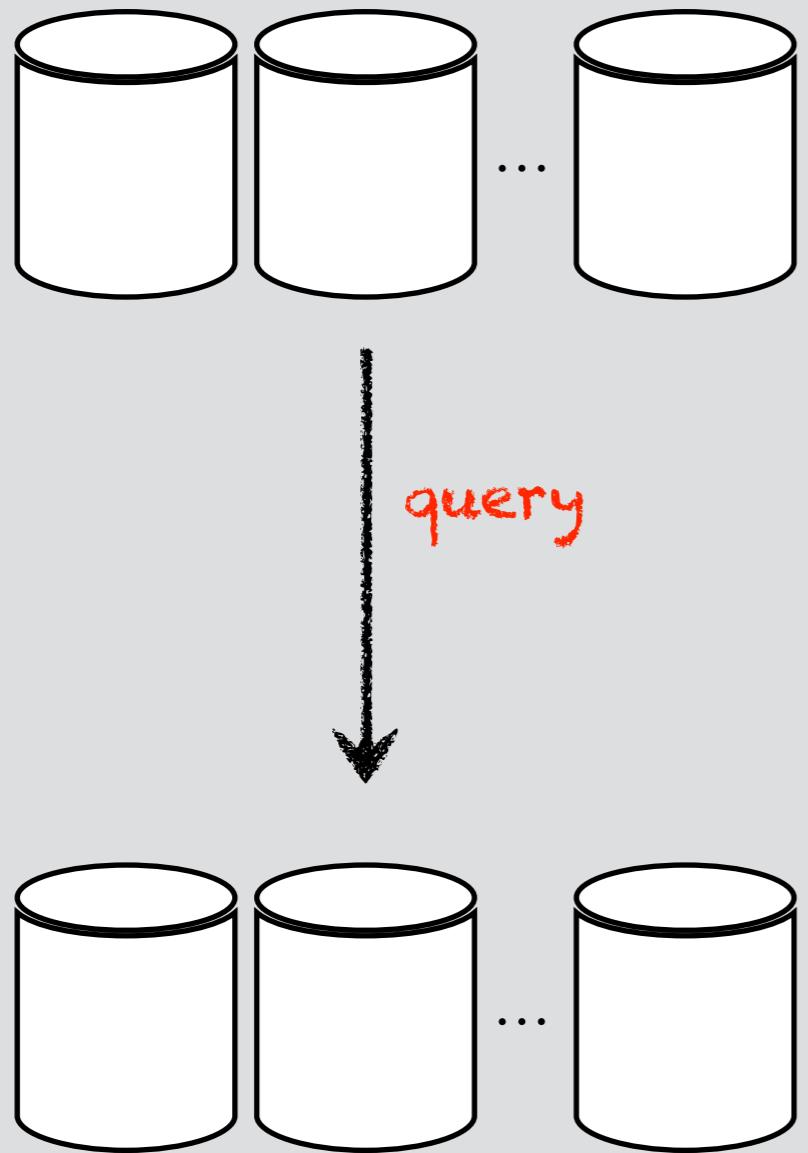


partial representation (c-tables)

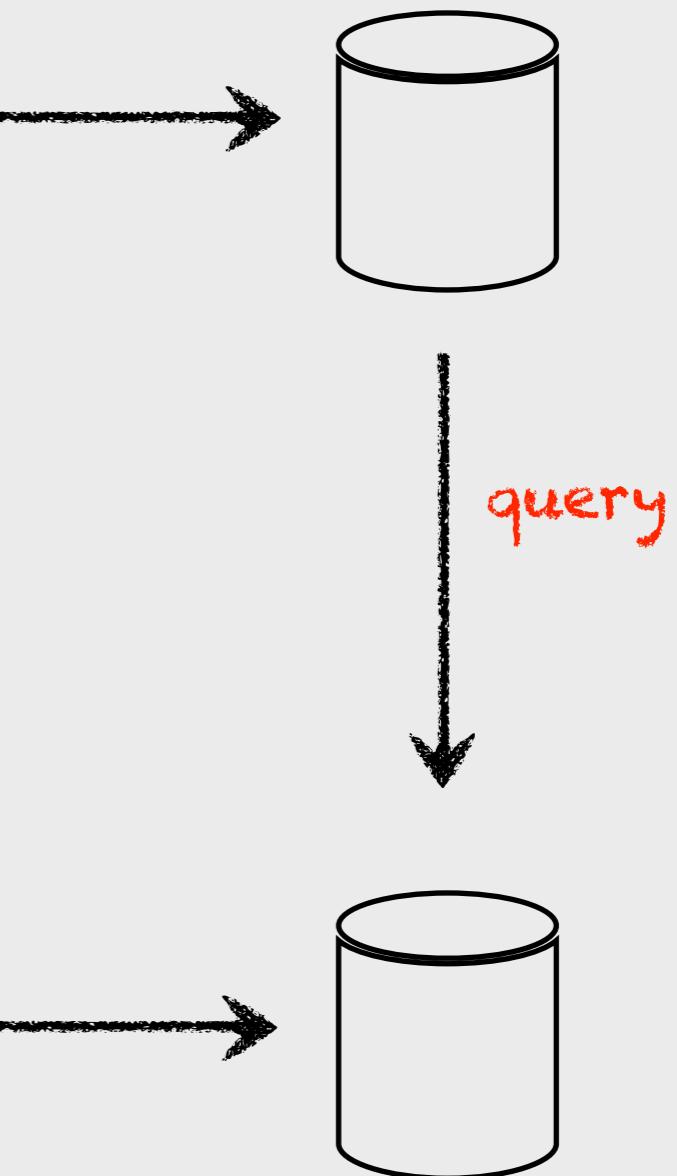
difference (between regular- and c- tables) not visible to the query

loss-less modeling

all definite instances (regular tables)



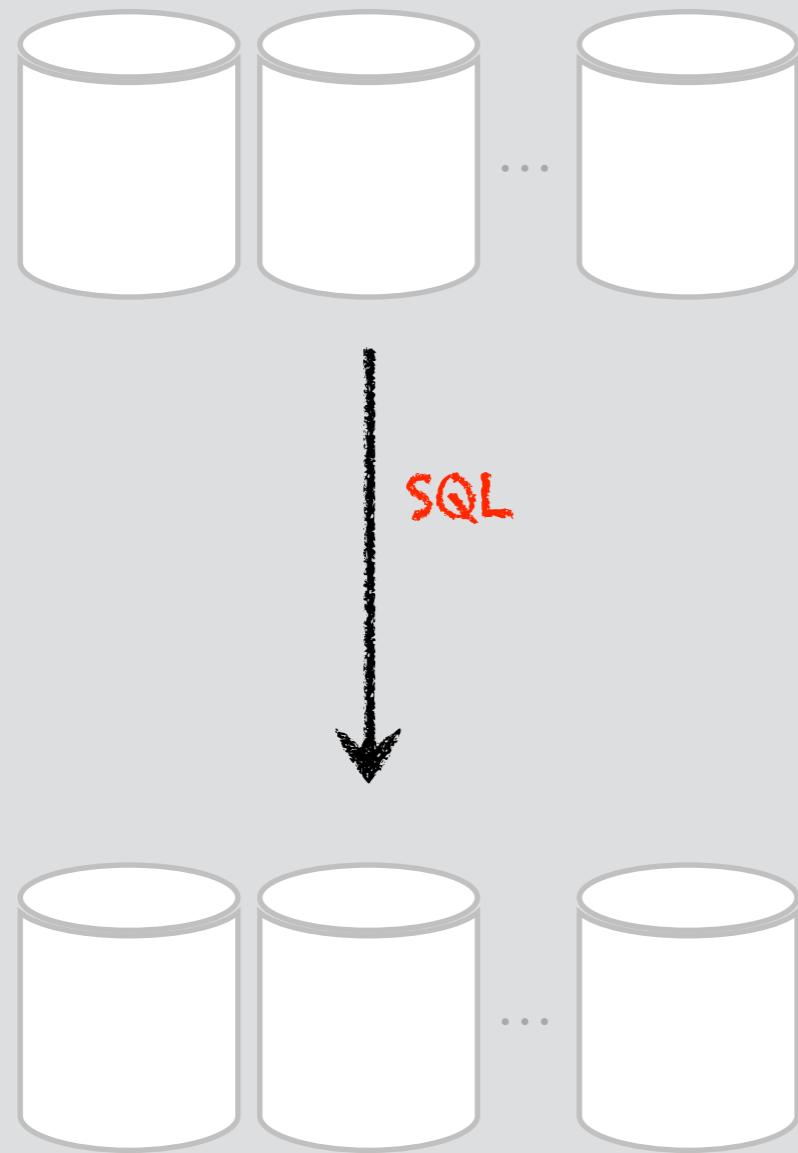
partial representation (c-tables)



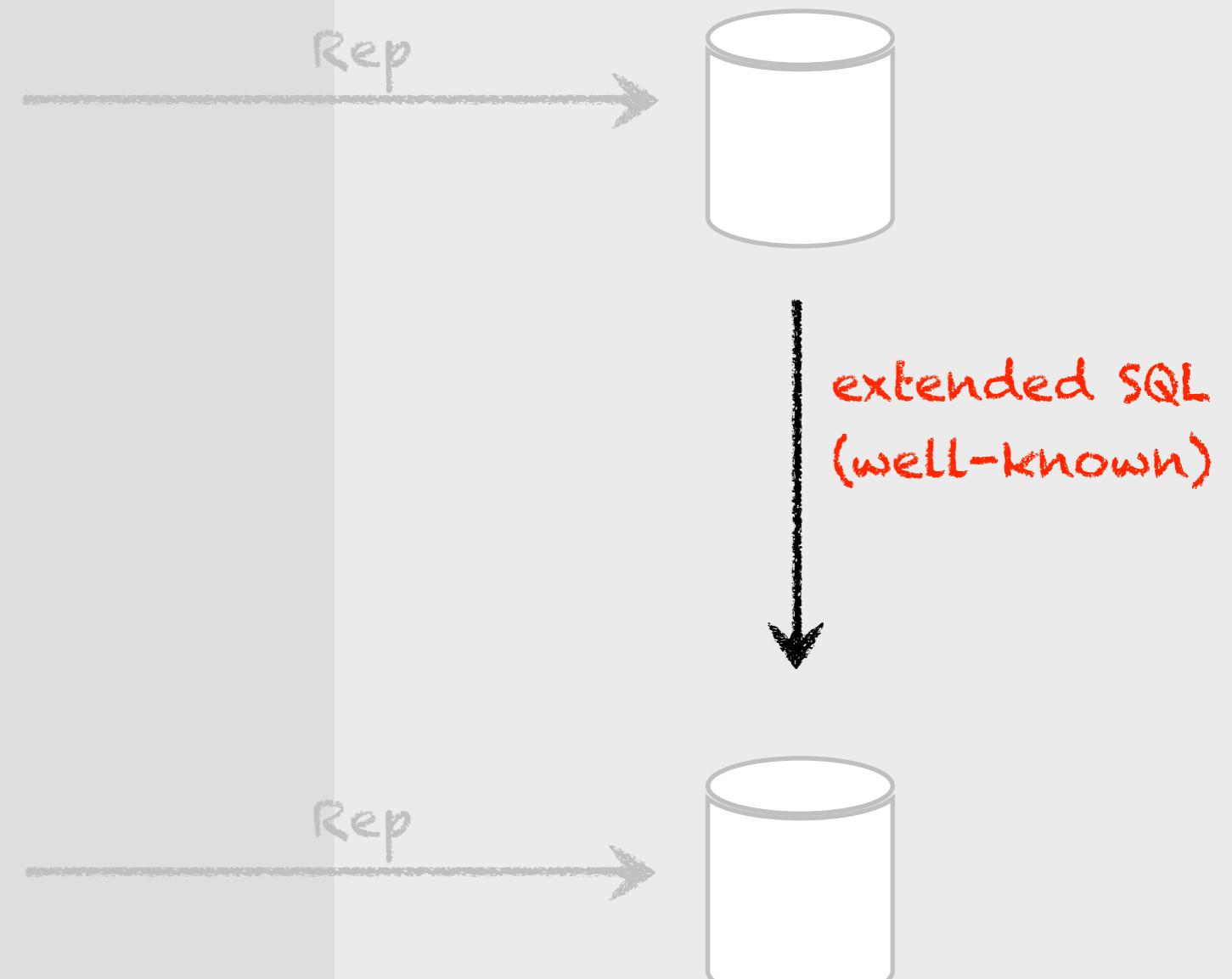
difference (between regular- and c- tables) not visible to the **query**

loss-less modeling with SQL?

all definite instances (regular tables)



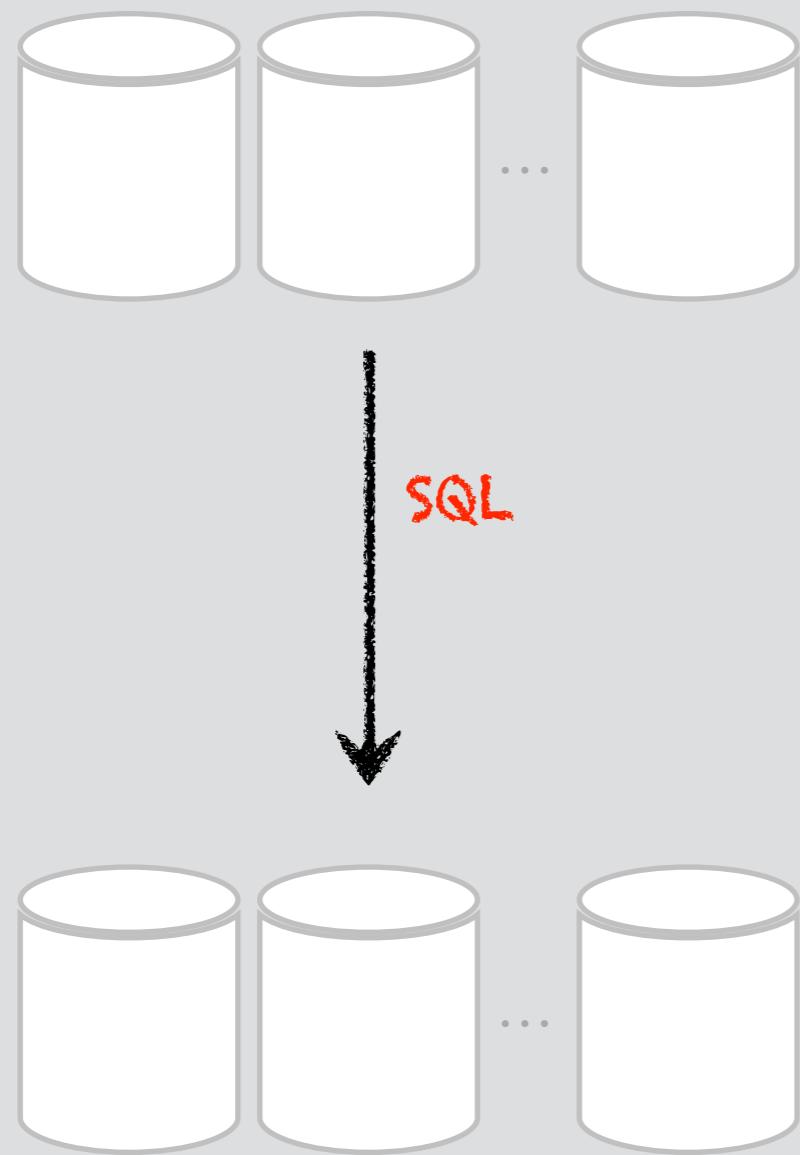
partial representation (c-tables)



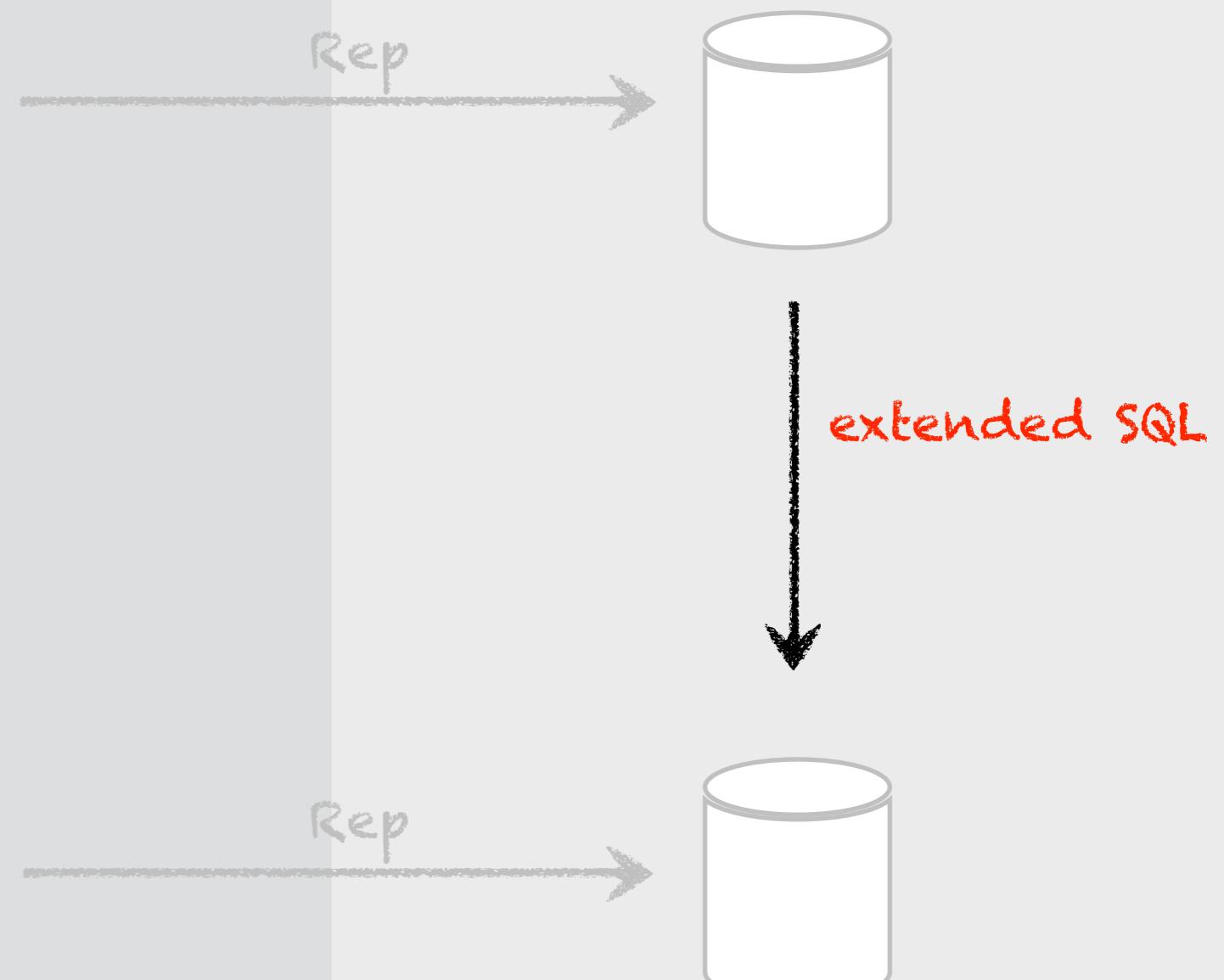
difference (between regular- and c- tables) not visible to the **SQL**

loss-less modeling with SQL?

all definite instances (regular tables)



partial representation (c-tables)



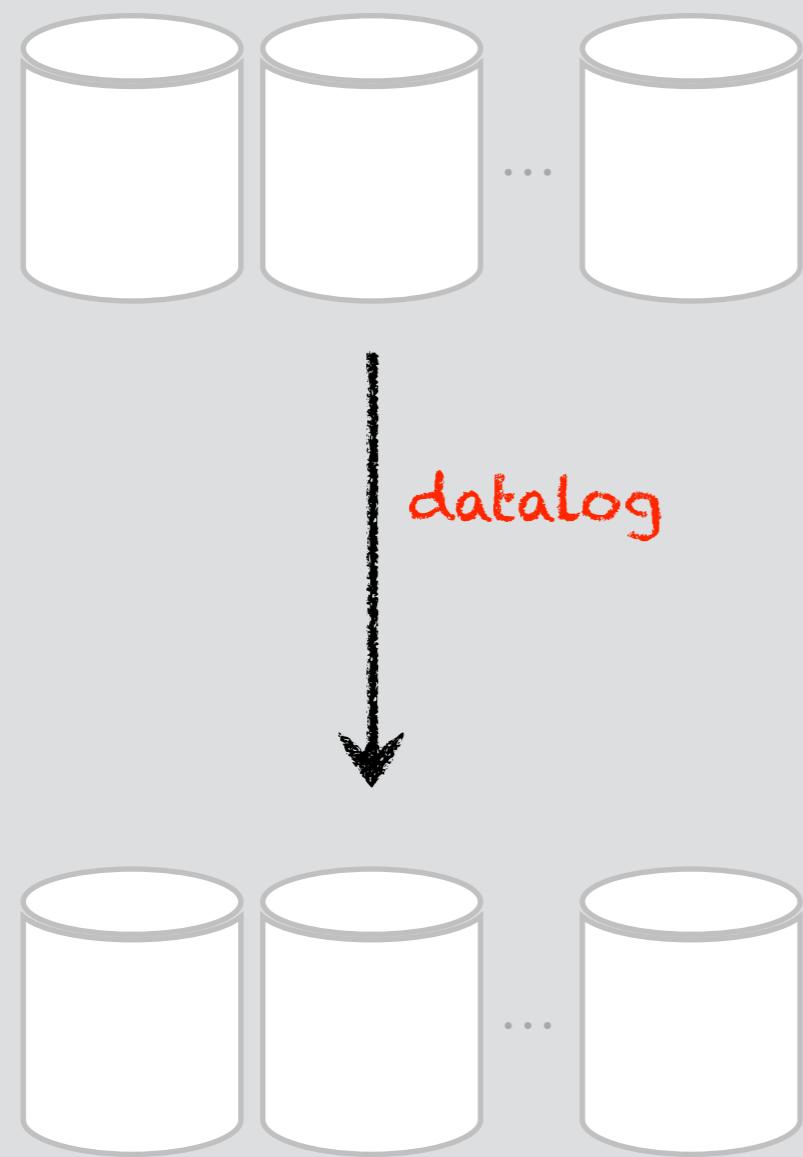
ad hoc data retrieval



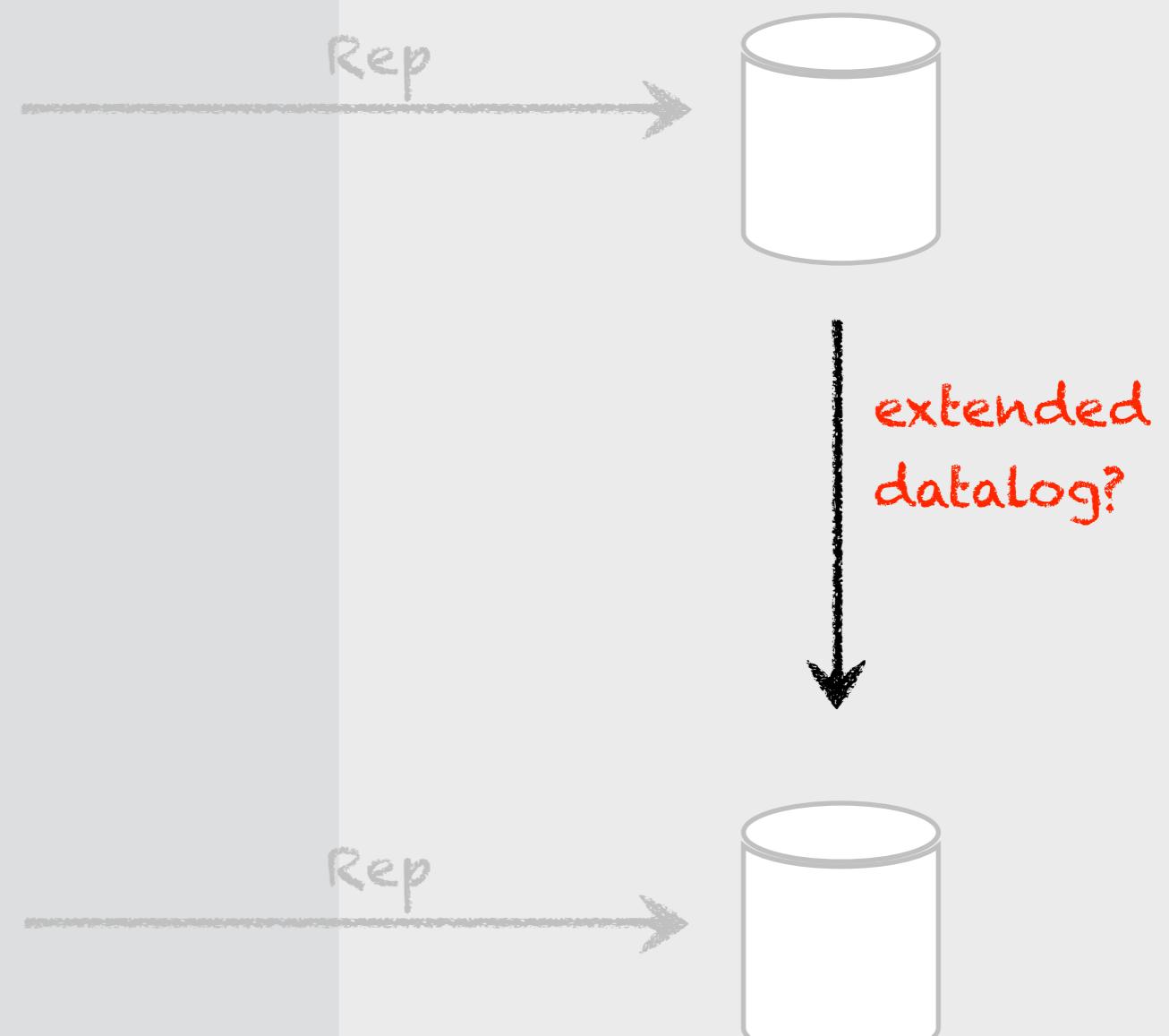
static analysis

loss-less modeling with fauré-log

all definite instances (regular tables)



partial representation (c-tables)



ad hoc data retrieval



static analysis

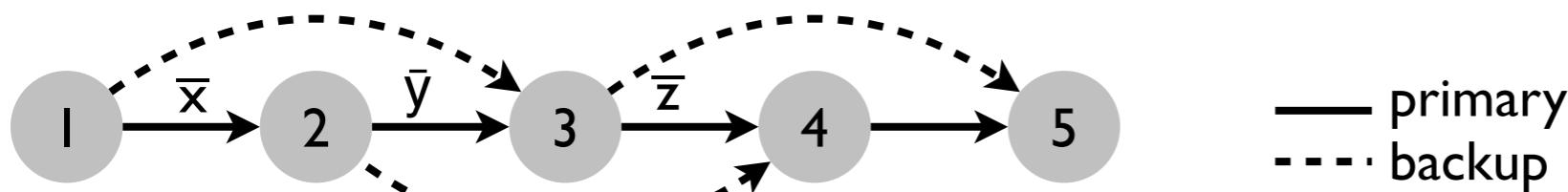
from datalog to fauré-log

	datalog	fauvé-log
syntax (rules q)	$H(u) :- B_1(u_1), \dots, B_n(u_n).$	$H(u) [(\wedge_{i=1}^n \varphi_i) \wedge (\wedge_{i=1}^m C_i)] :- B_1(u_1) [\varphi_1], \dots, B_n(u_n) [\varphi_n], C_1, \dots, C_m.$
semantics	$q(\mathbf{I}) = \{v(u) \mid v(u_i) \in \mathbf{I}\}, \quad \mathbf{I}$ is a database over schema R	

notions and definitions

u_i (free tuples)	contains symbols in $\text{var}(q)$ and $\text{dom}(R)$	
$\text{dom}(R)$ (attribute domain over schema R)	constants	constants U $\{\bar{x}, \bar{y}, \bar{z}, \dots\}$
v (valuation)	$v: \text{var}(q) \rightarrow \text{dom}(R)$ (i.e., $\{x, y, z, \dots\} \rightarrow \text{constants U } \{\bar{x}, \bar{y}, \bar{z}, \dots\}$)	
$\text{var}(q)$ (variables)	$\{x, y, z, \dots\}$	

Fauré-log queries



recursive fauré-log

```
/* reachability query */

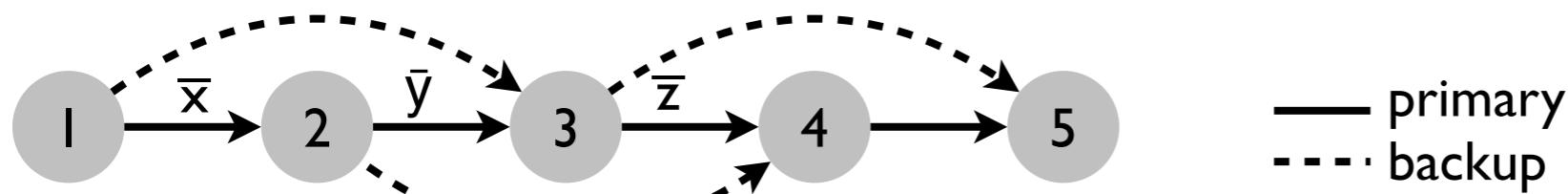
R(f, n1, n2) [φ] :- F(f, n1, n2) [φ].
R(f, n1, n2) [φF ∧ φR] :- F(f, n1, n3) [φF], R(f, n3, n2) [φR].
```

F	node	node	
1	2		$\bar{x}=1$
1	3		$\bar{x}=0$
2	3		$\bar{y}=1$
2	4		$\bar{y}=0$
...			



R	src	dest	
1	2		$\bar{x}=1$
...			
1	5		$\bar{x}=1 \wedge \bar{y}=1 \wedge \bar{z}=1$
1	5		$\bar{x}=0 \wedge \bar{z}=1$
1	5		$\bar{x}=0 \wedge \bar{z}=0$
1	5		$\bar{x}=1 \wedge \bar{y}=0$
2	3		$\bar{y}=1$
...			

Fauré-log queries



failure patterns over R

$T_1(f, n_1, n_2)[\phi \wedge \bar{x} + \bar{y} + \bar{z} = 1] :- R(f, n_1, n_2)[\phi], \bar{x} + \bar{y} + \bar{z} = 1.$

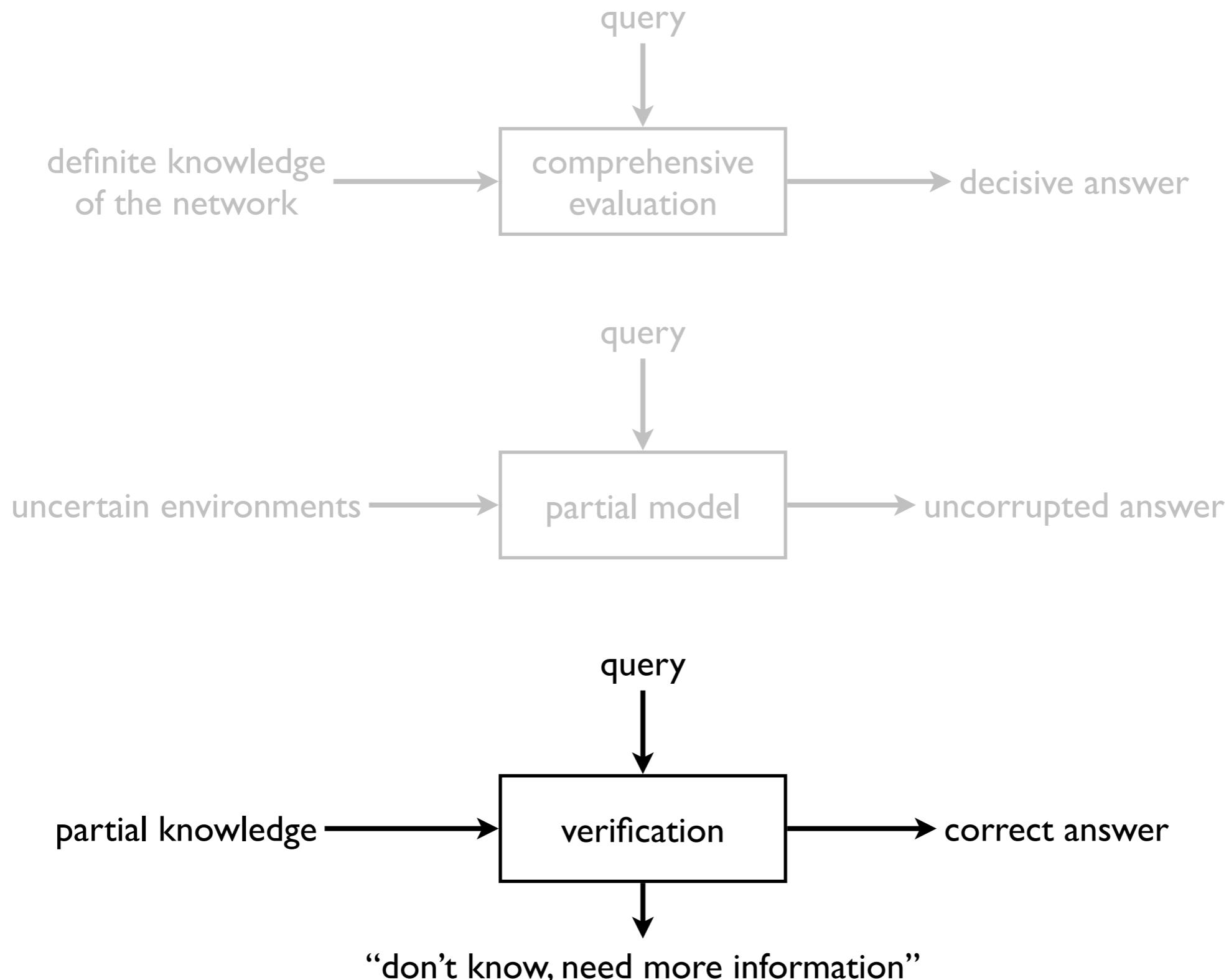
% reachability under 2-link failure

$T_2(f, 2, 5)[\phi \wedge \bar{y} = 0] :- T_1(f, 2, 5)[\phi], \bar{y} = 0.$ %
reachability between 2 and 5 under 2-link failure, one of the failure must be (2,3)

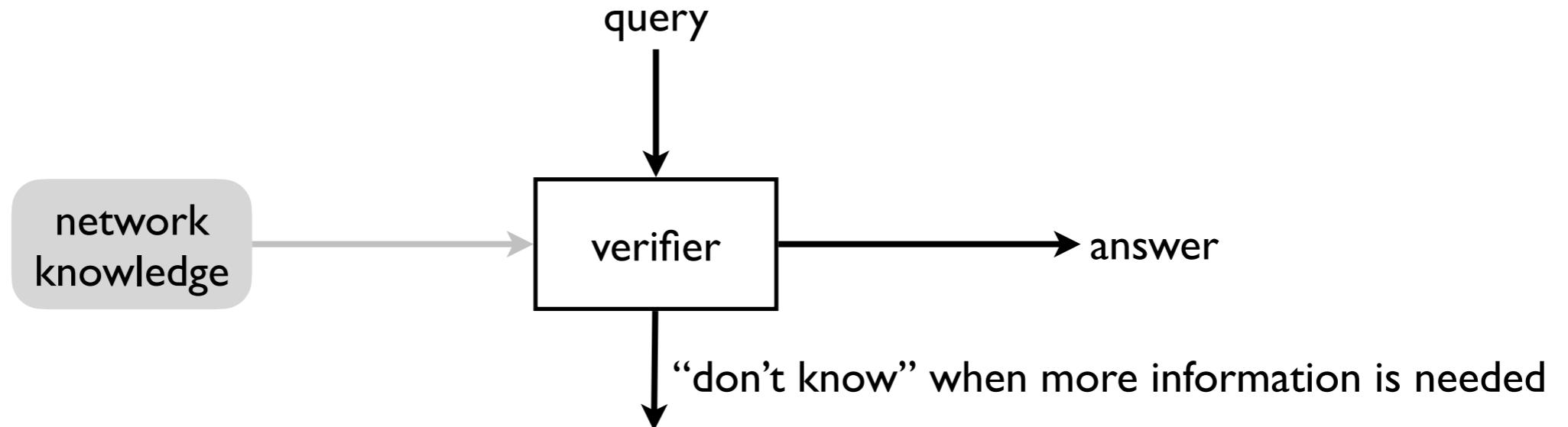
$T_3(f, 1, n_2)[\phi \wedge \bar{y} + \bar{z} < 2] :- R(f, 1, n_2)[\phi], \bar{y} + \bar{z} < 2.$ %
reachability to 1 with at least 1-link failure

R	src	dest	
	1	2	$\bar{x}=1$
			...
	1	5	$\bar{x}=1 \wedge \bar{y}=1 \wedge \bar{z}=1$
	1	5	$\bar{x}=0 \wedge \bar{z}=1$
	1	5	$\bar{x}=0 \wedge \bar{z}=0$
	1	5	$\bar{x}=1 \wedge \bar{y}=0$
2	3		$\bar{y}=1$
			...

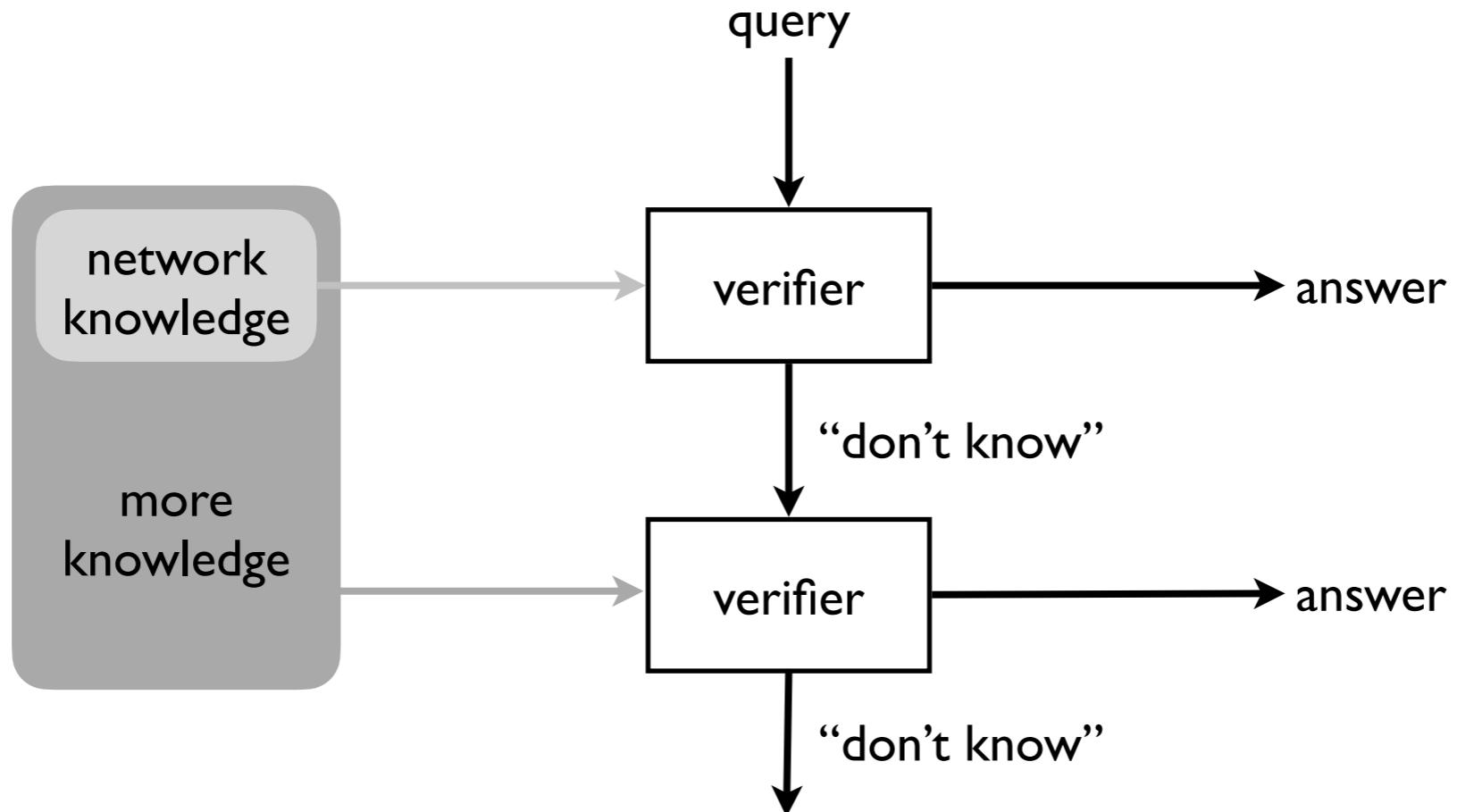
relative-complete verification



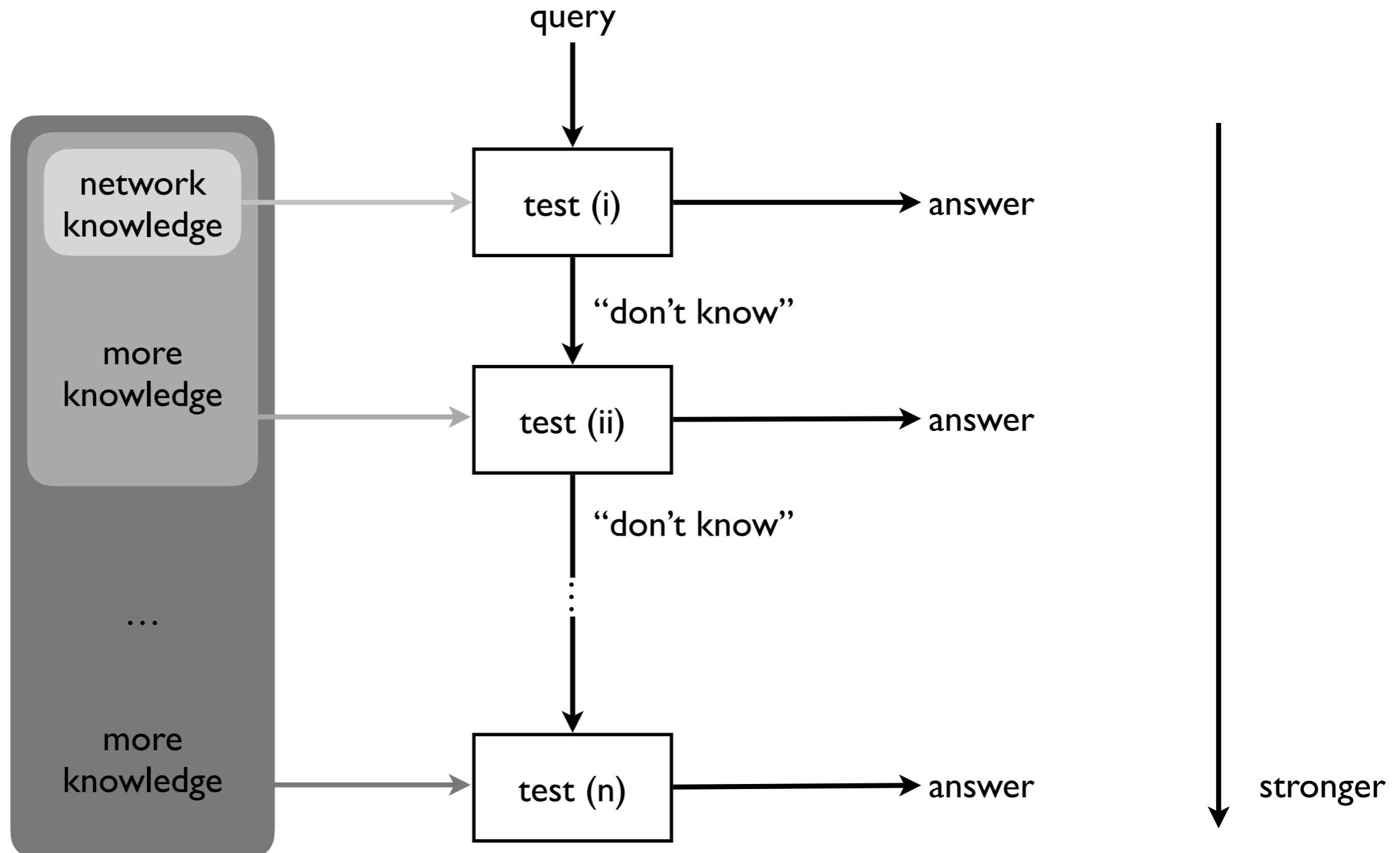
relative-complete verification



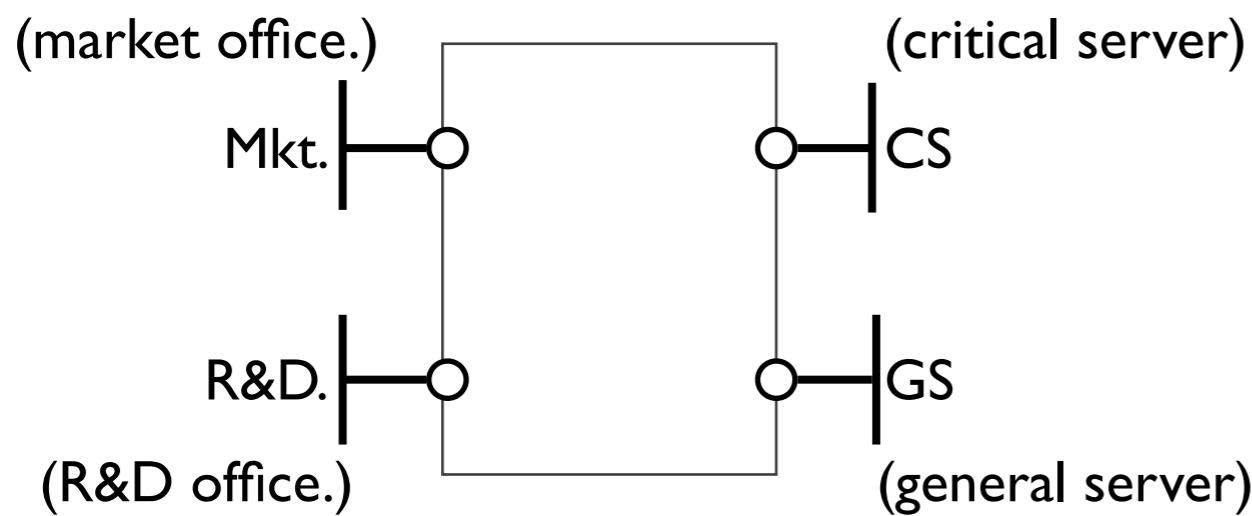
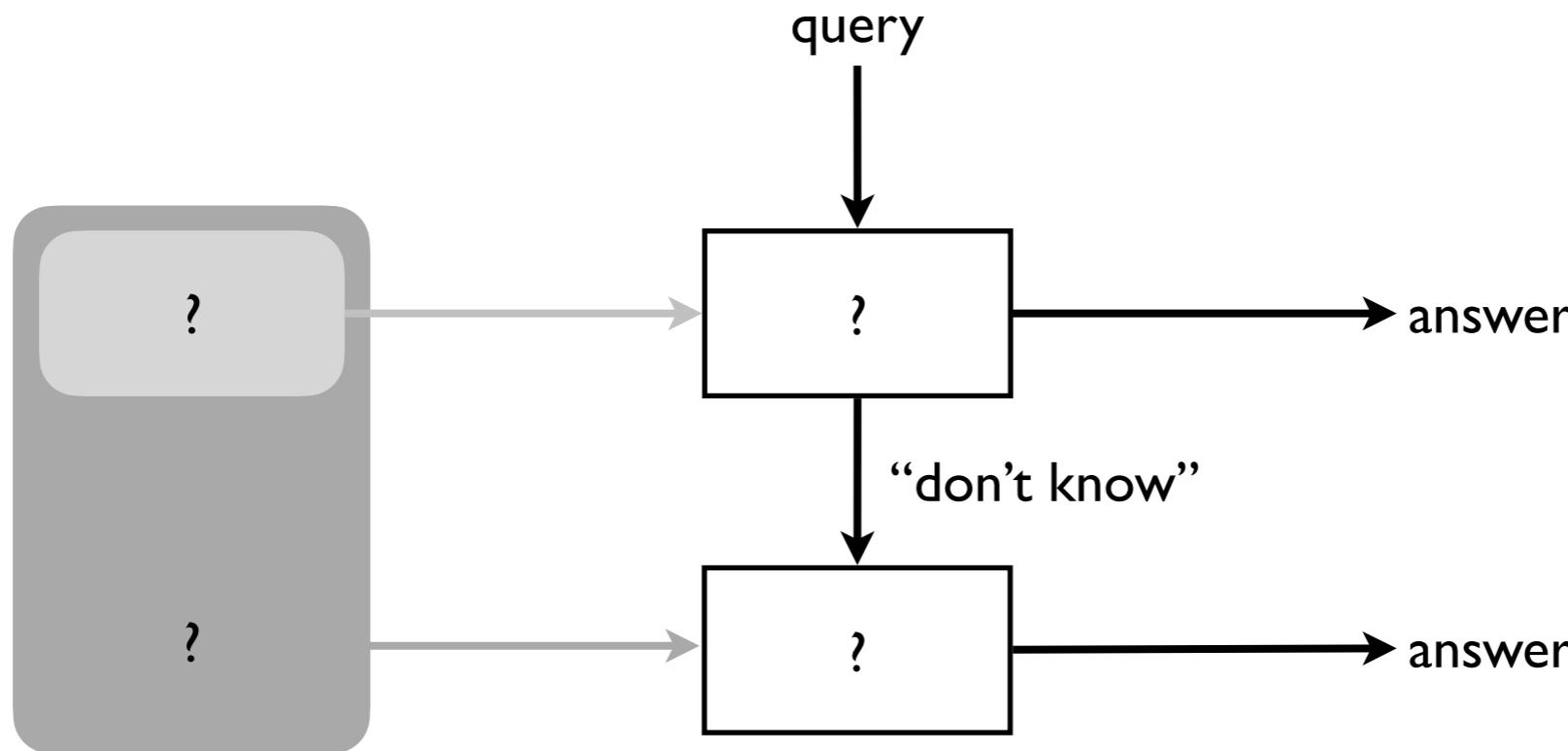
relative-complete verification



relative-complete verification



example relative-complete verification



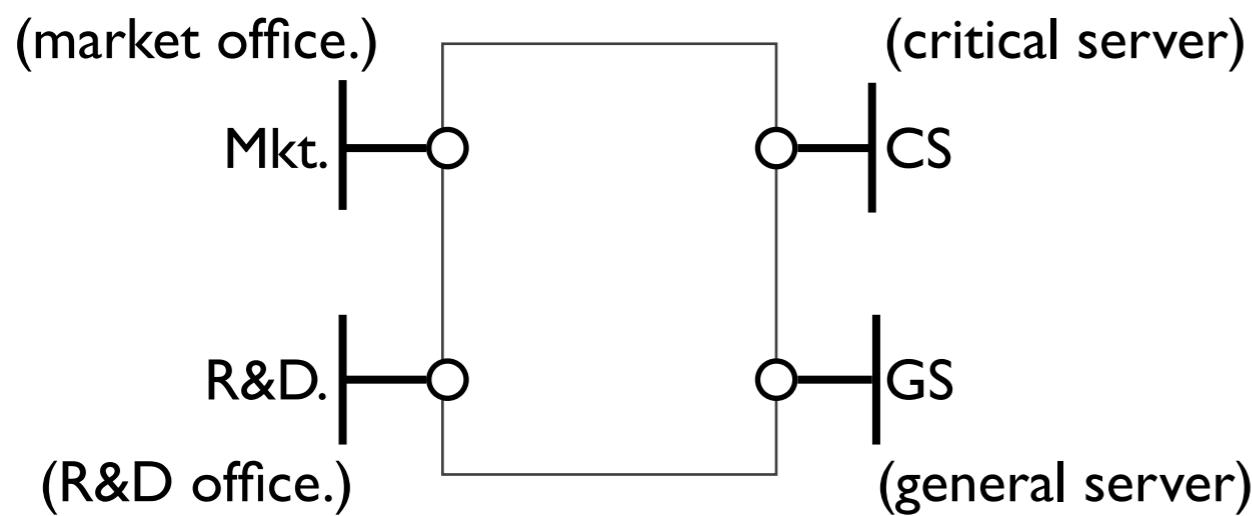
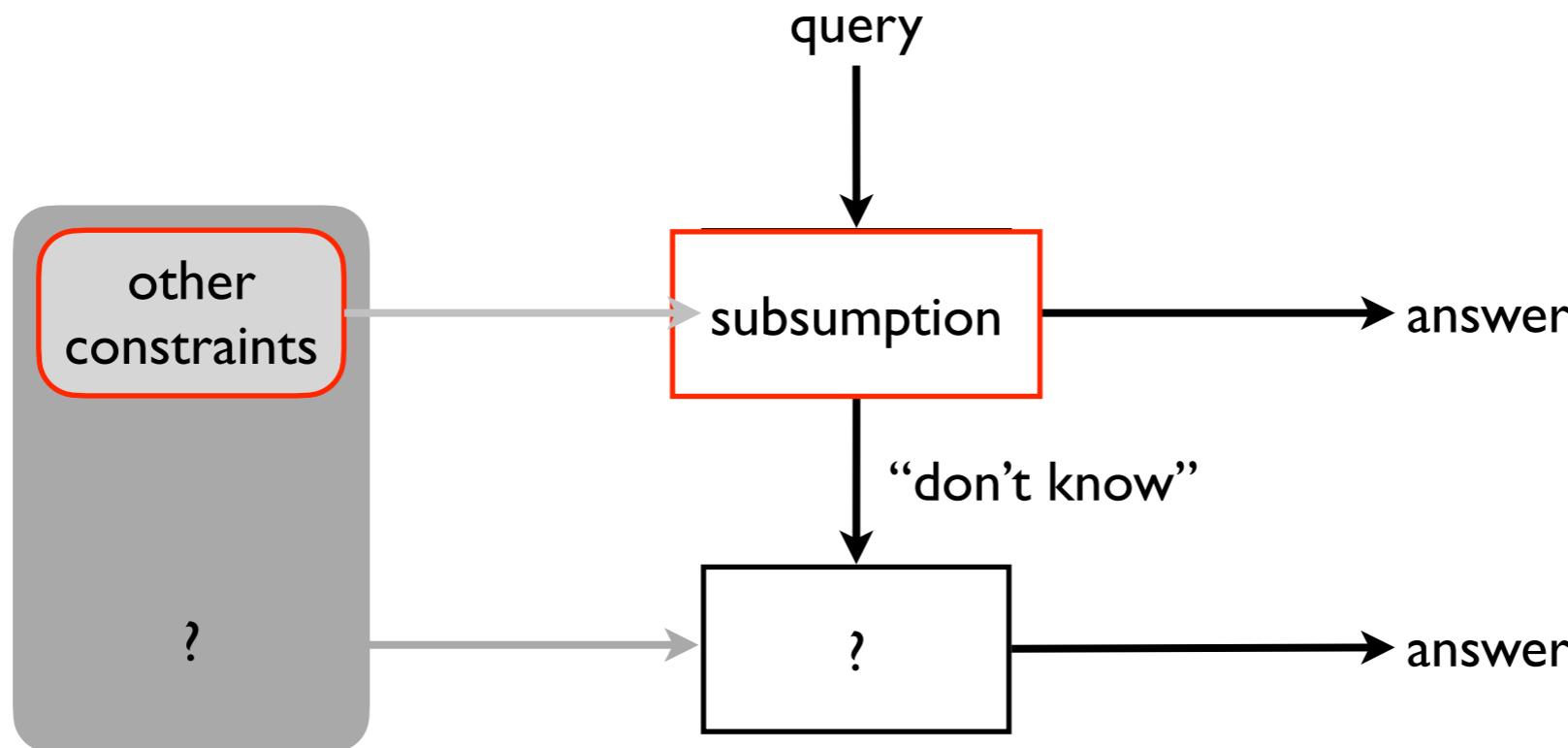
verification task

- invariants (T_1, T_2) continue to hold after updates

other teams

- security team maintains C_s
- TE team maintains C_{lb}

example relative-complete verification



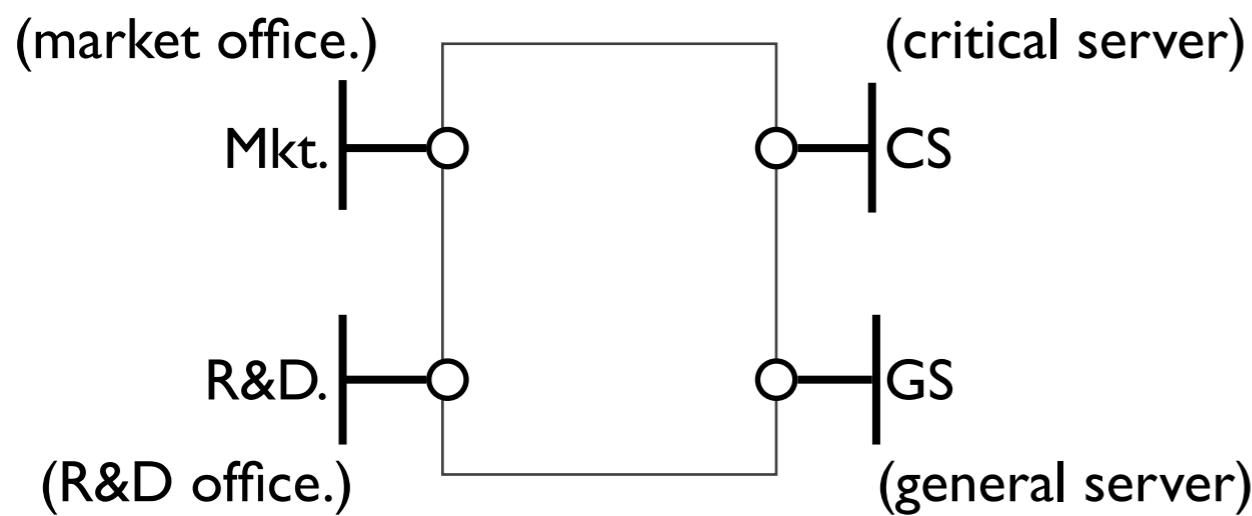
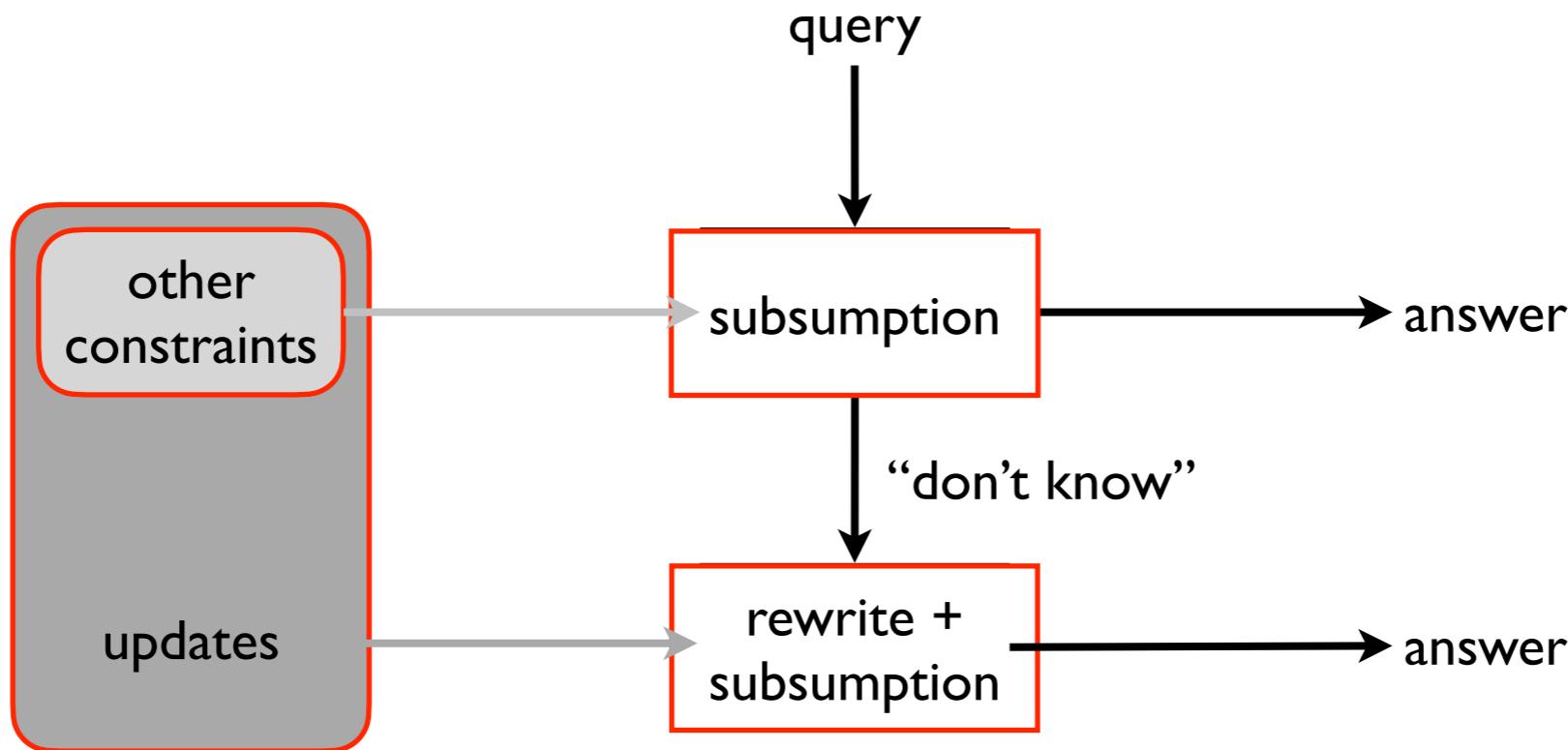
verification task

- invariants (T_1, T_2) continue to hold after updates

other teams

- security team maintains C_s
- TE team maintains C_{lb}

example relative-complete verification



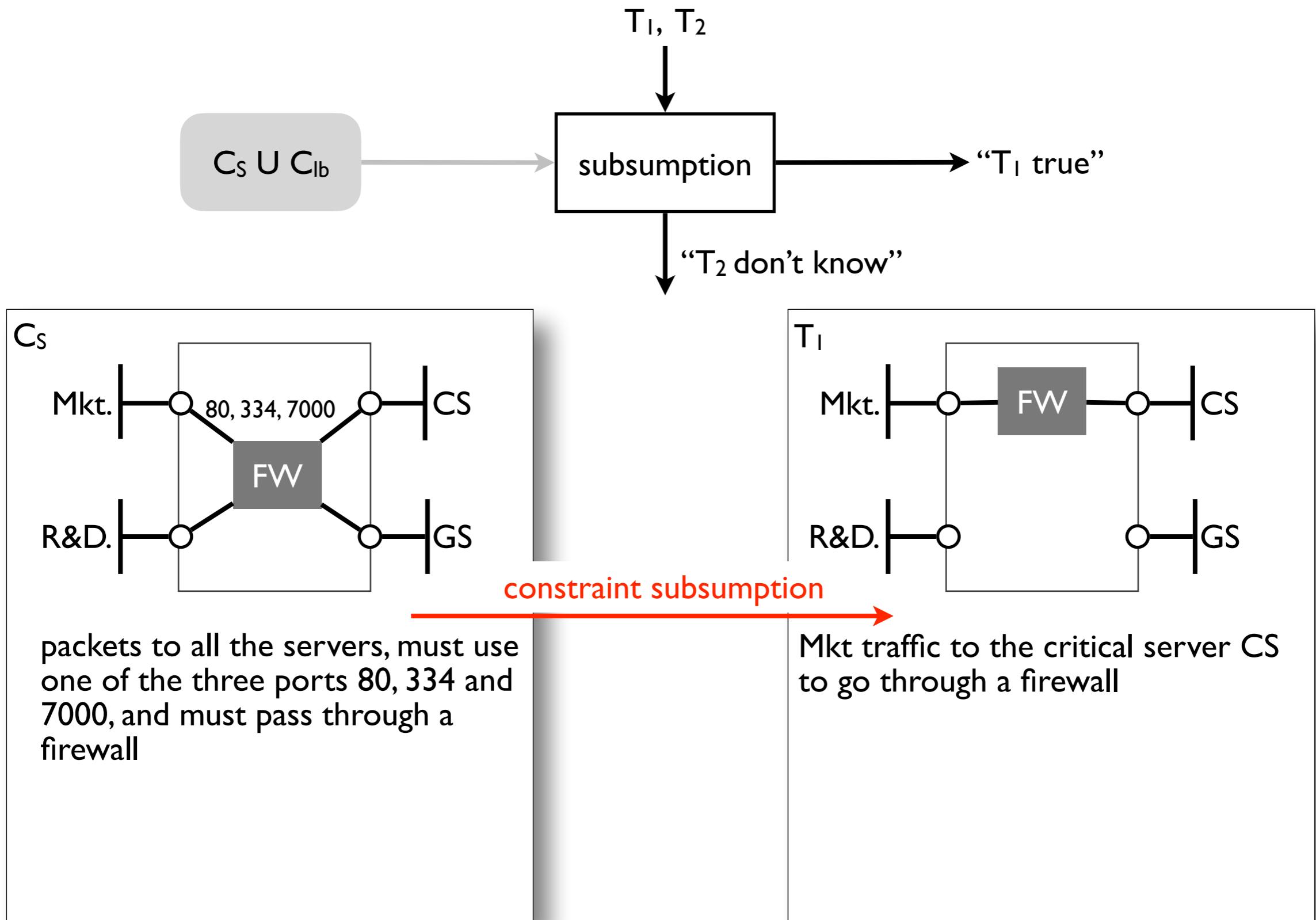
verification task

- invariants (T_1, T_2) continue to hold after updates

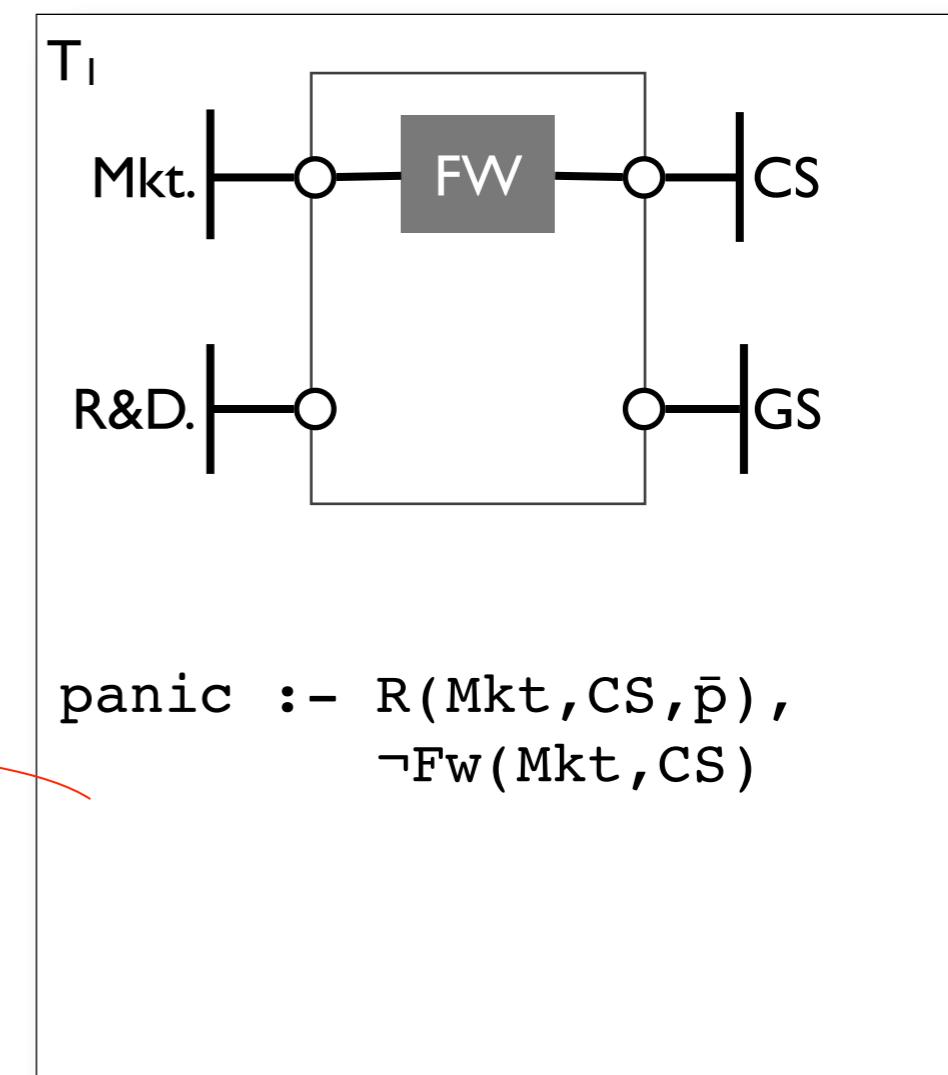
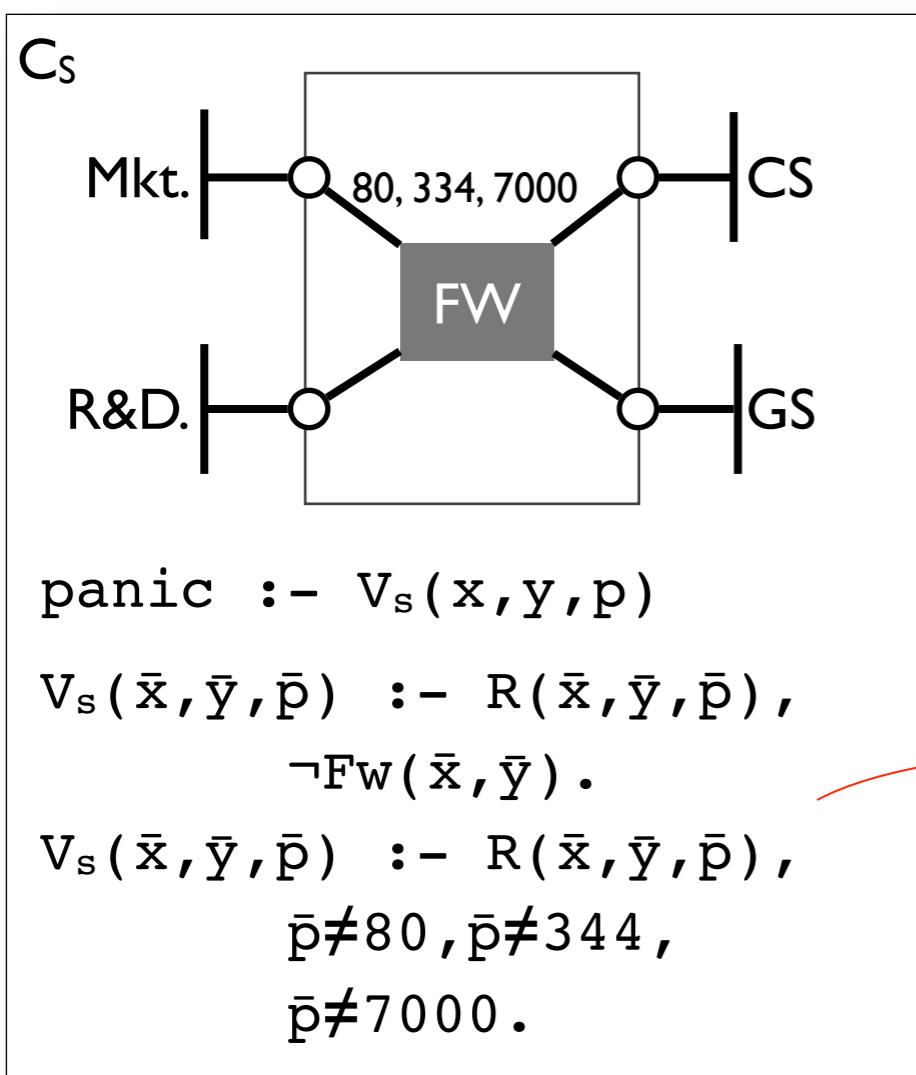
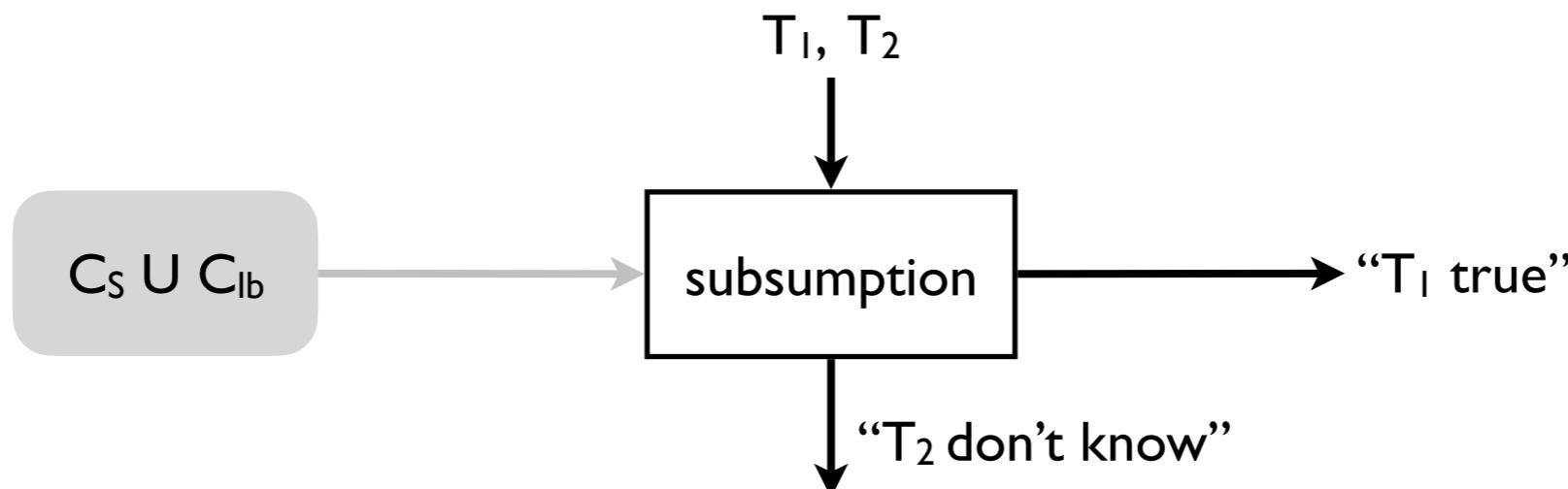
other teams

- security team maintains C_s
- TE team maintains C_{lb}

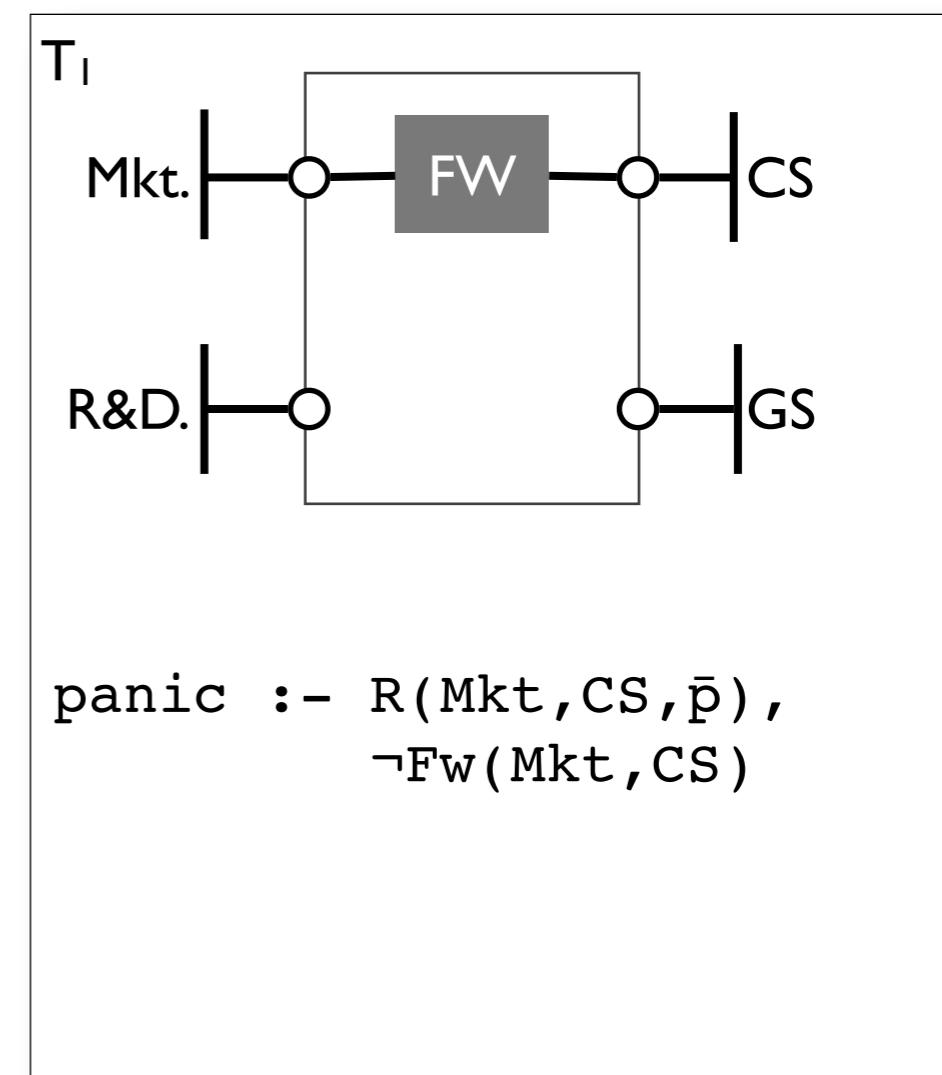
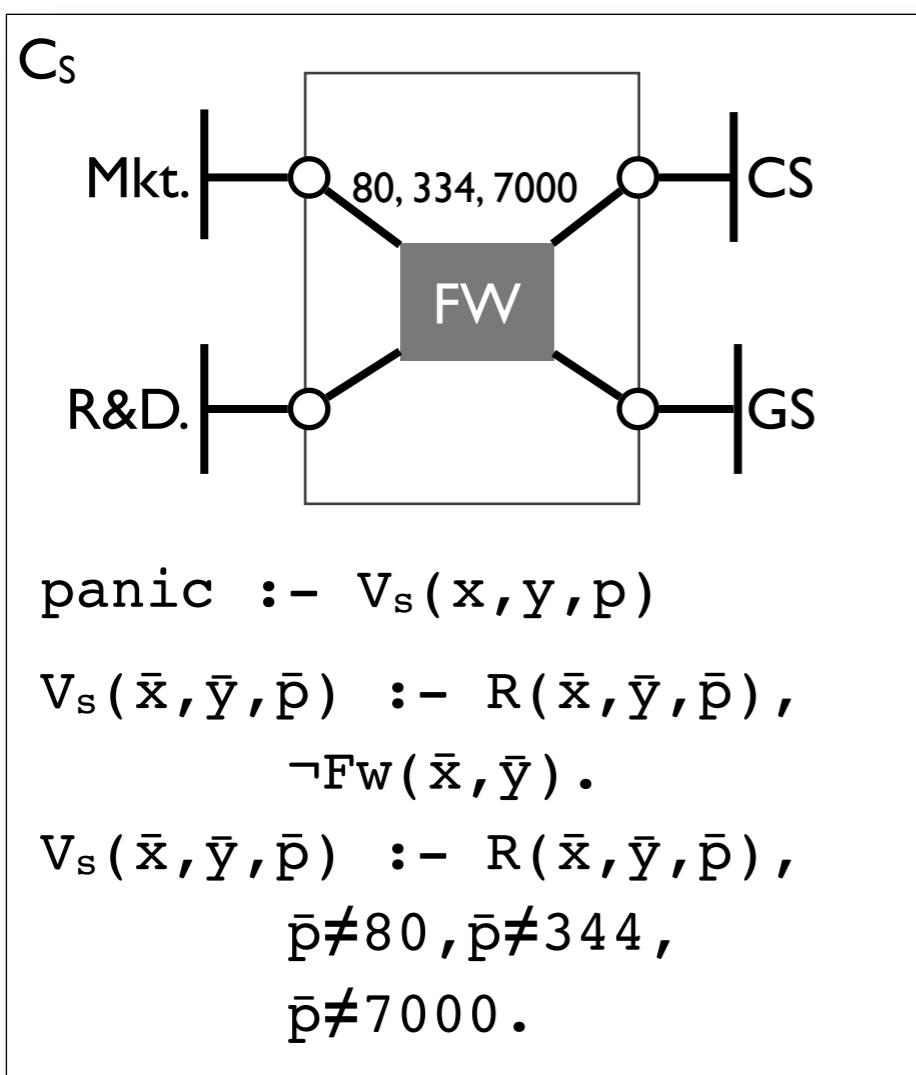
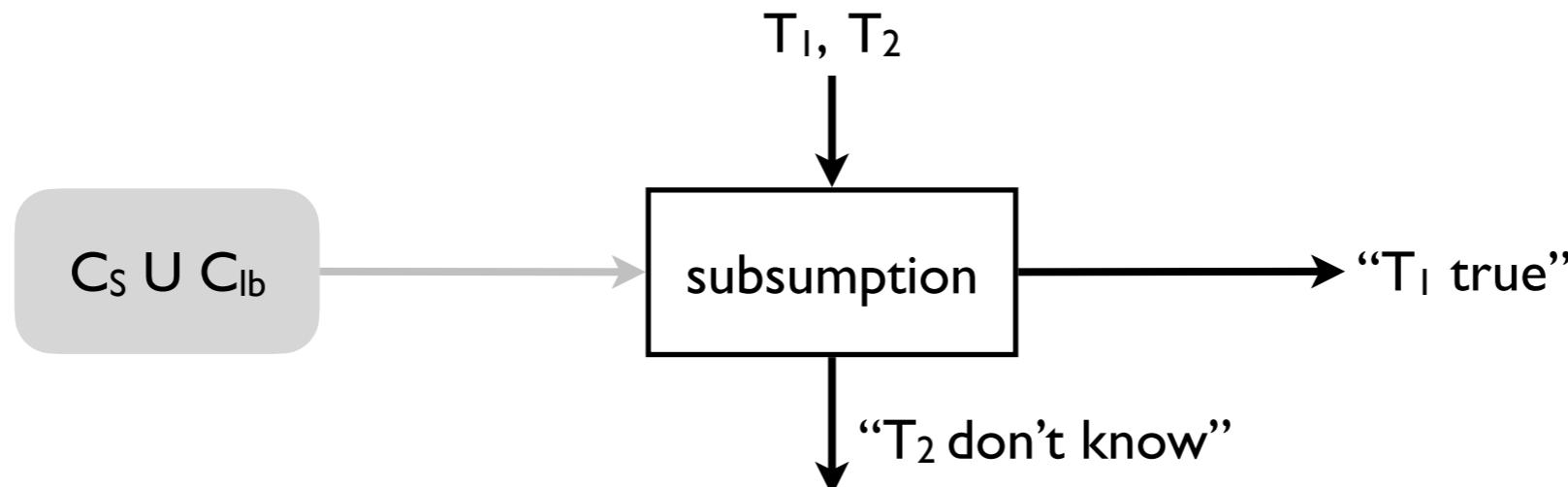
category (i) test: using only constraints



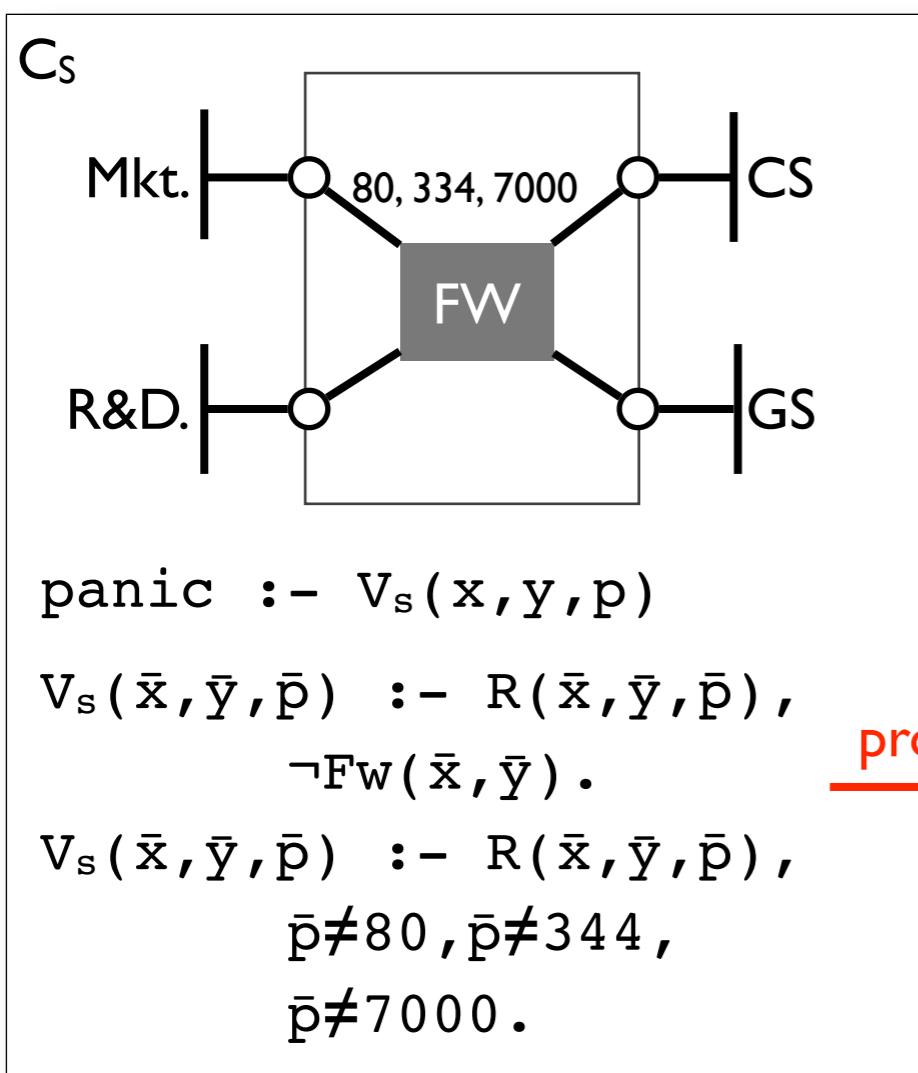
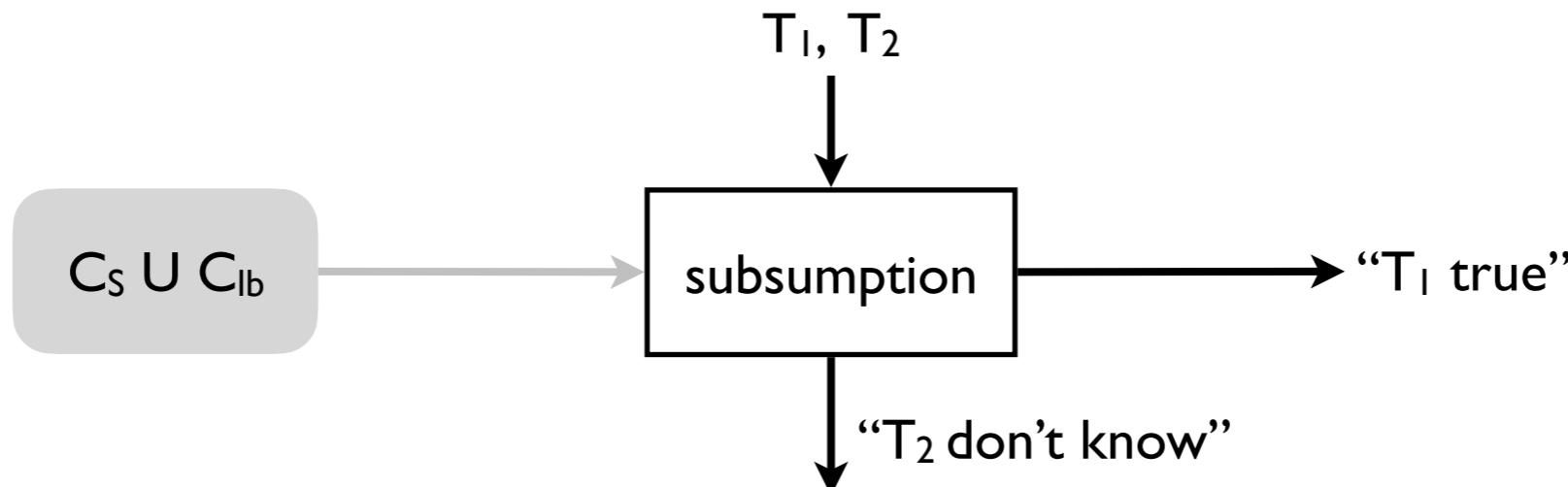
category (i) test: using only constraints



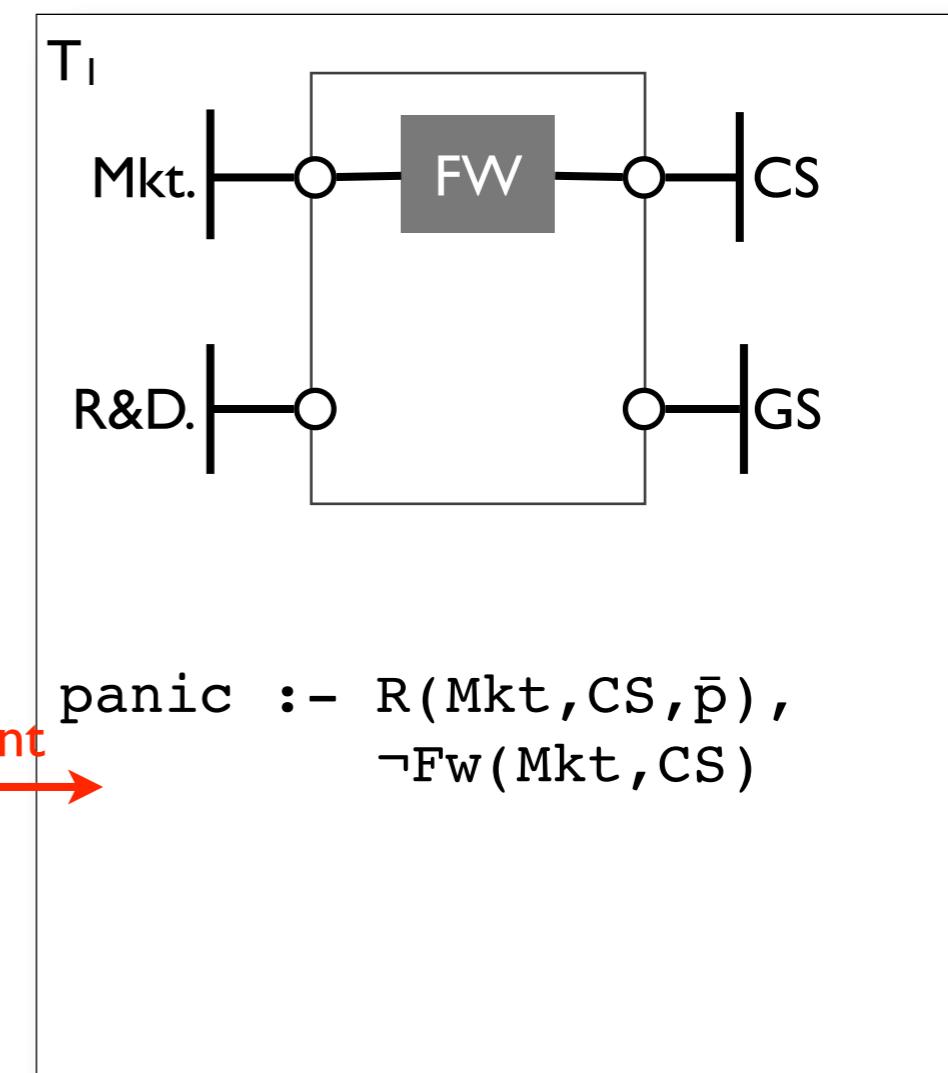
category (i) test: using only constraints



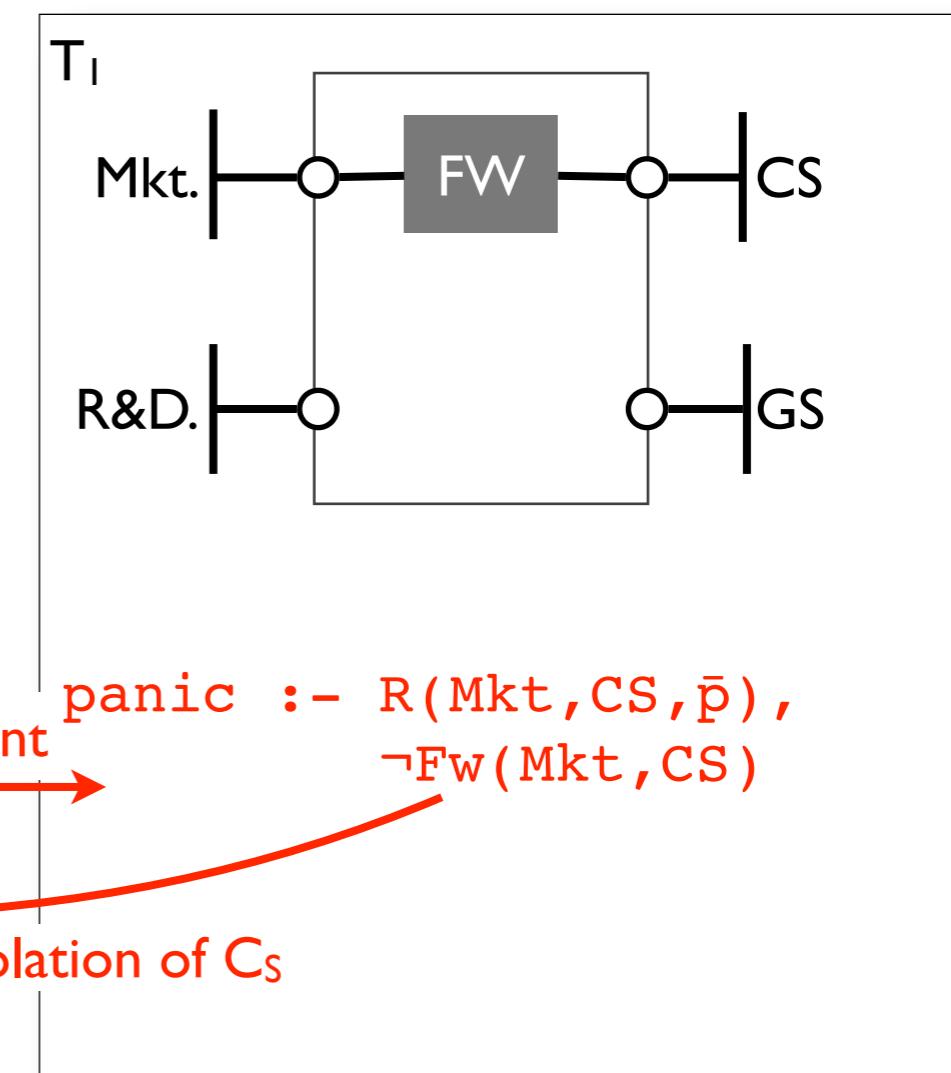
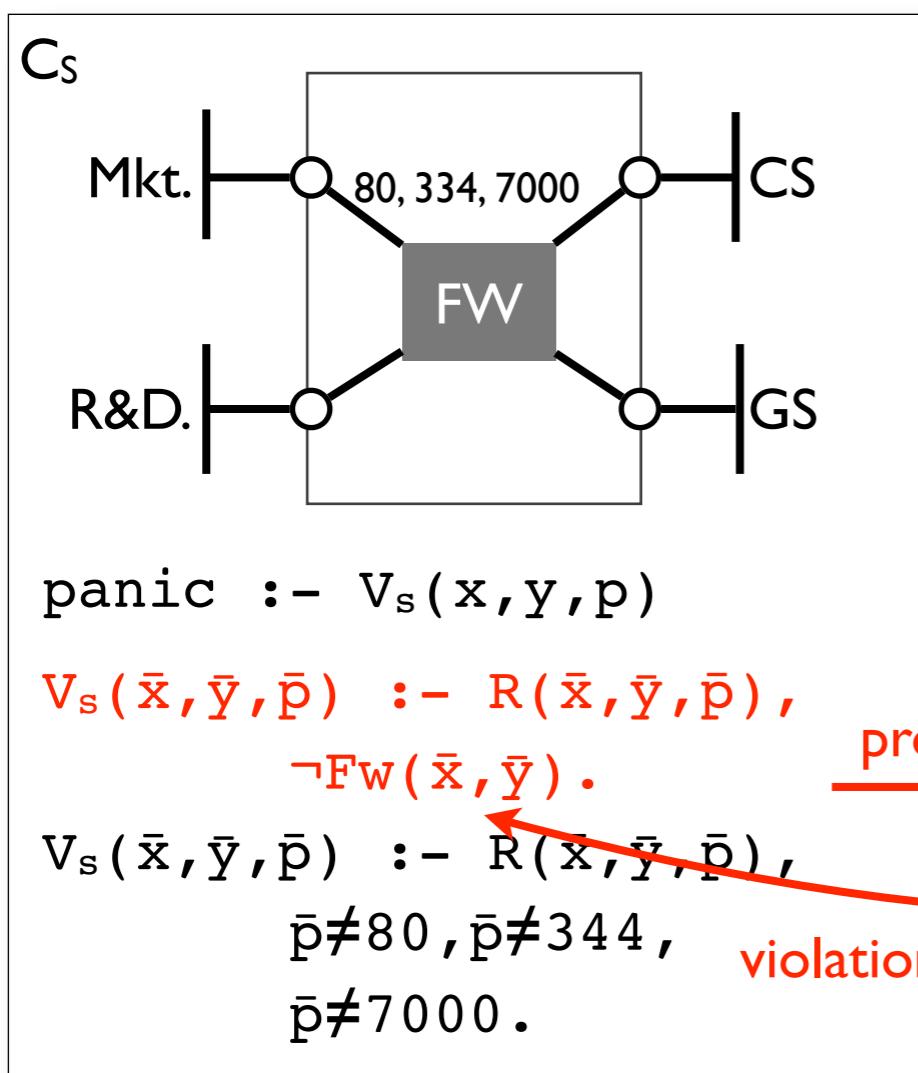
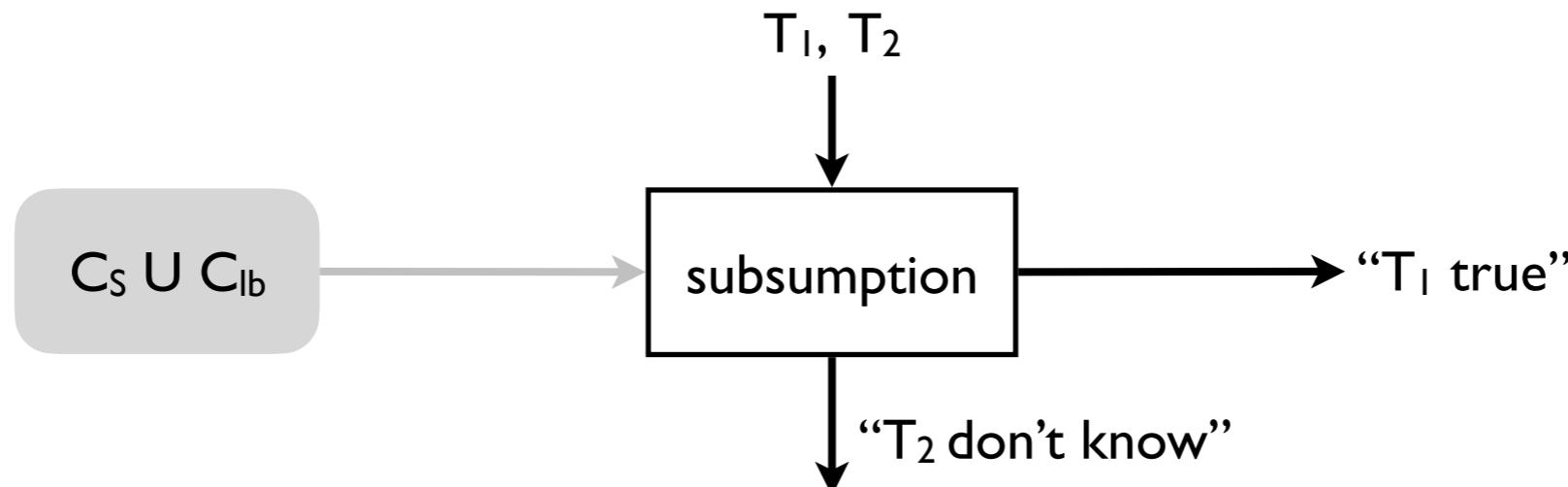
category (i) test: using only constraints



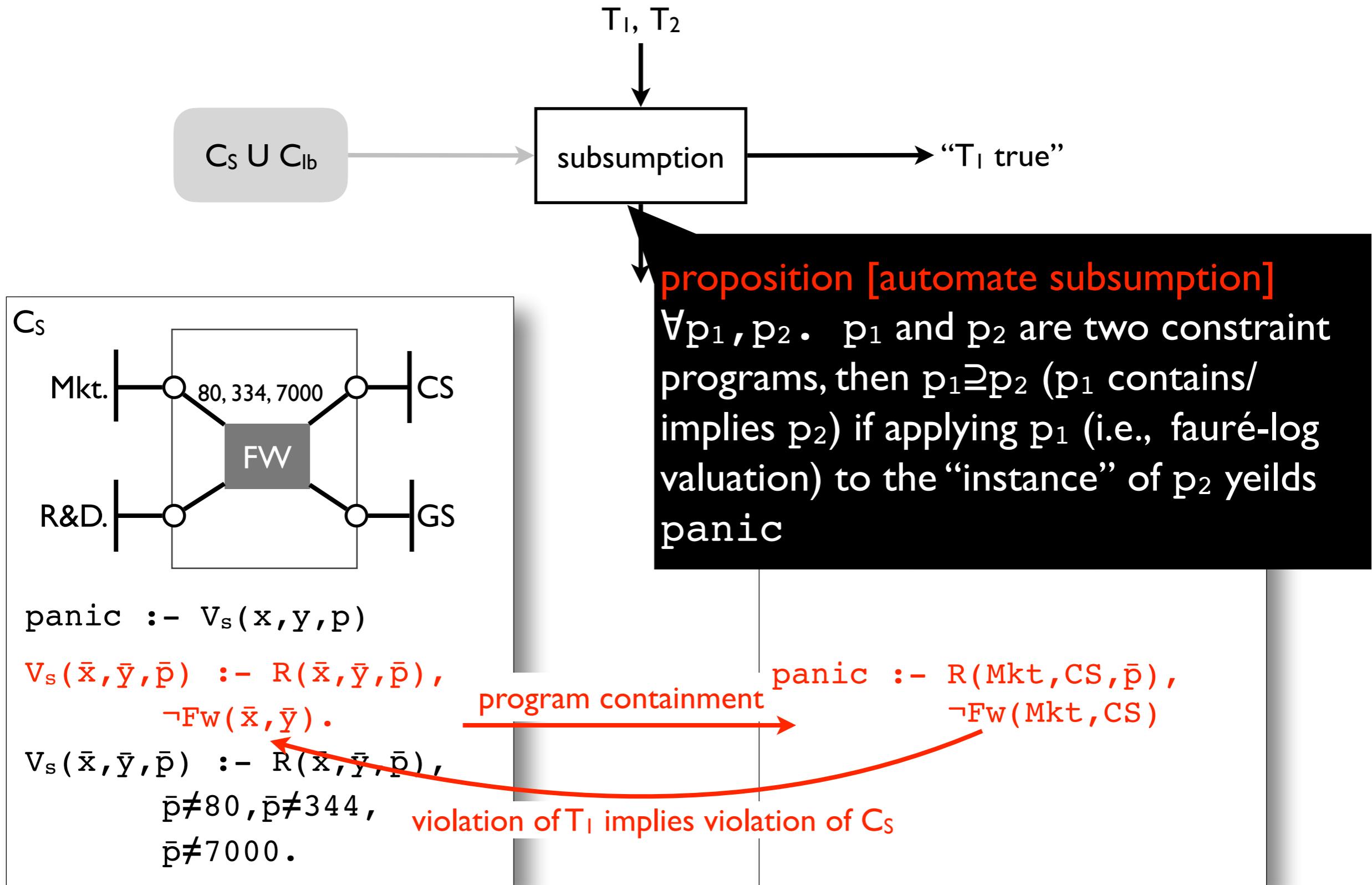
program containment



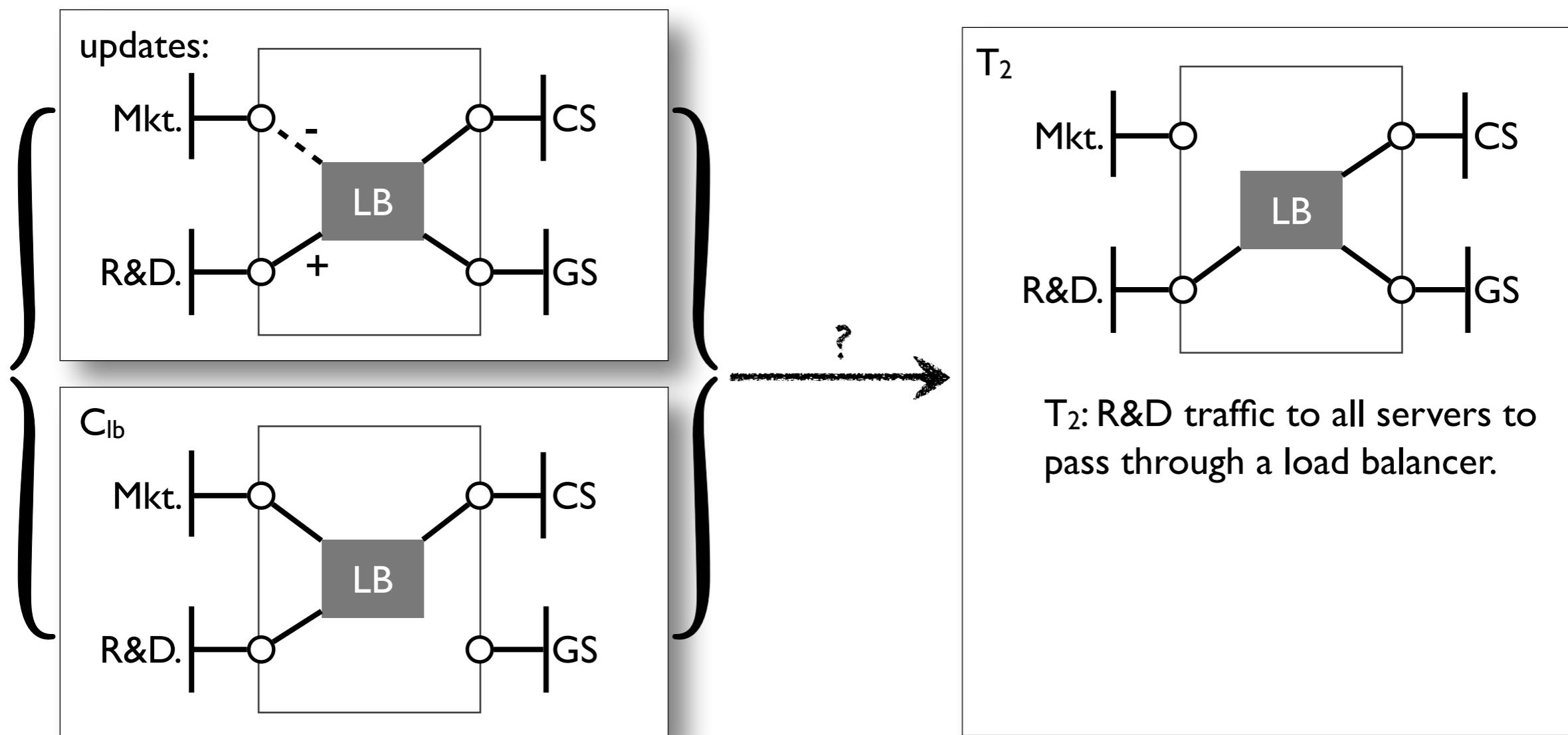
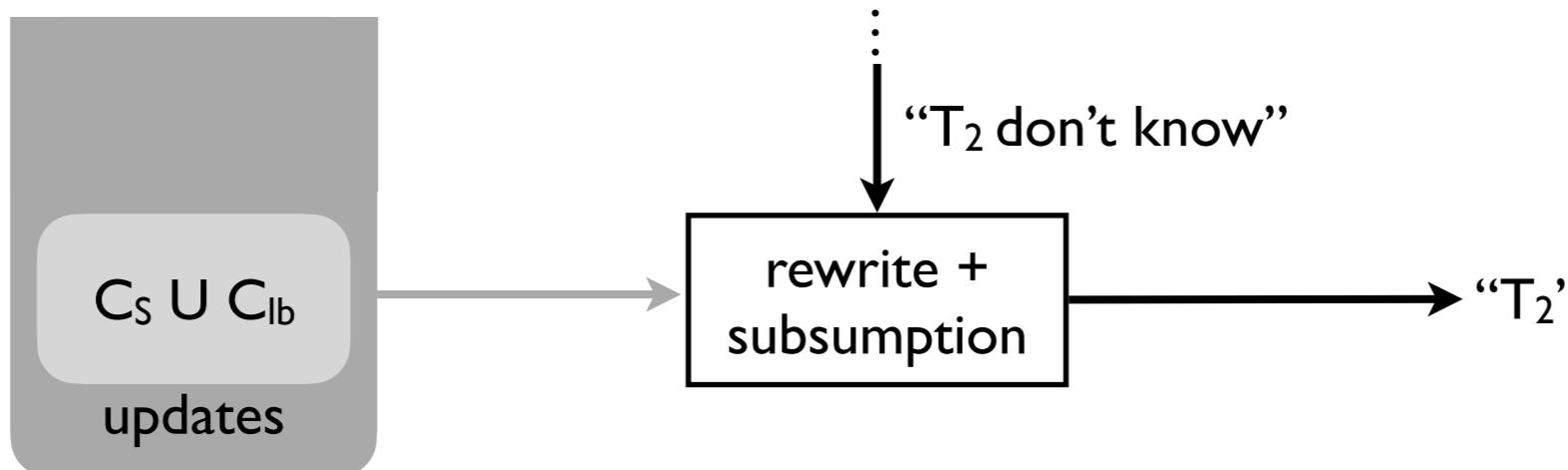
category (i) test: using only constraints



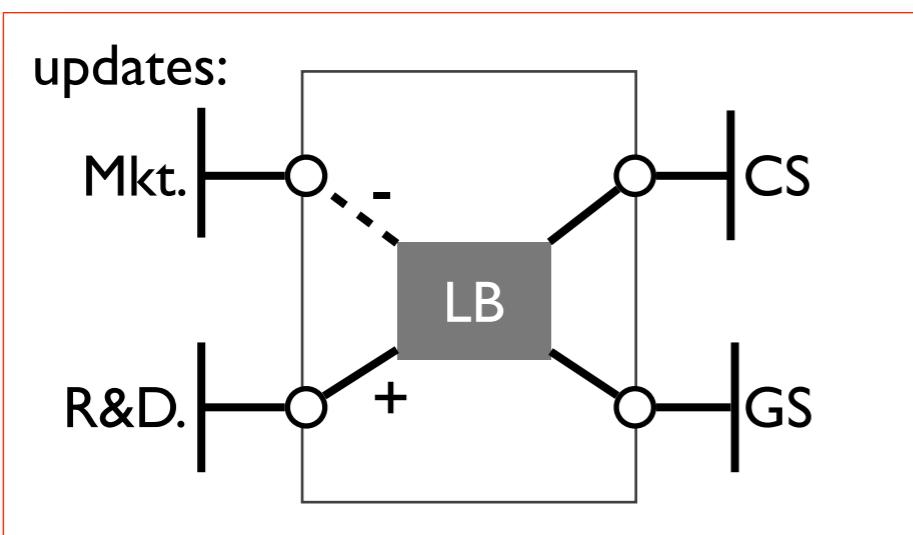
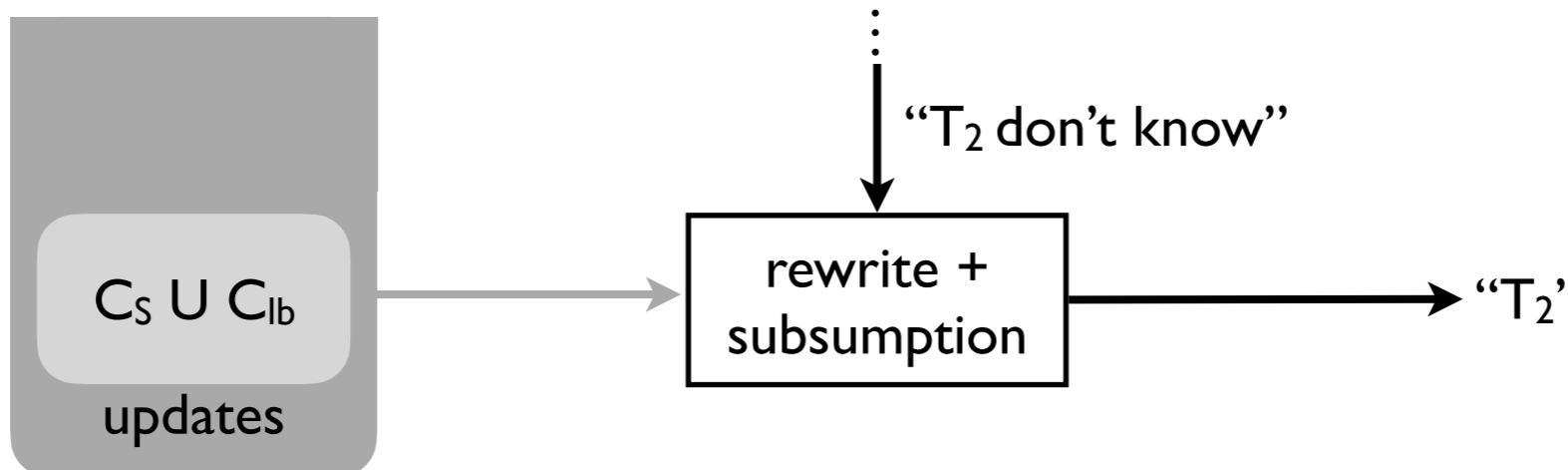
category (i) test: using only constraints



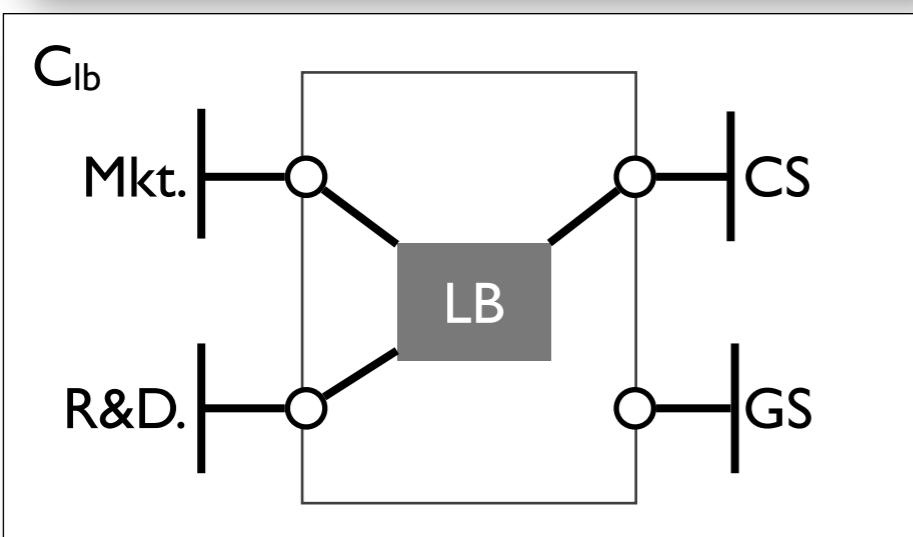
category (ii) test: using constraints & updates



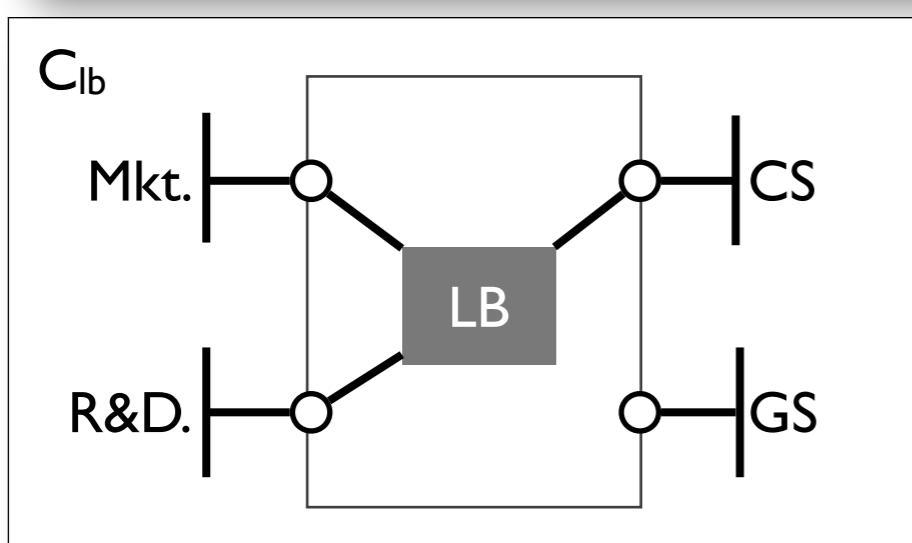
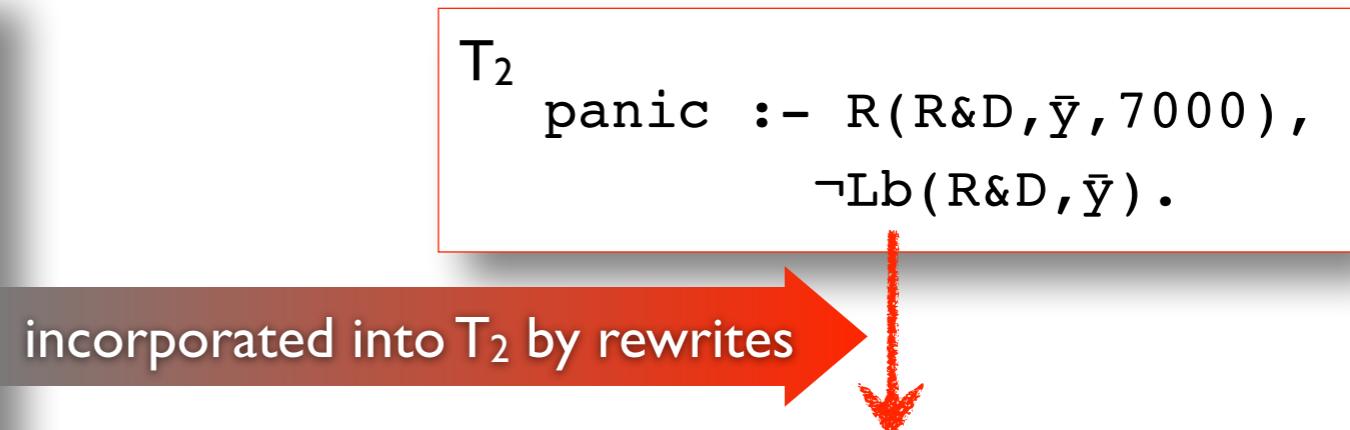
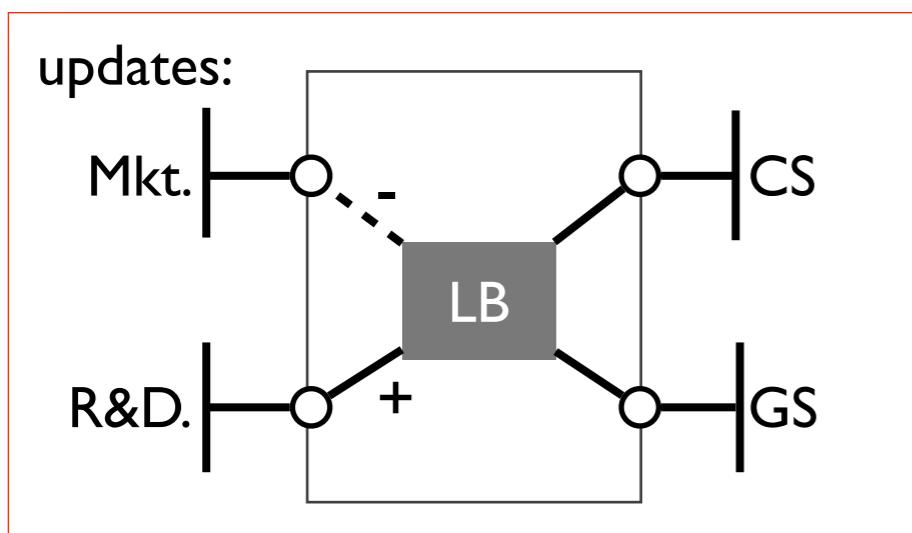
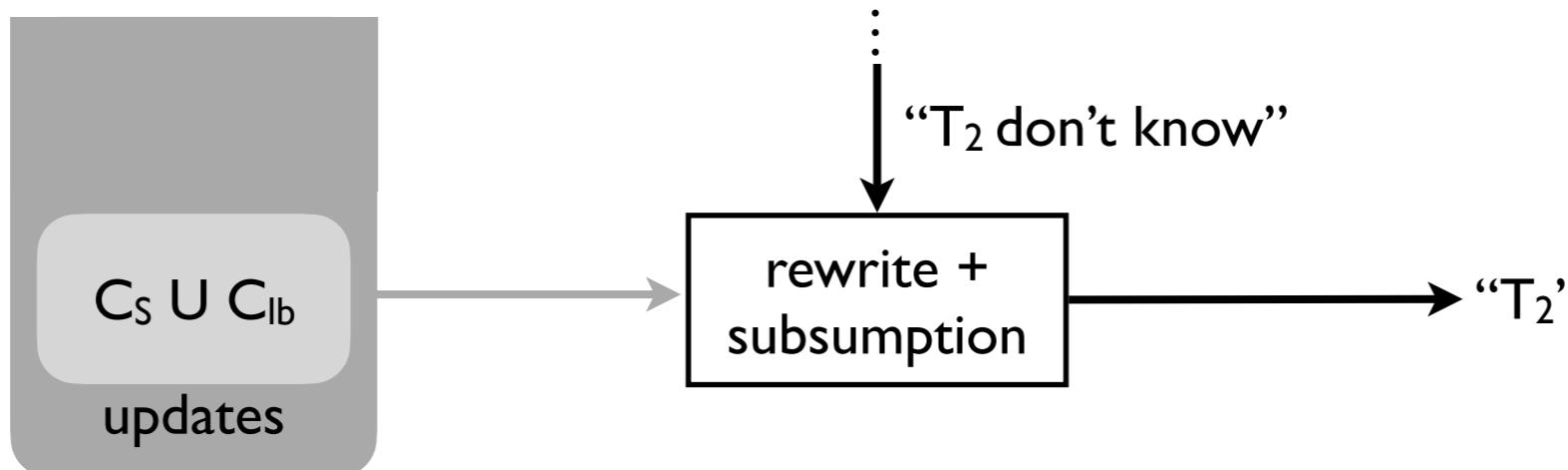
category (ii) test: using constraints & updates



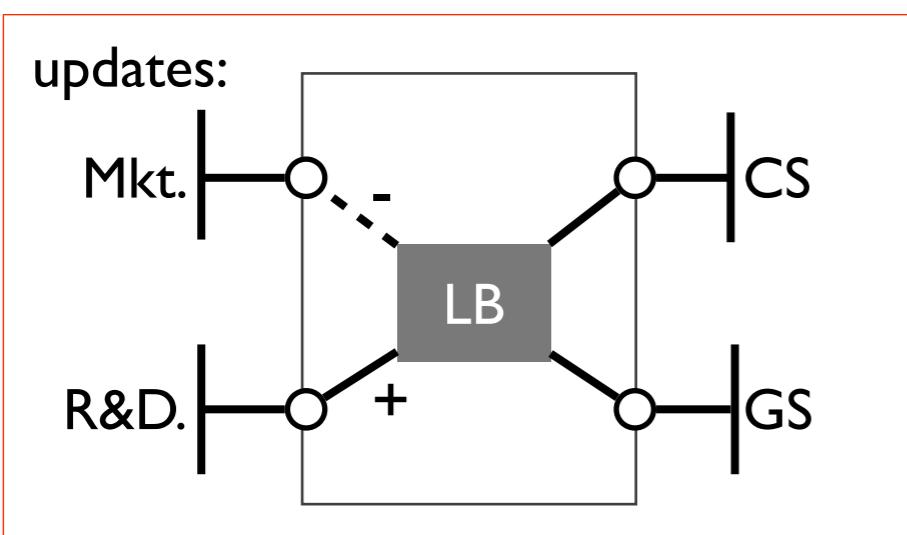
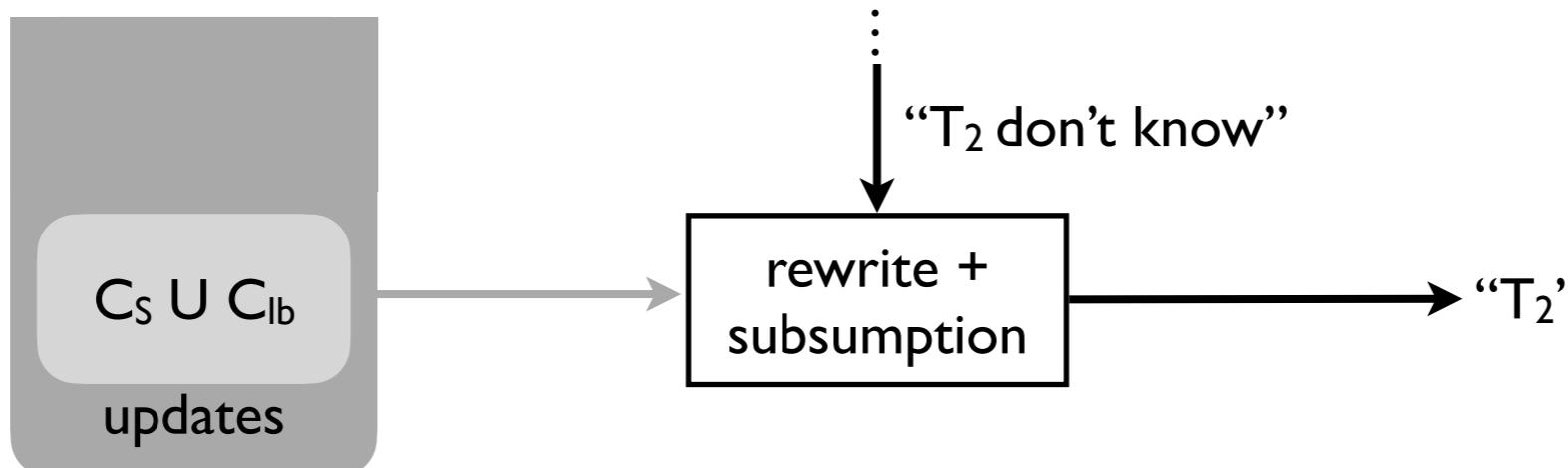
T₂ panic :- R(R&D, y, 7000),
 neg(Lb(R&D, y)).



category (ii) test: using constraints & updates



category (ii) test: using constraints & updates



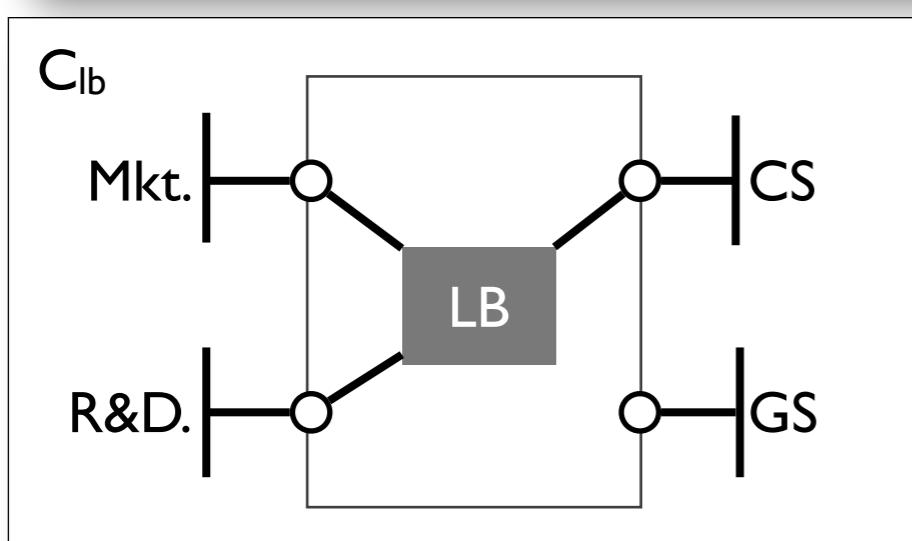
incorporated into T₂ by rewrites

T₂
 $\text{panic} :- \text{R}(\text{R\&D}, \bar{y}, 7000),$
 $\neg \text{Lb}(\text{R\&D}, \bar{y}).$

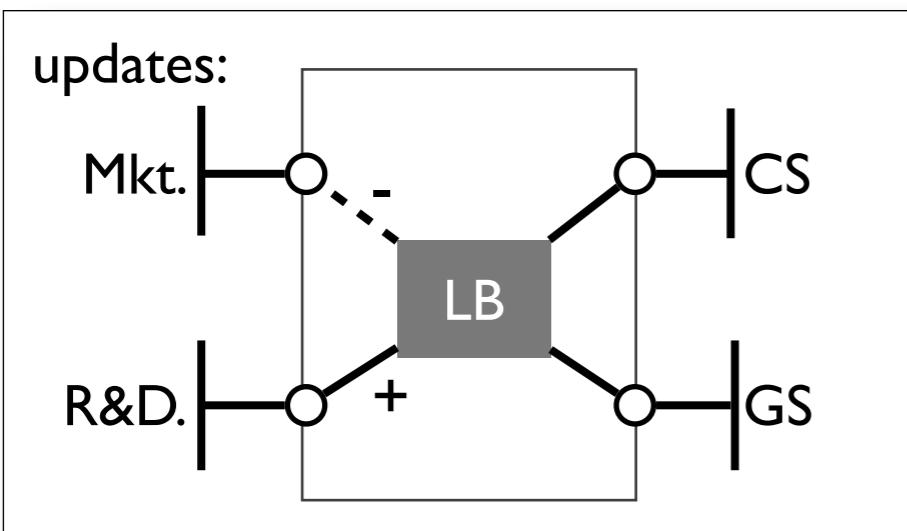
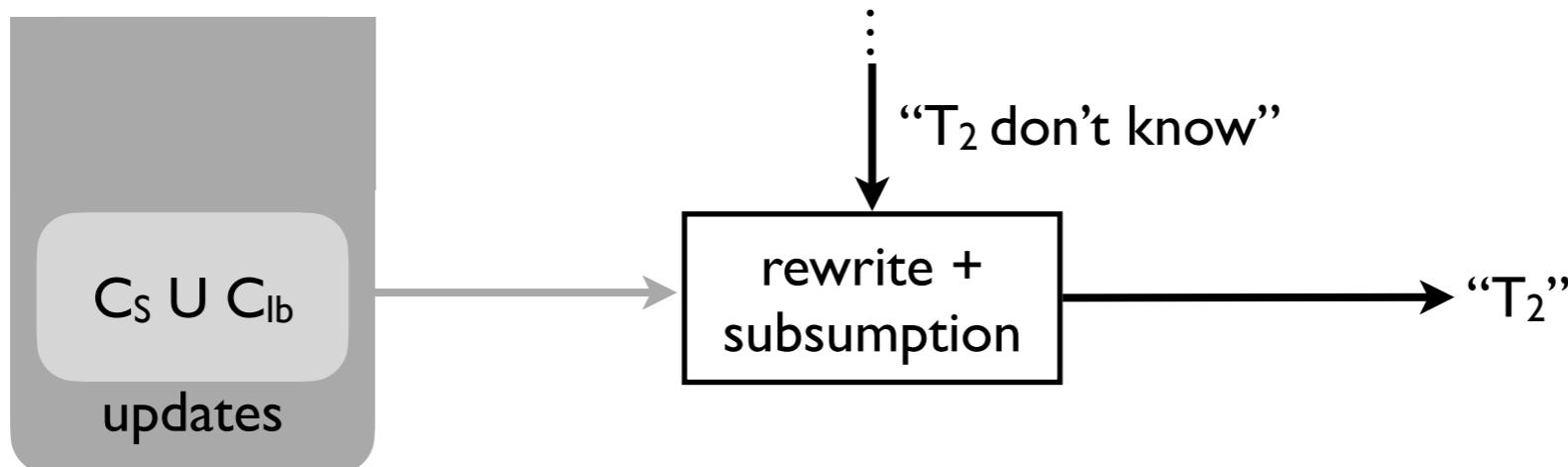
T_{2'}
/ add (R&D,GS) to LB */*
 $\text{Lb}(\text{R\&D}, \text{GS}).$
 $\text{Lb}_1(\bar{x}, \bar{y}) :- \text{Lb}(\bar{x}, \bar{y}).$

/ delete (Mkt,CS) LB */*
 $\text{Lb}_2(\bar{x}, \bar{y}) :- \text{Lb}_1(\bar{x}, \bar{y}) [\bar{x} \neq \text{Mkt}]$
 $\text{Lb}_2(\bar{x}, \bar{y}) :- \text{Lb}_1(\bar{x}, \bar{y}) [\bar{y} \neq \text{CS}]$

*/*panic after updates */*
 $\text{panic} :- \text{R}(\text{R\&D}, \bar{y}, 7000),$
 $\neg \text{Lb}_2(\text{R\&D}, \bar{y}).$



category (ii) test: using constraints & updates



C_{lb}

```

panic :- Vt(x,y,p)
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), ȳ ≠ Mkt, ȳ ≠ R&D.
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), ¬Lb(ȳ,CS)
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), Ȳ ≠ 7000
  
```

T_2

```

panic :- R(R&D,ȳ,7000),
        ¬Lb(R&D,ȳ).
  
```

T_2'

```

/* add (R&D,GS) to LB */
Lb(R&D,GS).
Lb1(ȳ,ȳ) :- Lb(ȳ,ȳ)
  
```

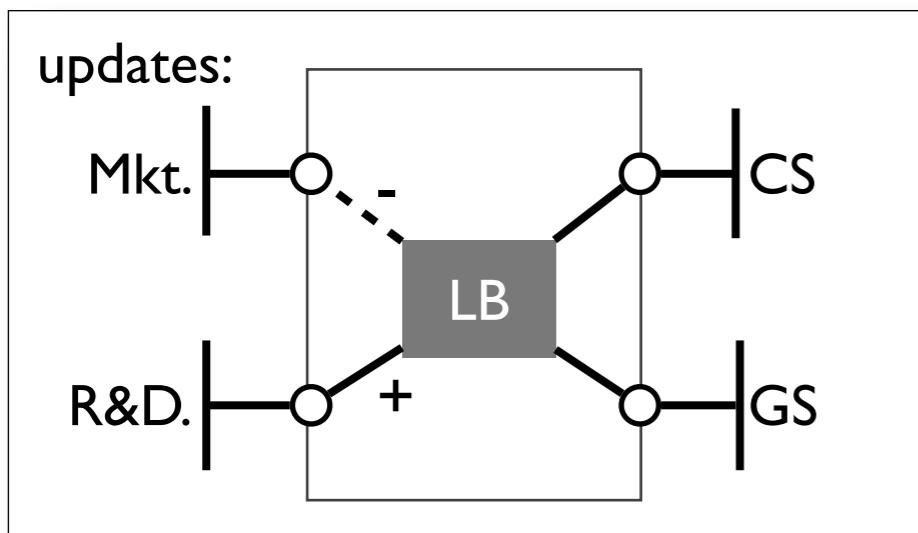
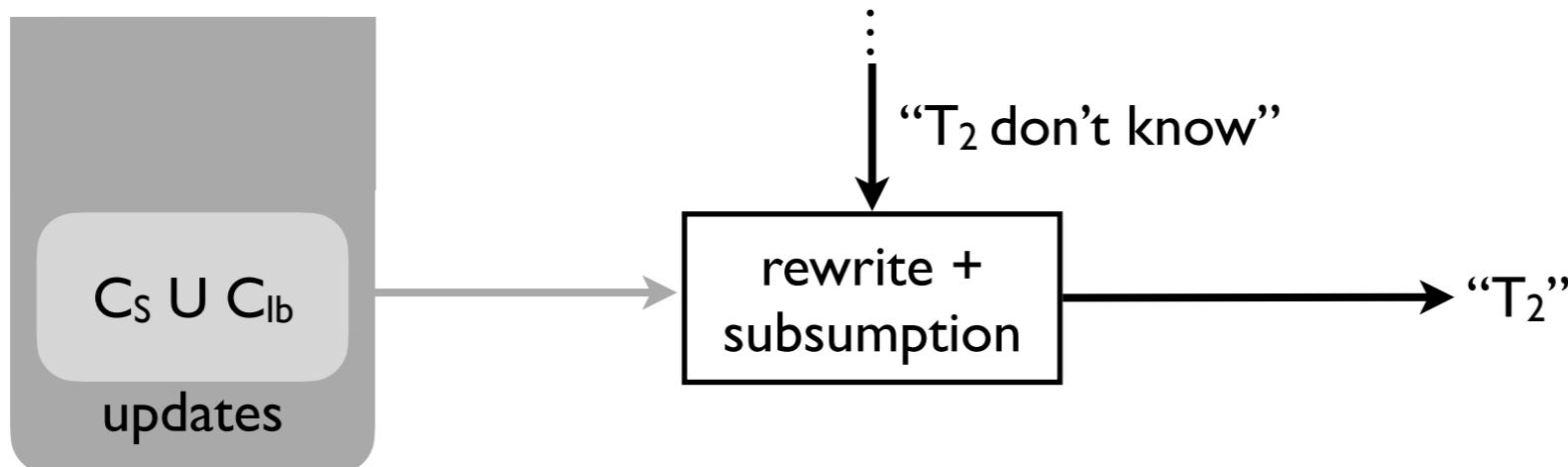
```

/* delete (Mkt,CS) LB */
Lb2(ȳ,ȳ) :- Lb1(ȳ,ȳ) [ȳ ≠ Mkt]
Lb2(ȳ,ȳ) :- Lb1(ȳ,ȳ) [ȳ ≠ CS]
  
```

```

/*panic after updates*/
panic :- R(R&D,ȳ,7000),
        ¬Lb2(R&D,ȳ)
  
```

category (ii) test: using constraints & updates



C_{lb}

```

panic :- Vt(x,y,p)
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), ȳ ≠ Mkt, ȳ ≠ R&D.
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), ¬Lb(ȳ,CS)
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), Ȳ ≠ 7000
  
```

subsumes →

T_2

```

panic :- R(R&D,ȳ,7000),
¬Lb(R&D,ȳ).
  
```

T_2'

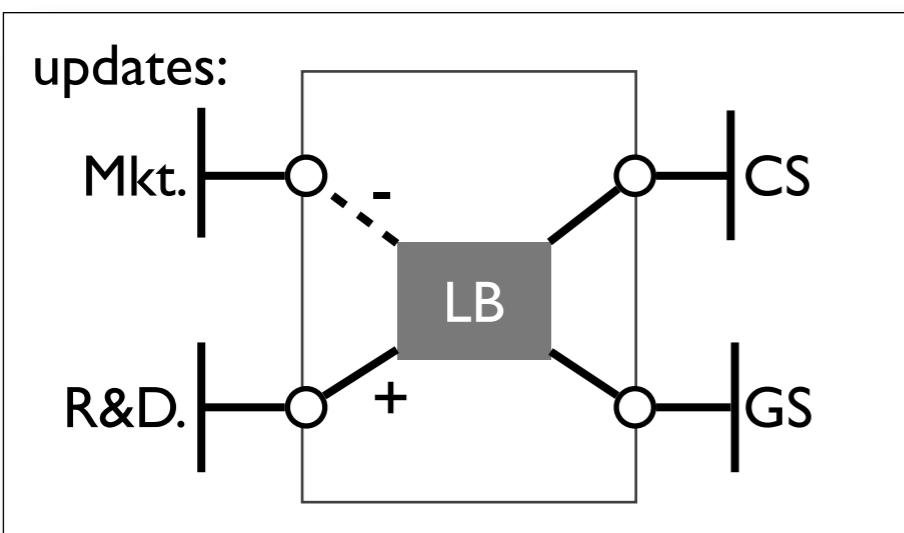
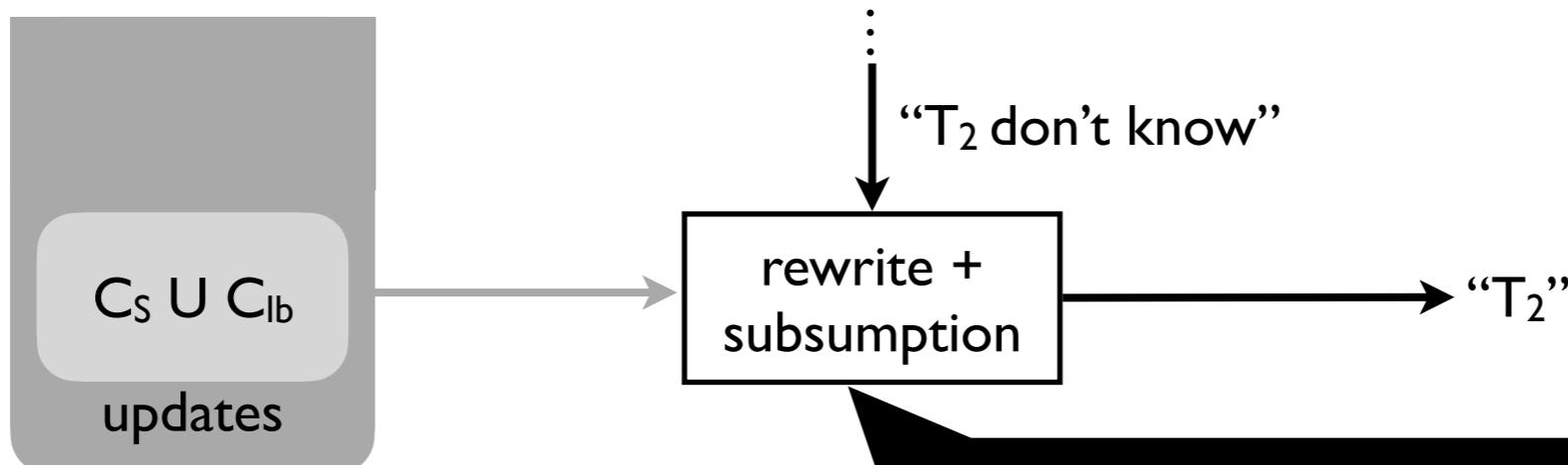
```

/* add (R&D,GS) to LB */
Lb(R&D,GS).
Lb1(ȳ,ȳ) :- Lb(ȳ,ȳ)

/* delete (Mkt,CS) LB */
Lb2(ȳ,ȳ) :- Lb1(ȳ,ȳ) [ȳ ≠ Mkt]
Lb2(ȳ,ȳ) :- Lb1(ȳ,ȳ) [ȳ ≠ CS]

/* panic after updates */
panic :- R(R&D,ȳ,7000),
¬Lb2(R&D,ȳ)
  
```

category (ii) test: using constraints & updates



proposition

given a constraint C and an update U,
incorporate U into C by rewriting C to
C': C holds after the update U iff C'
holds before the update

C_{lb}

```
panic :- Vt(x,y,p)
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), ȳ ≠ Mkt, ȳ ≠ R&D.
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), ¬Lb(ȳ,CS)
Vt(ȳ,CS,Ȳ) :- R(ȳ,CS,Ȳ), Ȳ ≠ 7000
```

subsumes

```
Lb(R&D,GS).
Lb1(ȳ,ȳ) :- Lb(ȳ,ȳ)

/* delete (Mkt,CS) LB */
Lb2(ȳ,ȳ) :- Lb1(ȳ,ȳ)[ȳ ≠ Mkt]
Lb2(ȳ,ȳ) :- Lb1(ȳ,ȳ)[ȳ ≠ CS]

/*panic after updates*/
panic :- R(R&D,ȳ,7000),
        ¬Lb2(R&D,ȳ)
```

preliminary result

practical implementation in SQL

- shallow embedding of fauré-log in PostgreSQL + Z3

evaluation

- realistic topology (inferred from BGP announcements)
- synthetic link failures
- representative queries
 - q₄-q₅ (all pair-wise reachability), q₆-q₈ (various failure patterns)

#prefix	q ₄ – q ₅		q ₆			q ₇			q ₈		
	sql	sql	Z3	#tuples	sql	Z3	#tuples	sql	Z3	#tuples	
1000	0.625s	0.85s(0.11%)	796.35s	42425	0.08s(22.86%)	0.27s	16	0.15s(1.17%)	12.64s	828	
10000	5.75s	8.96s	-	418224	0.27s(7.33%)	3.41s	194	1.8s(1.27%)	137.05s	8706	
100000	54.85s	113.48s	-	4435862	1.66s(6.18%)	25.22s	1387	34.67s(1.71%)	1941.04s	86360	
922067	816.4s	4169.02s	-	46503247	11.1s(3.71%)	288.17s	16490	267.05s	-	858180	

recap — partial analysis

classical network analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

a departure from the
complete approach

uncertain environments

query

loss-less modeling

uncorrupted answer

partial knowledge

query

relative-complete
verification

correct answer

“don’t know, need more information”

recap — realization

classical network analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

Fauré

uncertain environments
as c-tables

Fauré-log

loss-less modeling

uncorrupted answer

partial knowledge
as fauré-log

Fauré-log

relative-complete
verification

correct answer

“don’t know, need more information”

recap — realization

classical network analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

Fauré

uncertain environments
as **c-tables**

Fauré-log

query c-tables

loss-less modeling

uncorrupted answer

partial knowledge
as **fauré-log**

Fauré-log

relative-complete
verification

correct answer

“don’t know, need more information”

recap — realization

classical network analysis

definite knowledge
of the network

query

comprehensive
evaluation

decisive answer

Fauré

uncertain environments
as **c-tables**

Fauré-log

query c-tables

loss-less modeling

uncorrupted answer

partial knowledge
as **fauré-log**

Fauré-log

simplify static analysis

relative-complete
verification

correct answer

“don’t know, need more information”

thank you

<https://github.com/ravel-net/Faure>