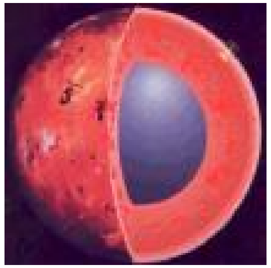
Data Exchange: Computing Cores in Polynomial Time



Georg Gottlob

Oxford University

Joint work with Alan Nash, UCSD

This talk is based on two recent papers:

G....: Computing Cores for Data Exchange: New Algorithms and Practical Solutions PODS 2005

G.... & Nash: Data Exchange: Computing Cores in Polynomial Time. Submitted to PODS 2006.

Detailed joint extended version of both papers:

G.... & Nash: Efficient Core Computation in Data Exchange. Available from the authors (Draft).

Talk Structure

Introduction & basics

Computing Cores

- for weakly acyclic TGDs as target dependencies
- for EGDs and weakly acyclic TGDs as target dependencies

Further results (time permitting)

Cores

Instance:

$\{ \; p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) \; \}$

Logical meaning

$\exists X, Y, U, V$:

$p(X,Y) \;\&\; p(X,b) \;\&\; p(a,b) \;\&\; p(U,c) \;\&\; p(U,V) \;\&\; q(a,c,d)$

Cores

endomorphism h: {Y → b}

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{ p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

Cores

endomorphism h: {Y → b}

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{ p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

REDUNDANT!

$\exists X,Y\ p(X,Y)\ \&\ p(X,b)$

$\Updownarrow$

$\exists X\ p(X,b)$

Cores

endomorphism h: {Y → b}

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{ ~~p(X,Y)~~, p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

REDUNDANT!

$\exists X,Y\ p(X,Y)$

$\Uparrow$

$\exists X\ p(X,b)$

Cores

endomorphism h: {Y → b}

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

⇔

h(I) = { p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

Cores

endomorphism h: {Y → b}

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

⇔

h(I) = { p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

X → a

V → c

h(I) can be further reduced by endomorphism g: {X → a, V→ c}

Cores

endomorphism h: {Y → b}

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

⇔

h(I) = { p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{ p(a,b), p(U,c), q(a,c,d) }

h(I) can be further reduced by endomorphism g: {X → a, V→ c}

Cores

$I = \{ p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) \}$

$\Leftrightarrow$

$h(I) = \{ p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) \}$

$\Leftrightarrow$

$f(I) = g(h(I)) = g \circ h(I) = \{ p(a,b), p(U,c), q(a,c,d) \}$

endomorphism f: {X → a, Y→b, V→ c}

Cores

$$I \;=\; \{\, p(X,Y),\ p(X,b),\ p(a,b),\ p(U,c),\ p(U,V),\ q(a,c,d) \,\}$$

$\Leftrightarrow$

$$h(I) \;=\; \{\, p(X,b),\ p(a,b),\ p(U,c),\ p(U,V),\ q(a,c,d) \,\}$$

$\Leftrightarrow$

$$f(I) = g(h(I)) = g \circ h(I) = \{\, p(a,b),\ p(U,c),\ q(a,c,d) \,\}$$

no refinement by endomorphisms possible !

endomorphism f: {X → a, Y→b, V→ c}

Cores

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

⇔

h(I) = { p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

⇔

f(I)= g(h(I)) = g○h(I)= { p(a,b), p(U,c), q(a,c,d) }

Core(I)

unique up to variable-renaming!

endomorphism f: {X → a, Y→b, V→ c}

Blocks

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

Blocks: Connected components in the variable-graph

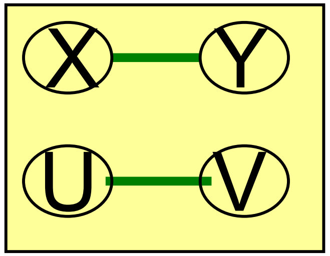
Atom-Blocks: corresponding sets of atoms

Blocks

I = { p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{X,Y} {U,V} blocksize(I)=2

Blocks: Connected components in the variable-graph



variable-graph

blocksize(I) = size of largest block of I

[Fagin, Kolaitis, Popa PODS'03]:

- Computing core(I) is NP-hard in general.
- It is tractable for bounded blocksize b:

core(I) can be computed in time $n * O(|dom(I)|^{b+2}) = O(n^{b+3})$

[G. PODS'05]

- Computing core(I) tractable for bounded treewidth or hypertree-width of variable-graph

=> new bound: $O(n^{b/2+3})$

based on hypertree decompositions. (→ end of talk, time permitting)

Dependencies

Tuple generating dependencies TGDs:

$$\forall X\ \forall Y\ \forall Z\ p(X,Y)\ \&\ q(Y,Z) \rightarrow \exists U\ \exists V\ \ r(X,U)\ \&\ p(Z,V)$$

Equality generating dependencies EGDs:

$$\forall X\ \forall Y\ \forall Z\ p(X,Y)\ \&\ p(X,Z) \rightarrow Y{=}Z$$

We usually omit universal quantifiers…

TGDs can be cyclic in which case the Chase may not terminate

Cyclic TGD:

$$p(X,Y)\ \&\ q(Y,Z) \rightarrow \exists\ U,V\ \ r(X,U)\ \&\ p(Z,V)$$

We restrict ourselves to setting of

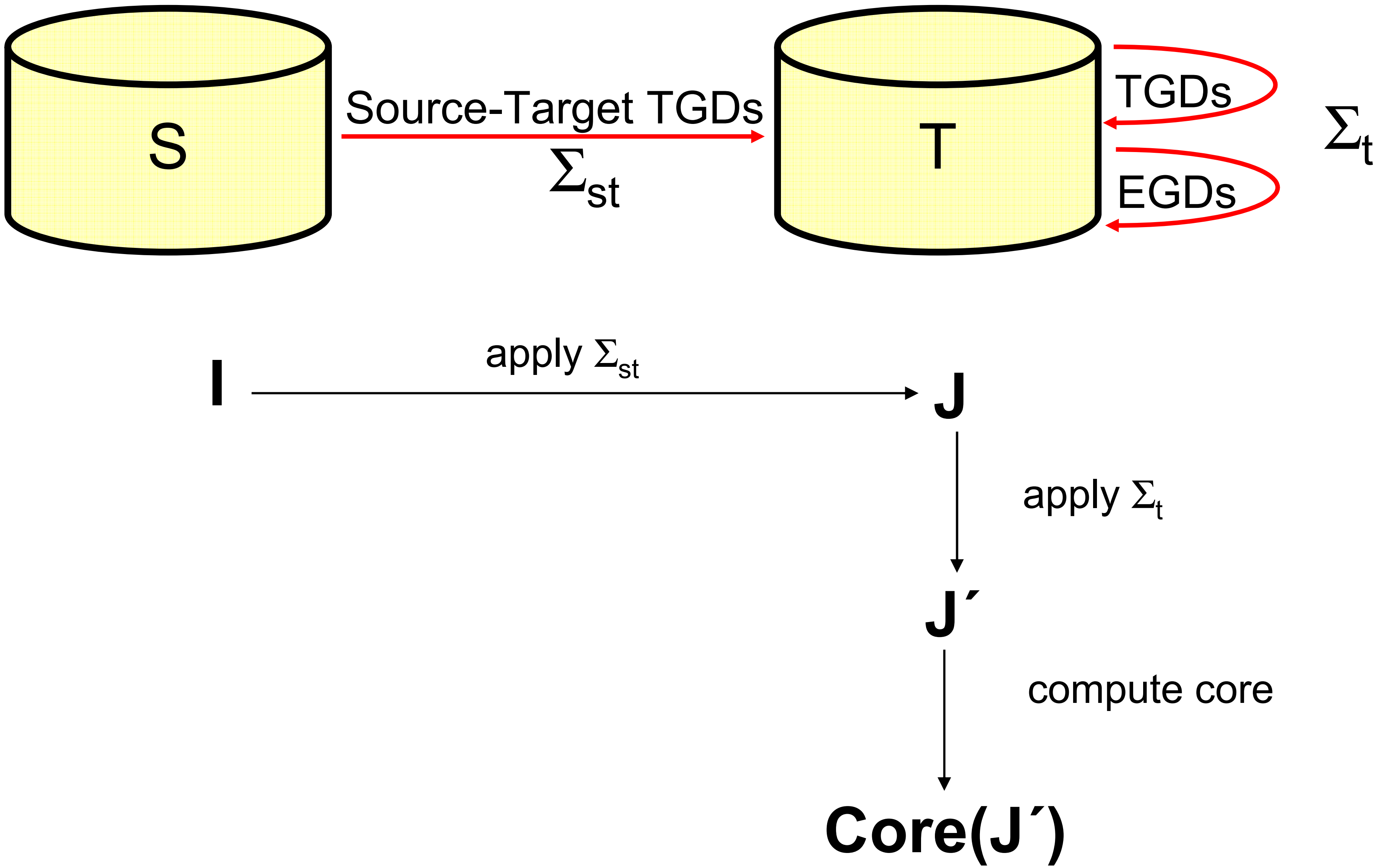
weakly acyclic sets of TGDs + arbitrary EGDs

([Fagin, Kolaitis, Popa 03], [Deutsch,Tannen 03])

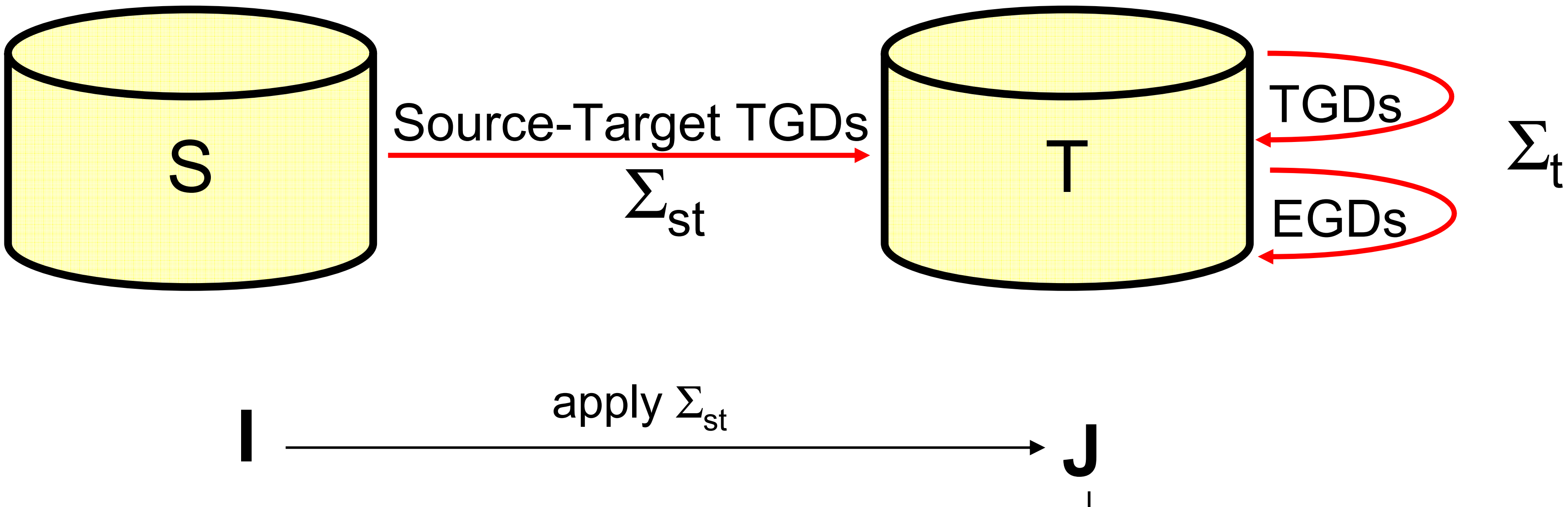
This covers the overwhelming part of relevant constraints:

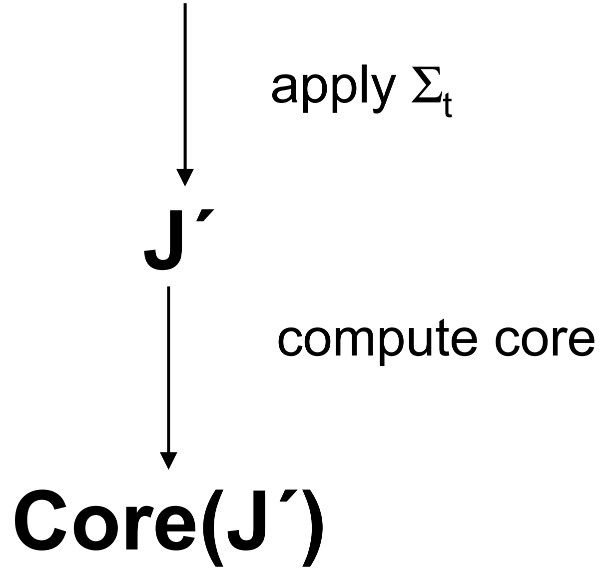
- Functional dependencies
- w.a. inclusion dependencies
- referential integrity
- foreign key constraints…
- …

Data Exchange



Data Exchange

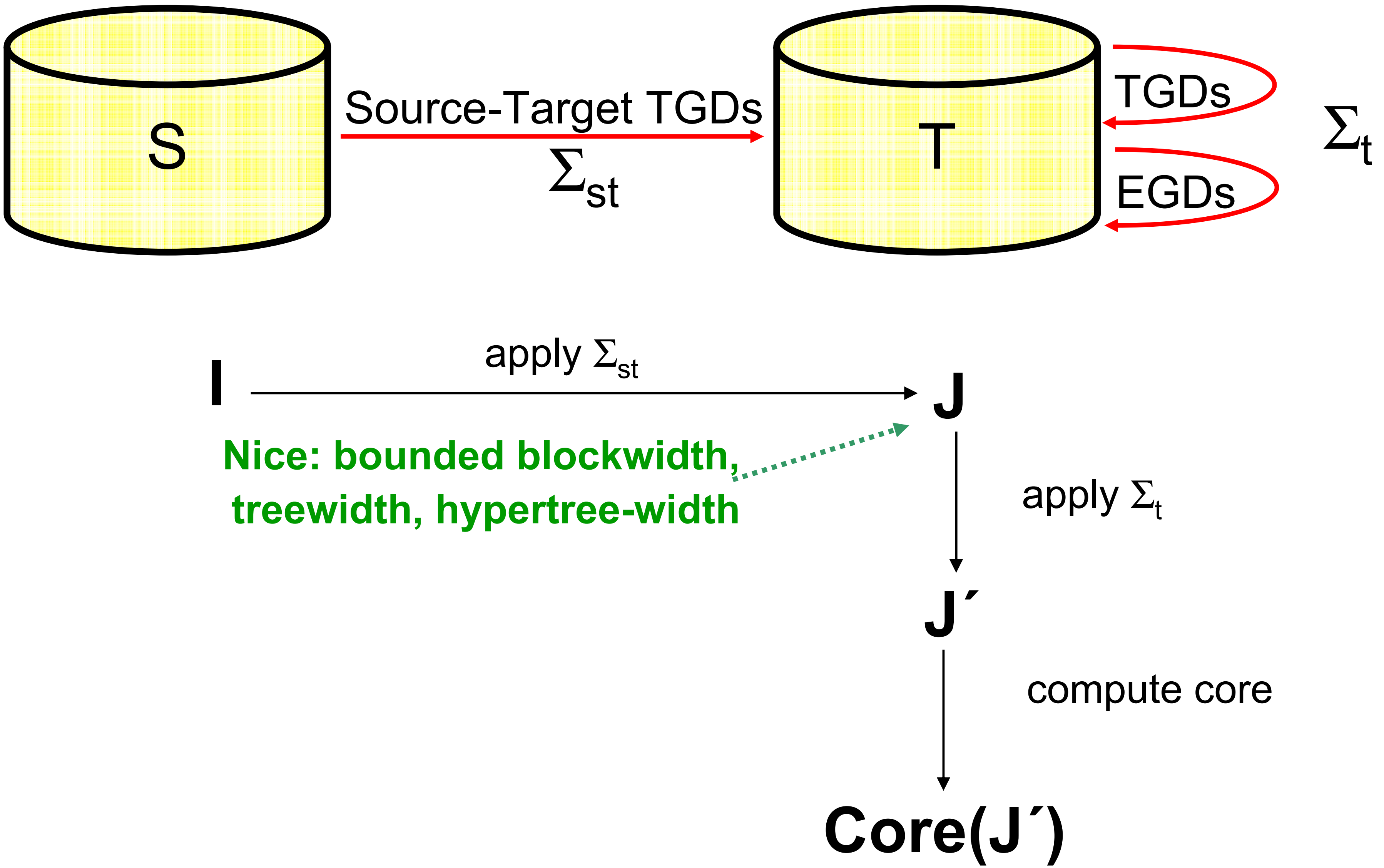




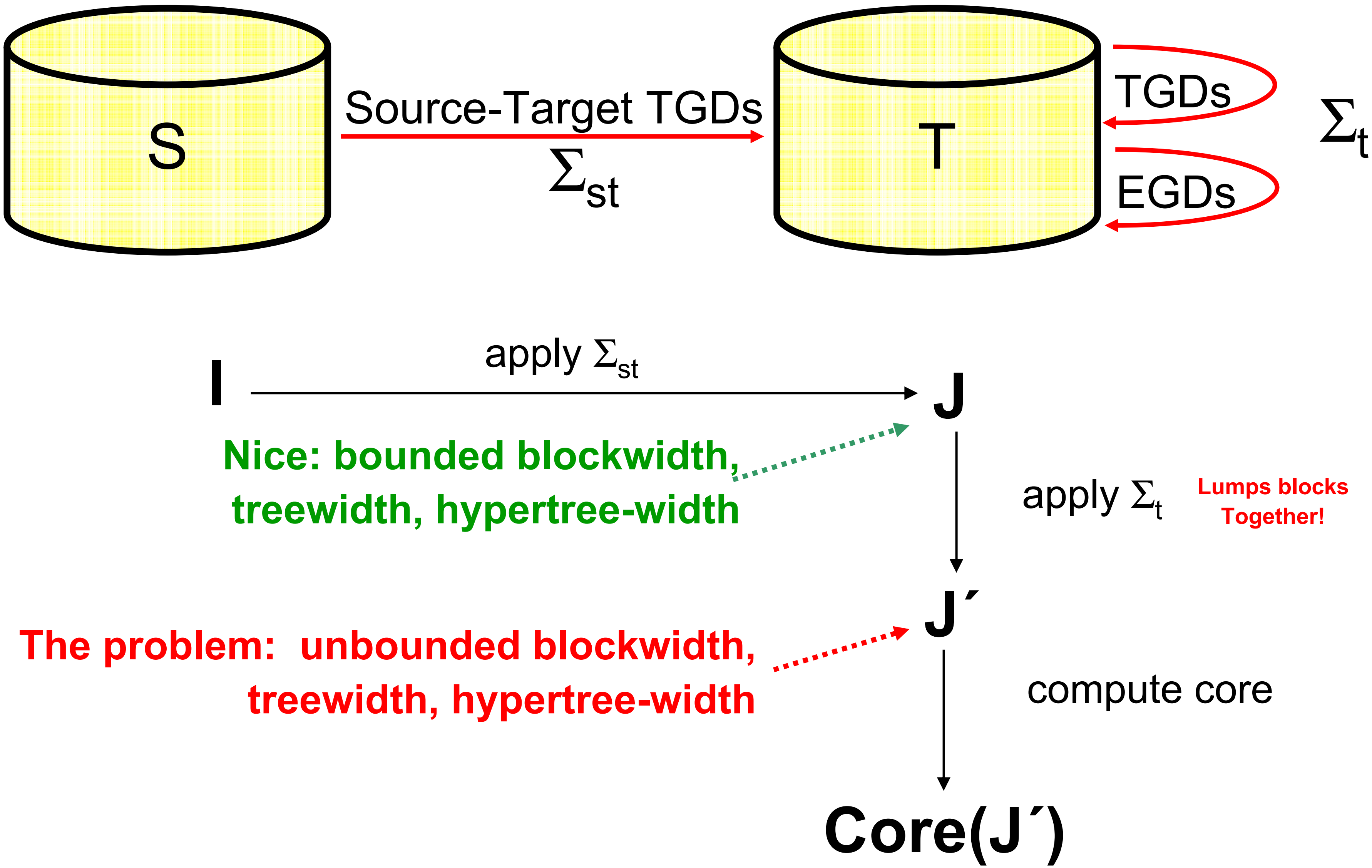
Open Problem [F.K.P.03]:

Can Core(J') be computed in polytime if Σ_t consists of w.a. TGDs and EGDs?

Data Exchange



Data Exchange



TGDs (even full TGDs) destroy blockwidth

{ p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{X,Y}   {U,V}   blocksize(I)=2

TGDs (even full TGDs) destroy blockwidth

{ p(X,Y), p(X,b), p(a,b), p(U,c), p(U,V), q(a,c,d) }

{X,Y} {U,V} blocksize=2

TGD: p(R,S) & p(R',S') → p(R,R')

p(X,U)

{X,Y,U,V} blocksize=4

Efficient Core Computation

- Fagin, Kolaitis, and Popa [PODS 2003]
  - Target dependencies are empty or contain only EGDs
    (blocks method and rigidity)

Efficient Core Computation

- Fagin, Kolaitis, and Popa [PODS 2003]
  - Target dependencies are empty or contain only EGDs
    (blocks method and rigidity)
- G….. [PODS 2005]
  - Target dependencies without existential quantification (= full)
  - Target dependencies with a single atom in the premise
    (they preserve hypertree-width)

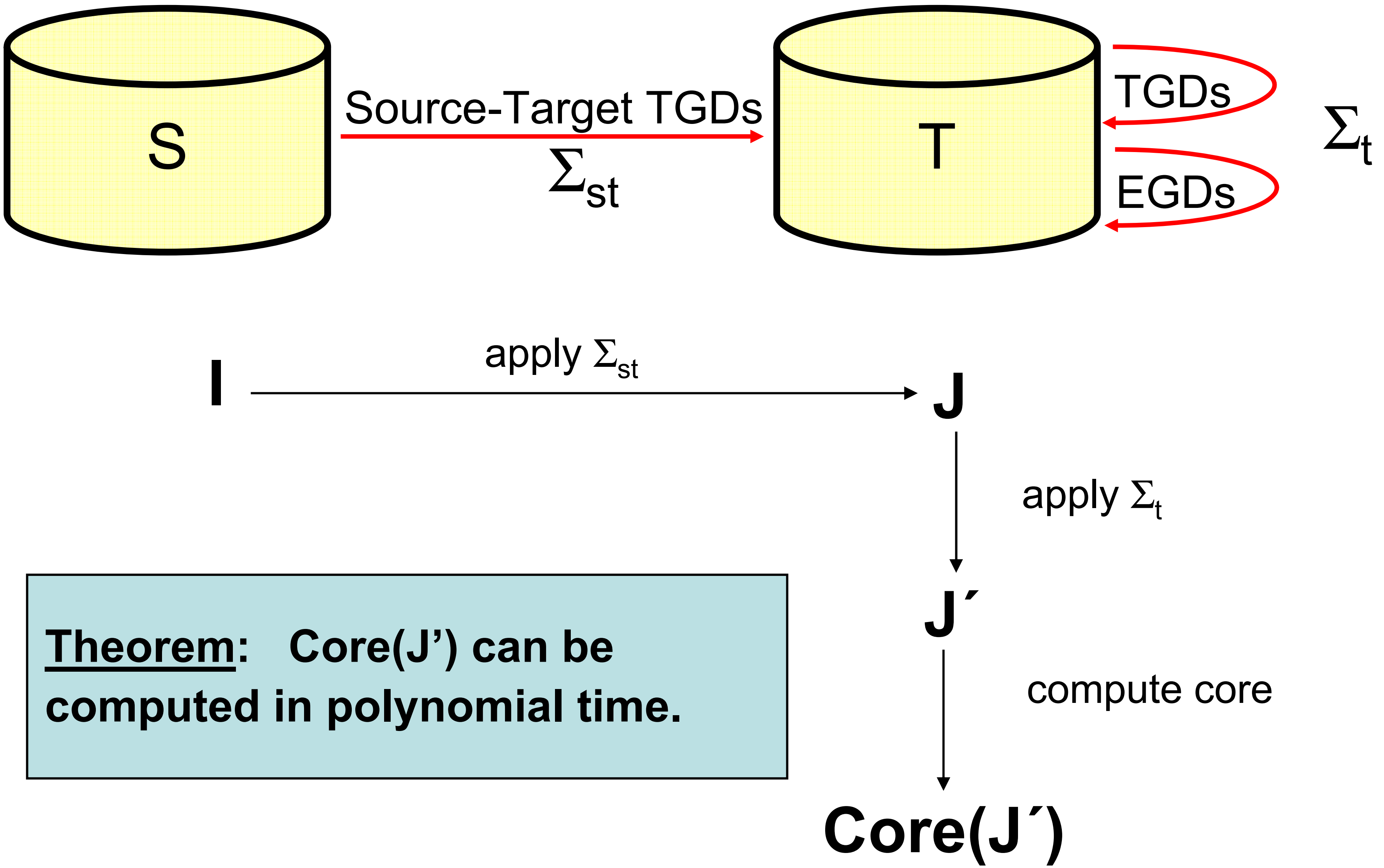
Efficient Core Computation

- Fagin, Kolaitis, and Popa [PODS 2003]
  - Target dependencies are empty or contain only EGDs
    (blocks method and rigidity)
- G….. [PODS 2005]
  - Target dependencies without existential quantification (= full)
  - Target dependencies with a single atom in the premise
    (they preserve hypertree-width)
- G…., Nash [PODS 2006]
  - General target dependencies for which the chase is known to terminate
  (weakly acyclic or new conditions)

Efficient Core Computation

- Fagin, Kolaitis, and Popa [PODS 2003]
  - Target dependencies are empty or contain only EGDs
    (blocks method and rigidity)
- G….. [PODS 2005]
  - Target dependencies without existential quantification (= full)
  - Target dependencies with a single atom in the premise
    (they preserve hypertree-width)
- G…., Nash [PODS 2006]
  - General target dependencies for which the chase is known to terminate (weakly acyclic or new conditions)
- In summary: Whenever we know we can compute universal solutions in PTIME, we know we can compute their cores in PTIME

Data Exchange



<u>Theorem</u>: Core(J’) can be computed in polynomial time.