Fancy types for provenance

James Cheney
University of Edinburgh

The provenance crisis

The provenance crisis

SEPTEMBER 9, 2008

UAL Shares Fall as Old Story Surfaces Online

The mysterious appearance on the Internet of a nearly six-year-old news story about UAL Corp.'s 2002 bankruptcy-court filing caused investors to dump the stock Monday.

After trading near $12.50 a share early Monday, stock in United Airlines' parent quickly fell to $3 on the Nasdaq Stock Market on heavy volume before trading was halted and the company issued a statement saying that reports of a new Chapter 11 filing were "completely untrue."

Once trading resumed 90 minutes later, UAL shares rebounded, but they still closed off 11% for the day at $10.92. Nasdaq, a unit of Nasdaq OMX Group Inc., ...

The provenance crisis

SEPTEMBER 9, 2008

UAL Shares Fall as Old Story Surfaces Online

The mysterious appearance on the Internet of a nearly six-year-old news story about UAL Corp.'s 2002 bankruptcy-court filing caused investors to dump the stock Monday.

After trading near $12.50 a share early Monday, stock in United Airlines' parent quickly fell to $3 on the Nasdaq Stock Market on heavy volume before trading was halted and the company issued a statement saying that reports of a new Chapter 11 filing were "completely untrue."

Once trading resumed 90 minutes later, UAL shares rebounded, but they still closed off 11% for the day at $10.92. Nasdaq, a unit of Nasdaq OMX Group Inc., ...

The provenance crisis

UAL Shares Fall as Old Story Surfaces Online - WSJ.com - Mozilla Firefox

SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

Until recently, Geoffrey Chang's career was on a trajectory most young scientists only dream about. In 1999, at the age of 28, the protein crystallographer landed a faculty position at the prestigious Scripps Research Institute in San Diego, California. The next year, in a ceremony at the White House, Chang received a 2001 *Science* paper, which described the structure of a protein called MsbA, isolated from the bacterium *Escherichia coli*. MsbA belongs to a huge and ancient family of molecules that use energy from adenosine triphosphate to transport molecules across cell membranes. These so-called ABC transporters perform many

The provenance crisis

UAL Shares Fall as Old Story Surfaces Online - WSJ.com - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://online.wsj.com/article/SB12

SEPTEMBER 9, 2008

UAL Shares Fall as Old Story Surfaces Online

SCIENTIFIC PUBLISHING

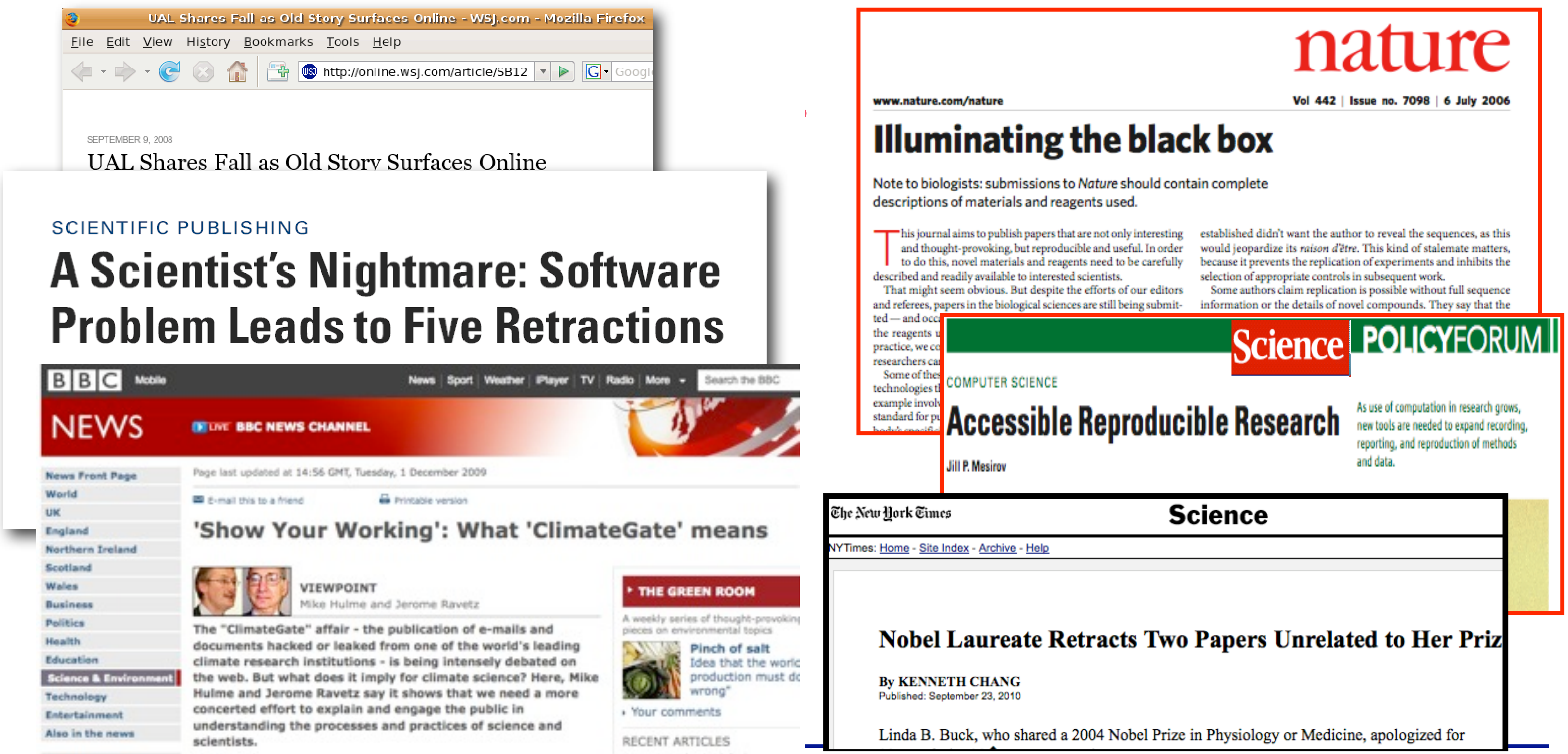
A Scientist's Nightmare: Software Problem Leads to Five Retractions

Until recently, Geoffrey Chang's career was on a trajectory most young scientists only dream about. In 1999, at the age of 28, the protein crystallographer landed a faculty position at the prestigious Scripps Research Institute in San Diego, California. The next year, in a ceremony at the White House, Chang received a 2001 *Science* paper, which described the structure of a protein called MsbA, isolated from the bacterium *Escherichia coli*. MsbA belongs to a huge and ancient family of molecules that use energy from adenosine triphosphate to transport molecules across cell membranes. These so-called ABC transporters perform many

The provenance crisis

The provenance crisis

The provenance crisis



Where to start?

Where to start?

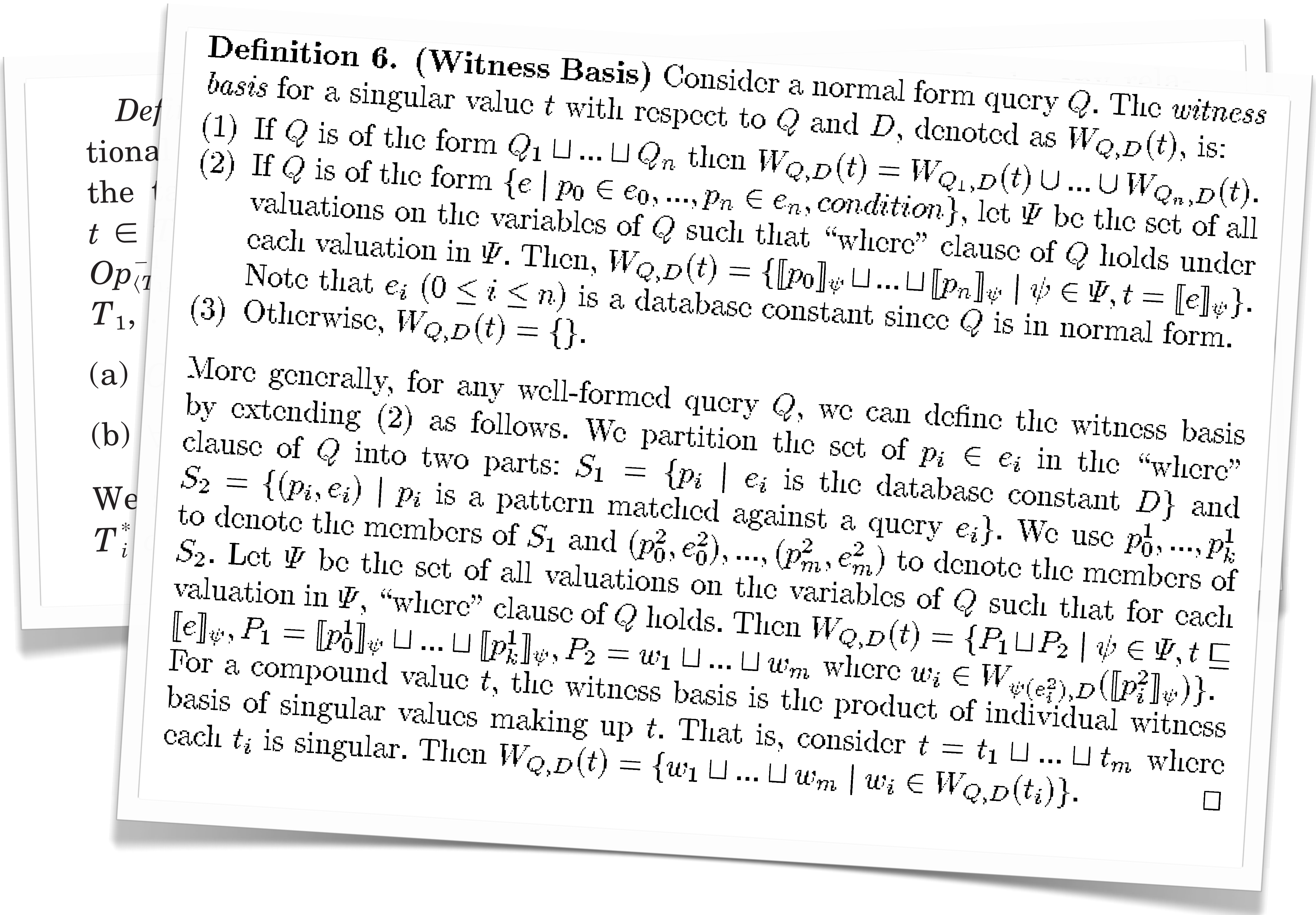
Definition 4.1 (Tuple Derivation for an Operator). Let Op be any relational operator over tables $T_1, \ldots, T_m$, and let $T = Op(T_1, \ldots, T_m)$ be the table that results from applying Op to $T_1, \ldots, T_m$. Given a tuple $t \in T$, we define *t's derivation in* $T_1, \ldots, T_m$ *according to* Op to be $Op^{-1}_{\langle T_1, \ldots T_m \rangle}(t) = \langle T^*_1, \ldots T^*_m \rangle$, where $T^*_1, \ldots, T^*_m$ are *maximal* subsets of $T_1, \ldots, T_m$ such that

(a) $Op(T^*_1, \ldots T^*_m) = \{t\}$.

(b) $\forall T^*_i : \forall t^* \in T^*_i$: $Op(T^*_1, \ldots, \{t^*\}, \ldots, T^*_m) \neq \emptyset$.

We also say that $Op^{-1}_{T_i}(t) = T^*_i$ is *t's derivation in* T_i, and each tuple t^* in T^*_i *contributes to* t, for $i = 1..m$.

Where to start?



Definition 6. (Witness Basis) Consider a normal form query Q. The *witness basis* for a singular value t with respect to Q and D, denoted as $W_{Q,D}(t)$, is:

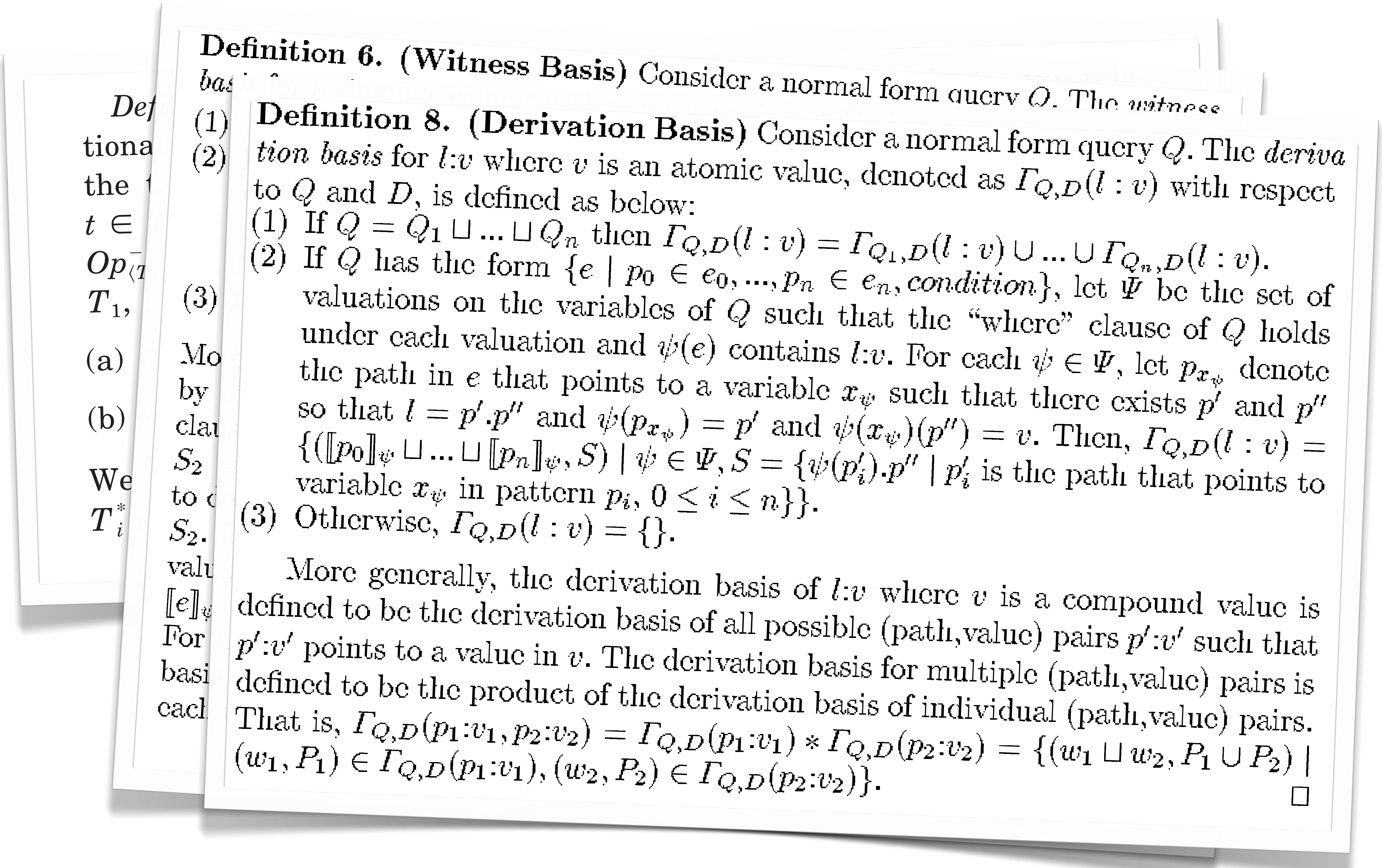
(1) If Q is of the form $Q_1 \sqcup ... \sqcup Q_n$ then $W_{Q,D}(t) = W_{Q_1,D}(t) \cup ... \cup W_{Q_n,D}(t)$.

(2) If Q is of the form $\{e \mid p_0 \in e_0, ..., p_n \in e_n, condition\}$, let Ψ be the set of all valuations on the variables of Q such that "where" clause of Q holds under each valuation in Ψ. Then, $W_{Q,D}(t) = \{[\![p_0]\!]_\psi \sqcup ... \sqcup [\![p_n]\!]_\psi \mid \psi \in \Psi, t = [\![e]\!]_\psi\}$. Note that e_i $(0 \leq i \leq n)$ is a database constant since Q is in normal form.

(3) Otherwise, $W_{Q,D}(t) = \{\}$.

More generally, for any well-formed query Q, we can define the witness basis by extending (2) as follows. We partition the set of $p_i \in e_i$ in the "where" clause of Q into two parts: $S_1 = \{p_i \mid e_i$ is the database constant $D\}$ and $S_2 = \{(p_i, e_i) \mid p_i$ is a pattern matched against a query $e_i\}$. We use $p_0^1, ..., p_k^1$ to denote the members of S_1 and $(p_0^2, e_0^2), ..., (p_m^2, e_m^2)$ to denote the members of S_2. Let Ψ be the set of all valuations on the variables of Q such that for each valuation in Ψ, "where" clause of Q holds. Then $W_{Q,D}(t) = \{P_1 \sqcup P_2 \mid \psi \in \Psi, t \sqsubseteq [\![e]\!]_\psi, P_1 = [\![p_0^1]\!]_\psi \sqcup ... \sqcup [\![p_k^1]\!]_\psi, P_2 = w_1 \sqcup ... \sqcup w_m$ where $w_i \in W_{\psi(e_i^2),D}([\![p_i^2]\!]_\psi)\}$. For a compound value t, the witness basis is the product of individual witness basis of singular values making up t. That is, consider $t = t_1 \sqcup ... \sqcup t_m$ where each t_i is singular. Then $W_{Q,D}(t) = \{w_1 \sqcup ... \sqcup w_m \mid w_i \in W_{Q,D}(t_i)\}$. □

Where to start?



Definition 6. (Witness Basis) Consider a normal form query Q. The *witness*

Definition 8. (Derivation Basis) Consider a normal form query Q. The *derivation basis* for $l{:}v$ where v is an atomic value, denoted as $\Gamma_{Q,D}(l:v)$ with respect to Q and D, is defined as below:

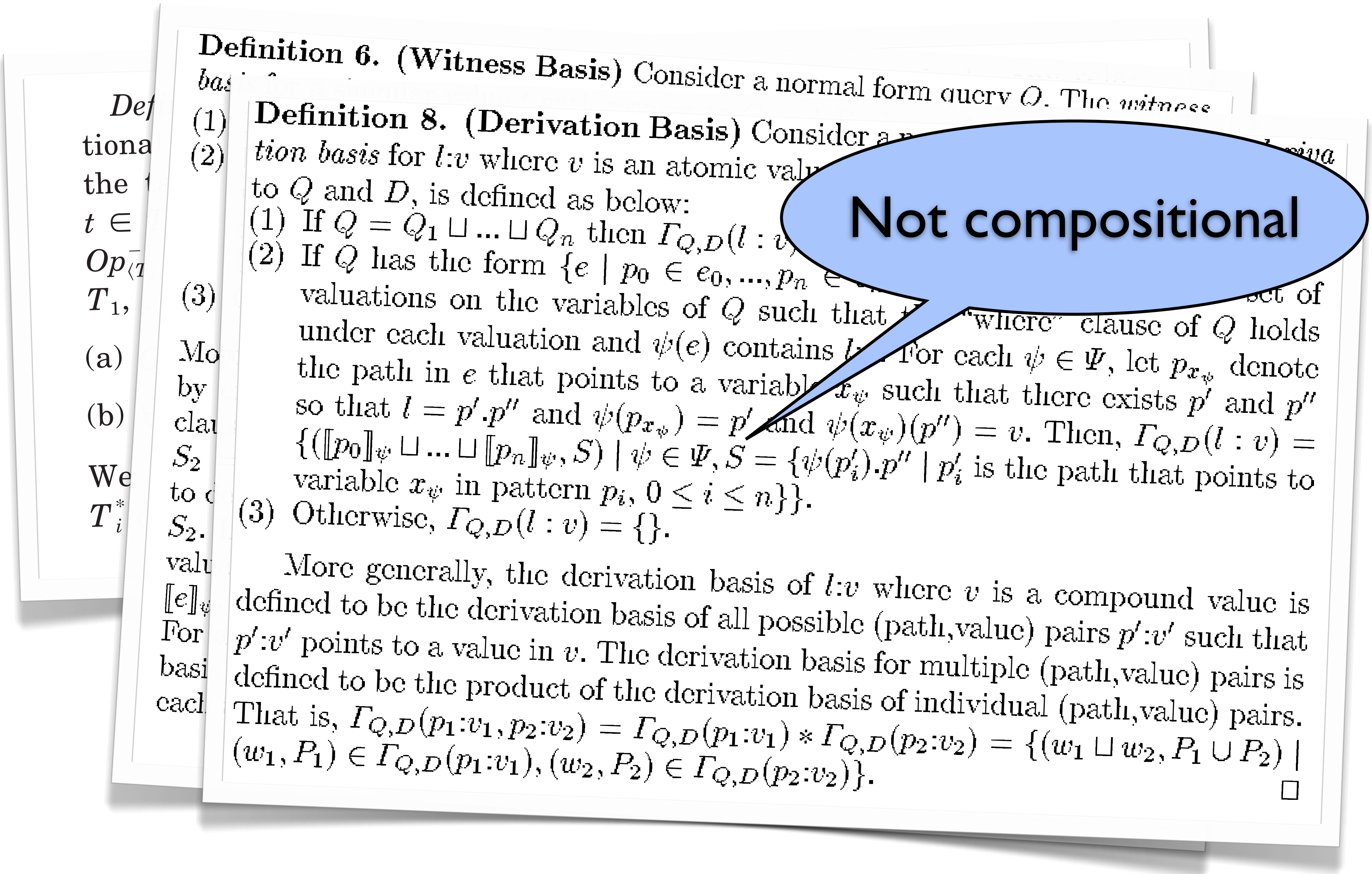
(1) If $Q = Q_1 \sqcup ... \sqcup Q_n$ then $\Gamma_{Q,D}(l:v) = \Gamma_{Q_1,D}(l:v) \cup ... \cup \Gamma_{Q_n,D}(l:v)$.

(2) If Q has the form $\{e \mid p_0 \in e_0, ..., p_n \in e_n, \mathit{condition}\}$, let Ψ be the set of valuations on the variables of Q such that the "where" clause of Q holds under each valuation and $\psi(e)$ contains $l{:}v$. For each $\psi \in \Psi$, let p_{x_ψ} denote the path in e that points to a variable x_ψ such that there exists p' and p'' so that $l = p'.p''$ and $\psi(p_{x_\psi}) = p'$ and $\psi(x_\psi)(p'') = v$. Then, $\Gamma_{Q,D}(l:v) = \{([\![p_0]\!]_\psi \sqcup ... \sqcup [\![p_n]\!]_\psi, S) \mid \psi \in \Psi, S = \{\psi(p'_i).p'' \mid p'_i$ is the path that points to variable x_ψ in pattern p_i, $0 \leq i \leq n\}\}$.

(3) Otherwise, $\Gamma_{Q,D}(l:v) = \{\}$.

More generally, the derivation basis of $l{:}v$ where v is a compound value is defined to be the derivation basis of all possible (path,value) pairs $p'{:}v'$ such that $p'{:}v'$ points to a value in v. The derivation basis for multiple (path,value) pairs is defined to be the product of the derivation basis of individual (path,value) pairs. That is, $\Gamma_{Q,D}(p_1{:}v_1, p_2{:}v_2) = \Gamma_{Q,D}(p_1{:}v_1) * \Gamma_{Q,D}(p_2{:}v_2) = \{(w_1 \sqcup w_2, P_1 \cup P_2) \mid (w_1, P_1) \in \Gamma_{Q,D}(p_1{:}v_1), (w_2, P_2) \in \Gamma_{Q,D}(p_2{:}v_2)\}$. □

Where to start?



Definition 8. (Derivation Basis) Consider a ... *tion basis* for $l{:}v$ where v is an atomic valu... to Q and D, is defined as below:

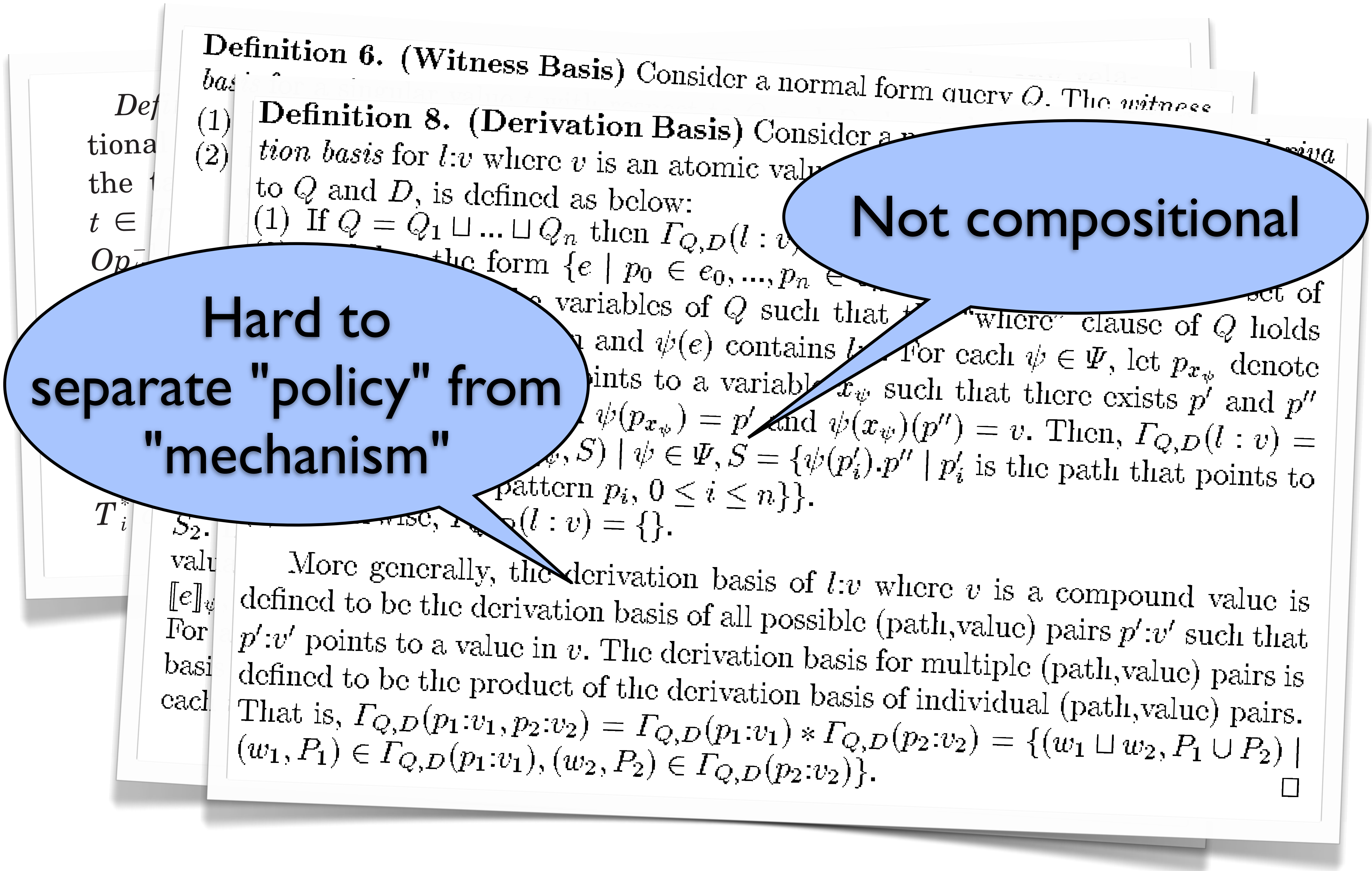
(1) If $Q = Q_1 \sqcup ... \sqcup Q_n$ then $\Gamma_{Q,D}(l : v$ ...

(2) If Q has the form $\{e \mid p_0 \in e_0, ..., p_n \in$ ... valuations on the variables of Q such that ... "where" clause of Q holds under each valuation and $\psi(e)$ contains l ... For each $\psi \in \Psi$, let p_{x_ψ} denote the path in e that points to a variable x_ψ such that there exists p' and p'' so that $l = p'.p''$ and $\psi(p_{x_\psi}) = p'$ and $\psi(x_\psi)(p'') = v$. Then, $\Gamma_{Q,D}(l : v) = \{([\![p_0]\!]_\psi \sqcup ... \sqcup [\![p_n]\!]_\psi, S) \mid \psi \in \Psi, S = \{\psi(p'_i).p'' \mid p'_i$ is the path that points to variable x_ψ in pattern p_i, $0 \leq i \leq n\}\}$.

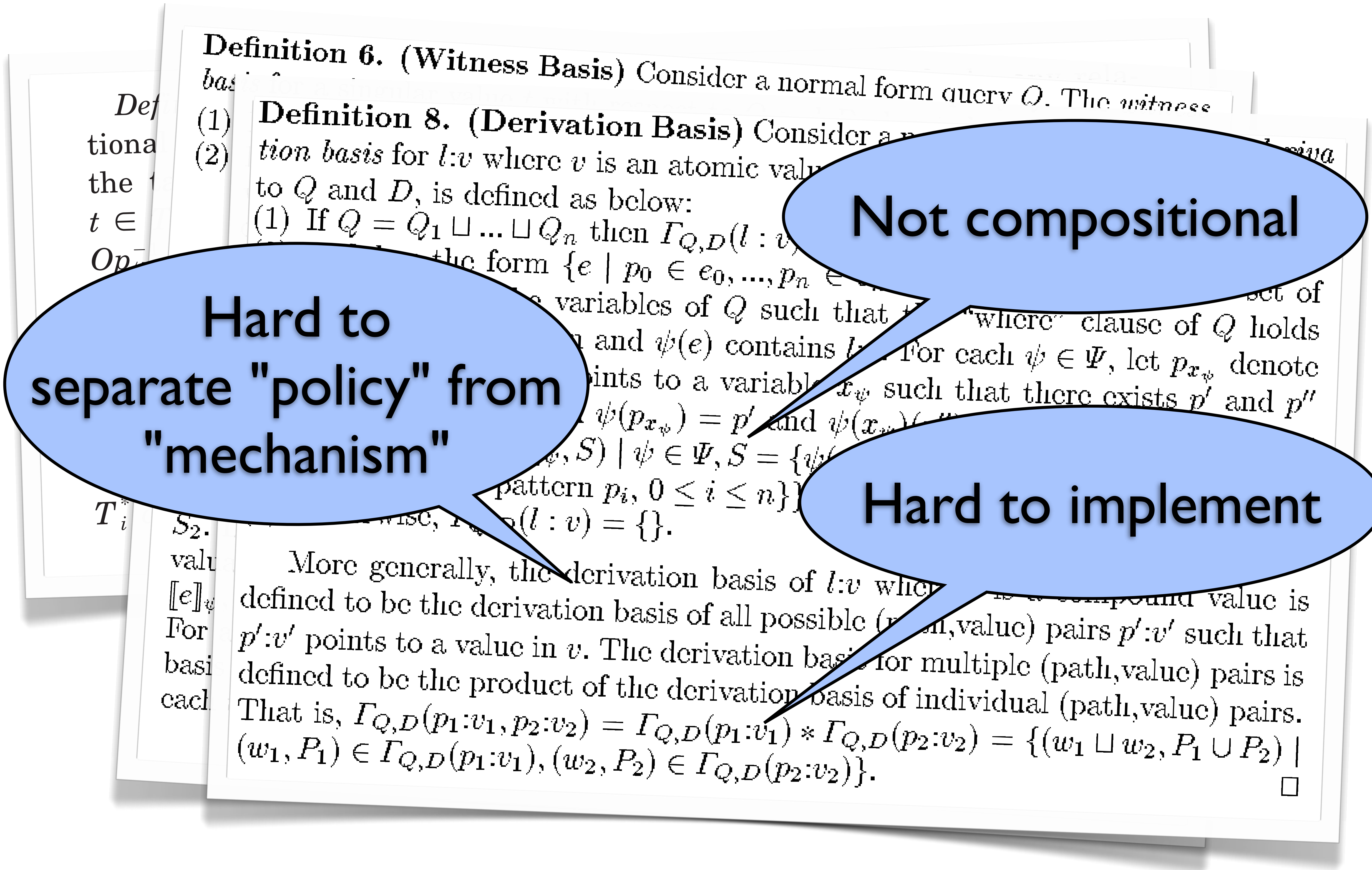
(3) Otherwise, $\Gamma_{Q,D}(l : v) = \{\}$.

More generally, the derivation basis of $l{:}v$ where v is a compound value is defined to be the derivation basis of all possible (path,value) pairs $p'{:}v'$ such that $p'{:}v'$ points to a value in v. The derivation basis for multiple (path,value) pairs is defined to be the product of the derivation basis of individual (path,value) pairs. That is, $\Gamma_{Q,D}(p_1{:}v_1, p_2{:}v_2) = \Gamma_{Q,D}(p_1{:}v_1) * \Gamma_{Q,D}(p_2{:}v_2) = \{(w_1 \sqcup w_2, P_1 \cup P_2) \mid (w_1, P_1) \in \Gamma_{Q,D}(p_1{:}v_1), (w_2, P_2) \in \Gamma_{Q,D}(p_2{:}v_2)\}$. □

Where to start?



Where to start?


Definition 6. (Witness Basis) Consider a normal form query Q. The *witness*

Definition 8. (Derivation Basis) Consider a

tion basis for $l{:}v$ where v is an atomic valu

to Q and D, is defined as below:

(1) If $Q = Q_1 \sqcup ... \sqcup Q_n$ then $\Gamma_{Q,D}(l : v$

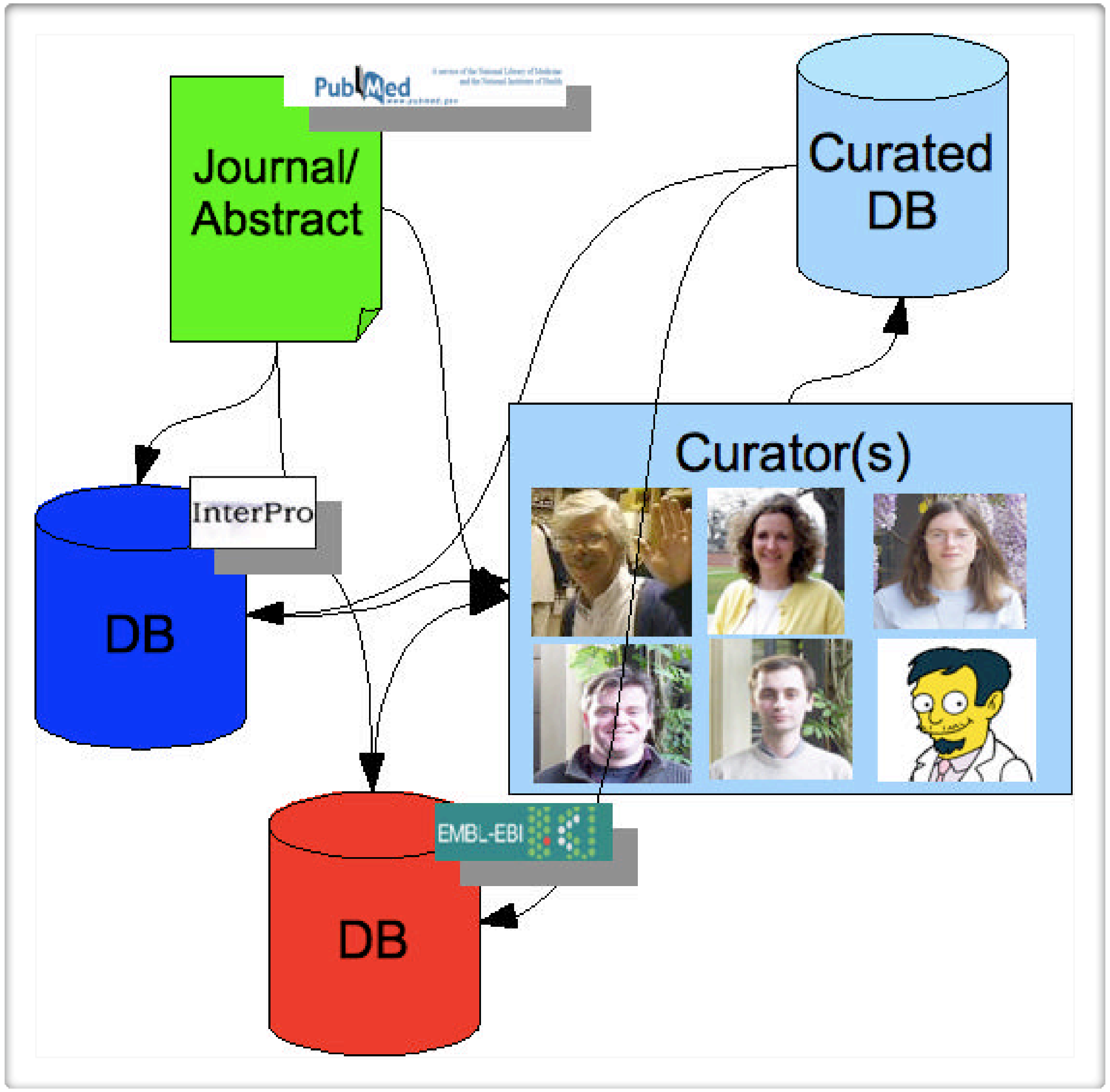
More generally, the derivation basis of $l{:}v$ whe

defined to be the derivation basis of all possible (path,value) pairs $p'{:}v'$ such that $p'{:}v'$ points to a value in v. The derivation basis for multiple (path,value) pairs is defined to be the product of the derivation basis of individual (path,value) pairs. That is, $\Gamma_{Q,D}(p_1{:}v_1, p_2{:}v_2) = \Gamma_{Q,D}(p_1{:}v_1) * \Gamma_{Q,D}(p_2{:}v_2) = \{(w_1 \sqcup w_2, P_1 \cup P_2) \mid (w_1, P_1) \in \Gamma_{Q,D}(p_1{:}v_1), (w_2, P_2) \in \Gamma_{Q,D}(p_2{:}v_2)\}$. □

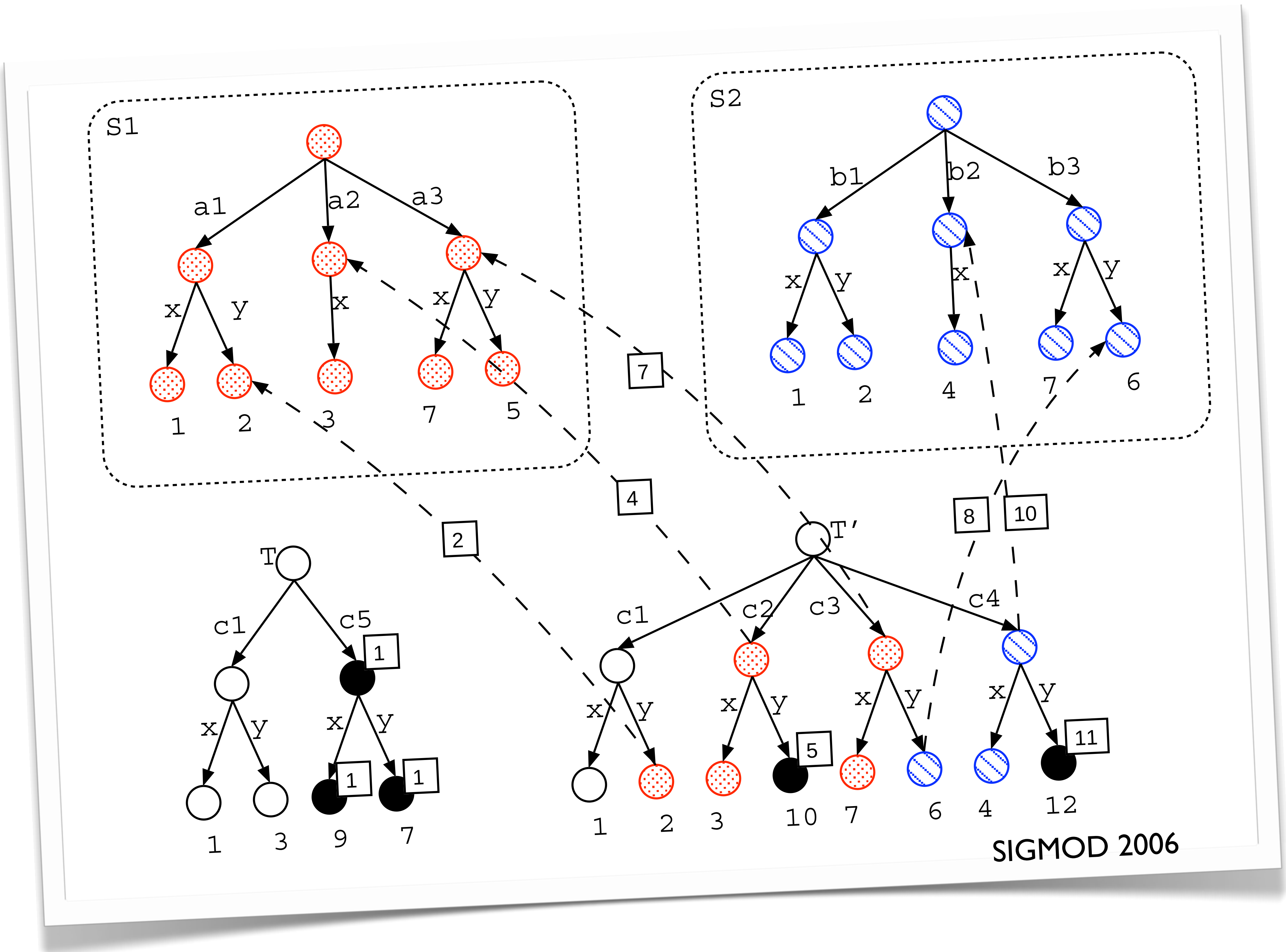
Databases and programming languages

- Database query languages **are** purely functional
  - optimization by equational rewriting basis of £10^9 DB industry
- Programming languages ideas can...
  - Help in analyzing, optimizing database queries (types, compilation, equational rewriting)
  - Integrate database or Web capabilities into higher-level languages (LINQ, Links)
- Database ideas can...
  - Lead to new programming idioms (Datalog, atomicity, STM)
  - Open up new problem spaces (high-level updates, **provenance**)

Provenance in curated databases



Provenance in curated databases



Provenance in curated databases

S

C	D
1	2
2	2
2	3

R'

A	B	C
1	2	2
1	2	3
2	3	4

R

A	B	C
1	2	2
5	6	3
2	3	4

```
update R
set (A,B) =
  (select S.C A, S.D B
   from S where S.A = 1)
where R.C = 3
```

ICDT 2007/TODS 2008

Formalization

- Consider types:

  $T ::= int \mid \ldots \mid T * T \mid T\ set$

- And expressions:
- $e ::= x \mid let\ x = e_1\ in\ e_2 \mid i \mid \ldots$
- $\quad \mid \ (e_1,e_2) \mid \pi_i(e)$
- $\quad \mid \ \varnothing \mid \{e\} \mid e_1 \cup e_2 \mid \bigcup\{e_2 \mid x \leftarrow e_1\}$

Type translation

- Translate:

P[int] = int * 'a option

P[T1 * T2] = P[T1] * P[T2] * 'a option

P[T set] = P[T] set * 'a option

Annotations 'a represent "pointers" to optional sources

Term Translation

Given x_1:T_1,...,x_n:T_n |- e : T

Want P[e] such that

x_1: P[T_1]...x_n:P[T_n] |- P[e] : P[T]

s.t. each SOME-pointer points to the "source"

Simple cases:

P[x] = x

P[i] = (i,NONE)

P[let x = e_1 in e_2] = let x = P[e_1] in P[e_2]

Pairs

$$P[(e_1,e_2)] = (P[e_1],P[e_2],NONE)$$

$$P[\pi_i(e)] = \pi_i(\pi_1(P[e]))$$

Sets

$P[\emptyset] = (\emptyset, \text{NONE})$

$P[e_1 \cup e_2] = \text{let } (v_1, \_) = P[e_1]$

$(v_2, \_) = P[e_2]$

$\text{in } (v_1 \cup v_2, \text{NONE})$

$P[\{e\}] = (\{P[e]\}, \text{NONE})$

$P[\cup\{e_2 \mid x \leftarrow e_1\}] = (\cup\{P[e_2] \mid x \leftarrow \pi_1(P[e_1])\}, \text{NONE})$

Dependency provenance

[CAA DBPL07,MSCS11]

Teams

name	team	gp	win	loss	tie	otl
string	string	int	int	int	int	int

name	team	gp	win	loss	tie	otl
Ottawa Senators	OTT	65	36	29	0	6
Dallas Stars	DAL	68	41	27	0	5
Detriot Red Wings	DET	65	42	23	0	6
Philadelphia Flyers	PHI	64	32	32	0	7
Pittsburgh Penguins	PIT	65	36	29	0	7
Calgary Flames	CGY	64	33	31	0	9
Washington Capitals	WAS	64	29	35	0	8
Tampa Bay Lightning	TAM	63	25	38	0	7
Colorado Avalanche	COL	65	33	32	0	6

Players

name	team	gp	g	a
string	string	int	int	int

name	team	gp	g	a
Malkin	PIT	65	36	48
Ovechkin	WAS	64	48	34
Lecavalier	TAM	63	32	46
Iginla	CGY	64	38	37
Spezza	OTT	59	25	50
Alfredsson	OTT	~42	35	40
Datsyuk	DET	65	23	51
Ribeiro	DAL	65	26	48
Zetterberg	DET	58	35	38
St Louis	TAM	63	21	50
Richards	PHI	62	23	43
Stastny	COL	50	19	33

Result

name	g
string	int

name	g
Iginla	38

cf. Dependency Core Calculus [Abadi et al. 1999]

Dependency provenance

[...97,MSCS11]

What input parts *may* given output depend upon?

		Players				Result	
name	[illegible]	[...]am	gp	g	a	name	g
string	[illegible]	[...]ring	int	int	int	string	int

name	team	gp	win	loss	tie	otl
Ottawa Senators	OTT	65	36	29	0	6
Dallas Stars	DAL	68	41	27	0	5
Detriot Red Wings	DET	65	42	23	0	6
Philadelphia Flyers	PHI	64	32	32	0	7
Pittsburgh Penguins	PIT	65	36	29	0	7
Calgary Flames	CGY	64	33	31	0	9
Washington Capitals	WAS	64	29	35	0	8
Tampa Bay Lightning	TAM	63	25	38	0	7
Colorado Avalanche	COL	65	33	32	0	6

name	team	gp	g	a
Malkin	PIT	65	36	48
O[illegible]echkin	WAS	64	48	34
Lecav[illegible]lier	TAM	63	32	46
Iginla	CGY	64	38	37
Spezza	OTT	59	25	50
Alfredsson	OTT	~42	35	40
Datsyuk	DET	65	23	51
Ribeiro	DAL	65	26	48
Zetterberg	DET	58	35	38
St Louis	TAM	63	21	50
Richards	PHI	62	23	43
Stastny	COL	50	19	33

name	g
Iginla	38

cf. Dependency Core Calculus [Abadi et al. 1999]

Dependency provenance

[CAA DBPL07,MSCS11]

"Good" players on winning teams

Teams

name	team	gp	win	loss	tie	[illegible]
Ottawa Senators	OTT	65	36	29	0	6
Dallas Stars	DAL	68	41	27	0	5
Detriot Red Wings	DET	65	42	23	0	6
Philadelphia Flyers	PHI	64	32	32	0	7
Pittsburgh Penguins	PIT	65	36	29	0	7
Calgary Flames	CGY	64	33	31	0	9
Washington Capitals	WAS	64	29	35	0	8
Tampa Bay Lightning	TAM	63	25	38	0	7
Colorado Avalanche	COL	65	33	32	0	6

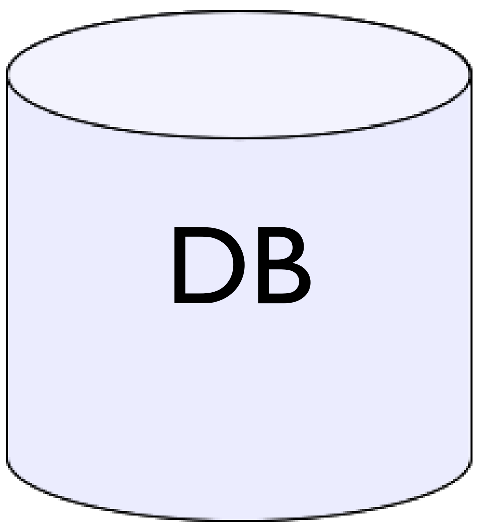
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	48
[illegible]	WAS	64	[illegible]	34
Lecavalier	TAM	63	32	[illegible]
Iginla	CGY	64	38	37
Spezza	OTT	59	25	50
Alfredsson	OTT	~42	35	40
Datsyuk	DET	65	23	51
Ribeiro	DAL	65	26	48
Zetterberg	DET	58	35	38
St Louis	TAM	63	21	50
Richards	PHI	62	23	43
Stastny	COL	50	19	33

Result

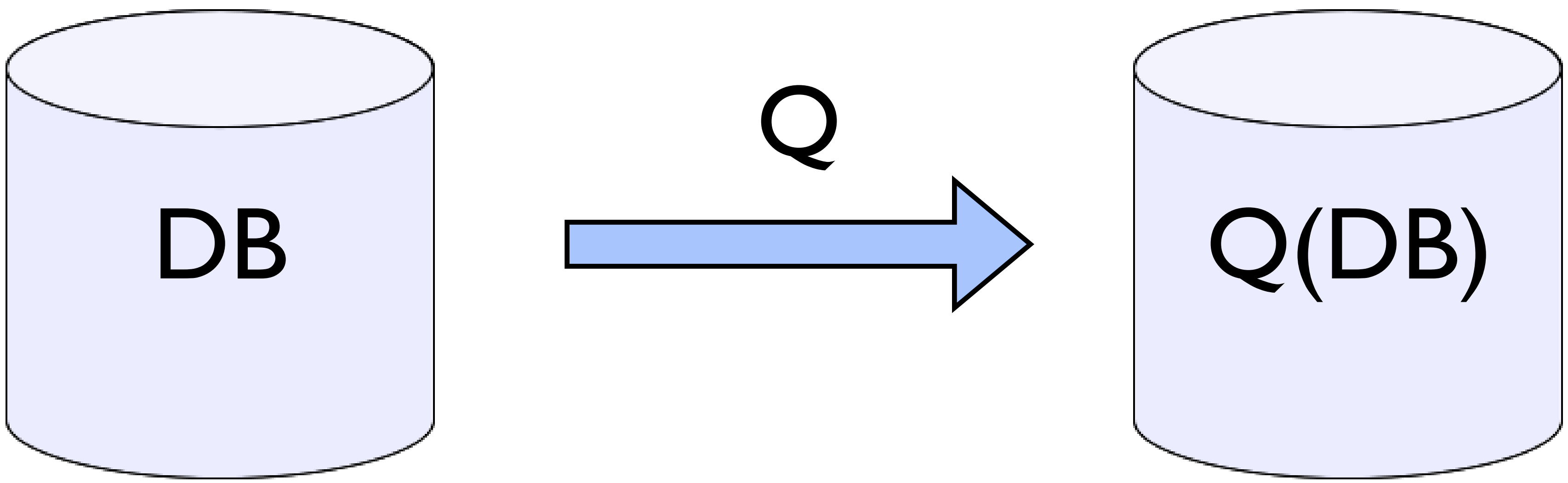
name	g
Iginla	38

cf. Dependency Core Calculus [Abadi et al. 1999]

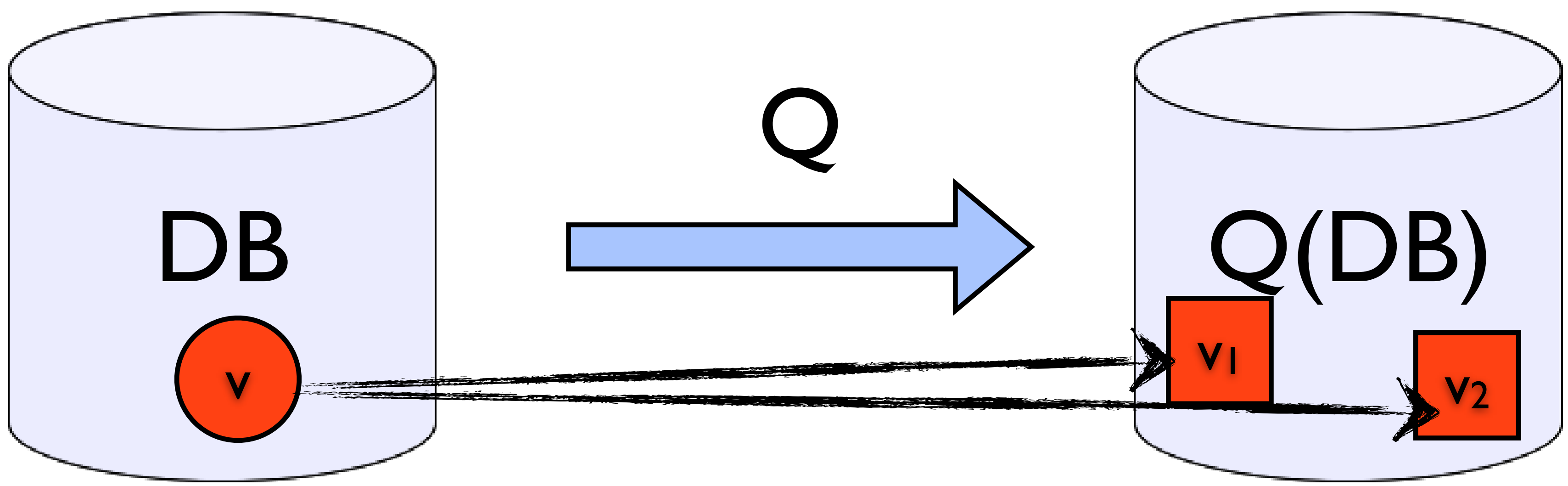
Basic idea



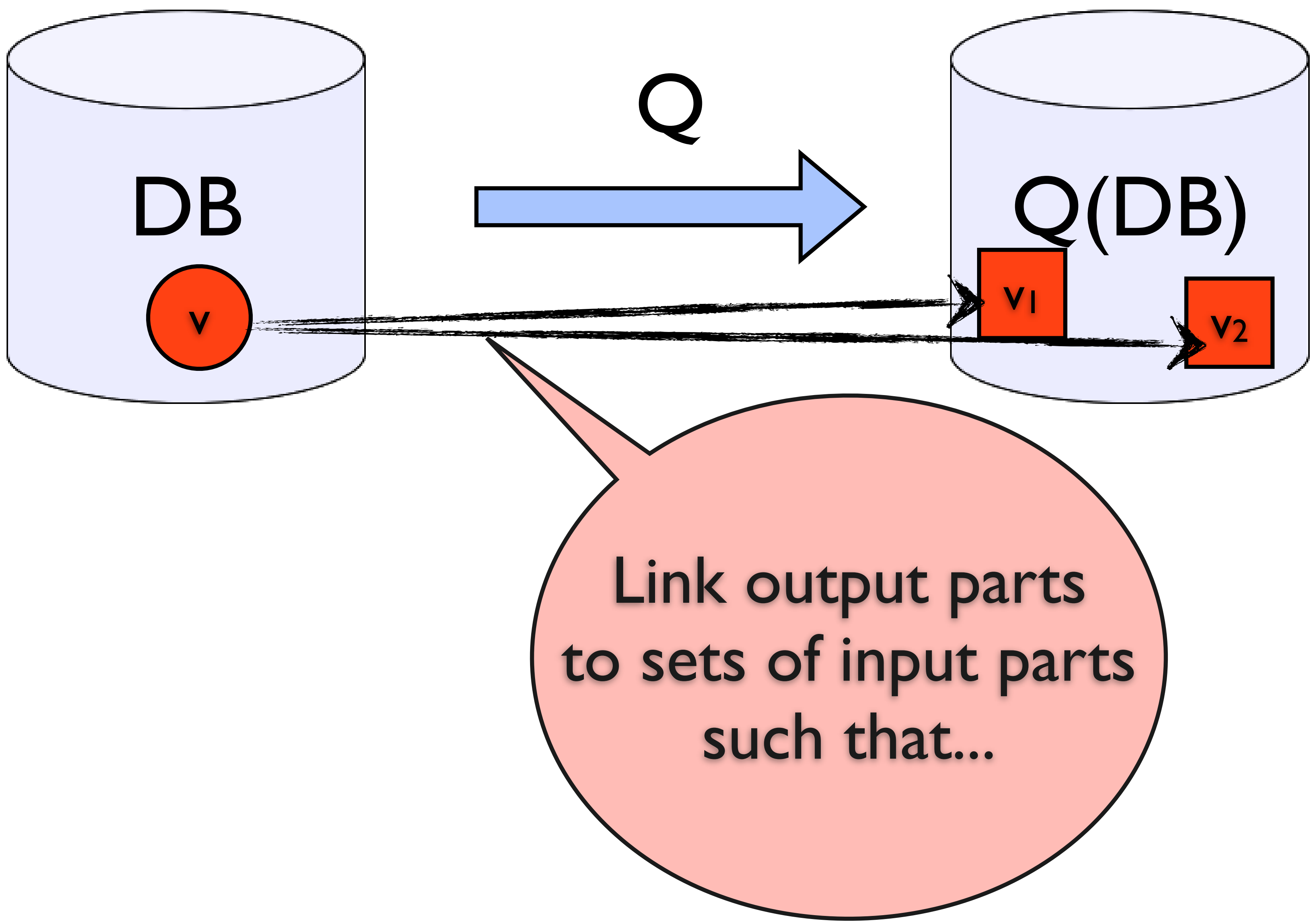
Basic idea



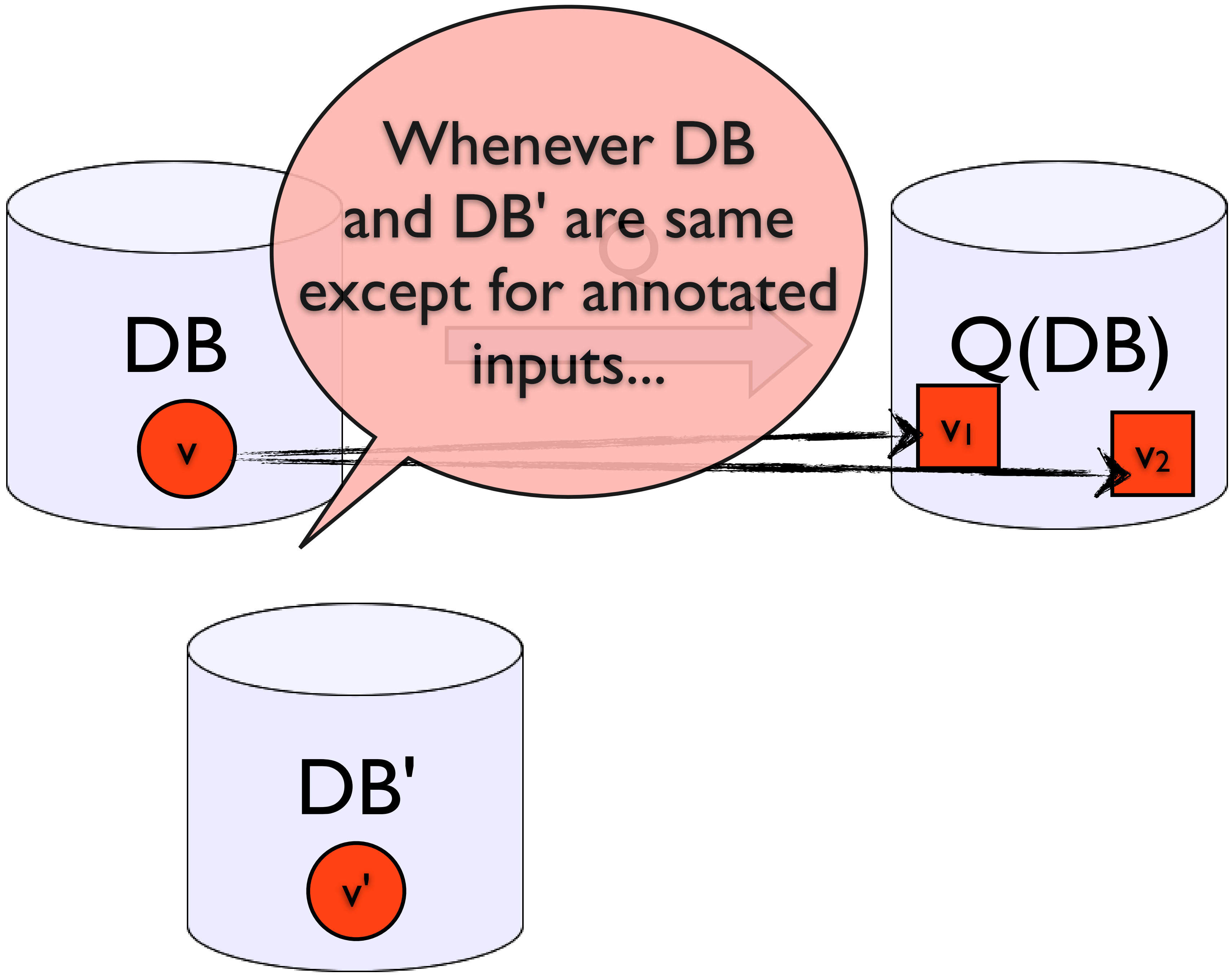
Basic idea



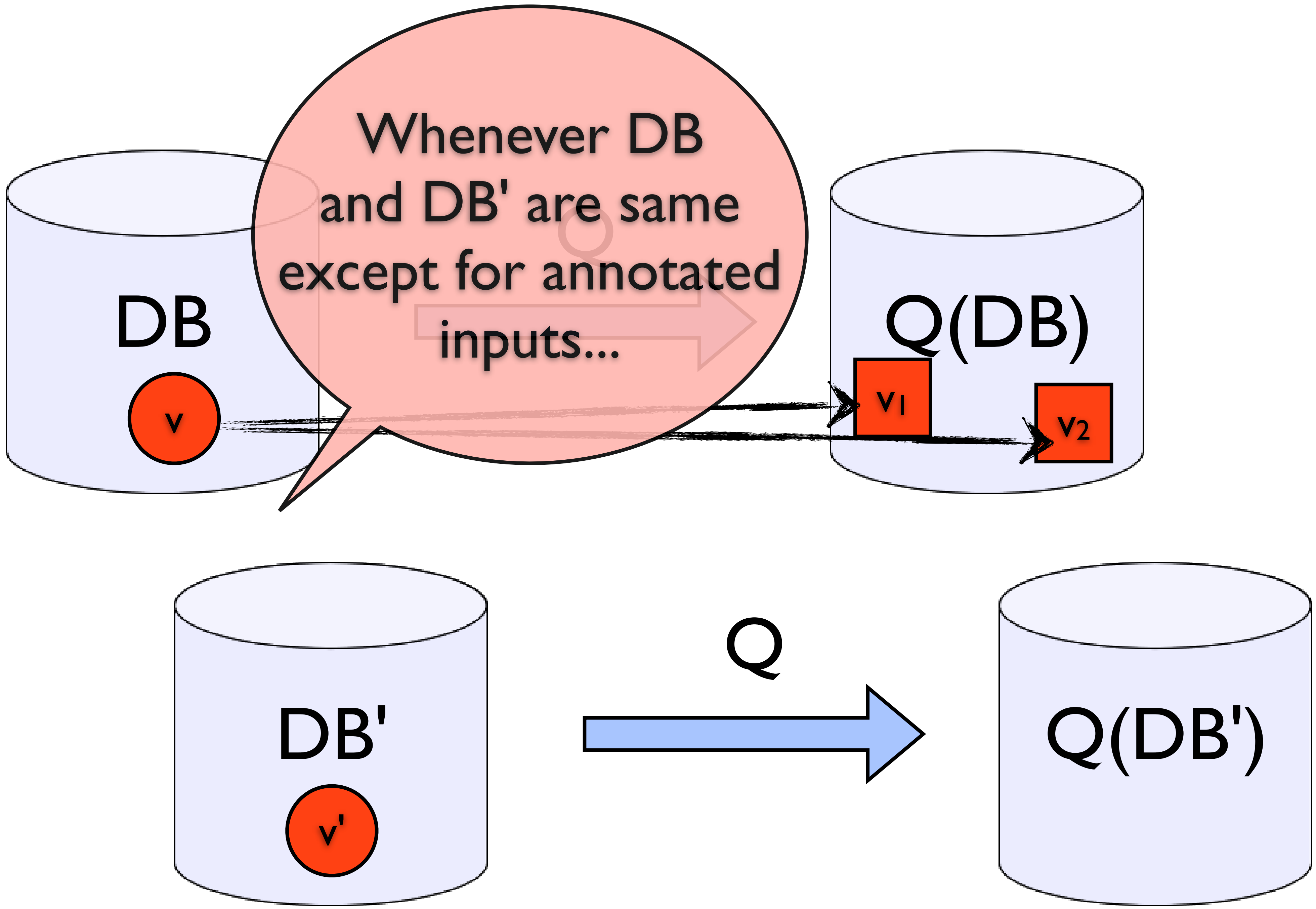
Basic idea


DB
v
Q
Q(DB)
v1
v2
Link output parts to sets of input parts such that...

Basic idea

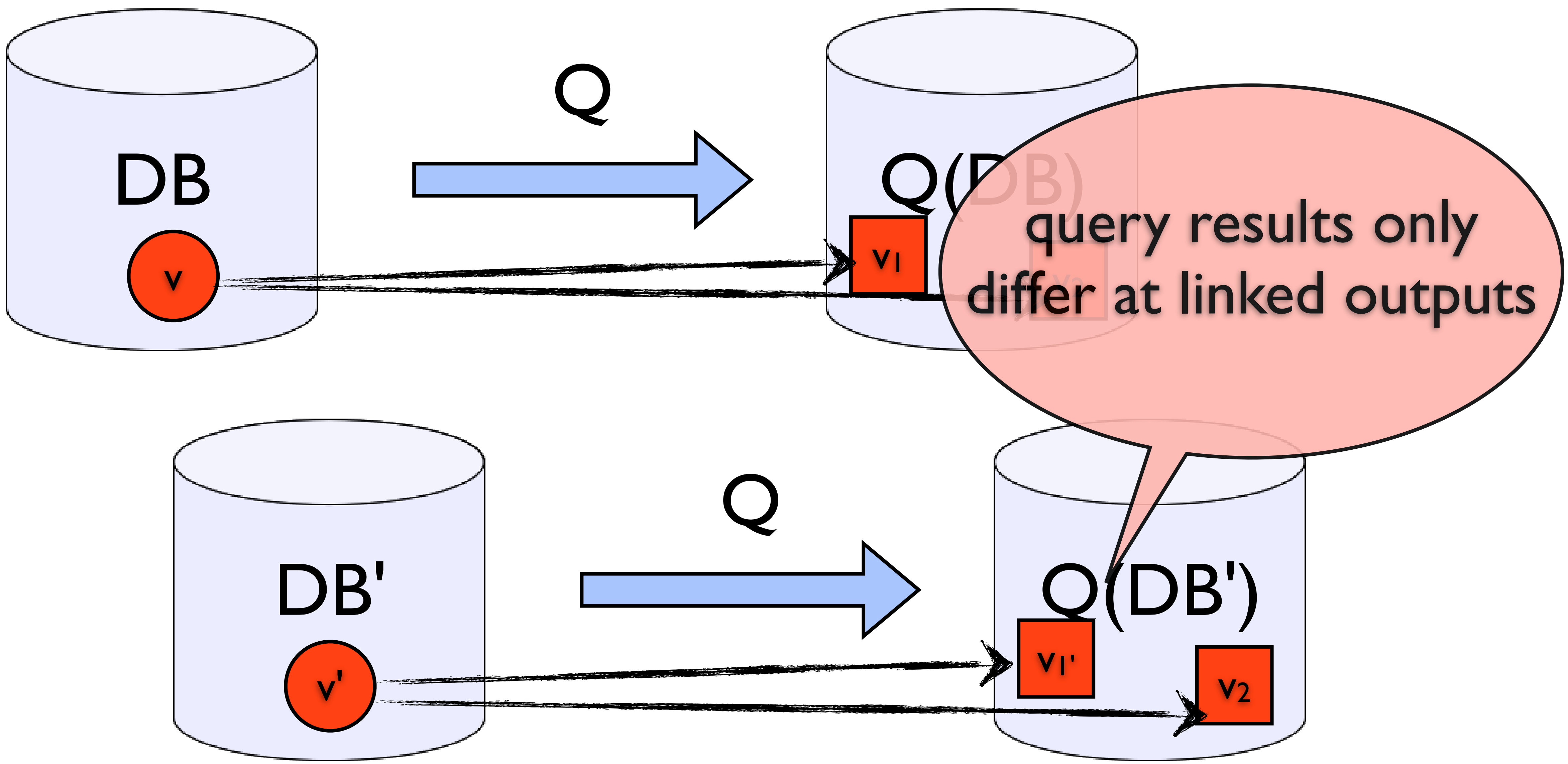

Whenever DB and DB' are same except for annotated inputs...
DB
v
Q
Q(DB)
v1
v2
DB'
v'

Basic idea

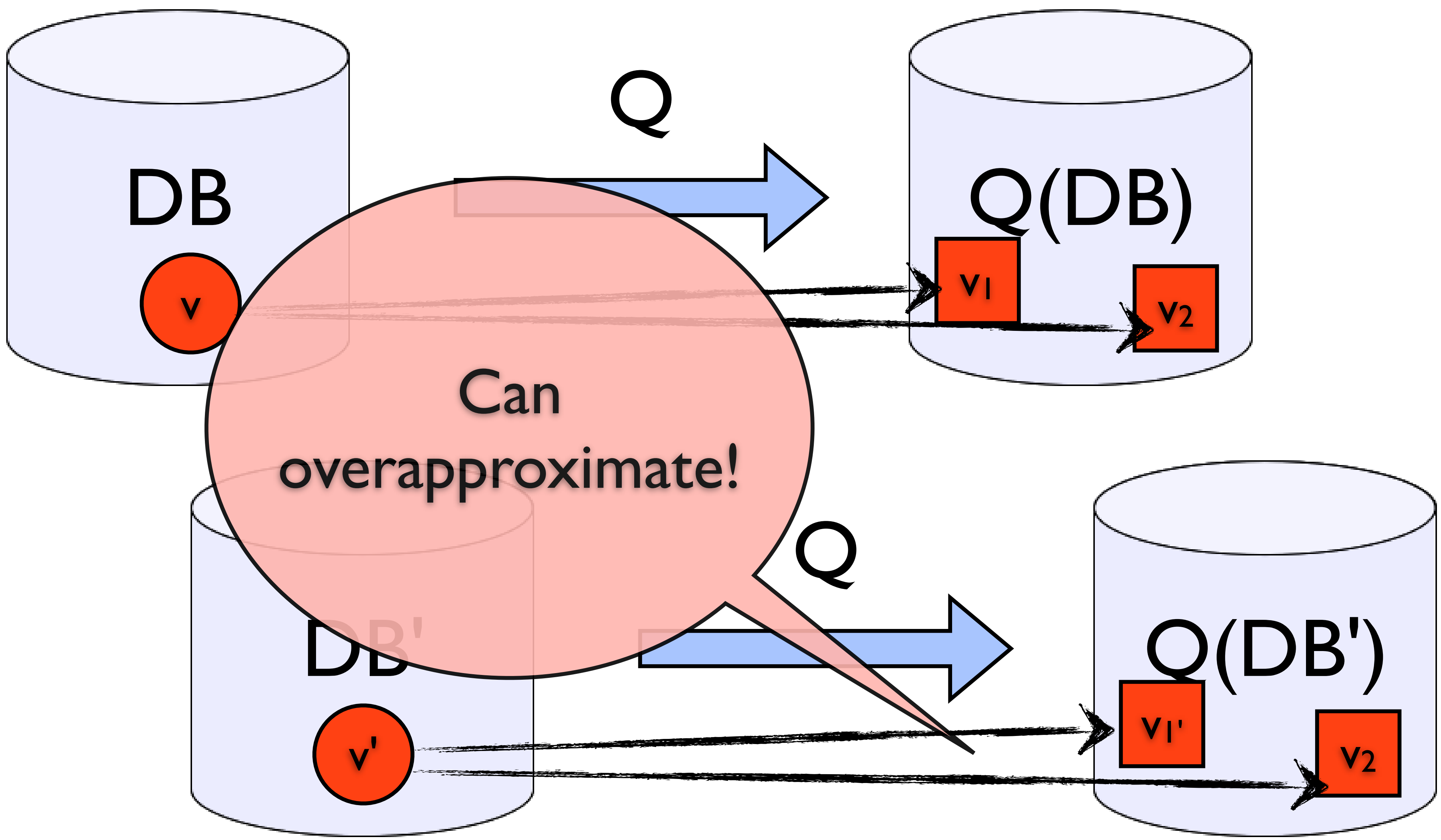


Whenever DB and DB' are same except for annotated inputs...
DB
v
Q
Q(DB)
v1
v2
DB'
v'
Q
Q(DB')

Basic idea


Q
DB
Q(DB)
v
v1
v2
query results only
differ at linked outputs
Q
DB'
Q(DB')
v'
v1'
v2

Basic idea



Formalization

- Consider types:

  T ::= int | ... | T * T | T set

- Translate:

$$P[\text{int}] = \text{int} * \text{'a set}$$

$$P[T_1 * T_2] = P[T_1] * P[T_2] * \text{'a set}$$

$$P[T\ \text{set}] = P[T]\ \text{set} * \text{'a set}$$

  Annotations 'a represent "pointers" to sets of sources

Term Translation

Given x_1:T_1,...,x_n:T_n |- e : T

Want P[e] such that

x_1: P[T_1]...x_n:P[T_n] |- P[e] : P[T]

s.t. all "dependencies" are captured.

Simple cases:

P[x] = x

P[i] = (i,∅)

P[let x = e_1 in e_2] = let x = P[e_1] in P[e_2]

Pairs

$P[(e_1,e_2)] = (P[e_1],P[e_2],\varnothing)$

$P[\pi_i(e)] = \text{let } (v,a) = P[e]$

$(v_i', b) = \pi_i(v)$

$\text{in } (v_i, a \cup b)$

Sets

$P[\emptyset] = (\emptyset,\emptyset)$

$P[e_1 \cup e_2] = \text{let } (v_1,a_1) = P[e_1]$

$(v_2,a_2) = P[e_2]$

$\text{in } (v_1 \cup v_2,\ a_1 \cup a_2)$

$P[\{e\}] = (\{P[e]\},\emptyset)$

$P[\cup\{e_2 \mid x \leftarrow e_1\}] = \text{let } (v,a) = P[e_1]$

$\text{in } (\cup\{P[e_2] \mid x \leftarrow v\},a)$

Question

- The translations seem to have a lot in common...
- Can we implement them "once and for all"
  - generic/dynamic typing?
  - dependent types?
- Can we implement them in a way that runs efficiently against database?

Links

- Currently supports superset of NRC core-language
  - Higher-order, impure features
  - Effect typing allows safe combination, query extraction [Cooper 2009]
- Ferry [Grust et al. 2010]: extending to support nested data
  - number of queries bounded by types, not data

Generic/Dependent Links?

- Ur/WEB also supports some generic web programming
- Would like to write something like this:

```
type family P a Int = (Int,a)
type family P a (b,c) =
    (P a b, P a c, a)
type family P a [b] = ([P a b],a)
```

Dependent/Generic Links?

- Would like to write something like this:

```
whereprov :: Exp e t' →
    Exp (P e (Maybe a))
        (P t' (Maybe a))
whereprov (Const c) =
    Pair (Const c) Nothing
whereprov (Var x) = Var x
...
```

Other ways forward?

- Haskell: GADTs + type families/type-level computaion + HaskellDB?
- Agda: dependent types ✓, but not DB?
- Idris: dependent types ✓; can we implement query normalization & DB communication as a EDSL?
- Ur/Web: Maybe already has enough GP, but still learning
- (ideally: compile Links to another language that has a mature compiler :)

Database Wiki

- Idea: Wiki-like Web interface to (semi)structured data

- Joint work with Buneman, Mueller, Lindley
- Prototype showcases prior research on provenance, archiving, annotation, security
  - to present at workshop on "Biological Wikis" [NETTAB 2010]

..se Wiki

..ce to (semi)structured data

Mueller, Lindley

research on provenance, ..rity

..Biological Wikis" [NETTAB 2010]

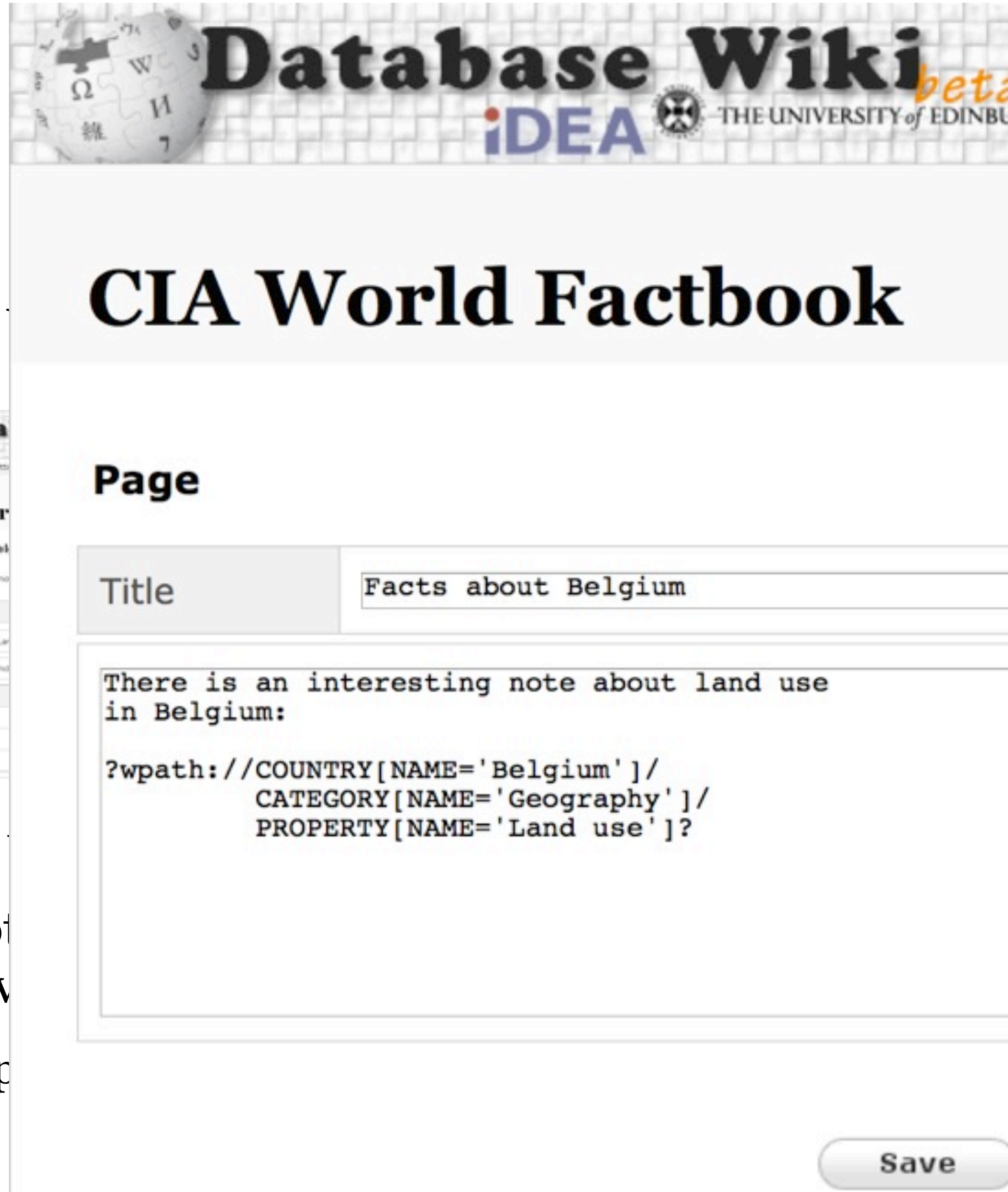
Database Wiki

- Idea: Wiki-like Web interface to (semi)structured data

- Joint work with Buneman, Mueller, Lindley
- Prototype showcases prior research on provenance, archiving, annotation, security
  - to present at workshop on "Biological Wikis" [NETTAB 2010]

- Idea: ... lata

- Joint ...
- Proto... ,
archiv...
  - to p... 2010]



Database Wiki

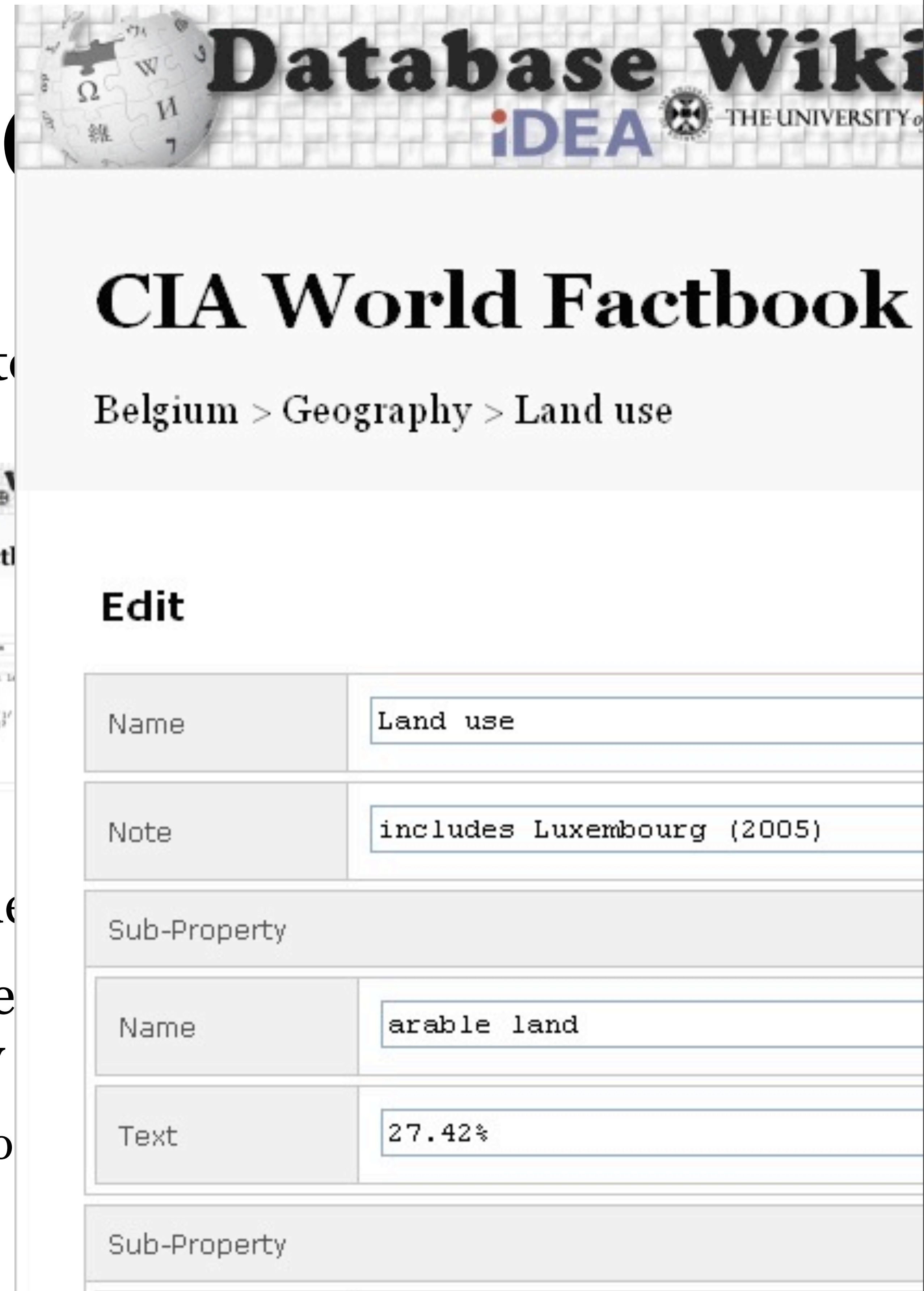
- Idea: Wiki-like Web interface to (semi)structured data

- Joint work with Buneman, Mueller, Lindley
- Prototype showcases prior research on provenance, archiving, annotation, security
  - to present at workshop on "Biological Wikis" [NETTAB 2010]

Database

- Idea: Wiki-like Web interface t

- Joint work with Buneman, Mue
- Prototype showcases prior rese archiving, annotation, security
  - to present at workshop on "Biolo



Database Wiki

- Idea: Wiki-like Web interface to (semi)structured data

- Joint work with Buneman, Mueller, Lindley
- Prototype showcases prior research on provenance, archiving, annotation, security
  - to present at workshop on "Biological Wikis" [NETTAB 2010]

Conclusions

- Provenance techniques can be defined as "type-dependent types/functions"
- Complex provenance transformations challenging to implement against real DBs
- Combining Links, Ur/WEB or LINQ with generic or dependent typing might be a good way to proceed