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

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 $11.50 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

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

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

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

Science Policy Forum

Nature: Illuminating the black box

New York Times: Nobel Laureate Retracts Two Papers Unrelated to Her Prize

The provenance crisis

The provenance crisis

Illustrates the need for provenance for reproducibility and verification of processes

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 $Op_{T_i}^{-1}(t) = T_i^*$ is $t$’s derivation in $T_i$, and each tuple $t^*$ in $T_i^*$ contributes to $t$, for $i = 1..m$. 
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 \( Q_1 \cup ... \cup Q_n \) then \( W_{Q,D}(t) = W_{Q_1,D}(t) \cup ... \cup W_{Q_n,D}(t) \).
2. If \( Q \) is of the form \( \{ c \mid p_0 \in c_0, ..., p_n \in c_n, \text{condition} \} \), let \( \Psi \) be the set of all valuations on the variables of \( Q \) such that “where” clause of \( Q \) holds under each valuation in \( \Psi \). Then, \( W_{Q,D}(t) = \{ [p_0]_{\psi} \cup ... \cup [p_n]_{\psi} \mid \psi \in \Psi, t = [c]_{\psi} \} \).

Note that \( c_i (0 \leq i \leq n) \) 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: \( S_1 = \{ p_i \mid e_i \text{ is the database constant } D \} \) and \( S_2 = \{ (p_i, e_i) \mid p_i \text{ is a pattern matched against a query } e_i \} \). We use \( p_1^1, ..., p_k^1 \) to denote the members of \( S_1 \) and \( (p_1^2, e_1^2), ..., (p_m^2, e_m^2) \) to denote the members of \( S_2 \). Let \( \Psi \) be the set of all valuations on the variables of \( Q \) such that for each valuation in \( \Psi \), “where” clause of \( Q \) holds. Then \( W_{Q,D}(t) = \{ P_1 \cup P_2 \mid \psi \in \Psi, t \subseteq [c]_{\psi}, P_1 = [p_1^1]_{\psi} \cup ... \cup [p_k^1]_{\psi}, P_2 = w_1 \cup ... \cup w_m \text{ where } w_i \in W_{\psi(e_i^2), D([p_i^2]_{\psi})} \} \).

For a compound value \( t \), the witness basis is the product of individual witness basis of singular values making up \( t \). That is, consider \( t = t_1 \cup ... \cup t_m \) where each \( t_i \) is singular. Then \( W_{Q,D}(t) = \{ w_1 \cup ... \cup w_m \mid w_i \in W_{Q,D}(t_i) \} \). \( \square \)
Where to start?

Definition 6. (Witness Basis) Consider a normal form query \( Q \). The witness basis for \( Q \) is defined as below:

1. For any \( t \in \text{Op}_1 \), \( \text{Witness}(Q) = \{ \text{Op}_1 \} \).
2. For any \( t \in \text{Op} \setminus \{ \text{Op}_1 \} \), \( \text{Witness}(Q) \) is defined as below:
   (a) \( \text{Witness}(Q) = \emptyset \).
   (b) \( \text{Witness}(Q) \) is defined to be \( \text{Witness}(Q) \).
   (c) \( \text{Witness}(Q) \) is defined to be \( \text{Witness}(Q) \).

Definition 8. (Derivation Basis) Consider a normal form query \( Q \). The derivation basis for \( Q \) is defined as below:

1. If \( Q = Q_1 \cup ... \cup Q_r \), then \( \Gamma_{Q,D}(Q : v) = \Gamma_{Q_1,D}(Q_1 : v) \cup ... \cup \Gamma_{Q_r,D}(Q_r : v) \).
2. If \( Q \) has the form \( \{ e \mid p_0 \in e_0, ..., p_n \in e_n, \text{condition} \} \), let \( \Psi \) 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 \( p_{x_{\psi}} \) denote the path in \( e \) that points to a variable \( x_{\psi} \) such that there exists \( p' \) and \( p'' \) so that \( l = p' \cdot p'' \) and \( \psi(p_{x_{\psi}}) = p' \) and \( \psi(x_{\psi})(p'') = v \). Then, \( \Gamma_{Q,D}(Q : v) = \{ ([p_0]_{\psi} \cup ... \cup [p_n]_{\psi}, S) \mid \psi \in \Psi, S = \psi(p_{x_{\psi}}) \cdot p'' \} \).
   \( \Gamma_{Q,D}(Q : v) \) is the path that points to variable \( x_{\psi} \) in pattern \( p_i \), \( 0 \leq i \leq n \).
3. Otherwise, \( \Gamma_{Q,D}(Q : v) = \emptyset \).

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, \( \Gamma_{Q,D}(p_1 : v_1, p_2 : v_2) = \Gamma_{Q,D}(p_1 : v_1) \times \Gamma_{Q,D}(p_2 : v_2) = \{ (w_1 \cup w_2, P_1 \cup P_2) \mid (w_1, P_1) \in \Gamma_{Q,D}(p_1 : v_1), (w_2, P_2) \in \Gamma_{Q,D}(p_2 : v_2) \} \).
Where to start?

Definition 6. (Witness Basis) Consider a normal form query $Q$. The witness basis for $Q$ is defined as follows:

1. If $Q = Q_1 \cup \ldots \cup Q_r$, then $O \in W(Q)$.
2. If $Q$ has the form $\{e \mid \rho_0 \in \epsilon_0, \ldots, \rho_m \in \epsilon_m\}$, the set of valuations on the variables of $Q$ such that the "where" clause of $Q$ holds under each valuation and $\psi(e)$ contains $Q$. For each $\psi \in \Psi$, let $p_x_\psi$ denote the path in $e$ that points to a variable $x_\psi$ 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, $O_{Q,D}(l: v) = \{([p_0]_\psi \cup \ldots \cup [p_n]_\psi, S) \mid \psi \in \Psi, S = \{\psi(p_i).p'' \mid p_i \text{ is the path that points to variable } x_\psi \text{ in pattern } p_i, 0 \leq i \leq n\}\}.
3. Otherwise, $O_{Q,D}(l: v) = \{\}$. 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, $O_{Q,D}(p_1:v_1, p_2:v_2) = O_{Q,D}(p_1:v_1) \times O_{Q,D}(p_2:v_2) = \{(w_1 \cup w_2, P_1 \cup P_2) \mid (w_1, P_1) \in O_{Q,D}(p_1:v_1), (w_2, P_2) \in O_{Q,D}(p_2:v_2)\}$. 

Not compositional
Where to start?

Hard to separate "policy" from "mechanism"

Not compositional
Where to start?

Definition 6. (Witness Basis) Consider a normal form query $Q$. The witness basis for $Q$ is defined as:

1. If $Q = Q_1 \cup \ldots \cup Q_k$, then $\Gamma_{Q,D}(l : v) = \{e \mid e_0 \in \epsilon_0, \ldots, e_n \in \epsilon_n\}$.

Definition 8. (Derivation Basis) Consider a normal form query $Q$. The derivation basis for $Q$ is defined as:

1. If $Q = Q_1 \cup \ldots \cup Q_k$, then $\Gamma_{Q,D}(l : v) = \{e \mid e_0 \in \epsilon_0, \ldots, e_n \in \epsilon_n\}$.

For each tuple $e$ in the witness basis, the derivation basis is defined as:

- If $e = e_0 \circ e_1 \circ \ldots \circ e_n$, then $\Gamma_{Q,D}(l : v) = \{e_0 \circ e_1 \circ \ldots \circ e_n\}$.

More generally, the derivation basis of a query $Q$ is defined to be the derivation basis of all possible (path,value) pairs $p' \cdot v'$ such that $p' \cdot v'$ 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, $\Gamma_{Q,D}(p_1:v_1, p_2:v_2) = \Gamma_{Q,D}(p_1:v_1) \times \Gamma_{Q,D}(p_2:v_2) = \{(w_1 \cup w_2, P_1 \cup P_2) \mid (w_1, P_1) \in \Gamma_{Q,D}(p_1:v_1), (w_2, P_2) \in \Gamma_{Q,D}(p_2:v_2)\}$. 

Not compositional

Hard to separate "policy" from "mechanism"

Hard to implement
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

Figure 4: An example of executing the update in Figure 3. The upper two trees are unchanged; black nodes represent inserted or deleted nodes; dashed lines indicate provenance links. Boxed numbers indicate fields is ignored for inserts and deletes. Note that information about each transaction, such as commit time and user information, may be wasteful in terms of space, because it introduces one provenance link. However, it retains the maximum possible information about the user's actions. In fact, the exact update operation describing the user's sequence of actions can be recovered from the provenance table.

Figure 5: The provenance tables for the update operation of the net changes resulting from a transaction. Whether or not transactional provenance is used, much of the storage cost for the provenance of a transaction is proportional to the number of nodes touched in the input and output of the transaction. That is, the number of transactional provenance nodes deleted from the input, and in the output.
Provenance in curated databases

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

Thursday, November 25, 2010
Formalization

• Consider types:
  \[ T ::= \text{int} \mid \ldots \mid T \times T \mid T \text{ set} \]

• And expressions:
  \[ e ::= x \mid \text{let } x = e_1 \text{ in } e_2 \mid i \mid \ldots \]

  \[ \mid (e_1,e_2) \mid \pi_i(e) \]

  \[ \mid \emptyset \mid \{e\} \mid e_1 \cup e_2 \mid \bigcup\{e_2 \mid x \leftarrow e_1\} \]
Type translation

- Translate:
  
  \[
P[\text{int}] = \text{int} \times 'a \text{ option}
  
  P[T_1 \times T_2] = P[T_1] \times P[T_2] \times 'a \text{ option}
  
  P[T \text{ set}] = P[T] \text{ set} \times 'a \text{ option}
  \]

  Annotations 'a represent "pointers" to optional sources
Term Translation

Given $x_1:T_1,\ldots,x_n:T_n \vdash e : T$

Want $P[e]$ such that

$x_1: P[T_1]\ldots x_n: P[T_n] \vdash P[e] : P[T]$

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

Simple cases:

$$P[x] = x$$

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

$$P[\text{let } x = e_1 \text{ in } e_2] = \text{let } x = P[e_1] \text{ in } P[e_2]$$
Pairs

\[ P[(e_1,e_2)] = (P[e_1],P[e_2],\text{NONE}) \]

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

---

<table>
<thead>
<tr>
<th>Teams</th>
<th>Players</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>name</strong></td>
<td><strong>team</strong></td>
<td><strong>gp</strong></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Ottawa Senators</td>
<td>OTT</td>
<td>65</td>
</tr>
<tr>
<td>Dallas Stars</td>
<td>DAL</td>
<td>68</td>
</tr>
<tr>
<td>Detroit Red Wings</td>
<td>DET</td>
<td>65</td>
</tr>
<tr>
<td>Philadelphia Flyers</td>
<td>PHI</td>
<td>64</td>
</tr>
<tr>
<td>Pittsburgh Penguins</td>
<td>PIT</td>
<td>65</td>
</tr>
<tr>
<td>Calgary Flames</td>
<td>CGY</td>
<td>64</td>
</tr>
<tr>
<td>Washington Capitals</td>
<td>WAS</td>
<td>64</td>
</tr>
<tr>
<td>Tampa Bay Lightning</td>
<td>TAM</td>
<td>63</td>
</tr>
<tr>
<td>Colorado Avalanche</td>
<td>COL</td>
<td>65</td>
</tr>
<tr>
<td>St Louis</td>
<td>TAM</td>
<td>63</td>
</tr>
<tr>
<td>Stastny</td>
<td>COL</td>
<td>50</td>
</tr>
</tbody>
</table>

---

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

[CAA DBPL07, MSCS11]

What input parts may given output depend upon?

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

[CAA DBPL07, MSCS11]

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

<table>
<thead>
<tr>
<th>name</th>
<th>team</th>
<th>gp</th>
<th>win</th>
<th>loss</th>
<th>tie</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ottawa Senators</td>
<td>OTT</td>
<td>65</td>
<td>36</td>
<td>29</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Dallas Stars</td>
<td>DAL</td>
<td>68</td>
<td>41</td>
<td>27</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Detroit Red Wings</td>
<td>DET</td>
<td>65</td>
<td>42</td>
<td>23</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Philadelphia Flyers</td>
<td>PHI</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Pittsburgh Penguins</td>
<td>PIT</td>
<td>65</td>
<td>36</td>
<td>29</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Calgary Flames</td>
<td>CGY</td>
<td>64</td>
<td>33</td>
<td>31</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>Washington Capitals</td>
<td>WAS</td>
<td>64</td>
<td>29</td>
<td>35</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Tampa Bay Lightning</td>
<td>TAM</td>
<td>63</td>
<td>25</td>
<td>38</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Colorado Avalanche</td>
<td>COL</td>
<td>65</td>
<td>33</td>
<td>32</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Ovechkin</td>
<td>WAS</td>
<td>64</td>
<td>43</td>
<td>38</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Lecavalier</td>
<td>TAM</td>
<td>63</td>
<td>32</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iginla</td>
<td>CGY</td>
<td>64</td>
<td>38</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spezza</td>
<td>OTT</td>
<td>59</td>
<td>25</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alfredsson</td>
<td>OTT</td>
<td>42</td>
<td>35</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datsyuk</td>
<td>DET</td>
<td>65</td>
<td>23</td>
<td>51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ribeiro</td>
<td>DAL</td>
<td>65</td>
<td>26</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zetterberg</td>
<td>DET</td>
<td>58</td>
<td>35</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>St Louis</td>
<td>TAM</td>
<td>63</td>
<td>21</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Richards</td>
<td>PHI</td>
<td>62</td>
<td>23</td>
<td>43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stastny</td>
<td>COL</td>
<td>50</td>
<td>19</td>
<td>33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"Good" players on winning teams

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

DB $\xrightarrow{Q} Q(DB)$
Basic idea
Basic idea

Link output parts to sets of input parts such that...
Basic idea

Whenever DB and DB' are same except for annotated inputs...
Basic idea

Whenever DB and DB' are same except for annotated inputs...
Basic idea

query results only differ at linked outputs
Basic idea

Can overapproximate!
Formalization

- Consider types:
  \[ T ::= \text{int} \mid \ldots \mid T \times T \mid T \text{ set} \]

- Translate:
  \[
  \begin{align*}
  P[\text{int}] &= \text{int} \times \text{'a set} \\
  P[T_1 \times T_2] &= P[T_1] \times P[T_2] \times \text{'a set} \\
  P[T \text{ set}] &= P[T] \text{ set} \times \text{'a set}
  \end{align*}
  \]

Annotations 'a represent "pointers" to sets of sources
Term Translation

Given $x_1: T_1, ..., x_n: T_n \vdash e : T$
Want $P[e]$ such that

$x_1: P[T_1], ..., x_n: P[T_n] \vdash P[e] : P[T]$

s.t. all "dependencies" are captured.

Simple cases:

\[
P[x] = x
\]

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

\[
P[\text{let } x = e_1 \text{ in } e_2] = \text{let } x = P[e_1] \text{ in } P[e_2]
\]
Pairs

\[ P[(e_1,e_2)] = (P[e_1], P[e_2], \emptyset) \]

\[ P[\pi_i(e)] = \text{let } (v,a) = P[e] \]
\[ (v_i', b) = \pi_i(v) \]
\[ \text{in } (v_i, a \cup b) \]
Sets

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

\[ P[e_1 \cup e_2] = \text{let } (v_1, a_1) = P[e_1] \]
\[ \quad (v_2, a_2) = P[e_2] \]
\[ \quad \text{in } (v_1 \cup v_2, a_1 \cup a_2) \]

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

\[ P[\bigcup\{e_2 \mid x \leftarrow e_1\}] = \text{let } (v, a) = P[e_1] \]
\[ \quad \text{in } (\bigcup\{P[e_2] \mid x \leftarrow v\}, a) \]
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:

```haskell
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:

  \[
  \text{whereprov} :: \text{Exp } e \; t' \rightarrow \\
  \quad \text{Exp } (P \; e \; (\text{Maybe } a)) \\\n  \quad (P \; t' \; (\text{Maybe } a)) \\
  \text{whereprov} \; (\text{Const } c) = \\
  \quad \text{Pair } (\text{Const } c) \; \text{Nothing} \\
  \text{whereprov} \; (\text{Var } x) = \text{Var } x \\
  \ldots
  \]
Other ways forward?

- Haskell: GADTs + type families/type-level computation + 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 :)

Thursday, November 25, 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 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]

CIA World Factbook

Facts about Belgium

There is an interesting note about land use in Belgium:

<table>
<thead>
<tr>
<th>Land use</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Land use</td>
</tr>
<tr>
<td>Note</td>
<td>includes Luxembourg (2005)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Text</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>arable land</td>
<td>27.42%</td>
<td></td>
</tr>
<tr>
<td>permanent crops</td>
<td>0.69%</td>
<td></td>
</tr>
<tr>
<td>other</td>
<td>71.89%</td>
<td></td>
</tr>
</tbody>
</table>

Mueller, Lindley

research on provenance, archity

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: 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 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 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 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