Symbolic Query Exploration

Pavel Grigorenko
(joint work with Margus Veanes,
Peli de Halleux and Nikolai Tillmann)

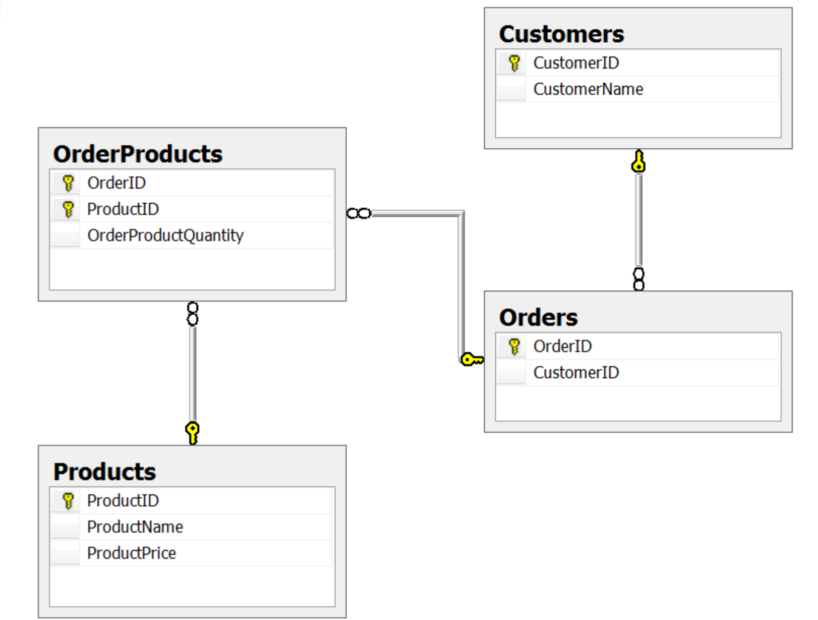
Institute of Cybernetics
Tallinn University of Technology

Theory Days at Mäetaguse
2 October 2009

Overview

- Motivation
  - Test data generation for SQL queries
  - Parameter generation for queries and store procedures
- Goal
  - Investigate model generation with Z3
- Approach
  - Satisfiability modulo background theory $\mathcal{T}^{\Sigma}$
  - Mapping SQL to $\mathcal{T}^{\Sigma}$

Sample DB



Customers

- 🔑 CustomerID
- CustomerName

OrderProducts

- 🔑 OrderID
- 🔑 ProductID
- OrderProductQuantity

Orders

- 🔑 OrderID
- CustomerID

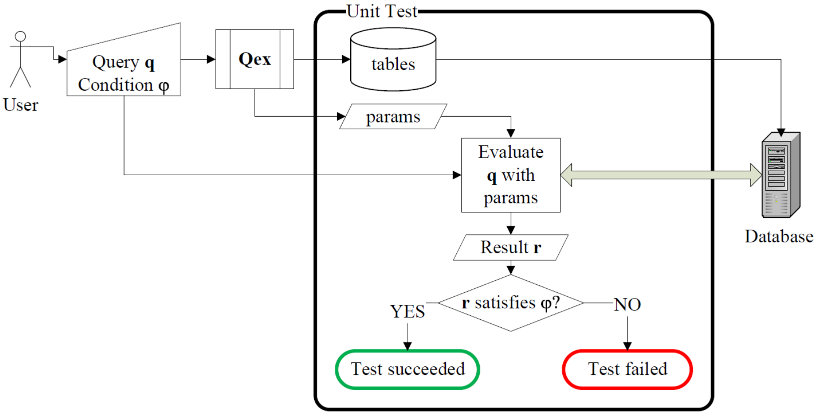
Products

- 🔑 ProductID
- ProductName
- ProductPrice

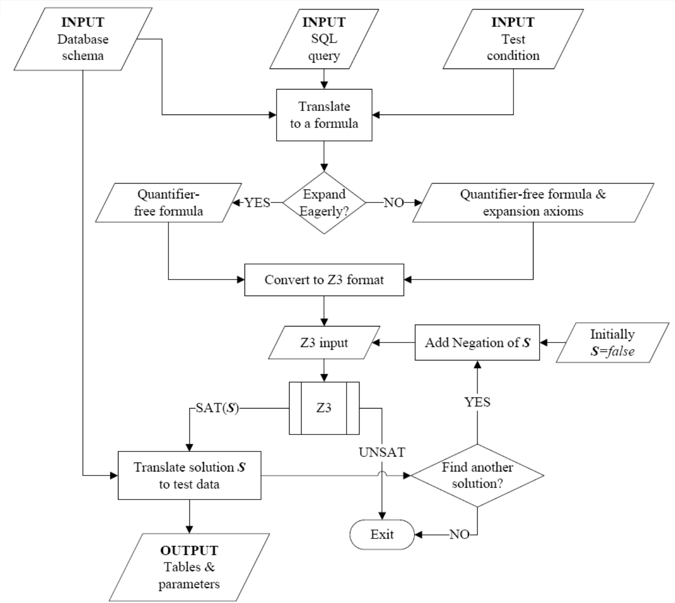
Query

```
SELECT C.CustomerID, SUM(OP.OrderProductQuantity
                         * P.ProductPrice)
FROM OrderProducts AS OP
JOIN Orders AS O ON OP.OrderID = O.OrderID
JOIN Products AS P ON OP.ProductID = P.ProductID
JOIN Customers AS C ON O.CustomerID = C.CustomerID
WHERE @value > 1
GROUP BY C.CustomerID
HAVING SUM(OP.OrderProductQuantity
           * P.ProductPrice) > 100 + @value
```

Unit test


Unit Test
User
Query q
Condition φ
Qex
tables
params
Evaluate
q with
params
Database
Result r
r satisfies φ?
YES
NO
Test succeeded
Test failed

Qex internal workflow



Expressions in $\mathcal{T}^{\Sigma}$

$$
\begin{array}{lcl}
T^{\sigma} & ::= & x^{\sigma} \mid \mathit{Default}^{\sigma} \mid \mathit{Ite}(T^{\mathbb{B}}, T^{\sigma}, T^{\sigma}) \mid \mathit{TheElementOf}(T^{\mathbb{S}(\sigma)}) \mid \\
 & & \pi_i(T^{\mathbb{T}(\sigma_0,\ldots,\sigma_i=\sigma,\ldots)}) \\
T^{\mathbb{T}(\sigma_0,\ldots,\sigma_k)} & ::= & \langle T^{\sigma_0}, \ldots, T^{\sigma_k} \rangle \\
T^{\mathbb{Z}} & ::= & k \mid T^{\mathbb{Z}} + T^{\mathbb{Z}} \mid k * T^{\mathbb{Z}} \mid \Sigma_i(T^{\mathbb{S}(\mathbb{T}(\sigma_0,\ldots,\sigma_i=\mathbb{Z},\ldots))}) \\
T^{\mathbb{R}} & ::= & r \mid T^{\mathbb{R}} + T^{\mathbb{R}} \mid k * T^{\mathbb{R}} \mid \Sigma_i(T^{\mathbb{S}(\mathbb{T}(\sigma_0,\ldots,\sigma_i=\mathbb{R},\ldots))}) \mid \mathit{AsReal}(T^{\mathbb{Z}}) \\
T^{\mathbb{B}} & ::= & \mathit{true} \mid \mathit{false} \mid \neg T^{\mathbb{B}} \mid T^{\mathbb{B}} \wedge T^{\mathbb{B}} \mid T^{\mathbb{B}} \vee T^{\mathbb{B}} \mid T^{\sigma} = T^{\sigma} \mid \\
 & & T^{\mathbb{S}(\sigma)} \subseteq T^{\mathbb{S}(\sigma)} \mid T^{\sigma} \in T^{\mathbb{S}(\sigma)} \mid T^{\mathbb{Z}} \leq T^{\mathbb{Z}} \mid T^{\mathbb{R}} \leq T^{\mathbb{R}} \\
T^{\mathbb{S}(\sigma)} & ::= & X^{\mathbb{S}(\sigma)} \mid \{T^{\sigma} \mid_{\bar{x}} T^{\mathbb{B}}\} \mid T^{\mathbb{S}(\sigma)} \cup T^{\mathbb{S}(\sigma)} \mid T^{\mathbb{S}(\sigma)} \cap T^{\mathbb{S}(\sigma)} \mid \\
 & & T^{\mathbb{S}(\sigma)} \setminus T^{\mathbb{S}(\sigma)} \\
F & ::= & T^{\mathbb{B}} \mid \exists x\, F \mid \exists X\, F
\end{array}
$$

Semantics

- S is a state for a term t such that $FV(t) \subseteq Dom(S)$
- For a given term t, t^S is the *interpretation* of t in S
- For a given formula φ, $S \models \varphi$ means that φ^S is *true*

$$
\begin{aligned}
Ite(\varphi, t_1, t_2)^S &= \begin{cases} t_1^S, & \text{if } S \models \varphi; \\ t_2^S, & \text{otherwise.} \end{cases} \\
\{t_0 \mid_{x^\sigma} \varphi\}^S &= \{t_0^{S \uplus \{x \mapsto a\}} : a \in \mathcal{U}^\sigma, S \uplus \{x \mapsto a\} \models \varphi\} \\
\Sigma_i(t_1)^S &= \sum_{a \in t_1^S} \pi_i(a)
\end{aligned}
$$

Semantics

- S is a state for a term t such that $FV(t) \subseteq Dom(S)$
- For a given term t, t^S is the *interpretation* of t in S
- For a given formula φ, $S \models \varphi$ means that φ^S is *true*

$$\begin{aligned}
Ite(\varphi, t_1, t_2)^S &= \begin{cases} t_1^S, & \text{if } S \models \varphi; \\ t_2^S, & \text{otherwise.} \end{cases} \\
\{t_0 \mid_{x^\sigma} \varphi\}^S &= \{t_0^{S \uplus \{x \mapsto a\}} : a \in \mathcal{U}^\sigma, S \uplus \{x \mapsto a\} \models \varphi\} \\
\Sigma_i(t_1)^S &= \sum_{a \in t_1^S} \pi_i(a)
\end{aligned}$$

Example

Integer multiplication:

$$n * m \stackrel{\text{def}}{=} \Sigma_0(\{\langle m, x\rangle \mid 0 \leq x < n\}) = \sum_{x=0}^{n-1} \pi_0(\langle m, x\rangle) = \sum_{x=0}^{n-1} m$$

Semantics

Tables are *bags* (multisets)!

A bag b with elements $\{a_i\}_{i<n}$ each having *multiplicity* $m_i > 0$ in b for $i < n$, is represented as a set of pairs $\{\langle a_i, m_i\rangle\}_{i<n}$, thus having the sort $\mathbb{S}(\mathbb{T}(\sigma, \mathbb{Z}))$ for some basic sort σ (*domain sort of b*).
Let $\mathbb{M}(\sigma)$ be the type $\mathbb{S}(\mathbb{T}(\sigma, \mathbb{Z}^+))$ with the *constraint*:

$$\forall X^{\mathbb{M}(\sigma)}\, \forall x^\sigma\, y^\sigma\, ((x \in X \wedge y \in X \wedge x.0 = y.0) \Rightarrow x.1 = y.1).$$

Semantics

Tables are *bags* (multisets)!

A bag b with elements $\{a_i\}_{i<n}$ each having *multiplicity* $m_i > 0$ in b for $i < n$, is represented as a set of pairs $\{\langle a_i, m_i\rangle\}_{i<n}$, thus having the sort $\mathbb{S}(\mathbb{T}(\sigma, \mathbb{Z}))$ for some basic sort σ (*domain sort of b*).
Let $\mathbb{M}(\sigma)$ be the type $\mathbb{S}(\mathbb{T}(\sigma, \mathbb{Z}^+))$ with the *constraint*:

$$\forall X^{\mathbb{M}(\sigma)}\, \forall x^\sigma\, y^\sigma\, ((x \in X \wedge y \in X \wedge x.0 = y.0) \Rightarrow x.1 = y.1).$$

$$\begin{aligned}
AsBag(Y^{\mathbb{S}(\sigma)}) &\stackrel{\text{def}}{=} \{\langle y, 1\rangle \mid y \in Y\} \\
AsSet(X^{\mathbb{M}(\sigma)}) &\stackrel{\text{def}}{=} \{y.0 \mid y \in X\} \\
\Sigma_i^{\mathrm{b}}(X^{\mathbb{M}(\mathbb{T}(\sigma_0,\ldots,\sigma_i,\ldots))}) &\stackrel{\text{def}}{=} \Sigma_0(\{\langle x.1 * x.0.i, x.0\rangle \mid x \in X\})
\end{aligned}$$

Bags

Example

$$\begin{aligned}
\varphi[x^{\mathbb{Z}}] &= x < 4 \\
q[X^{\mathbb{M}(\mathbb{T}(\mathbb{Z},\mathbb{Z},\mathbb{Z}))}] &= \{\langle x.0.0, \Sigma^{\mathrm{b}}_1(\{y \mid y \in X \wedge x.0.0 = y.0.0 \wedge \varphi[y.0.2]\})\rangle \\
&\qquad \mid x \in X \wedge \varphi[x.0.2]\} \\
t &= \{\langle\langle 0,2,1\rangle, 2\rangle, \langle\langle 1,2,3\rangle, 1\rangle, \langle\langle 1,2,4\rangle, 1\rangle\}
\end{aligned}$$

$$\begin{aligned}
q[t] &= \{\langle x.0.0, \Sigma^{\mathrm{b}}_1(\{y \mid y \in t \wedge x.0.0 = y.0.0 \wedge \varphi[y.0.2]\})\rangle \\
&\qquad \mid x \in t \wedge \varphi[x.0.2]\} \\
&= \{\langle 0, \Sigma^{\mathrm{b}}_1(\{y \mid y \in t \wedge 0 = y.0.0 \wedge \varphi[y.0.2]\})\rangle, \\
&\qquad \langle 1, \Sigma^{\mathrm{b}}_1(\{y \mid y \in t \wedge 1 = y.0.0 \wedge \varphi[y.0.2]\})\rangle\} \\
&= \{\langle 0, \textstyle\sum_{a \in \{\langle\langle 0,2,1\rangle,2\rangle\}} \pi_1(a) * \pi_1(\pi_0(a))\rangle, \\
&\qquad \langle 1, \textstyle\sum_{a \in \{\langle\langle 1,2,3\rangle,1\rangle\}} \pi_1(a) * \pi_1(\pi_0(a))\rangle\} \\
&= \{\langle 0, 4\rangle, \langle 1, 2\rangle\}
\end{aligned}$$

From SQL to $\mathcal{T}^{\Sigma}$

Translation

$\mathbf{Q} : \mathrm{SQL} \rightarrow \mathcal{T}^{\Sigma}$

Strings

- All strings have a maximum length k; an encoding of a k-string is as a k-tuple of integers, each character a is encoded as an integer $c(a)$. For a string $a_0 \cdots a_l$, $l < k$ the encoding is $\langle c(a_0), \ldots, c(a_l), 0, \ldots, 0 \rangle$.
- For a collection D of strings, encode as $|D|$-enums.

Nullable values

Given a basic sort σ, let $?\sigma$ be the sort $\mathbb{T}(\sigma, \mathbb{B})$ with the constraint $\forall x^{?\sigma}\,(x.1 = \mathit{false} \Rightarrow x.0 = \mathit{Default}^{\sigma})$ and $\mathit{null}^{?\sigma} \stackrel{\mathrm{def}}{=} \mathit{Default}^{\mathbb{T}(\sigma,\mathbb{B})}$.
Operations that are defined for σ are lifted to $?\sigma$.
For example, for a numeric sort σ,

$$x^{?\sigma} + y^{?\sigma} \stackrel{\mathrm{def}}{=} \mathit{Ite}(x.1 \wedge y.1, \langle x.0 + y.0, \mathit{true} \rangle, \mathit{null}^{?\sigma}).$$

From SQL to $\mathcal{T}^{\Sigma}$

Select clauses

$$\mathbf{Q}(\texttt{SELECT l FROM t}) \overset{\text{def}}{=} \{\langle\langle x.0.l_0, \ldots, x.0.l_n\rangle, M(x)\rangle \mid x \in \mathbf{Q}(\texttt{t})\}$$

$$\text{where } M(x) = \Sigma_0(\{\langle y.1, y\rangle \mid y \in \mathbf{Q}(\texttt{t}) \wedge \bigwedge_{i=0}^{n} y.0.l_i = x.0.l_i\})$$

$$\mathbf{Q}(\texttt{SELECT DISTINCT l FROM t}) \overset{\text{def}}{=} \mathit{AsBag}(\mathit{AsSet}(\mathbf{Q}(\texttt{SELECT l FROM t})))$$

$$\mathit{AsSet}(\mathbf{Q}(\texttt{SELECT l FROM t})) = \{\langle y.l_0, \ldots, y.l_n\rangle \mid y \in \mathit{AsSet}(\mathbf{Q}(\texttt{t}))\}$$

From SQL to $\mathcal{T}^{\Sigma}$

Join clauses

$$\mathbf{Q}(\texttt{t1 INNER JOIN t2 ON c}) \overset{\text{def}}{=}$$
$$\{\langle x_1.0 \times x_2.0, x_1.1 * x_2.1\rangle \mid x_1 \in \mathbf{Q}(\texttt{t1}) \wedge x_2 \in \mathbf{Q}(\texttt{t2}) \wedge \mathbf{Q}(\texttt{c})[x_1.0, x_2.0]\}$$

$$x \times y \overset{\text{def}}{=} \langle \pi_0(x), \ldots, \pi_{m-1}(x), \pi_0(y), \ldots, \pi_{n-1}(y)\rangle$$

$$\mathit{AsSet}(\mathbf{Q}(\texttt{t1 INNER JOIN t2 ON c})) =$$
$$\{y_1 \times y_2 \mid y_1 \in \mathit{AsSet}(\mathbf{Q}(\texttt{t1})) \wedge y_2 \in \mathit{AsSet}(\mathbf{Q}(\texttt{t2})) \wedge \mathbf{Q}(\texttt{c})[y_1, y_2]\}$$

From SQL to $\mathcal{T}^{\Sigma}$

Grouping and aggregates

$$\begin{aligned}
&\texttt{t} = \texttt{SELECT a, SUM(b) AS d FROM t1 WHERE c1}\\
&\mathbf{Q}(\texttt{t GROUP BY a HAVING c2}) \overset{\text{def}}{=} \mathit{AsBag}(\{z \mid z \in G \wedge \mathbf{Q}(\texttt{c2})[z]\})\\
&\quad \text{where } G = \{\langle x.0.0, \Sigma_1^{\text{b}}(\{y \mid y \in \mathbf{Q}(\texttt{t}) \wedge y.0.0 = x.0.0\})\rangle \mid x \in \mathbf{Q}(\texttt{t})\}\\
&\mathit{Sum} \overset{\text{def}}{=} \Sigma_i^{\text{b}}\\
&\mathit{Count} \overset{\text{def}}{=} \Sigma_1\\
&\mathit{Min}(X^{\mathbb{S}(\sigma)}) \overset{\text{def}}{=} \mathit{TheElementOf}(\{y \mid y \in X \wedge \{z \mid z \in X \wedge z < y\} = \emptyset\})
\end{aligned}$$

From SQL to $\mathcal{T}^{\Sigma}$

Union

$$\mathbf{Q}(\texttt{q1 UNION q2}) \overset{\text{def}}{=} \mathit{AsBag}(\mathit{AsSet}(\mathbf{Q}(\texttt{q1})) \cup \mathit{AsSet}(\mathbf{Q}(\texttt{q2}))).$$

From SQL to $\mathcal{T}^{\Sigma}$

Simplifications

$$\begin{array}{rcl}
\mathit{AsSet}(\mathit{AsBag}(X^{\mathbb{S}(\sigma)})) & = & X \\
\Sigma_i^{\mathrm{b}}(\mathit{AsBag}(X^{\mathbb{S}(\sigma)})) & = & \Sigma_i(X) \\
\mathit{AsSet}(\{t \mid \varphi\}^{\mathbb{M}(\sigma)}) & = & \{t.0 \mid \varphi\} \\
\pi_i(\langle t_0, \ldots, t_i, \ldots \rangle) & = & t_i
\end{array}$$

Model Generation with SMT

Model Generation in $\mathcal{T}_{\mathbf{Q}}^{\Sigma}$

Given a quantifier free formula $\varphi[X]$ in $\mathcal{T}^{\Sigma}$, and a query q, decide if $\psi = \varphi[\mathbf{Q}(q)]$ is satisfiable, and if ψ is satisfiable, generate a model of ψ.

Two approaches to deal with *comprehensions* and *summations* in Z3

- Eager expansion
  - Provide finite bounds for tables and unwind
- Lazy expansion
  - Add expansion rules as axioms

Expansion of terms

1. Given a query $q[X]$
2. Create a symbolic table $t_X = \{\langle x_1, m_1\rangle, \ldots, \langle x_k, m_k\rangle\}$ (k and m_i are fixed, x_i are variables)
3. Expand $q[t_X]$ to $\varphi = \mathbf{Exp}(q[t_X])$
4. Generate a model for φ (if φ is sat.)
5. Increase k and m_i, repeat 1..4.

Eager expansion

Set describer

Consider a formula $\psi[\overline{X}]$ as an instance of the model generation problem, where every X in $\overline{X}$ is a bag variable.

- The constant $\mathit{Empty}^{\mathbb{S}(\sigma)}$ is a set describer.
- If $t^{\mathbb{S}(\sigma)}$ is a set describer then so is the term $\mathit{Set}(\varphi^{\mathbb{B}}, u^{\sigma}, t)$.

Given a state S for $\mathit{Set}(\varphi, u, t)$, the interpretation in S is,

$$\mathit{Set}(\varphi, u, t)^{S} = \mathit{Ite}(\varphi, \{u\}, \emptyset)^{S} \cup t^{S}, \quad \mathit{Empty}^{S} = \emptyset.$$

X is fixed in $\overline{X}$ and t_X is the set describer:

$$\mathit{Set}(\mathit{true}, \langle x_1, m_1 \rangle, \ldots \mathit{Set}(\mathit{true}, \langle x_k, m_k \rangle, \mathit{Empty}) \ldots)$$

where k and all the m_i's are some positive integer constants and each x_i is a variable.

Expansion rules

Expansion rule for *comprehensions*

$$\begin{aligned}
\mathbf{Exp}(\{t \mid_x x \in r \wedge \varphi\}) &\overset{\text{def}}{=} \mathbf{ExpC}(t, x, \mathbf{Exp}(r), \varphi) \\
\mathbf{ExpC}(t, x, \mathit{Empty}, \varphi) &\overset{\text{def}}{=} \mathit{Empty} \\
\mathbf{ExpC}(t[x], x, \mathit{Set}(\gamma, u, \mathit{rest}), \varphi[x]) &\overset{\text{def}}{=} \mathit{Set}(\gamma \wedge \mathbf{Exp}(\varphi[u]), \mathbf{Exp}(t[u]), \\
&\qquad\qquad \mathbf{ExpC}(t, x, \mathit{rest}, \varphi))
\end{aligned}$$

Expansion rules

Expansion rule for *comprehensions*

$$
\begin{aligned}
\mathbf{Exp}(\{t \mid_x x \in r \wedge \varphi\}) &\overset{\text{def}}{=} \mathbf{ExpC}(t, x, \mathbf{Exp}(r), \varphi)\\
\mathbf{ExpC}(t, x, \mathit{Empty}, \varphi) &\overset{\text{def}}{=} \mathit{Empty}\\
\mathbf{ExpC}(t[x], x, \mathit{Set}(\gamma, u, \mathit{rest}), \varphi[x]) &\overset{\text{def}}{=} \mathit{Set}(\gamma \wedge \mathbf{Exp}(\varphi[u]), \mathbf{Exp}(t[u]),\\
&\qquad\qquad \mathbf{ExpC}(t, x, \mathit{rest}, \varphi))
\end{aligned}
$$

Expansion rule for Σ_i

$$
\begin{aligned}
\mathbf{Exp}(\Sigma_i(t)) &\overset{\text{def}}{=} \mathbf{Sum}_i(\mathbf{Exp}(t), \mathit{Empty})\\
\mathbf{Sum}_i(\mathit{Empty}, s) &\overset{\text{def}}{=} 0\\
\mathbf{Sum}_i(\mathit{Set}(\gamma, u, \mathit{rest}), s) &\overset{\text{def}}{=} \mathit{Ite}(\gamma \wedge u \notin s, \pi_i(u), 0) \ +\\
&\quad + \mathbf{Sum}_i(\mathit{rest}, \mathit{Set}(\gamma, u, s))
\end{aligned}
$$

Lazy expansion

In addition to a quantifier free formula ψ, universally quantified *axioms* are provided in the form:

$$(\forall \bar{x}(\alpha),\ pat_\alpha),\ \ FV(\alpha) = FV(pat_\alpha) = \bar{x}$$

If ψ contains a subterm t and there exists a substitution θ such that $t = pat_\alpha\theta$, i.e., *t matches the pattern pat*$_\alpha$, then ψ is replaced during proof search by (a reduction of) $\psi \wedge \alpha\theta$.

Lazy sum

Axioms

$$\begin{aligned} \alpha_1 &= \forall s(\mathbf{Sum}_i(\mathit{Empty}, s) = 0) \\ \mathit{pat}_{\alpha_1} &= \mathbf{Sum}_i(\mathit{Empty}, s) \\ \alpha_2 &= \forall b\, u\, r\, s\, (\mathbf{Sum}_i(\mathit{Set}(b, u, r), s) = \\ &\qquad \mathit{Ite}(b \wedge u \notin s, \pi_i(u), 0) + \mathbf{Sum}_i(r, \mathit{Ite}(b, \{u\}, \emptyset) \cup s)) \\ \mathit{pat}_{\alpha_2} &= \mathbf{Sum}_i(\mathit{Set}(b, u, r), s) \end{aligned}$$

Example

$$\begin{aligned} & x \leq \mathbf{Sum}_1(\mathit{Set}(\mathit{true}, \langle 1, y \rangle, \mathit{Set}(\mathit{true}, \langle 1, z \rangle, \mathit{Empty})), \emptyset) \\ \overset{\alpha_2}{\Longrightarrow}\; & x \leq y + \mathbf{Sum}_1(\mathit{Set}(\mathit{true}, \langle 1, z \rangle, \mathit{Empty}), \{\langle 1, y \rangle\}) \\ \overset{\alpha_2}{\Longrightarrow}\; & x \leq y + \mathit{Ite}(z \neq y, z, 0) + \mathbf{Sum}_1(\mathit{Empty}, \{\langle 1, y \rangle, \langle 1, z \rangle\}) \\ \overset{\alpha_1}{\Longrightarrow}\; & x \leq y + \mathit{Ite}(z \neq y, z, 0) \end{aligned}$$

Experiments

```
q1: SELECT C.CustomerID, O.OrderID
FROM Orders AS O
JOIN Customers AS C ON
O.CustomerID = C.CustomerID
WHERE O.CustomerID > 2 AND
O.OrderID < 15
```

```
q2: SELECT C.CustomerID,
Count(O.OrderID)
FROM Orders AS O
JOIN Customers AS C ON
O.CustomerID = C.CustomerID
GROUP BY C.CustomerID HAVING
Count(O.OrderID) > 1
```

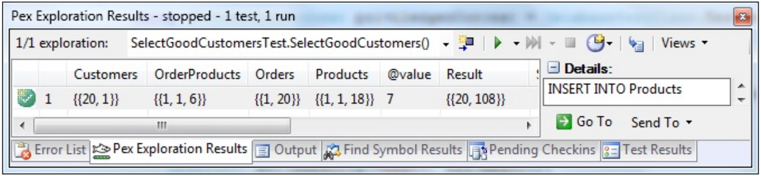
```
q3: DECLARE @value AS INT;
SELECT C.CustomerID, SUM(OP.OrderProductQuantity * P.ProductPrice)
FROM OrderProducts AS OP
JOIN Orders AS O ON OP.OrderID = O.OrderID
JOIN Products AS P ON OP.ProductID = P.ProductID
JOIN Customers AS C ON O.CustomerID = C.CustomerID
WHERE @value > 1
GROUP BY C.CustomerID
HAVING SUM(OP.OrderProductQuantity * P.ProductPrice) > 100 + @value
```

query	cond	*k*	check	t_{exp}	t_{z3}
q1	$res \neq \emptyset$	1	sat	.03	.001
		2	sat	.05	.005
		3	sat	.3	.02
		4	sat	1.4	.13
	$res = \emptyset$	1	sat	.03	.001
		2	sat	.05	.006
		3	sat	.3	.12
		4	sat	1.4	2

query	cond	*k*	check	t_{exp}	t_{z3}
q1	$\|res\| = 5$	1	unsat	.03	.001
		2	unsat	.05	.01
		3	unsat	.3	.16
		4	unsat	1.4	10
		5	sat	8.4	1.6
q2	$res \neq \emptyset$	1	unsat	.03	.001
		2	sat	.7	.006
		3	sat	26	.03
q3	$res \neq \emptyset$	1	sat	.34	.001
		2	sat	30	.03

Implementation

- ***Qex*** written in C#, uses ***T-SQL*** syntax for input queries
- Integration with ***Pex***



Pex Exploration Results - stopped - 1 test, 1 run

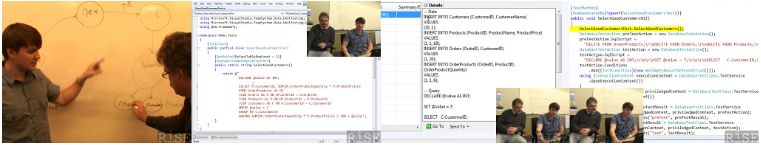
1/1 exploration: SelectGoodCustomersTest.SelectGoodCustomers()

		Customers	OrderProducts	Orders	Products	@value	Result
✓	1	{{20, 1}}	{{1, 1, 6}}	{{1, 20}}	{{1, 1, 18}}	7	{{20, 108}}

Details:

INSERT INTO Products

- Project homepage *http://research.microsoft.com/qex*
- Check ***Qex*** demo at *http://channel9.msdn.com/posts/Peli/Qex-Symbolic-Query-Exploration/*



Future work

- **Lazy** expansion (done?)
- Improved Database **constraint** support
- Queries with **side-effects**
- Improved **datatype** support
- Query **optimizations**

Future work

- **Lazy** expansion (done?)
- Improved Database **constraint** support
- Queries with **side-effects**
- Improved **datatype** support
- Query **optimizations**

Thanks! Questions?