Sound, Complete, and Tractable Linearizability Monitoring for Concurrent Collections

Michael Emmi  SRI International
Constantin Enea  Université Paris 7
Concurrent Objects

Abstract data type (Queue)

- enq: 1
- enq: 2
- deq: 1
- deq: 2

Blocking reference implementation

- enq: 1
- (blocked) enq: 2
- deq: 1
- deq: 2

Efficient nonblocking implementation

- enq: 1
- deq: 2
- deq: 1
- enq: 2
Linearizability

Effects of each invocation appear to occur instantaneously

Execution history

```
<table>
<thead>
<tr>
<th>enq: 1</th>
<th>deq: 2</th>
<th>deq: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enq: 2</td>
<td></td>
</tr>
</tbody>
</table>
```

Linearization admitted by Queue type

```
e:2  e:1  d:2  d:1
```

**Theorem** (Herlihy and Wing, 1990)
Linearizability implies observational refinement

**Theorem** (Filipovic et al, 2010)
Observational refinement implies linearizability
Theorem (Gibbons and Korach, 1997)
Checking linearizability is NP-hard
Result
Replace enumeration by monotonic deductive inference

**Execution history / partial linearization**

<table>
<thead>
<tr>
<th>enq: 1</th>
<th>deq: 2</th>
<th>deq: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>enq: 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Queue rule**

<table>
<thead>
<tr>
<th>if</th>
<th>d:X</th>
<th>d:Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>then</td>
<td>e:X</td>
<td>e:Y</td>
</tr>
</tbody>
</table>

**Theorem**
Checking linearizability is polynomial-time for *collection types*
Approach

Reduction to logical satisfiability problem

Theorem (Dowling and Gallier, 1984)

Horn satisfiability is solvable in linear time
Logical Characterization

Queue rules

\[
\begin{align*}
\text{if} & \quad d:X \quad d:Y \\
\text{then} & \quad e:X \quad e:Y
\end{align*}
\]

Execution history

\[
\begin{array}{ccc}
\text{e:1} & \text{d:2} & \text{d:1} \\
\text{e:2}
\end{array}
\]

Total order axioms

\[
\forall x \forall y \forall z (x < y \land y < z \Rightarrow x < z)
\]
\[
\forall x \forall y (x = y \lor x < y \lor y < x)
\]
\[
\ldots
\]

Queue axioms

\[
\forall x_1 \forall x_2 \forall y_1 \forall y_2 (\text{match}(x_1, x_2) \land \text{match}(y_1, y_2) \land x_1 < y_1 \Rightarrow x_2 < y_2)
\]
\[
\land \ldots
\]

Ground formula

\[
e:1 < d:2 \land d:2 < d:1 \land \text{match}(e:1, d:1) \land \text{match}(e:2, d:2)
\]
\[
\land \ldots
\]

Theorem

Linearizability equivalent to satisfiability for \textit{unambiguous} histories
Hornification

Original clause
A < B ∨ B < C ∨ ¬(A < B)

Translated clauses
A < B ∨ ¬(C < B) ∨ ¬(A < B)
¬(B < A) ∨ B < C ∨ ¬(A < B)
¬(B < A) ∨ ¬(C < B) ∨ ¬(A < B)
¬(B < A) ∨ ¬(C < B) ∨ B < A

Theorem
Translation is equisatisfiable and polynomial-time computable
Collection Types

Parametricity
\{3,5\} \text{ deq:3 } \{5\}
\{7,5\} \text{ deq:7 } \{5\}

Locality
\{3,5\} \text{ deq:3 } \{5\}
\{3,5\} \text{ size:2 } \{3,5\}

Value invariance
\{3,5\} \text{ deq:3 } \{5\}
\{3,5\} \text{ sum:8 } \{

Reducibility
\{3,5,7\} \text{ d:3; d:5; d:7 } \{
\text{2-reducibility}
\{3,5\} \text{ d:3; d:5 } \{
\{3,7\} \text{ d:3; d:7 } \{
\{5,7\} \text{ d:5; d:7 } \{

Theorem
Restriction to unambiguous histories is sound for collections
Bounded Violations

Theorem
Collections have bounded minimal violations
Logical Representation

Minimal violations

\[
\begin{align*}
d &: X & \rightarrow & e &: X \\
\rightarrow & \quad & e &: X & \rightarrow e &: Y \\
\rightarrow & \quad & d &: Y & \rightarrow d &: X
\end{align*}
\]

Quantified conjunction of negations

\[
\forall x_1 \forall x_2 \forall y_1 \forall y_2 \\
\neg (m(x_1, x_2) \land x_2 < x_1) \\
\land \neg (m(x_1, x_2) \land m(y_1, y_2) \land x_1 < y_1 \land y_2 < x_2) \\
\land \ldots
\]

Theorem

Collections have sound logical specifications

Corollary

Linearizability is PTIME computable for collections
Empirical Evaluation

Performance
Over 10 histories with 10K steps each

Scalability
Normalized by (collection size)^2

Observation
Orders of magnitude speedup
Related Work

Exponential enumeration
Wing and Gong, 1993. *Testing and Verifying Concurrent Objects*

NP Hardness

Asymptotically-equivalent optimizations
Shacham et al, 2011. *Testing atomicity of composed concurrent operations*
Horn et al, 2015. *Faster linearizability checking via P-compositionality*
Lowe, 2016. *Testing for linearizability*

Tractable approximation
Bouajjani et al, 2015. *Tractable refinement checking for concurrent objects*

Logical characterization
Emmi et al, 2015. *Monitoring refinement via symbolic reasoning*

Logical specification inference
Emmi et al, 2016. *Symbolic abstract data type inference*