Fast and Lean
Immutable Data Structures

Michael Steindorfer
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
Johannes Kepler University, Linz, Austria
set.add(34);
```java
set.add(34);

newSet = set.clone();
newSet.add(0);
```

<table>
<thead>
<tr>
<th>0</th>
<th>2</th>
<th>4098</th>
<th>32</th>
<th>...</th>
<th>34</th>
<th>...</th>
</tr>
</thead>
</table>

Full Copy on Write
Goal

Mimimize Performance Gap
Focus On

Scalable and Efficient Immutable General Purpose Data Structures (on the JVM)
Data Driven & Experimental
Hash Array Mapped Trie (HAMT)

Prefix Trees Encoding Hash Codes
Result #1
Cache Conscious, Canonical, and Compact HAMT
Result #2

Code Specialization for Memory Efficiency
2GB
Result #3

Heterogeneous Optimizations
Result #4

Performance Modeling of Complex, Cross-Cutting Optimization Strategies
\texttt{x.equals(y)} \iff \texttt{x == y}

Maximal Sharing a.k.a. Hash-Consing
Ongoing Research

A Product Family of Collection Data Structures
Future Work

Dynamic Adaptability of Collection Data Structures