

# Atomic Dynamic Upgrades Using Software Transactional Memory

Luís Pina      João Cachopo  
luis.pina@ist.utl.pt      joao.cachopo@ist.utl.pt

INESC-ID/Instituto Superior Técnico

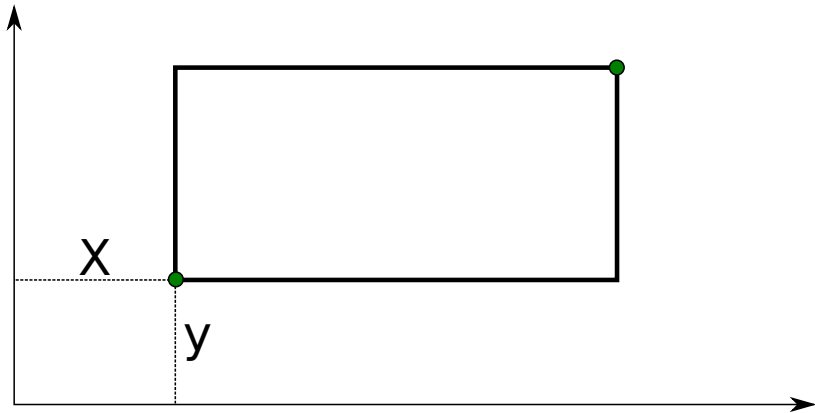
June 03, 2012

Fourth Workshop on Hot Topics in Software Upgrades

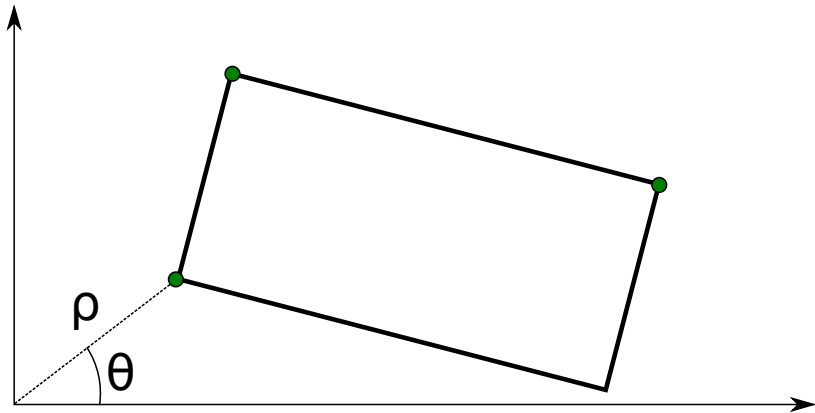
# Atomic Dynamic Upgrades

- Immediate upgrade semantics
- Lazy state migration
- Low disruption when upgrading

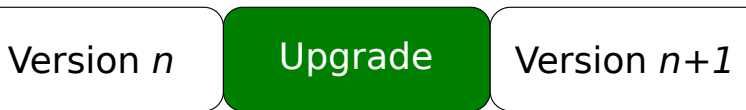
## Application Example



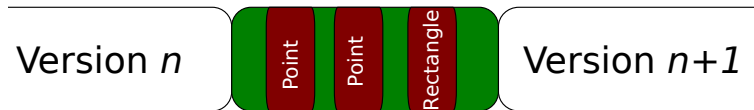
## Upgrade Example



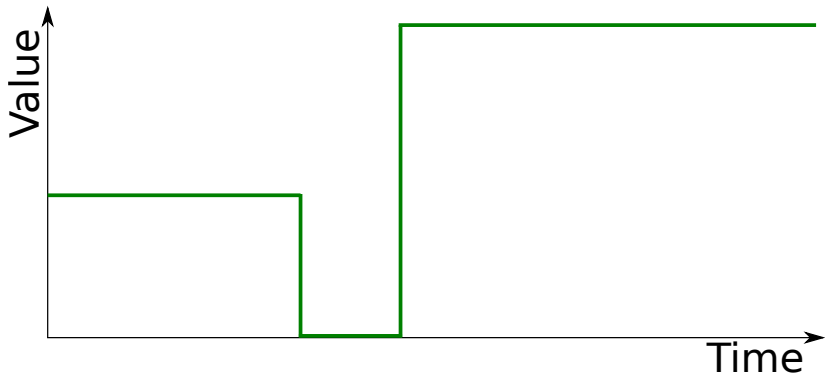
## Immediate Semantics



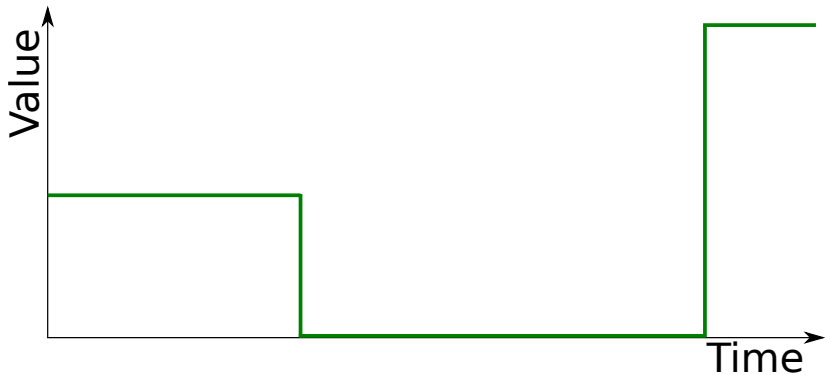
## Immediate Semantics



## Immediate Semantics

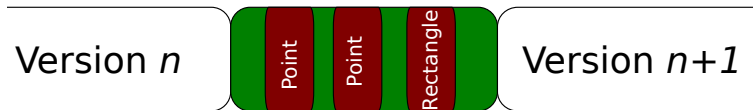


## Immediate Semantics





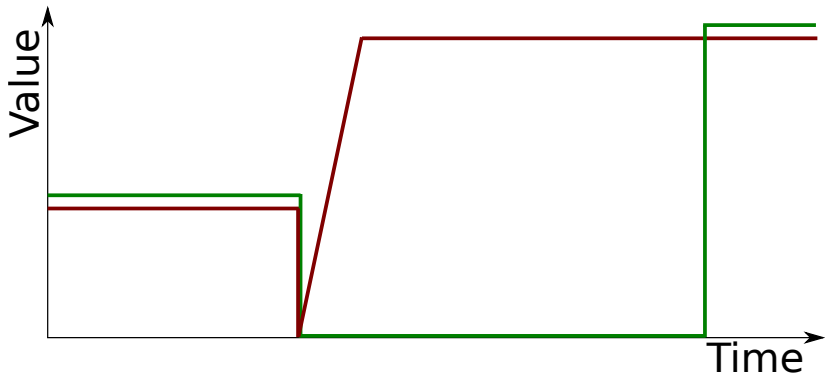
## Lazy Semantics



## Lazy Semantics



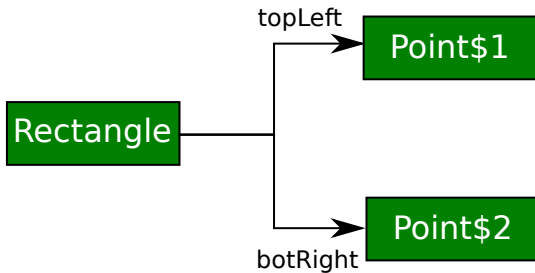
## Lazy Semantics



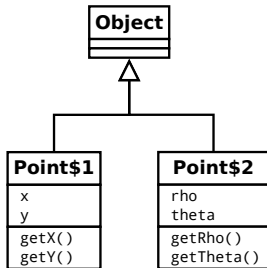
## Conversion Ordering



## Conversion Ordering



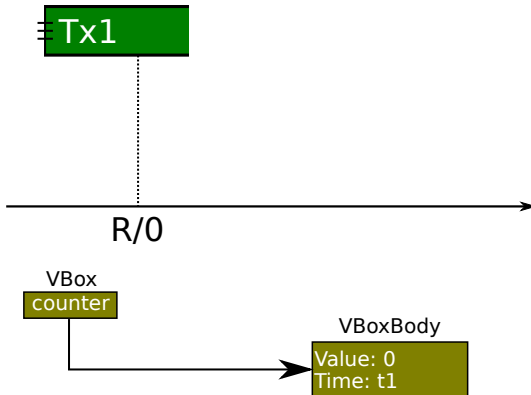
## Conversion Ordering



Keep old versions

# Java **Versioned** Software Transactional Memory

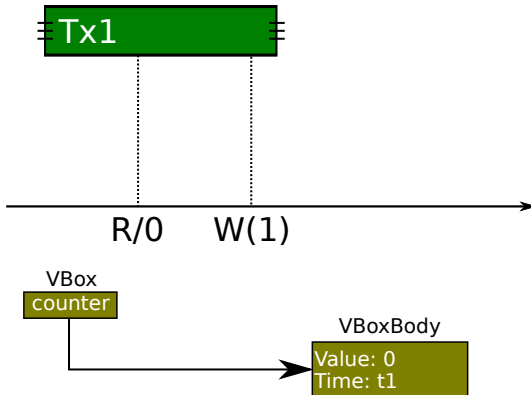
Shared Counter





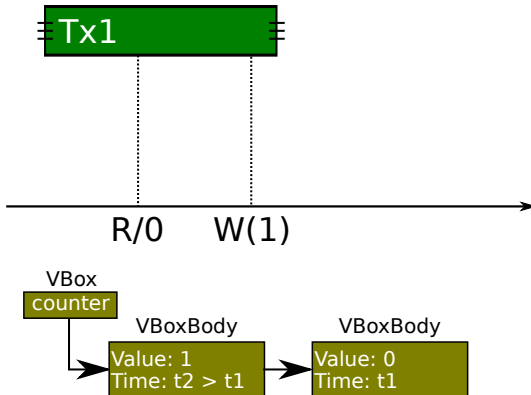
# Java **Versioned** Software Transactional Memory

Shared Counter



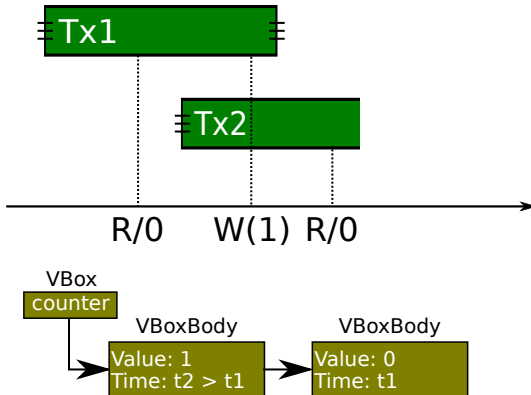
# Java **Versioned** Software Transactional Memory

Shared Counter



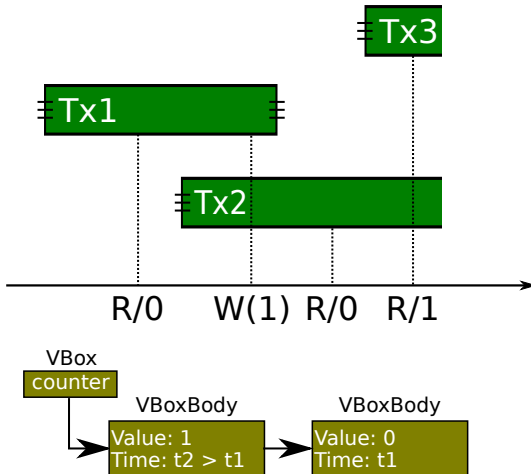
# Java **Versioned** Software Transactional Memory

Shared Counter



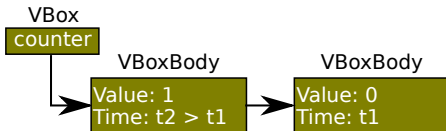
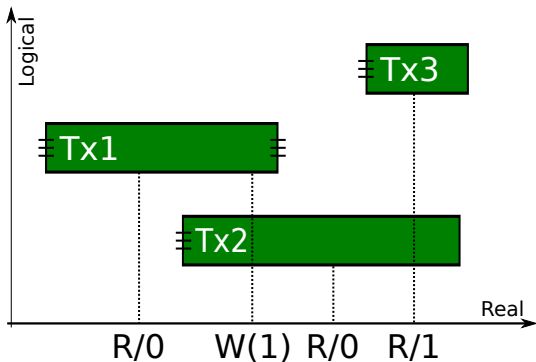
# Java **Versioned** Software Transactional Memory

Shared Counter



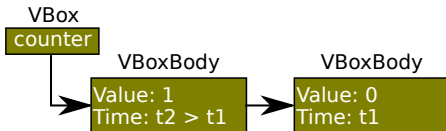
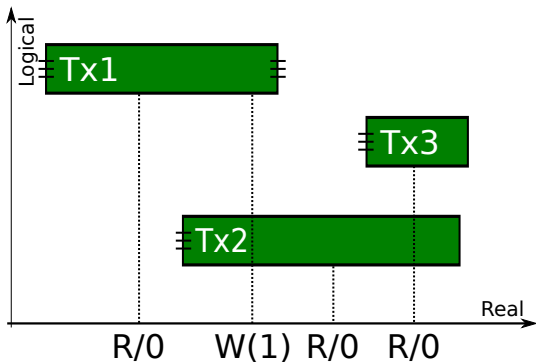
# Java **Versioned** Software Transactional Memory

Shared Counter

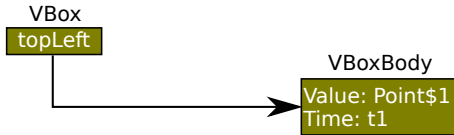
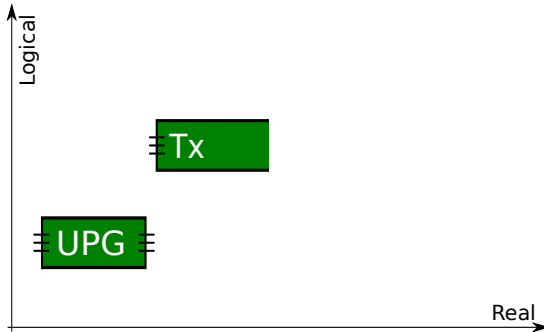


# Java **Versioned** Software Transactional Memory

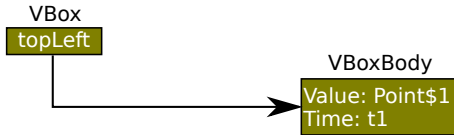
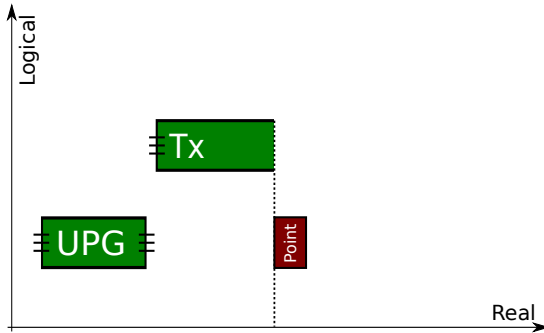
Shared Counter



# Atomic Upgrade Semantics

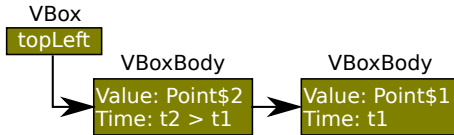
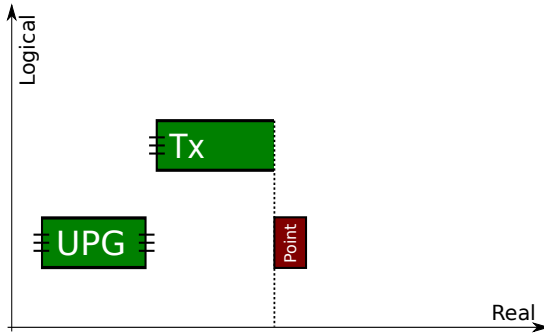


# Atomic Upgrade Semantics

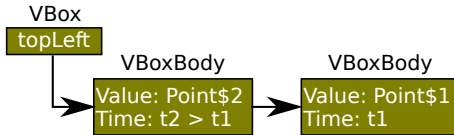
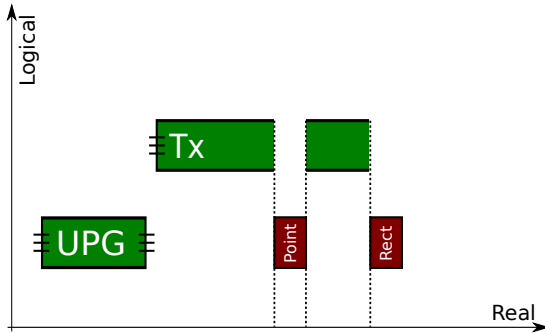




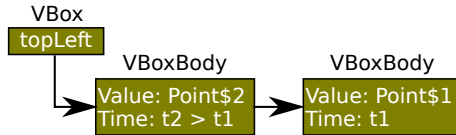
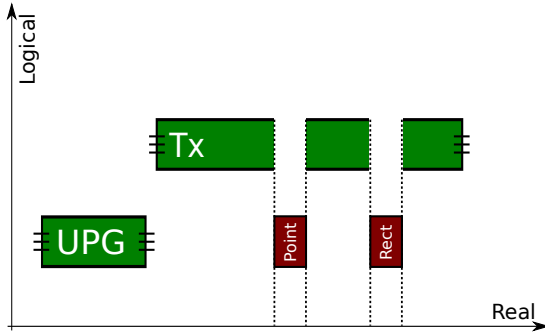
# Atomic Upgrade Semantics



# Atomic Upgrade Semantics



# Atomic Upgrade Semantics



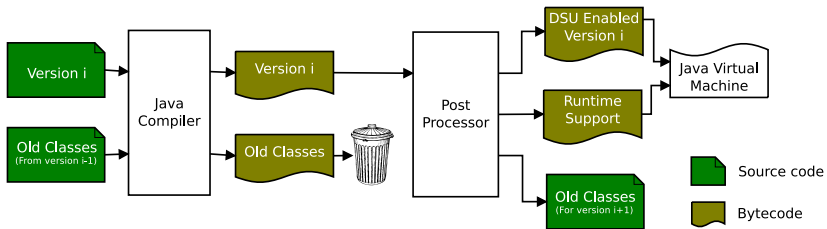
# Implementation

## Upgradable Types

Upgradable Code (e.g. Point)
Execution Platform (e.g. java.lang.String)

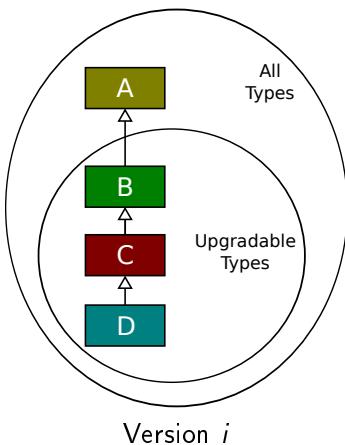
# Implementation

## Development Process



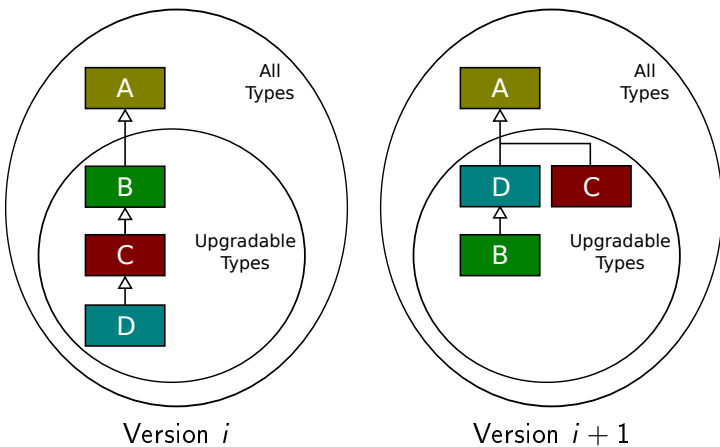
# Implementation

Flexibility



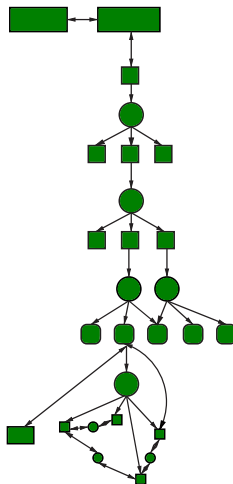
# Implementation

## Flexibility



## Experimental Evaluation

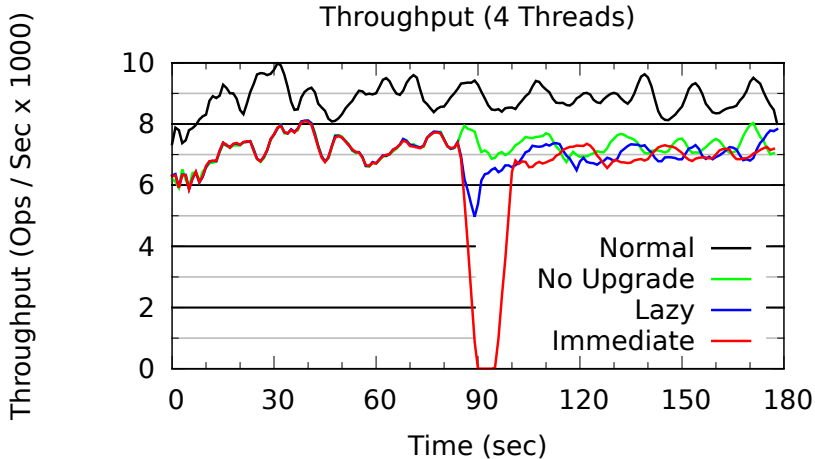
## STMBench7





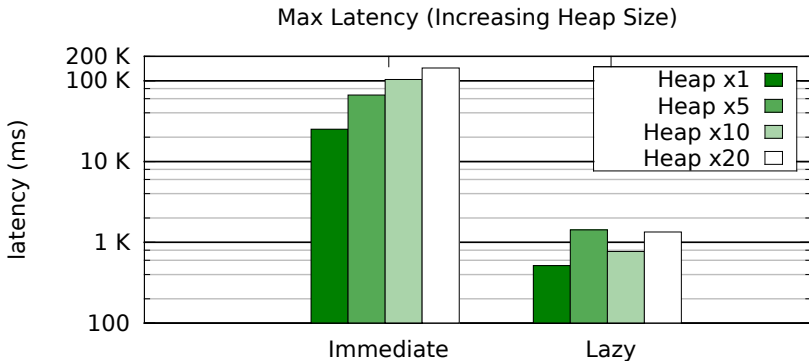
# Experimental Evaluation

Throughput



# Experimental Evaluation

Latency



# Conclusion

- Immediate semantics
- Lazy implementation
- Unaffected maximum latency

Thank you!

Q&A