

Range Fixes and Their Application on Software Configuration

Yingfei Xiong, Peking University, 2012

Based on an ICSE'12 paper with
Arnaud Hubaux (U. Namur),
Steven She and Krzysztof Czarnecki (U. Waterloo)

We produce errors everyday

```
public static void main(String[] args) {  
    output("Hello, world");  
}
```

 The method `output(String)` is undefined for the type `Main`

1 quick fix available:

■ [Create method 'output\(String\)'](#)

Press 'F2' for focus

We use fixes everyday

```
public static void main(String[] args) {  
    output("Hello, world");  
}
```

```
private static void output(String msg) {  
    // TODO Auto-generated method stub  
}
```

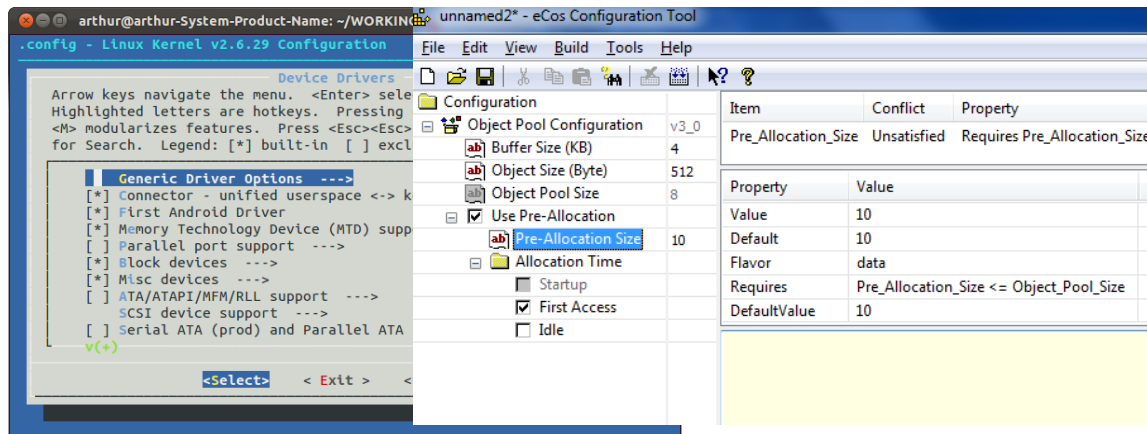
How much do we know about fixes?

- How much are fixes needed?
- What fixes are desirable?
- Can we generate fixes automatically?

Study domain: operating system configuration

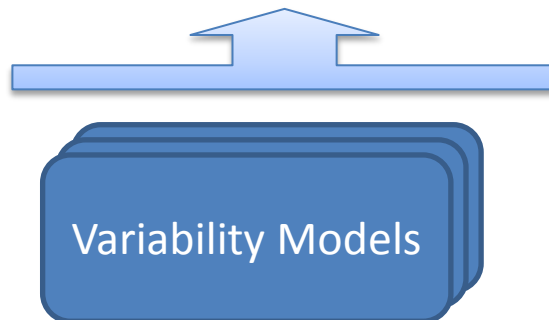


Configuration



Linux Kconfig,
eCos CDL,

...



eCos Configurator - Errors

The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The left pane displays a tree view of the configuration hierarchy: Configuration > Object Pool (v3_0) > Buffer Size (KB) (4) > Object Size (Byte) (512) > Pool Size (8) > Preload (checked) > Preload Size (10). The right pane shows a table of properties for the selected 'Preload Size' item.

Item	Property
PreloadSize	Requires PreloadSize <= PoolSize

Property	Value
Value	10
Default	10
Flavor	data
Requires	PreloadSize <= PoolSize
DefaultValue	10

Contributions

- How much are fixes needed?
 - A survey revealing manual fixes take minutes
- What fixes are desirable?
 - A new type of fix, range fix, and evaluated desirable properties of fixes
- Can we generate fixes automatically?
 - An algorithm generating range fixes in tens of milliseconds

How much are fixes needed?

A survey showing manual fixes take
minutes

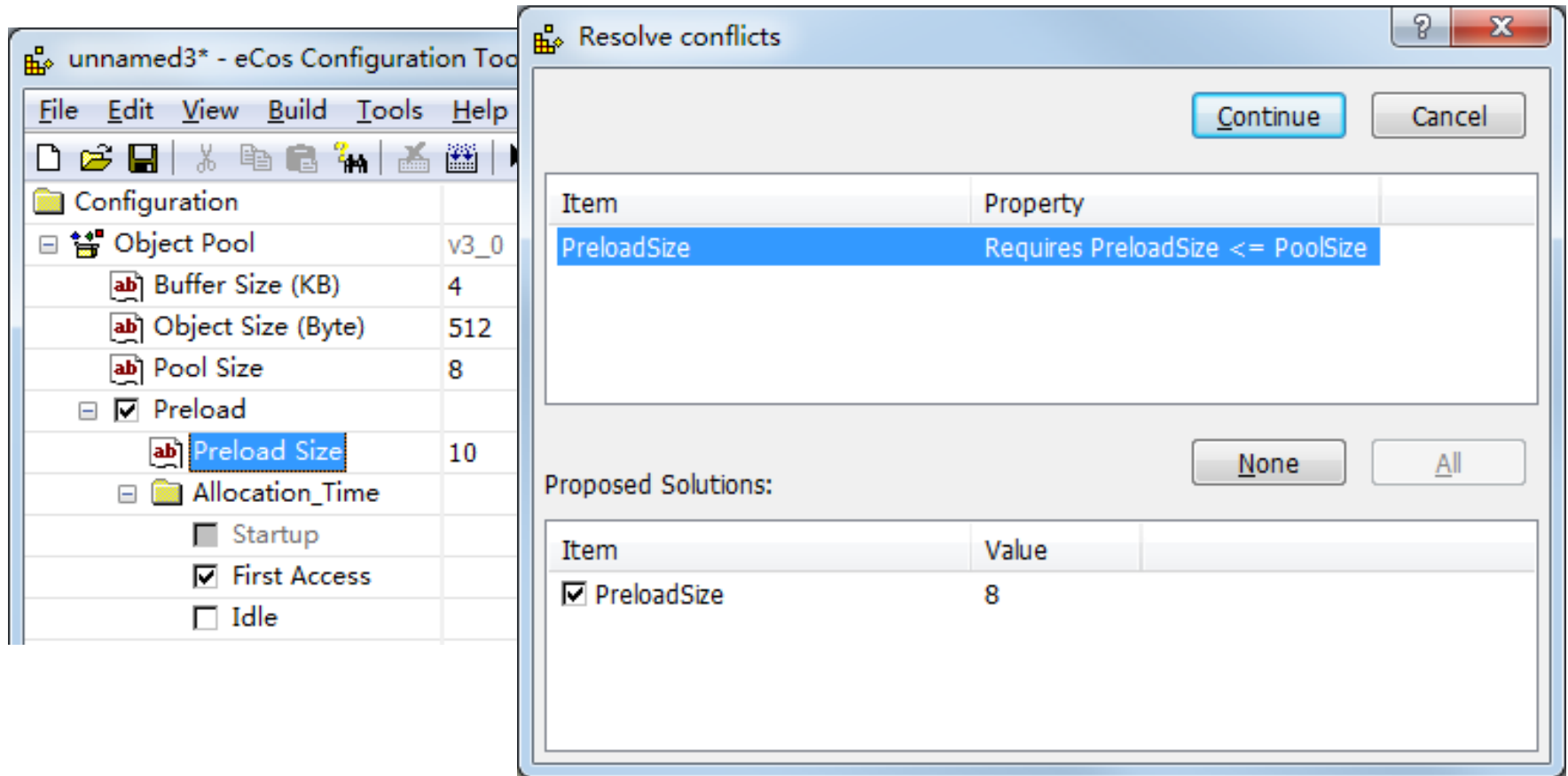
Survey

- 97 Linux users and 9 eCos users
- Resolving a violation is hard
 - 20% Linux users need "a few dozen minutes" to resolve a violation in average
 - 56% eCos users consider violation resolution to be a problem

What fixes are desirable?

A new type of fixes, range fixes, and
evaluated desirable properties

eCos Configurator



eCos configurator has built-in fixes

Fix Incompleteness

The image shows a screenshot of the eCos Configuration Tool and a 'Resolve conflicts' dialog box. The configuration tool on the left shows a tree view with 'Object Pool' expanded, showing 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), 'Pool Size' (8), and 'Preload' (checked). The 'Preload Size' is set to 10. The 'Resolve conflicts' dialog on the right shows a conflict between 'PreloadSize' and 'PoolSize'. The dialog has a 'Continue' button and a 'Cancel' button. A red box highlights the 'PreloadSize' property with the text 'Increase to any value ≥ 10 '. The 'Proposed Solutions' section shows a table with 'Item' and 'Value' columns. The 'PreloadSize' item is checked with a value of 8. A red box highlights the value 8 with the text 'Further decrease to any value ≤ 8 '. A red box highlights the 'Preload' checkbox with the text 'Disable'.

unnamed3* - eCos Configuration Tool

File Edit View Build Tools Help

Configuration

- Object Pool (v3_0)
 - Buffer Size (KB) 4
 - Object Size (Byte) 512
 - Pool Size 8
 - ☒ Preload
 - Preload Size 10
 - Allocation_Time
 - ☐ Startup
 - ☒ First Access
 - ☐ Idle

Resolve conflicts

Continue Cancel

Item	Property
PreloadSize	Requires PreloadSize \leq PoolSize

Proposed Solutions:

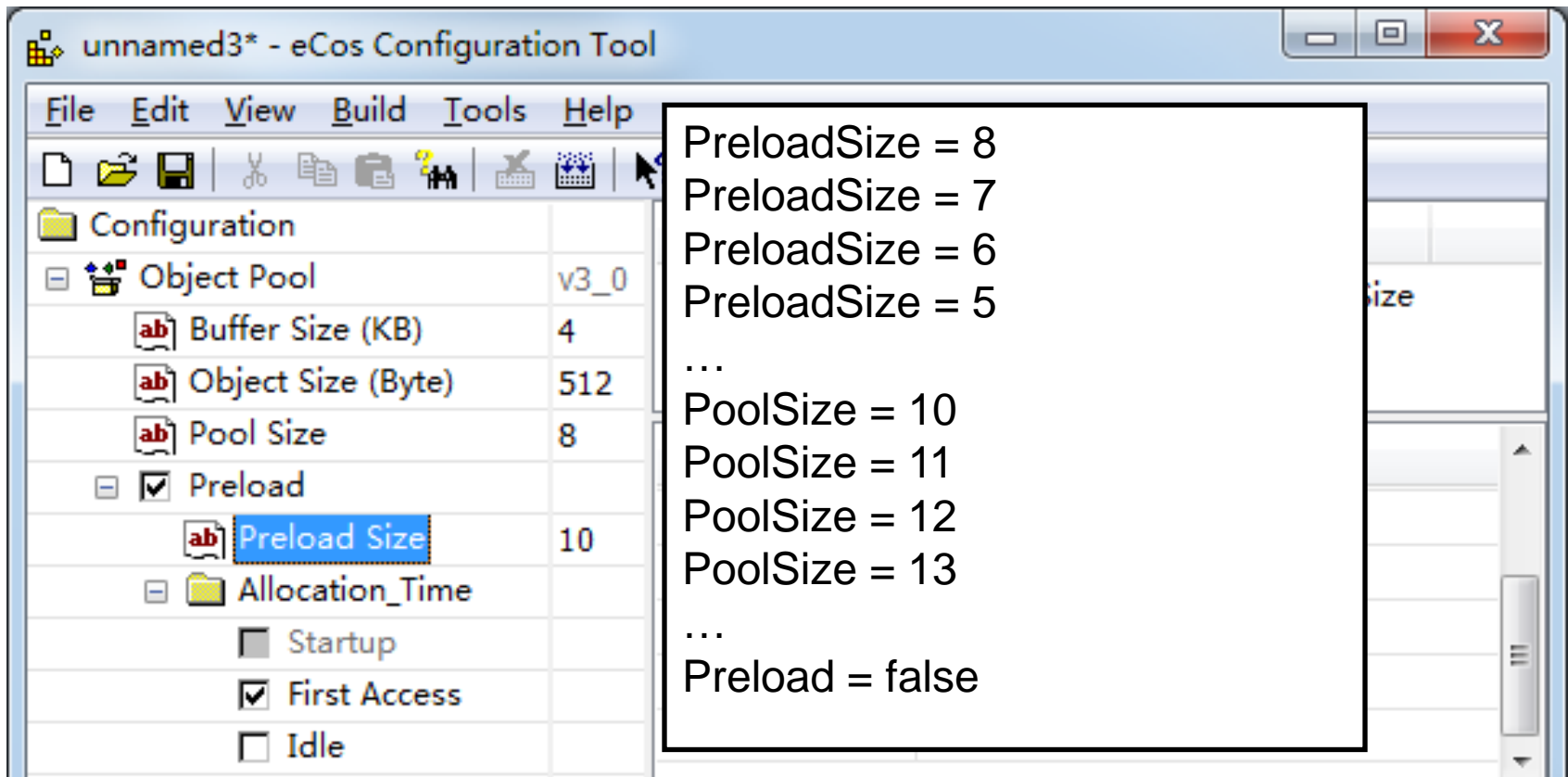
Item	Value
<input checked="" type="checkbox"/> PreloadSize	8

Disable

Further decrease to any value ≤ 8

78% eCos users have encountered situations where the proposed fix is not useful

How to complete fixes



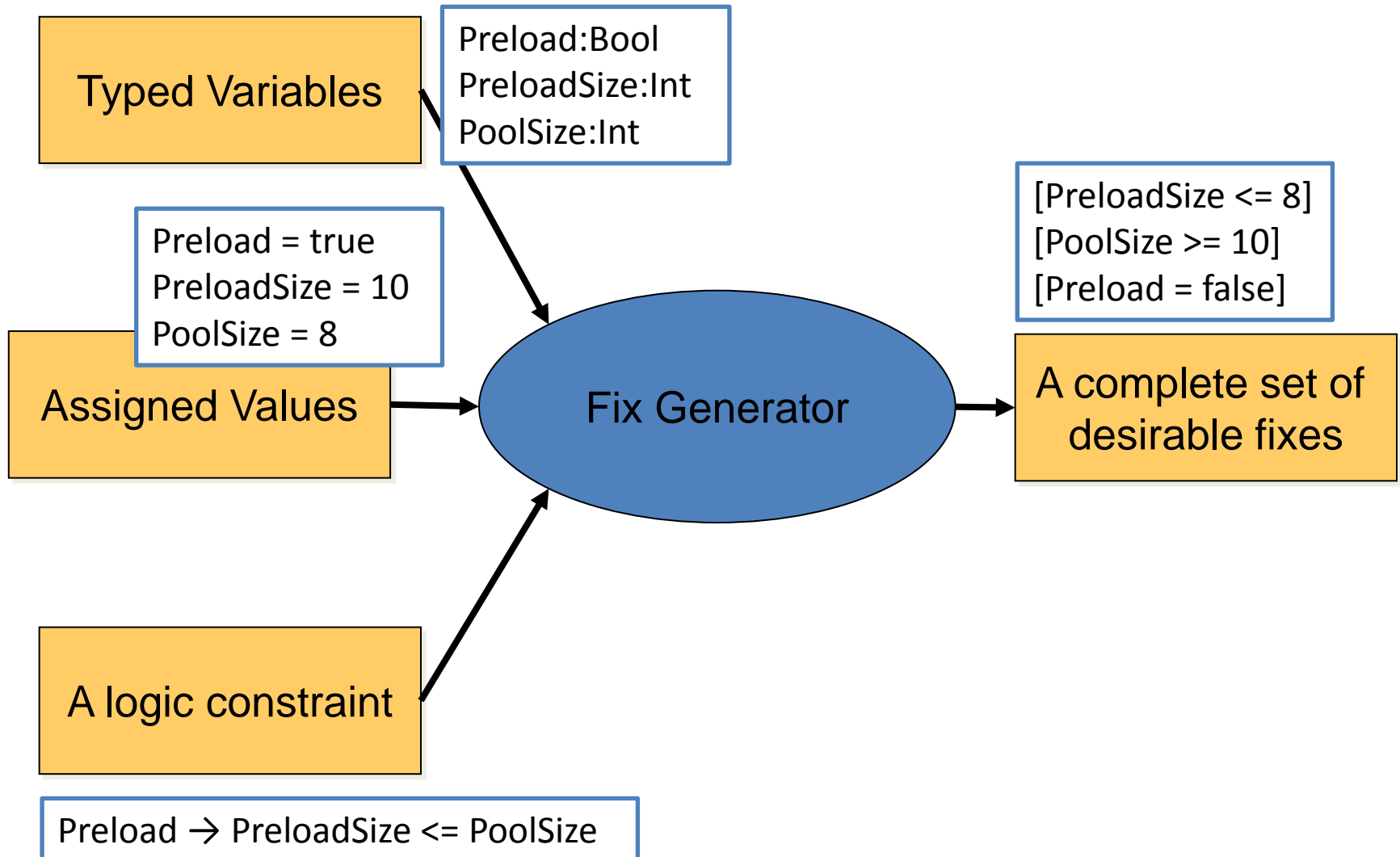
Our Proposal – Range Fixes

The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The left pane displays the configuration tree with 'Object Pool' expanded, showing 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), 'Pool Size' (8), 'Preload' (checked), 'Preload Size' (10), and 'Allocation_Time' (Startup, First Access checked, Idle unchecked). The right pane shows the 'PreloadSize' property with the value 'Requires PreloadSize <= PoolSize'. A text box overlay contains the following text:

[PreloadSize <= 8]
[PoolSize >= 10]
[Preload = false]

Fix Generation Problem

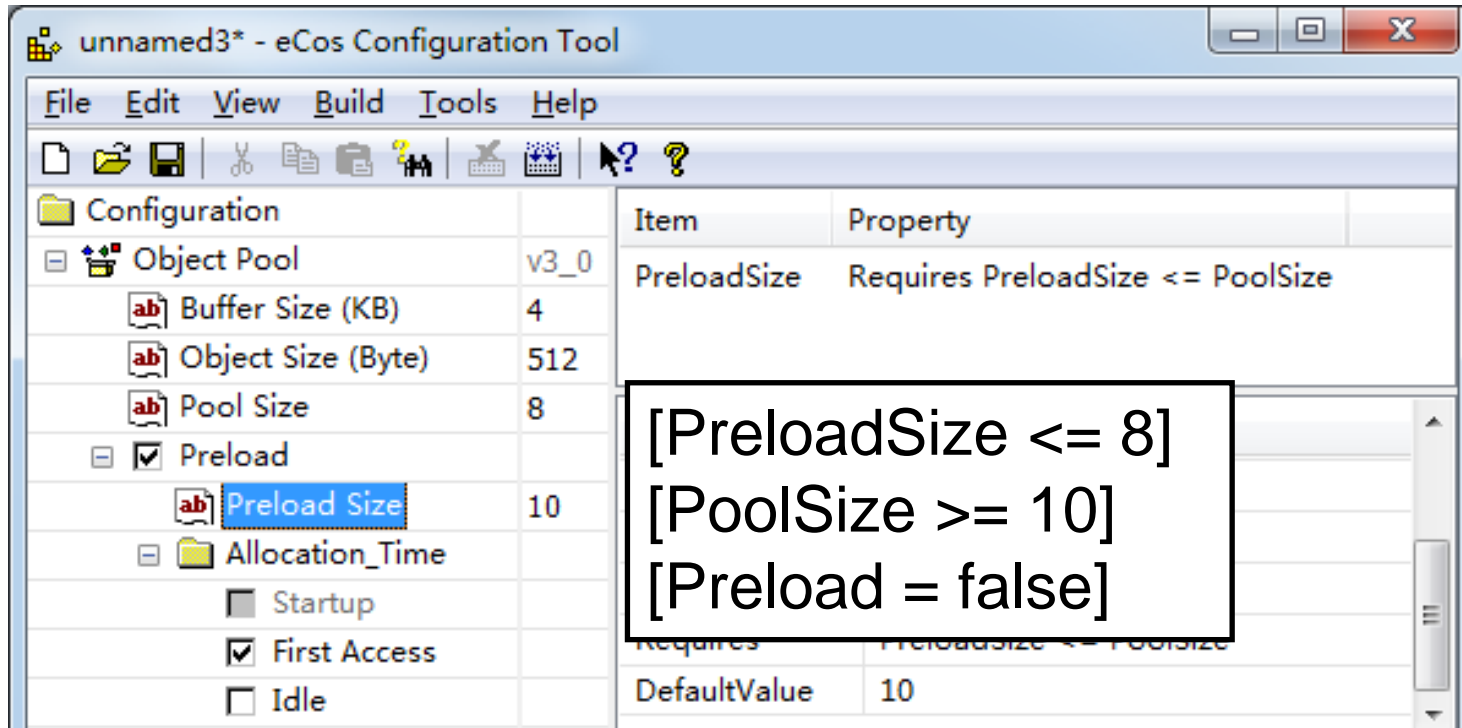
– a General Definition



Desired Properties of Fixes

Correctness	Minimality of variables	Maximality of ranges
Any change represented by a range fix will satisfy the constraint	There is no way to change a subset of variables to satisfy the constraint	A range fix represents the maximal ranges over the variables
A desirable one: [PreloadSize <=8]		
Undesirable ones		
[PreloadSize <= 9]	[PreloadSize <=8, Preload = false]	[PreloadSize <=7]

Constraint Interaction



Constraint Interaction

The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The left pane displays the configuration tree with 'Object Pool' expanded, showing 'Pool Size' set to 12. A red circle highlights the value 12, with a callout arrow pointing to a text box that says 'Increase PoolSize'. The right pane shows a table of properties and values. The top table has columns 'Item' and 'Property', with a row for 'PoolSize' having the property 'Requires PoolSize == BufferSize * 1024 / ObjectSize'. The bottom table has columns 'Property' and 'Value', with rows for 'File', 'Macro', 'Value', 'Default', 'Flavor', and 'Requires'. The 'Value' row shows '12', and the 'Requires' row shows 'PoolSize == BufferSize * 1024 / ObjectSize'. A red callout arrow points from the 'Requires' row to a text box that says 'Causing another error'. Another red callout arrow points from the 'Value' row to a text box that says 'Interacting constraint'.

unnamed3* - eCos Configuration Tool

File Edit View Build Tools Help

Configuration

- Object Pool (v3_0)
 - Buffer Size (KB) 4
 - Object Size (Byte) 512
 - Pool Size 12
 - Preload
 - Preload Size 10
 - Allocation_Time
 - Startup
 - First Access
 - Idle

Item	Property
PoolSize	Requires PoolSize == BufferSize * 1024 / ObjectSize

Property	Value
File	unnamed3_install/include/pkgconf\hal.h
Macro	PoolSize
Value	12
Default	0
Flavor	data
Requires	PoolSize == BufferSize * 1024 / ObjectSize

Causing another error

Increase PoolSize

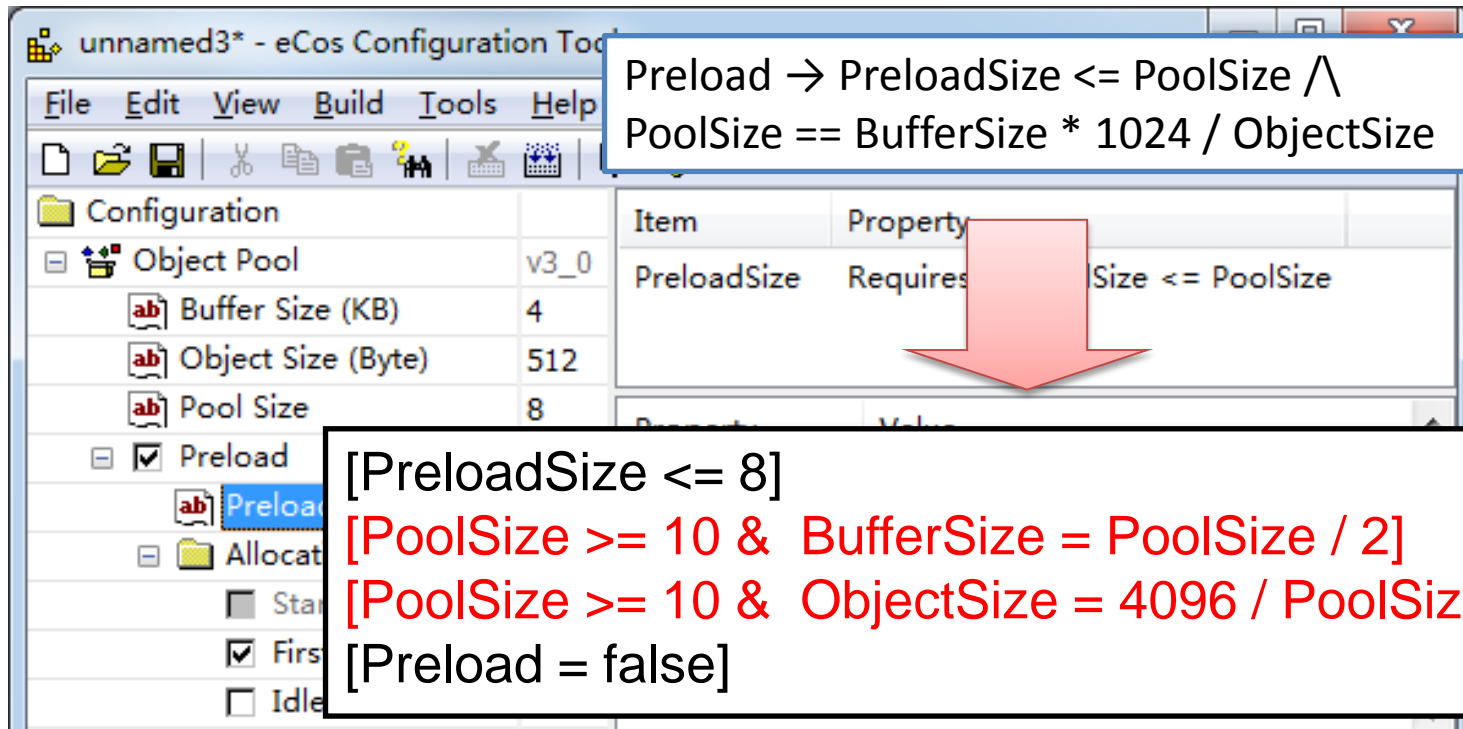
Interacting constraint

Three Strategies

- Ignorance
 - Elimination
 - Propagation
-
- Summarized from existing approaches

Propagation Strategy

Make a conjunction of all satisfied constraints plus the violated one



The screenshot shows the 'unnamed3* - eCos Configuration Tool' window. The 'Configuration' tree on the left includes 'Object Pool' (v3_0) with sub-items 'Buffer Size (KB)' (4), 'Object Size (Byte)' (512), and 'Pool Size' (8). Below these is 'Preload' (checked) and 'Allocation' (unchecked). The 'Properties' table on the right shows 'PreloadSize' with a 'Requires' property 'Size <= PoolSize'. A red arrow points from this property to a list of constraints in a white box.

Preload \rightarrow $\text{PreloadSize} \leq \text{PoolSize} \wedge$
 $\text{PoolSize} == \text{BufferSize} * 1024 / \text{ObjectSize}$

[PreloadSize <= 8]
[PoolSize >= 10 & BufferSize = PoolSize / 2]
[PoolSize >= 10 & ObjectSize = 4096 / PoolSize]
[Preload = false]

Evaluation

- Source
 - Version histories from 5 open source projects
- Steps
 - Compare each pair of consecutive versions
 - Replay the user changes in different orders
 - Generate fixes for the violations and compare with user changes

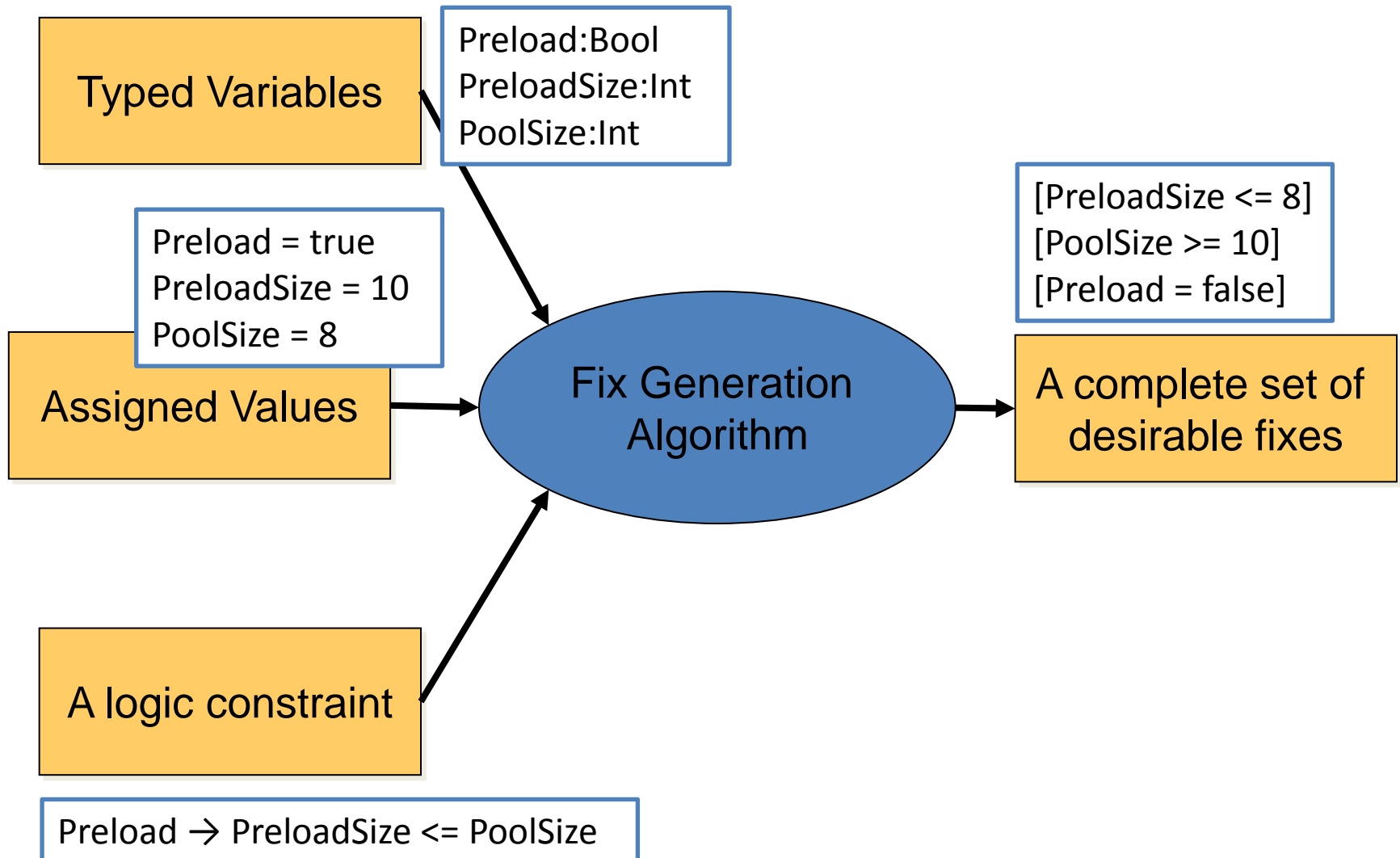
Results

- Coverage of User changes: 100%
- Complexity of fix lists
 - measured by adding up the number of variables in each fix
 - Median: 2
 - Maximum: 58
 - 83% of the fix lists contain less than 10 variables

Can we generate fixes automatically?

An algorithm generating range fixes
in tens of milliseconds

Interface of our algorithm



Algorithm Outline

- Step 1: find the variables to change
 - Basic idea: translating to an SMT problem
 1. treat configurations also as constraints
 2. ask an SMT solver for unsatisfiable cores
 3. combine the unsatisfiable cores
- Step 2: find the range of the variables
 - Basic idea: simplify the constraint
 1. replace unchangeable variables with their current values
 2. simplify the constraint and convert to CNF

Performance of the algorithm

- Published results
 - Average: 50ms
 - Maximum: 250ms
- We have recently improved the performance

Thank you for your attention!

EccFixer: <http://gsd.uwaterloo.ca/eccfixer>