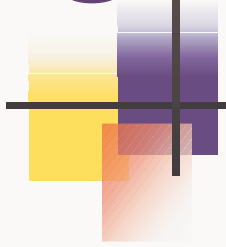




Towards the verification of C with KeY

By Christoph Gladisch
University Koblenz-Landau



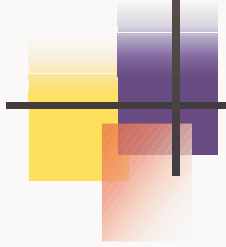
Goal

- Extend KeY for the verification of C
(not C++)
- First C0 then MISRA C



Tasks

- New parser, AST-converter, GUI, schematic types for the taclet language
- New verification rules



First attempt

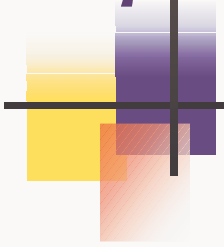
1. Find differences between C and Java
2. Extend KeY and write verification rules



New approach

1. Write verification rules for C
2. If differences to Java are found then:
 - extend the tactic mechanism
 - goto 1.

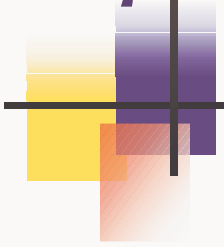
Differences between C and Java



- **Assignments** by copy
- **Pointers** of local variables and substructures
- Explicit object **deletion**

(differences concern expressions, statements are similar enough)

Differences between C and Java



$a, b \in C$ struct	$a, b \in C$ struct pointer	$a, b \in \text{Java class}$
$b.c := d;$	$b \rightarrow c := d;$	$b.c := d;$
$a := b;$	$a := b;$	$a := b;$
$b.c := e;$	$b \rightarrow c := e;$	$b.c := e;$
$a.c := d$	$a \rightarrow b := e$	$a.c := e$

- Deep copy
- Aliasing



Assignments by copy vs. by reference

There are two way how to handle the problem:

- Unfolding all implicit updates to one big parallel update
- Creating new update rules for „Lazy evaluation“

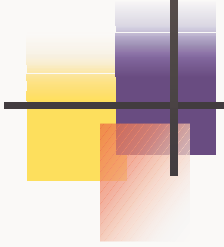


Assignments by copy vs. by reference

Finding new update rules for assignments by copy involves finding rules for

- Application of a single update to an expression
 $\langle X := a \rangle Z$
- Parallel update application
 $\langle X := a, Y := b \rangle Z$
- Application of an update to another update
 $\langle X := a \rangle \langle Y := Z \rangle$
- Generalisation of the rules

Assignments by copy vs. by reference



Rule for the application of a parallel update to an expression

if $X = Y \wedge X \sqsubseteq Z \wedge Y \sqsubseteq Z$ then $\langle X := a, Y := b \rangle Z \rightsquigarrow \langle Y := b \rangle Z$

Example

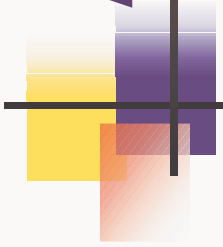
$\langle c.x := a, c.x := b \rangle c \rightsquigarrow \langle c.x := b \rangle c$



All cases

case	rewrite to	example
$X = Z$	a	$\langle c \triangleleft a \rangle c \rightsquigarrow a$
$X \sqsubseteq Z$	$\langle X \triangleleft a \rangle Z$	$\langle c.x \triangleleft a \rangle c \rightsquigarrow \langle c.x \triangleleft a \rangle c$
$X \sqsupset Z$	$((X \triangleleft a)Z').x =$ a.x where $Z = Z'.x$	$\langle c \triangleleft a \rangle c.x \rightsquigarrow ((c \triangleleft a)c).x$
$X \boxtimes Z$	Z	$\langle c \triangleleft a \rangle d \rightsquigarrow d$

Table 1. Application of a simple update to a complex identifier $\langle X \triangleleft a \rangle Z$



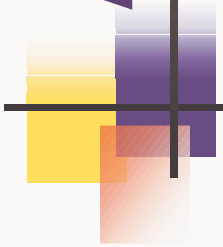
All cases

$X = Y \wedge$

subcase	rewrite to	example
$X = Z \wedge Y = Z$	$\langle Y \triangleleft b \rangle Z = b$	$\langle c \triangleleft a, c \triangleleft b \rangle c \rightsquigarrow b$
$X \sqsubset Z \wedge Y \sqsubset Z$	$\langle Y \triangleleft b \rangle Z$	$\langle c.x \triangleleft a, c.x \triangleleft b \rangle c \rightsquigarrow \langle c.x \triangleleft b \rangle c$
$X \sqsupset Z \wedge Y \sqsupset Z$	$\langle Y \triangleleft b \rangle Z = (\langle Y \triangleleft b \rangle Z') \cdot x$ $=_{b,x}$ where $Z'.x = Z$	$\langle c \triangleleft a, c \triangleleft b \rangle c.x \rightsquigarrow b.x$
$X \boxtimes Z \wedge Y \boxtimes Z$	$\langle Y \triangleleft b \rangle Z = Z$	$\langle c \triangleleft a, c \triangleleft b \rangle d \rightsquigarrow d$

$X \sqsubset Y \wedge$

subcase	rewrite to	example
In any case	$\langle Y \triangleleft b \rangle Z$	$\langle c.x \triangleleft a, c \triangleleft b \rangle c.x \rightsquigarrow b.x$ $\langle c.x \triangleleft a, c \triangleleft b \rangle c \rightsquigarrow b$ $\langle c.x \triangleleft a, c \triangleleft b \rangle c.x.y \rightsquigarrow b.x.y$



All cases

subcase	rewrite to	example
$X = Z \wedge Y \supset Z$	$\langle X \sqsubseteq a, Y \sqsubseteq b \rangle Z$	$\langle c \sqsubseteq a, c.x \sqsubseteq b \rangle c \rightsquigarrow$ $\langle c \sqsubseteq a, c.x \sqsubseteq b \rangle c$
$X \supset Z \wedge Y \sqsubset Z$	$\langle X \sqsubseteq a, Y \sqsubseteq b \rangle Z$	$\langle c \sqsubseteq a, c.x.y \sqsubseteq b \rangle c.x \rightsquigarrow$ $\langle c \sqsubseteq a, c.x.y \sqsubseteq b \rangle c.x$
$X \supset Z \wedge Y = Z$	$\langle Y \sqsubseteq b \rangle Z = b$	$\langle c \sqsubseteq a, c.x \sqsubseteq b \rangle c.x \rightsquigarrow b$
$X \supset Z \wedge Y \supset Z$	$\langle Y \sqsubseteq b \rangle Z = (\langle Y \sqsubseteq b \rangle Z').y$ $= b.y$ where $Z'.y = Z$	$\langle c \sqsubseteq a, c.x \sqsubseteq b \rangle c.x.y \rightsquigarrow b.y$
$X \boxtimes Z \wedge Y \boxtimes Z$	$\langle Y \sqsubseteq b \rangle Z = Z$	$\langle c \sqsubseteq a, c.x \sqsubseteq b \rangle d \rightsquigarrow d$

$X \supset Y \wedge$

subcase	rewrite to	example
$X \sqsubset Z \wedge Y \boxtimes Z$	$\langle X \sqsubseteq a \rangle Z$	$\langle c \sqsubseteq a, d \sqsubseteq b \rangle c \rightsquigarrow \langle c \sqsubseteq a \rangle c$
$X \boxtimes Z \wedge Y \sqsubset Z$	$\langle Y \sqsubseteq b \rangle Z$	$\langle d \sqsubseteq a, c \sqsubseteq b \rangle c \rightsquigarrow \langle c \sqsubseteq b \rangle c$
$X \boxtimes Z \wedge Y \boxtimes Z$	$\langle Y \sqsubseteq b \rangle Z = Z$	$\langle c \sqsubseteq a, c.x \sqsubseteq b \rangle d \rightsquigarrow d$

$X \boxtimes Y \wedge$



All cases

Rewriting of $\langle X \sqsubseteq a \rangle \langle Y \sqsubseteq Z \rangle$.

$X = Z$

subcase	rewrite to	example
$X = Y \wedge Y = Z$	$\langle X \sqsubseteq a \rangle$	$\langle x \sqsubseteq a \rangle \langle x \sqsubseteq x \rangle \rightsquigarrow \langle x \sqsubseteq a \rangle$
$X \sqsubset Y \wedge Y \sqsupset Z$	forbidden, not defined	$\langle x.b \sqsubseteq a \rangle \langle x \sqsubseteq x.b \rangle$
$X \sqsupset Y \wedge Y \sqsubset Z$	forbidden, not defined	$\langle x \sqsubseteq a \rangle \langle x.b \sqsubseteq x \rangle$
$X \boxtimes Y \wedge Y \boxtimes Z$	$\langle X \sqsubseteq a, Y \sqsubseteq \langle X \sqsupset a \rangle Z \rangle \rightsquigarrow$ $\langle X \sqsubseteq a, Y \sqsubseteq a \rangle$	$\langle x \sqsubseteq a \rangle \langle d \sqsubseteq x \rangle \rightsquigarrow \langle x \sqsubseteq a, d \sqsubseteq a \rangle$

$X \sqsubset Z$

subcase	rewrite to	example
$X = Y \wedge Y \sqsubset Z$	forbidden, not defined	$\langle x.b \sqsubseteq a \rangle \langle x.b \sqsubseteq x \rangle$
$X \sqsubset Y \wedge Y = Z$	$\langle Y \sqsubseteq Z \rangle$	$\langle x.b \sqsubseteq a \rangle \langle x \sqsubseteq x \rangle \rightsquigarrow$ $\langle x.b \sqsubseteq a \rangle$
$X \sqsupset Y \wedge Y \sqsubset Z$	forbidden, not defined	$\langle x.b \sqsubseteq a \rangle \langle x.b.c \sqsubseteq x \rangle$
$X \boxtimes Y \wedge Y \boxtimes Z$	$\langle X \sqsubseteq a, Y \sqsubseteq Z, \langle Z \sqsubseteq Y \rangle X \sqsubseteq a \rangle \rightsquigarrow$ $\langle X \sqsubseteq a, Y \sqsubseteq Z, x.Y \sqsubseteq a \rangle$ where $X = x.b$	$\langle x.b \sqsubseteq a \rangle \langle d \sqsubseteq x \rangle \rightsquigarrow$ $\langle x.b \sqsubseteq a, d \sqsubseteq x, d.b \sqsubseteq a \rangle$



All cases

subcase	rewrite to	example
$X \sqsupset Y \wedge Y \sqsupset Z$	forbidden, not defined	$\langle x \triangleleft a \rangle \langle x \triangleleft x.b \rangle$
$X \sqsubset Y \wedge Y = Z$	$\langle X \triangleleft a \rangle$	$\langle x \triangleleft a \rangle \langle x.b \triangleleft x.b \rangle \rightsquigarrow \langle x \triangleleft a \rangle$
$X \sqsupset Y \wedge Y \sqsubset Z$	forbidden, not defined	$\langle x \triangleleft a \rangle \langle x.b \triangleleft x.b.c \rangle$
$X \boxtimes Y \wedge Y \boxtimes Z$	$\langle X \triangleleft a, Y \triangleleft (Y \triangleleft Z) Z \rangle \rightsquigarrow$ $\langle X \triangleleft a, Y \triangleleft a.b \rangle$ where $Z = Z'.b$	$\langle x \triangleleft a \rangle \langle d \triangleleft x.b \rangle \rightsquigarrow$ $\langle x \triangleleft a, d \triangleleft a.b \rangle$

$X \sqsupset Z$

subcase	rewrite to	example
$X = Y \wedge Y \boxtimes Z$	$\langle Y \triangleleft Z \rangle$	$\langle x \triangleleft x \rangle \langle x \triangleleft d \rangle \rightsquigarrow \langle x \triangleleft d \rangle$
$X \sqsupset Y \wedge Y \boxtimes Z$	$\langle X \triangleleft a, Y \triangleleft Z \rangle$	$\langle x \triangleleft a \rangle \langle x.b \triangleleft d \rangle \rightsquigarrow \langle x \triangleleft a, x.b \triangleleft d \rangle$
$X \sqsubset Y \wedge Y \boxtimes Z$	$\langle Y \triangleleft Z \rangle$	$\langle x.b \triangleleft a \rangle \langle x \triangleleft d \rangle \rightsquigarrow \langle x.b \triangleleft d \rangle$
$X \boxtimes Y \wedge Y = Z$	$\langle X \triangleleft a \rangle$	$\langle x \triangleleft a \rangle \langle d \triangleleft d \rangle \rightsquigarrow \langle x \triangleleft a \rangle$
$X \boxtimes Y \wedge Y \sqsupset Z$	forbidden, not defined	$\langle x \triangleleft a \rangle \langle d \triangleleft d.a \rangle$
$X \boxtimes Y \wedge Y \sqsubset Z$	forbidden, not defined	$\langle x \triangleleft a \rangle \langle d.a \triangleleft d \rangle$
$X \boxtimes Y \wedge Y \boxtimes Z$	$\langle X \triangleleft a, Y \triangleleft Z \rangle$	$\langle x \triangleleft a \rangle \langle e \triangleleft d \rangle \rightsquigarrow \langle x \triangleleft a, e \triangleleft d \rangle$

$X \boxtimes Z$



Assignments by copy vs. by reference

Pointers and aliasing not considered. Why are there so many rules?

- There are more relation between expressions which have to be considered
- An update doesn't represent a single update, but a whole set of recursive actions

Pointers and the addressOf-operator



Where do pointers come from in C0 and MISRA C?

`new` – C and Java

`&` - additionally in C

Additional operator

`*` - dereference operator

Pointers and the addressOf- operator

```
1. a := 1;  
2. p := &a;  
3. *p := 2;
```

$\Rightarrow a := 2$



Pointers and the addressOf-operator

Additional object layer. Treat variables as objects.

v : object \rightarrow value

Source code Logic
 a \Rightarrow $v(a)$



Pointers and the addressOf-operator

addressOf-operator `&` and the dereference-operators `*`

`&: object` \rightarrow pointer is defined as `&(v(X)) := X`

`*: pointer` \rightarrow $\begin{cases} \text{pointer} \\ \text{object} \end{cases}$ defined as `*(X) := v(X)`

They are inverse operations `*(&(X)) \doteq X`.

Pointers and the addressOf-operator

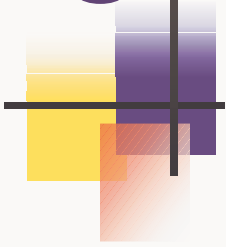
Example

```
1. a := 1;  
2. p := &a;  
3. *p := 2;
```



```
 $\langle v(a) := 1 \rangle$   
 $\langle v(p) := \&(v(a)) \rangle$   
 $\langle *(v(p)) := 2 \rangle$ 
```

Pointers and the addressOf-operator



$\langle v(a) := 1 \rangle$ $\langle v(p) := \&(v(a)) \rangle$ $\langle *(v(p)) := 2 \rangle$	\Rightarrow	$\langle v(a) := 1 \rangle$ $\langle v(p) := a \rangle$ $\langle v(v(p)) := 2 \rangle$
---	---------------	--

$\langle v(a) := 1 \rangle \langle v(p) := a \rangle \langle v(v(p)) := 2 \rangle \rightsquigarrow 2$



Object deletion

$c: \text{object} \rightarrow \{\text{true}, \text{false}\}$.

```
p_a := new int;     $\Uparrow$     OK  
delete p_a;
```

```
p_a := &var;       $\Uparrow$     INVALID  
delete p_a;
```

But how to distinguish?



Object deletion

1. $p := \text{new int}; \implies \langle^v(p) := \text{obj}_{\text{int}}(\text{next}_{\text{int}})\rangle$
2. $p := \&\text{var}; \implies \langle^v(p) \triangleleft \&(v(\text{var}))\rangle \rightsquigarrow \langle^v(p) \triangleleft \text{var}\rangle$

$$\frac{\Phi, \exists n. \text{Expr} \doteq \text{obj}_{\mathcal{T}}(n) \vdash \langle^c(\text{Expr}) := \text{false}\rangle \Delta \dots}{\Phi \vdash \langle \text{delete Expr} \rangle \Delta}$$



Structures and deletion

- static objects
- dynamic objects
- subdynamic objects !?!

```
struct strA{int d}  
strB* s := new strA;  
int* pd := &(s->d) //&((*s).d)  
delete pd;
```



Structures and deletion

Problem

$$e(A.X) \doteq e(A) \rightarrow \langle e(A) := \text{false} \rangle^c(A.X) \doteq e(A)$$

Possible solution?

$$\text{for } z \bullet z \sqsubset A \bullet e(z) := \text{false}$$



Conclusion

- The system has to be extended
- Differences are:
 - Assignments by copy
 - Pointers of local variables and substructures
 - Explicit deletion
- It is more complicated than it looks like