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 taclet 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	$a,b \in$ Java class

`b.c:=d;`
`a   :=b;`
`b.c:=e;`
	`b->c:=d;`
	`a    :=b;`
	`b->c:=e;`
		`b.c:=d;`
		`a   :=b;`
		`b.c:=e;`

$a.c \doteq d$	$a\text{->}b \doteq e$	$a.c \doteq e$

- 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

$$\text{if } X = Y \wedge X \sqsubset Z \wedge Y \sqsubset Z \ \text{ then } \langle X := a, Y := b \rangle Z \rightsquigarrow \langle Y := b \rangle Z$$

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	$\langle c \trianglelefteq a \rangle c \rightsquigarrow a$
$X \sqsubset Z$	$\langle X \trianglelefteq a \rangle Z$	$\langle c.x \trianglelefteq a \rangle c \rightsquigarrow \langle c.x \trianglelefteq a \rangle c$
$X \sqsupset Z$	$(\langle X \trianglelefteq a \rangle Z').x=$ a.x where Z=Z'.x	$\langle c \trianglelefteq a \rangle c.x \rightsquigarrow (\langle c \trianglelefteq a \rangle c).x$
$X \boxtimes Z$	Z	$\langle c \trianglelefteq a \rangle d \rightsquigarrow d$

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

All cases

$X = Y \wedge$

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

$X \sqsubset Y \wedge$

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

All cases

$X \sqsupset Y \wedge$

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

$X \boxtimes Y \wedge$

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

All cases

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

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

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

All cases

$X \sqsupset Z$

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

$X \boxtimes Z$

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

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
- **`&`** - additionaly in C

Additional operator

- **`*`** - dereference operator

Pointers and the addressOf-operator

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

$\Rightarrow a \doteq 2$

Pointers and the addressOf-operator

Additional object layer. Treat variables as objects.

$$^{v}\!: \text{object} \rightarrow \text{value}$$

Source code		Logic
`a`	$\Longrightarrow$	$^{v}(a)$

Pointers and the addressOf-operator

addressOp-operator & and the dereference-operators *

$\&\colon \text{object} \rightarrow \text{pointer}$ is defined as $\&({}^{v}(X)) := X$

$*\colon \text{pointer} \rightarrow \left\{ \begin{matrix} \text{pointer} \\ \text{object} \end{matrix} \right.$ 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;
```

$\Longrightarrow$

$$\langle {}^{v}(a) := 1 \rangle$$
$$\langle {}^{v}(\mathrm{p}) := \&({}^{v}(a)) \rangle$$
$$\langle *({}^{v}(\mathrm{p}))) := 2 \rangle$$

Pointers and the addressOf-operator

$$\begin{array}{l}\langle\, {}^{v}(a) := 1\rangle\\ \langle\, {}^{v}(\mathrm{p}) := \&({}^{v}(a))\rangle\\ \langle\, *\,({}^{v}(\mathrm{p}))) := 2\rangle\end{array} \Longrightarrow \begin{array}{l}\langle\, {}^{v}(a) := 1\rangle\\ \langle\, {}^{v}(\mathrm{p}) := a\rangle\\ \langle\, {}^{v}({}^{v}(\mathrm{p}))) := 2\rangle\end{array}$$

$$\langle\, {}^{v}(a) := 1\rangle\langle\, {}^{v}(\mathrm{p}) := a\rangle\langle\, {}^{v}({}^{v}(\mathrm{p}))) := 2\rangle\, {}^{v}(a) \leadsto 2$$

Object deletion

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

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

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

But how to distinguish?

Object deletion

1. `p := new int;` $\Longrightarrow$ $\langle {}^{v}(\mathrm{p}) := \mathrm{obj}_{\mathrm{int}}(\mathrm{next}_{\mathrm{int}})\rangle$
2. `p := &var;` $\Longrightarrow$ $\langle {}^{v}(\mathrm{p}) \lhd \&({}^{v}(\mathrm{var}))\rangle \leadsto \langle {}^{v}(\mathrm{p}) \lhd \mathrm{var}\rangle$

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

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

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

Possible solution?

$$\mathrm{for}\ z \bullet z \sqsubset A \bullet {}^{c}(z) := \mathrm{false}$$

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