MACHINE
xOwl_SubClassOf
INCLUDES
xOwl
SETS
$state = \{safe, not\_safe\}$
DEFINITIONS
$SET\_PREF\_DEFAULT\_SETSIZE == 25$;
$SET\_PREF\_MAX\_OPERATIONS == 10$;
$SET\_PREF\_SHOW\_EVENTB\_ANY\_VALUES == \mathbf{FALSE}$;

$inheritence == ((subClassExp ; classOfClassExp)^{-1} ; (superClassExp ; classOfClassExp))$;
$disjunctionExp == classOfClassExp \otimes disjointClassExp$;
$disjunction == (\mathbf{ran}(disjunctionExp) ; \mathbf{ran}(disjunctionExp)^{-1})$;
$super(cl) == \mathbf{closure}(inheritence)[\{cl\}]$;

VARIABLES
subClass, *superClass*, *check*
INVARIANT
$subClass \subseteq Class \wedge superClass \subseteq Class$
$\wedge\ subClass \cap superClass = \emptyset$
$\wedge\ check : (Class \times Class) \pfun state$
INITIALISATION
$subClass, superClass, check := \emptyset , \emptyset , \emptyset$
OPERATIONS
$ss \leftarrow \mathbf{selectClasses}(iri1, iri2) =$
PRE $iri1 \in STRING \wedge iri2 \in STRING$ **THEN**
ANY
cl1, *cl2*
WHERE
$cl1 \in Class \wedge name(cl1) = iri1 \wedge cl1 \notin subClass \cup superClass$
$\wedge\ cl2 \in Class \wedge name(cl2) = iri2 \wedge cl2 \notin subClass \cup superClass$
$\wedge\ cl2 \notin super(cl1)$
$\wedge\ (\mathbf{closure1}(inheritence \cup \{(cl1 \mapsto cl2)\}) \cap \mathbf{id}(Class) = \emptyset)$
THEN
$subClass := subClass \cup \{cl1\}$ ||
$superClass := superClass \cup \{cl2\}$ ||
IF $super(cl1) \times super(cl2) \cap disjunction = \emptyset$ **THEN**
$ss := safe$
ELSE
$ss := not\_safe$
END ;
$\mathbf{check}(cl1 \mapsto cl2) := ss$
END
END ;
$\mathbf{createSubClassOf}(cl1,cl2) =$
PRE $cl1 \in subClass \wedge cl2 \in superClass$ **THEN**

```
        ANY axiom, exp1, exp2 WHERE
            axiom ∈ AXIOM - Axiom
             ∧ {exp1, exp2} ⊆ CLASSEXPRESSION - ClassExpression
        THEN
            ClassExpression_NEW(exp1);
            ClassExpression_NEW(exp2);
            SubClassOf_NEW(axiom, exp1, exp2);
            ClassExpression_SetClass(exp1,cl1) ;
            ClassExpression_SetClass(exp2,cl2) ;
            IF check(cl1 ↦ cl2) = not_safe THEN
                ANY annotation WHERE
                    annotation ∈ ANNOTATION - Annotation
                THEN
                    Annotation_NEW(annotation) ;
                    AddAxiomAnnotations(axiom, annotation) ;
                    SetAnnotationValue(annotation, "inconsistent")
                END
            END ;
            BEGIN
                subClass := subClass - {cl1} ||
                superClass := superClass - {cl2}
            END
        END
    END ;
addSubClassOfAxiom =
    ANY
        onto, axiom
    WHERE
        onto ∈ XOwl ∧ axiom ∈ Axiom - dom(axioms)
    THEN
        AddAxioms(onto,axiom)
    END
END
```