**SignAnalysis.cpp**

An inter-procedural simplified sign analysis.

This analysis maps numeric variables to a sign (negative, positive or zero), if it is statically determined to be singular, or else bottom (represented by null.

Flow functions are non-distributive for statements involving sums or products of two variables.

|  |
| --- |
| **Fields inherited from class** [**InterProceduralAnalysis**](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html) |
| [contexts](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#contexts), [contextTransitions](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html" \l "contextTransitions), [reverse](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#reverse), [worklist](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#worklist) |

**Functions: soot vs llvm:**

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**boundaryValue**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#boundaryValue(soot.SootMethod))([SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true" \o "class or interface in soot) method) |

Sign SignAnalysis::boundaryValue(Function\* entryPoint) {

return topValue();

}

Returns the initial data flow value at the program entry points. For forward analyses this is the IN value at the ENTRY to each entry method, while for backward analyses this is the OUT value at the EXIT to each entry method.

Note that this method will be called exactly once per entry point specified by the program representation.

**Specified by:**

[boundaryValue](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#boundaryValue(M)) in class [InterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Parameters:**

method - an entry point specified by the program representation

**Returns:**

the data flow value at the boundary

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**callEntryFlowFunction**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#callEntryFlowFunction(vasco.Context,%20soot.SootMethod,%20soot.Unit,%20java.util.Map))([Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>> context, [SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true) calledMethod, [Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true) unit, [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> inValue) |

Processes the inter-procedural flow function for a method call at the start of the call, to handle parameters.

**Specified by:**

[callEntryFlowFunction](http://padhye.org/vasco/apidocs/vasco/ForwardInterProceduralAnalysis.html#callEntryFlowFunction(vasco.Context,%20M,%20N,%20A)) in class [ForwardInterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/ForwardInterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Parameters:**

context - the value context at the call-site

calledMethod - the target (or one of the targets) of this call site

unit - the statement containing the method call

inValue - the data flow value before the call

**Returns:**

the data flow value at the entry to the called procedure

Sign SignAnalysis::callEntryFlowFunction(std::reference\_wrapper<Context<Function\*, BasicBlock\*, Sign>> context, Function\* target\_method, Instruction\* Insn, Sign in\_value) {

CallInst\* call = dyn\_cast<CallInst>(Insn);

unsigned ii = call->getNumArgOperands();

Sign entry\_value;

for(unsigned i=0;i<ii;i++) {

Value\* op = call->getArgOperand(i);

Function::arg\_iterator AI = target\_method->arg\_begin();

if(ConstantInt \*CI = dyn\_cast<ConstantInt>(op)){

if(CI->isZero()) {

entry\_value[AI[i].getName()] = SIGN::ZERO;

} else if(CI->isNegative()) {

entry\_value[AI[i].getName()] = SIGN::NEGATIVE;

} else {

entry\_value[AI[i].getName()] = SIGN::POSITIVE;

}

} else {

if(in\_value.find(op->getName()) != in\_value.end()) {

entry\_value[AI[i].getName()] = in\_value[op->getName()];

} else {

assert(false);

}

}

}

return entry\_value;

}

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**callExitFlowFunction**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#callExitFlowFunction(vasco.Context,%20soot.SootMethod,%20soot.Unit,%20java.util.Map))([Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>> context, [SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true) calledMethod, [Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true) unit, [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> exitValue) |

Processes the inter-procedural flow function for a method call at the end of the call, to handle return values.

**Specified by:**

[callExitFlowFunction](http://padhye.org/vasco/apidocs/vasco/ForwardInterProceduralAnalysis.html#callExitFlowFunction(vasco.Context,%20M,%20N,%20A)) in class [ForwardInterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/ForwardInterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Parameters:**

context - the value context at the call-site

calledMethod - the target (or one of the targets) of this call site

unit - the statement containing the method call

exitValue - the data flow value at the exit of the called procedure

**Returns:**

the data flow value after the call (returned component)

Sign SignAnalysis::callExitFlowFunction(std::reference\_wrapper<Context<Function\*, BasicBlock\*, Sign>> context, Function\* target\_method, Instruction\* Insn, Sign exit\_value) {

Sign return\_value;

Value\* LHS = dyn\_cast<Value>(Insn);

return\_value[LHS->getName()] = exit\_value[RETURN];

DBG(errs() << "callExitFlowFunction " << SIGN\_toString(exit\_value[RETURN]) << "\n";)

return return\_value;

}

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**copy**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#copy(java.util.Map))([Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> src) |

Sign SignAnalysis::copy(Sign src) {

return Sign(src);

}

Returns a copy of the given data flow value.

**Specified by:**

[copy](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#copy(A)) in class [InterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Parameters:**

src - the data flow value to copy

**Returns:**

a new data flow value which is a copy of the argument

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**meet**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#meet(java.util.Map,%20java.util.Map))([Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> op1, [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> op2) |

Sign SignAnalysis::meet(Sign op1, Sign op2) {

Sign result(op1);

for(auto &x : op2) {

if (op1.find(x.first) != op1.end()) {

SIGN sign1 = x.second;

SIGN sign2 = op1[x.first];

result[x.first] = SIGN\_meet(sign1, sign2);

} else {

result[x.first] = x.second;

}

}

return result;

}

Returns the meet of two data flow values.

**Specified by:**

[meet](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#meet(A,%20A)) in class [InterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Parameters:**

op1 - the first operand

op2 - the second operand

**Returns:**

a new data flow which is the result of the meet operation of the two operands

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**normalFlowFunction**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#normalFlowFunction(vasco.Context,%20soot.Unit,%20java.util.Map))([Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>> context, [Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true) unit, [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> inValue) |

Processes the intra-procedural flow function of a statement that does not contain a method call.

**Specified by:**

[normalFlowFunction](http://padhye.org/vasco/apidocs/vasco/ForwardInterProceduralAnalysis.html#normalFlowFunction(vasco.Context,%20N,%20A)) in class [ForwardInterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/ForwardInterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Parameters:**

context - the value context at the call-site

unit - the statement whose flow function to process

inValue - the data flow value before the statement

**Returns:**

the data flow value after the statement

Sign SignAnalysis::normalFlowFunction(std::reference\_wrapper<Context<Function\*, BasicBlock\*, Sign>> context, BasicBlock\* node, Sign in\_value) {

DBG(errs() << "In normalFlowFunction for function " << node->getParent()->getName() << ":" << node->getName() << "\n";)

Sign out\_value(in\_value);

for(auto &I: \*node) {

Instruction\* Insn = &I;

std::pair<StringRef, SIGN> p = signOf(Insn, out\_value);

if(p.first != BRANCH)

out\_value[p.first] = p.second;

for(auto &x : out\_value) {

DBG(errs() << x.first << " " << SIGN\_toString(x.second) << "; ";)

}

DBG(errs() << "\n\n";)

}

DBG(errs() << "\nIn normalFlowFunction out\_value: " << "\n";)

for(auto &x : out\_value) {

DBG(errs() << "\t " << x.first << " " << SIGN\_toString(x.second) << "\n";)

}

return out\_value;

}

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> | [**topValue**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#topValue())() |

Sign SignAnalysis::topValue() {

return Sign();

}

Returns an empty map.

**Specified by:**

[topValue](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html#topValue()) in class [InterProceduralAnalysis](http://padhye.org/vasco/apidocs/vasco/InterProceduralAnalysis.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>>

**Returns:**

the default data flow value (lattice top)

|  |  |  |
| --- | --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true" \o "class or interface in soot),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> |  | [**callLocalFlowFunction**](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.html#callLocalFlowFunction(vasco.Context,%20soot.Unit,%20java.util.Map))([Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[SootMethod](http://www.sable.mcgill.ca/soot/doc/soot/SootMethod.html?is-external=true),[Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true),[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)>> context, [Unit](http://www.sable.mcgill.ca/soot/doc/soot/Unit.html?is-external=true) unit, [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Local](http://www.sable.mcgill.ca/soot/doc/soot/Local.html?is-external=true),[SignAnalysis.Sign](http://padhye.org/vasco/apidocs/vasco/soot/examples/SignAnalysis.Sign.html)> inValue) |

(The soot one) Processes the intra-procedural flow function for a method call at the call-site itself, to handle propagation of local values that are not involved in the call.

**Parameters:**

context - the value context at the call-site

node - the statement containing the method call

inValue - the data flow value before the call

**Returns:**

the data flow value after the call (local component)