**ContextTransitionTable**

**Type Parameters:**

M - the type of a method

N - the type of a node in the CFG

A - the type of a data flow value

A record of transitions between contexts at call-sites.

The context transition table records a bidirectional one-to-many mapping of call-sites to called contexts parameterised by their methods.

If a call-site transition is not traversed in an analysis (e.g. a call to a native method) then it is listed as a "default site" which this table also records.

**Vasco methods and their llvm equivalents:**

|  |  |
| --- | --- |
| void | [**addTransition**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#addTransition(vasco.CallSite,%20vasco.Context))([CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)> callSite, [Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)> targetContext) |

Adds a transition to the table.

If the target context is specified as null, the source call-site is considered a "default site" from which transitions are unknown. This is used to model unpredictable targets, for example, when encountering calls to native methods.

Any previous transitions from the source call site to other contexts of the same called method are deleted.

**Parameters:**

callSite - the call-site which is the source of the transition

targetContext - the value context which is the target of the call-site

void ContextTransitionTable<M, N, A>::addTransition(CallSite<M, N, A> call\_site, Context<M, N, A> target\_context) {

if(!target\_context.is\_null) {

M target\_method = target\_context.getMethod();

if(transitions.find(call\_site) != transitions.end() && transitions[call\_site].find(target\_method) != transitions[call\_site].end()) {

Context<M, N, A> old\_target = transitions[call\_site][target\_method];

callers[old\_target].erase(std::remove(callers[old\_target].begin(), callers[old\_target].end(), call\_site), callers[old\_target].end());

}

transitions[call\_site][target\_method] = target\_context;

callers[target\_context].push\_back(call\_site);

}

else {

if(transitions.find(call\_site) != transitions.end()) {

for (auto e: transitions[call\_site]) {

Context<M, N, A> old\_target = transitions[call\_site][e.first];

callers[old\_target].erase(std::remove(callers[old\_target].begin(), callers[old\_target].end(), call\_site), callers[old\_target].end());

}

}

default\_call\_sites.push\_back(call\_site);

}

}

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>,[Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>>> | [**getCallers**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#getCallers())() |

std::unordered\_map<Context<M, N, A>, std::vector<CallSite<M, N, A>>> ContextTransitionTable<M, N, A>::getCallers(void) {

return callers;

}

Returns an unmodifiable view of the mapping from contexts to their callers.

|  |  |
| --- | --- |
| [Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>> | [**getCallers**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#getCallers(vasco.Context))([Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)> target) |

std::vector<CallSite<M, N, A>> ContextTransitionTable<M, N, A>::getCallers(Context<M, N, A> target) {

return callers[target];

}

Returns the callers of a value context.

**Parameters:**

target - the target value context

**Returns:**

a set of call-sites which transition to the given target context

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>,[Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>>> | [**getCallSitesOfContexts**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#getCallSitesOfContexts())() |

std::unordered\_map<Context<M, N, A>, std::vector<CallSite<M, N, A>>> ContextTransitionTable<M, N, A>::getCallSitesOfContexts(void) {

return call\_sites\_of\_contexts;

}

Returns an unmodifiable view of a mapping from calling contexts to all their call-sites.

|  |  |
| --- | --- |
| [Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>> | [**getDefaultCallSites**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#getDefaultCallSites())() |

std::vector<CallSite<M, N, A>> ContextTransitionTable<M, N, A>::getDefaultCallSites(void) {

return default\_call\_sites;

}

Returns an unmodifiable view of the set of call-sites marked "default".

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html" \o "type parameter in ContextTransitionTable),[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>> | [**getTargets**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#getTargets(vasco.CallSite))([CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)> callSite) |

std::unordered\_map<M,Context<M, N, A>> ContextTransitionTable<M, N, A>::getTargets(CallSite<M, N, A> call\_site) {

return transitions[call\_site];

}

Returns the targets of a call-site.

**Parameters:**

callSite - the source of the transition

**Returns:**

a map of target methods to target contexts

|  |  |
| --- | --- |
| [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>,[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html" \o "type parameter in ContextTransitionTable),[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>>> | [**getTransitions**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#getTransitions())() |

std::unordered\_map<CallSite<M, N, A>, std::unordered\_map<M, Context<M, N, A>>> ContextTransitionTable<M, N, A>::getTransitions(void) {

return transitions;

}

Returns an unmodifiable view of context transitions.

|  |  |
| --- | --- |
| [Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>> | [**reachableSet**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#reachableSet(vasco.Context,%20boolean))([Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)> source, boolean ignoreFree) |

Computes a reachable set of value contexts from a particular source by traversing the context transition table. Note that the source context itself is only reachable from itself if it there is a recursive call to it (i.e. a context is not reachable to itself by default).

**Parameters:**

source - the source context

**Returns:**

a set of contexts reachable from source

**Data structures:**

**(TO be defined in the ContextTransitionTable.h)**

|  |  |
| --- | --- |
| protected  [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>,[Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>>> | [**callers**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#callers) |

std::unordered\_map<Context<M, N, A>, std::vector<CallSite<M, N, A>>> callers;

A map from contexts to a set of call-sites that transition to it.

|  |  |
| --- | --- |
| protected  [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>,[Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>>> | [**callSitesOfContexts**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#callSitesOfContexts) |

std::unordered\_map<Context<M, N, A>, std::vector<CallSite<M, N, A>>> call\_sites\_of\_contexts;

A map of contexts to call-sites present within their method bodies.

|  |  |
| --- | --- |
| protected  [Set](http://download.oracle.com/javase/6/docs/api/java/util/Set.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>> | [**defaultCallSites**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#defaultCallSites) |

std::vector<CallSite<M, N, A>> default\_call\_sites;

A set of call-sites from which transitions are unknown.

|  |  |
| --- | --- |
| protected  [Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[CallSite](http://padhye.org/vasco/apidocs/vasco/CallSite.html" \o "class in vasco)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>,[Map](http://download.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html" \o "type parameter in ContextTransitionTable),[Context](http://padhye.org/vasco/apidocs/vasco/Context.html)<[M](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[N](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html),[A](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html)>>> | [**transitions**](http://padhye.org/vasco/apidocs/vasco/ContextTransitionTable.html#transitions) |

std::unordered\_map<CallSite<M, N, A>, std::unordered\_map<M, Context<M, N, A>>> transitions;

A map from call-sites to contexts, parameterised by the called method.