TaskMiner Research Project, Call Site Annotation Plan

The Goal: Automatically annotate function calls with OpenMP pragmas and compare the annotated code with manual annotated code.

Plan: Use an intra-procedural analysis to determine which parameters and global variables a function access and classify the access among: *in*, *out* and *inout*. This should only consider leaf functions.

Example:

```
void foo(int* p1, int* p2) {
	int isOdd = *p1 % 2;

	if (isOdd) {
		*p2 = *p1 - 1;
	}
	else {
		*p2 = *p1;
	}
}
```

The algorithm will produce the follow output, note that local variables are irrelevant:

Input	Output	InOut
{p1}	{p2}	{}

Based on that information the analysis will proceed to annotate call sites with OpenMP task directives. Further analysis may be necessary to prove that *p1* and *p2* do not alias at the call site location, if that is not possible we need to be conservative and merge the access attributes of *p1* and *p2*. After that the following output is desired:

```
int main() {
	#pragma omp parallel
	#pragma omp task depend(in:p1, out:p2)
	foo(p1, p2);
}
```