

### Concept Chapter3 Chapter 3. Concept

From a high point of view, this work tries to weaken the harsh requirements on SCoPs in order to make Pollys loop optimizations applicable on a wider range of programs. Apart from the implementation work, which will be described in the next chapter, immense effort has been made on the concepts and key ideas behind. We believe that these ideas and the knowledge gained during the work is very valuable not only for future work on SPolly or one of its bases but also for other approaches facing similar situations. On the way to a working version many pitfalls have been encountered that should be avoided in the future, perhaps with similar approaches we worked out.

#### SPolly In A Nutshell

**Speculative SCoPs** A speculative valid SCoPs (sSCoPs) is defined similar to a valid SCoPs but with weakened conditions. General SCoPs need to fulfill all constraints listed in fig:SCoPconstraints, while sSCoPs renounce the restrictions on aliasing and partially those on function calls.

#### sSCoP extraction

[ TODO ref to VMAD cause their size explodes ] The sSCoP extraction was designed to simplify the method versioning for functions containing several speculative valid SCoPs. It creates a new sub function for every sSCoP and inserts a call in the former place. Later on this call could be inlined again but this is not implemented yet. On the one hand side the creation of profiling and parallel versions as well the later function exchanging becomes a lot easier and cheaper this way. On the other hand it is the first step to multiple specialized versions, e.g., with constant instead of variable loop bounds.

#### sSCoP Versions

##### Introduced tests

##### Region Scores

frame=none wrapfigure[r0.4 [complete static sSCoP] Primitives/Code/sSCoPstatic.c lst:sSCoPstatic

[Branch within a sSCoP] Primitives/Code/sSCoPbranch.c lst:sSCoPbranch

[irreversible call within a sSCoP] Primitives/Code/sSCoPprintf.c lst:sSCoPprintf example sSCoPs

fig:ScoredSCoPs