

SCL Transformation

Deprecated since 0.12

This article is deprecated. The described features are no longer available in current releases.

- [Core SCChart to SCL Transformation](#)
- [SCL Transformation Optimization](#)

Core SCChart to SCL Transformation

coming soon...

SCL Transformation Optimization

The (naive) generated SCL code can be optimized. Several optimizations are implemented or planned.

For the following optimizations two assumptions are made:

- A transition does not require an explicit goto instruction
- A thread terminates, iff its parallel block is exited (via the next par statement or join)

The optimization level can be specified in the CoreToSCLTransformation class.

Optimization	Description
OPTIMIZE_GOTO	Removes all goto instructions, that target a label, that follows that goto.
OPTIMIZE_LABEL	Removes all unreferenced labels.
OPTIMIZE_SELFLOOP	If a state has only one outgoing non-default transition, the expression of that transition is negated, the transition target is changed to itself and the implicit selfloop is deleted. So, if the original expression is true, the sequential control flow is used to proceed and no goto jump is necessary.
OPTIMIZE_STATEPOSITION (temporary deactivated)	In some cases label and goto instructions can be removed, if the states in a region are ordered differently. This optimization checks for redundant jumps and changes the order of the states if possible and necessary.
OPTIMIZE_SINGLETRANSITIONS (planned)	
OPTIMIZE_WTRANSITIONS (planned)	