Editor's report for the Concepts TS

Andrew Sutton <asutton@uakron.edu>

Date: 2015-10-02

Document number: N4554

Acknowledgments

We wish to thank all of the reviewers, including both the Core and Evolution Working groups, and those who contributed national body comments. Many thanks to those who have sent pull requests or emailed me privately to identify wording issues or suggest fixes.

History

The current workding draft is P0121R0. This version of the text includes the handful editorial changes made since the 2015-07-20 teleconference. This is the same version of the document that is published on the ISO web site (although without the ISO cover page).

The following sections summarize the significant changes that have been applied to the Concepts TS since the 2015-01-12 meeting in Skillman, NJ. More recent changes are appear first.

Since the 2015-07-20 teleconference

The following changes have been applied since the Jul 28 teleconference.

- Update the `__cpp_concepts` value to reflect the date of acceptance.
- Editorial fixes (missing `typename` in an example).

Since the 2015-06-29 teleconference

The following changes were made after Jun 29 telconference.

- [dcl.spec.auto] Allows *constrained-type-specifier*s in variable types and return types.
- [temp.constr.decl, temp.constr.resolve] Requires concept resolution for *template-id*s in *constraint-expression*s when they name a concept.

- [temp.constr.conv] Remove example showing equivalence to `is_convertible`.

- [temp.param] Fix a grammar ambiguity in *constrained-parameter*.

- [expr.prim.general] Clarifications about placeholders designated by *constrained-type-specifier*s.

- Fix an issue raised by Hubert Tong in the last teleconference. As specified, the associated constraints of a template declaration would not be equivalent to the *constraint-expression* of a requires clause (comparing expressions and constraints).

The proposed resolution is to define "associated constraints" in terms of constraint-expressions. Changes are somewhat significant.

- The formulation of associated constraints is written in terms of logical AND expressions.

- Normalization is redefined so that it transforms an expression into a constraint. Effectively, the "inner item" from the previous version are now the "top level items". A new rule is added to ensure that we recursively normalize nested-requirements.

- Satisfaction and ordering are defined in terms of a declaration's "normalized constraints" (the result of normalizing the associated constraints).

- Template *constrained-parameter*s and *template-introduction*s now introduce *constraint-expression*s instead of predicate constraints.

- Update wording in [dcl.spec.auto] and [dcl.fct] to refer to expressions instead of constraints. Also, define the constraints for variable/return type deduction in terms of an expression instead of a constraint.

- Various editorial issues.

Changes since the 2015-02 mailing.

- [temp.constr.order], fix the definition of "at least as constrained".

- [temp.constr.resolve], support concept resolution template-ids during constraint normalization.

- Replace [intro.feature] with the suggested wording by SG10 (given in SD-6).

- [expr.prim.general] Fix errors in the examples.

- [expr.prim.requires] Make this note normative wording.

- [temp.param] Revise constrained-parameter syntax so that packs do not accept default arguments.

- Numerous editorial fixes and clarifications.