JTC1/SC22/WG14 N2770
Title: Remove UB for incomplete types of function parameters
Author: Martin Uecker, University of Göttingen
Date: 2021-07-11

After removal of K&R function definitions, there appear to be some statements left that make no sense anymore, e.g. incomplete types of parameters in function definitions are simultaneously a constraint violation and undefined behavior via "shall" statements in the constraints and the semantics sections, respectively. It is proposed to remove this case of UB and to make two other minor changes.

Proposed Change 1:

6.7.6.3 Function declarators
Constraints

3 After adjustment, the parameters in a parameter type list in a function declarator that is part of a definition of that function shall not have incomplete type.

6.9.1 Function definitions
Semantics

7 The declarator in a function definition specifies the name of the function being defined and the types (and optionally the names) of all the parameters; the declarator also serves as a function prototype for later calls to the same function in the same translation unit. The type of each parameter is adjusted as described in 6.7.6.3~~**; the resulting type shall be a complete object type**~~.

J.2 Undefined behavior

1 The behavior is undefined in the following circumstances:

~~**— An adjusted parameter type in a function definition is not a complete object type (6.9.1).**~~

Proposed Change 2:

6.5.2.2 Function calls

103) A function can change the values of its parameters, but these changes cannot affect the values of the arguments. On the other hand, it is possible to pass a pointer to an object, and the function can then change the value of the object pointed to. A parameter declared to have array or function type is adjusted to have a pointer type as described in ~~**6.9.1.**~~**6.7.6.3.**

Proposed Change 3:

6.5.2.2 Function calls

~~**8 No other conversions are performed implicitly; in particular, the number and types of arguments are not compared with those of the parameters in a function definition that does not include a function prototype declarator.**~~
FOOTNOTE TO 7) No other conversions are performed implicitly.

Acknowledgment: I want to thank Jens Gustedt for helpful comments on this paper.