Daniel Rao

CS-354

Homework (5)

P344: 7.1

I think the answer boils down to the matter of opinion. Name equivalence is more of an abstract concept. It allows the programmer to decide specifically exactly which type should be considered compatible, and that which are should be incompatible, rather than simply basing the distinction based on weather the type share the same implementation.

Structural equivalence isn't bad either. For example taking structs in C. Like if we have two structs that represent two related piece of hardware that has exact same hardware configuration, can be set equal to one another. If I want to have more them one modular design where one structs checks a certain bit in a field of a register and another checks a different bit. Structural equivalence checks

if the contents of both structs are the same else it would result in an error.

7.2 All 4 arrays are structurally equivalent.

Under name equivalence, array D is incompatible with others. Under struct equivalence A and B are incompatible with C, under loose name equivalence A, B and C are all mutually compatible.