		example	atomic/1	compound/1	compound_name_arity/3	compound_name_arguments/3	functor/3	=../2 (univ)
atomic		foo	true	false	"ERROR: Type error" on disassembly: compound_name_arity(foo,X,Y).	"ERROR: Type error" on disassembly: compound_name_arguments(foo,X,Y).	functor(foo,foo,0).	foo =.. [foo]
compound	arity 0	foo()	false	true	compound_name_arity(foo(),foo,0).	compound_name_arguments(foo(),foo,[]).	"ERROR: Domain error" on disassembly: functor(foo(),X,Y).	"ERROR: Domain error" on disassembly: foo() =.. L.
	arity 1	foo(1)	false	true	compound_name_arity(foo(1),foo,1).	compound_name_arguments(foo(1),foo,[1]).	functor(foo(1),foo,1).	foo(1) =.. [foo,1]
	arity 2	foo(1,2)	false	true	compound_name_arity(foo(1,2),foo,2).	compound_name_arguments(foo(1,2),foo,[1,2]).	functor(foo(1,2),foo,2).	foo(1,2) =.. [foo,1,2]

Approach for disassembly:

1) Check whether it is a compound term with compound/1
2) Disassemble with compound_name_arity/3 or compound_name_arguments/3

Approach for assembly:

- Always use compound_name_arity/3 to construct a skeleton, e.g. foo(_3958, _3960)
- Always use compound_name_arguments/3 to construct an parameter-adorned compound term

These must throw domain error on disassembly
of arity 0 compound terms because there is no way to
distinguish a dissembled "foo" and "foo()":
Disassembly would become a surjective mapping.

version of 2020-05-15