

Before type-casting an *e* variable ("as_a"), we often want to check the validity of the operation (this is quite similar in concept to \$cast in SystemVerilog). The reason is simple, in case the casting operation failed we would end up with a fatal error at run-time that otherwise could have been avoided. But how?

Here's one elegant way to do it:

{code}

```
type ttt: [E1=3, E2=10];
```

```
extend sys {  
  run() is also {  
    var x: byte = 3; // try x=7 too  
    if x.as_a(ttt) in all_values(ttt) then { out( *** X IS IN RANGE *** ) };  
  };  
};
```

{/code}

In this small example we were trying to convert (type-cast) *x* from *byte* to *ttt*. The method `all_values()` returns a list of all possible scalar types (in our case - for type *ttt*). Try this code with *x=3* and with *x=7* and see what happens.