Exercise 1.7 Subtleties of `foo = bah` and `foo := @(bah)`

```
{a,b,c,d,e,f,h#}::Indices.

foo := B_{b} A_{a}.
bah := A_{a} C_{c}.

# cdbBeg(print.0107)
print("foo = "+str(foo))
print("bah = "+str(bah)+"\n")

print("type foo = "+str(type(foo)))
print("type bah = "+str(type(bah))+"\n")

print("id foo = "+str(id(foo)))
print("id bah = "+str(id(bah))+"\n")

bah = foo

print("foo = "+str(foo))
print("bah = "+str(bah)+"\n")

sort_product (foo)

print("bah = "+str(bah)+"\n")

print("id foo = "+str(id(foo)))
print("id bah = "+str(id(bah))+"\n")

bah := @(foo).

print("id foo = "+str(id(foo)))
print("id bah = "+str(id(bah))+"\n")
# cdbEnd(print.0107)
```

```
1  foo = B_{b} A_{a}
2  bah = A_{a} C_{c}
3
4  type foo = <class 'cadabra2.Ex'>
5  type bah = <class 'cadabra2.Ex'>
6
7  id foo = 4600022944
8  id bah = 4600023000
9
10 foo = B_{b} A_{a}
11 bah = B_{b} A_{a}
12
13 bah = A_{a} B_{b}
14
15 id foo = 4600022944
16 id bah = 4600022944
17
18 id foo = 4600022944
19 id bah = 4600023784
```

Note that the line numbers referenced in the following are those of the output above not those of the Cadabra source.

- Lines 7 and 8 show that the objects `foo` and `bah` point to distinct areas of memeory (i.e., they point to different objects).
- Lines 10 and 11 show the result of the statement `bah = foo`.
- Line 13 shows that `bah` has changed after the statement `sort_product (foo)`.
- Lines 15 and 16 verifies that `foo` and `bah` point to the same object (so changes in `foo` will be seen by `bah`, as just noted).
- Lines 18 and 19 shows that after `bah := @(foo)` the symbols `bah` and `foo` no longer point to the same object.