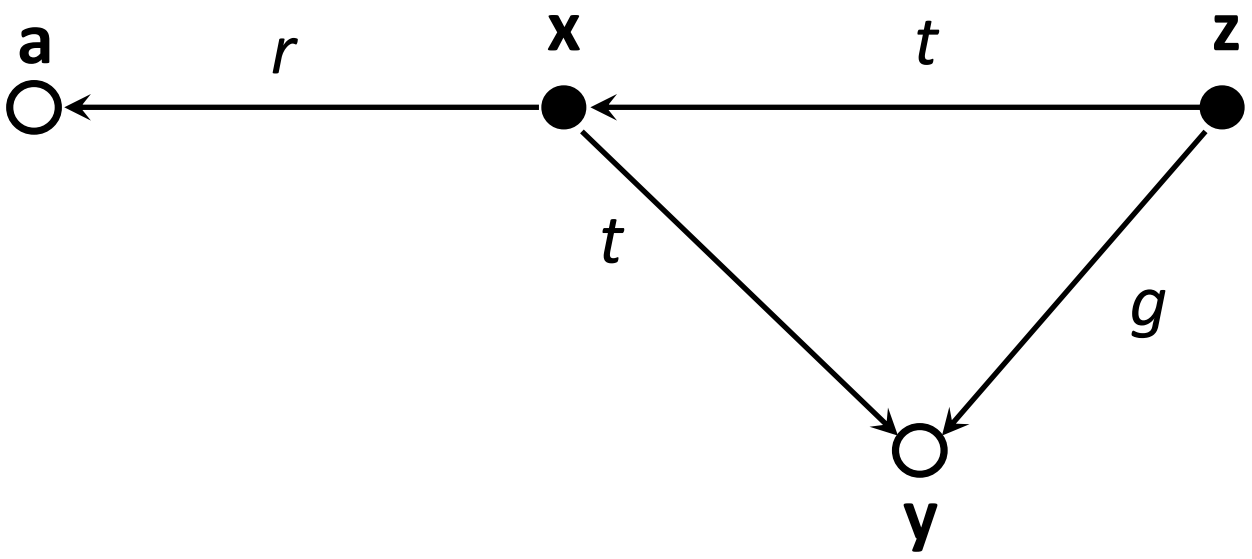
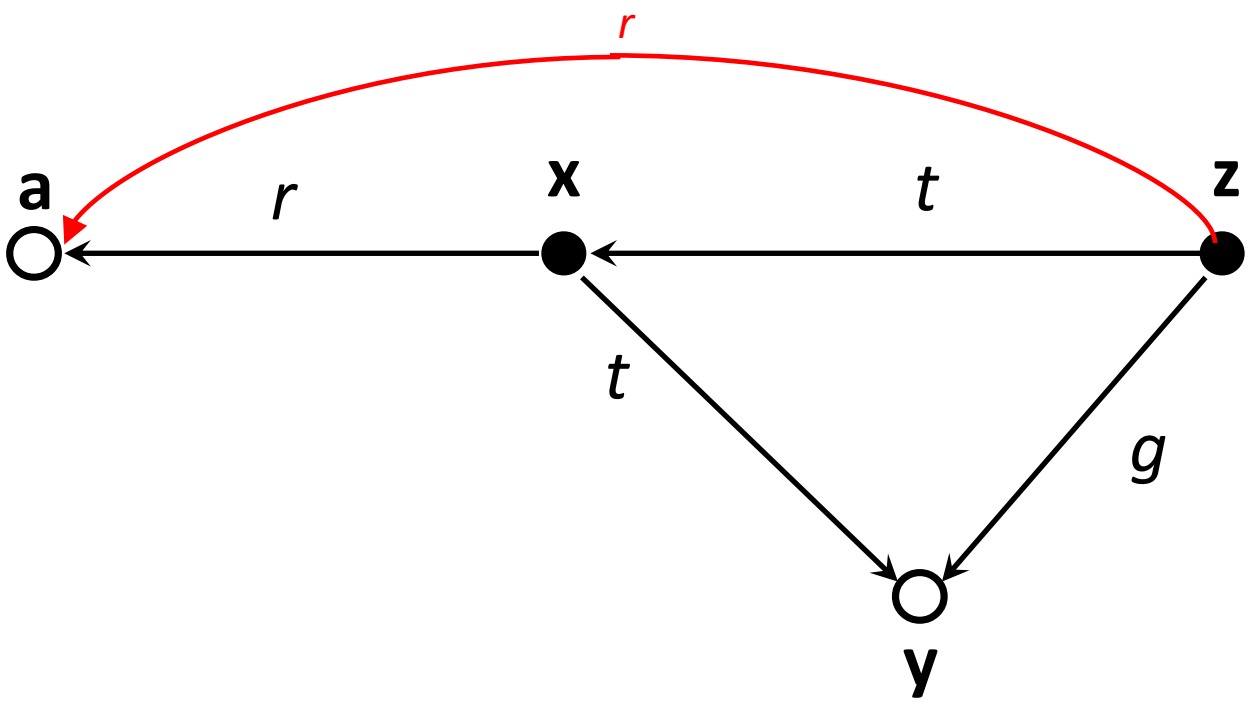
Example of Take-Grant Rule Applications

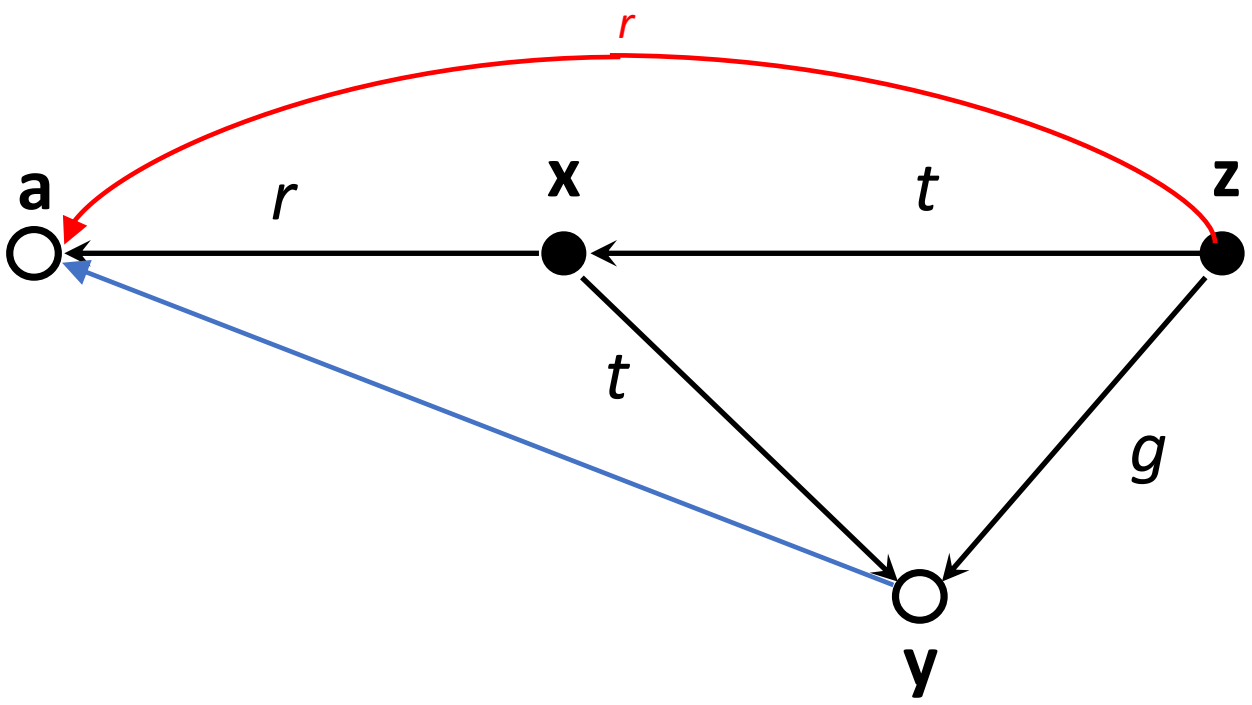
How can **y** obtain *r* rights over **a**?



z takes (*r* to a) from x



z grants (r to a) to y



So the Witness Is:

1. **z** takes (*r* to **a**) from **x**
2. **z** grants (*r* to **a**) to **w**

What Homework 1, Problem 3 Asks

Problem Statement

- Justify the statement: "Suppose two subjects s_1 and s_2 are created and the rights in $A[s_1, o_1]$ and $A[s_2, o_2]$ are tested. The same test for $A[s_1, o_1]$ and $A[s_1, o_2] = A[s_1, o_2] \cup A[s_2, o_2]$ will produce the same result."
- Would it be true if one could test for the absence of rights as well as for the presence of rights?

2 create subjects

	o_1	o_2	s_1	s_2
s_1				
s_2		r		

create subject s_1
create subject s_2
enter r into $A[s_2, o_2]$
if r in $A[s_2, o_2]$
then

. . .

[[IMPORTANT: these would be in commands]]

1 create subject

	o_1	o_2	s_1
s_1		r	

create subject s_1
enter r into $A[s_1, o_2]$
if r in $A[s_1, o_2]$
then

. . .

[[IMPORTANT: these would be in commands]]

Problem: prove the result of executing the two command sequences produces the same result

Test for Absence of Rights

- Current access control matrix model allows conditional tests of the form `if` *`r`* `in` *`A`*`[`*`s`*`,`*`o`*`]` but ***not*** `if` *`r`* `not in` *`A`*`[`*`s`*`,`*`o`*`]`
- The problem asks, what if ***both*** are allowed? Would the two command sequences still produce the same results?