Lab 6-7. Access Control: Take-Grant. Predicate can.share

https://profs.info.uaic.ro/~fltiplea/IS/AccessControl_02.pdf

Take-Grant (**TG**) is a model that helps in determining the protection rights (e.g., read or write) in a computer system. The model was introduced by Jones, Lipton, and Snyder in

"A linear time algorithm for deciding subject security" Journal of the ACM 24(3):455-464, 1977 (http://flint.cs.yale.edu/cs428/doc/liptonTG.pdf)

in order to show that it is possible to decide on the safety of a computer system even when the number of subjects and objects are very large, or unbound. This can be accomplished in linear time based on the initial size of the system.

It is of a DAC - Discretionary Access Control type, bazed on permission access rules on a system resources. In this model the users are allowed to transfer their rights at their own discretion.

TG is a protection model which consists of a set of subjects (users, processes etc) and objects (resources, directories, files, etc), a set of state-transitions rules, and a directed graph that implements these rules. The nodes in this digraph are the subjects and objects of the model. The directed edges between the nodes represent the rights that one node has over the linked node.

Denotation:
(x,y) is the set of access rights on the edge from node x to node y. If r is an element of (x,y), i.e., $r \in (x,y)$, then node x has the right r for node y.

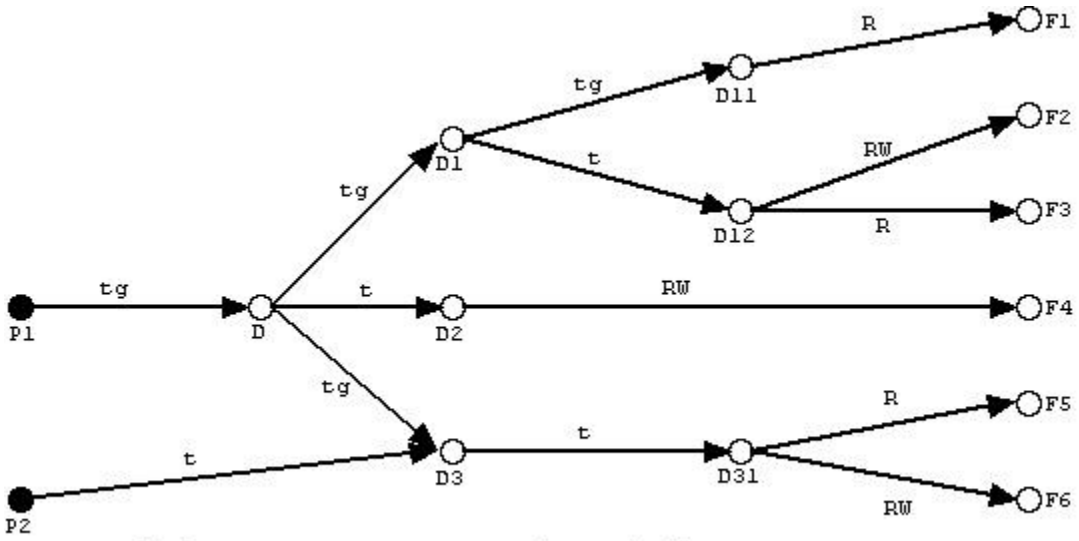
Take
The ***take right t:*** for a subject s to have the right t on an object x, it means that subject s can take any rights that x possesses, $t \in (x,y)$.

Grant
The ***grant right g***: for a subject s to have the right g over a subject/object x, it means that subject s can grant (share) any of the rights it possesses to subject/object x, , $g \in (x,y)$.

Example 1

The figure1 there is a graphical representation of a TG digraph. In this digraph **P1** and **P2** are subjects, and **D**s and **F**s are objects.

Remark If a subject/object has read (take) capability on an object, it can take any of the capabilities that this object has. Similarly, if a subject/object has write (grant) capabilities to an object, it can grant any of its capabilities to that object.



Take-grant representation of directory structure

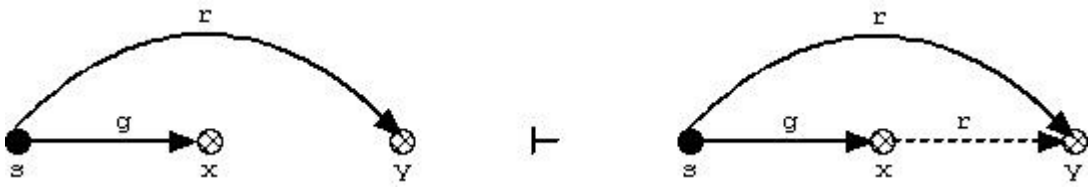
figure 1

State Transition Rules

Rule 1 (Take):
Let *s* be a subject, $t \in (s, x)$ and a right $r \in (x,y)$, where *x*, *y* are nodes. To add *r* to (*s*,*y*) use: ***s* take *r* for *y* from *x***.
This is shown in the following picture where nodes *x* and *y* can be either subjects or objects.

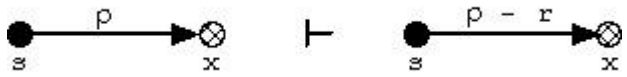
Rule 2 (Grant):

Let *s* be a subject, $g \in (s, x)$ and a right $r \in (s,y)$, where *x*, *y* are nodes. To add *r* to (*x*,*y*) use: ***s* grant *r* for *y* to *x***.
This is shown in the following picture where nodes *x* and *y* can be either subjects or objects.



Rule 3 (Create):
If *s* is a subject and *p* is a set of rights, then the command:
s **create** *p* for **new** {**subject** or **object**} *x* will add a new node *x* and sets (*s*,*x*) = *p*.

Rule 4 (Remove):
If *s* is a subject and *x* is a node, then the command:
s **remove** *r* for *x* will remove the right *r* from (*s*,*x*).
This is shown in the following picture where node *x* can be either a subject or an object.



Example 2

Describe the rules used to add a new object F7 in the directory D11 (in the TG digraph from figure 1) such that at the end D11 will have the Read/Write rights over F7.

1. *P1* **create** *RW* for **new object** *F7*
2. *P1* **take** *t* for *D1* from *D*
3. *P1* **take** *g* for *D11* from *D1*
4. *P1* **grant** *RW* for *F7* to *D11*

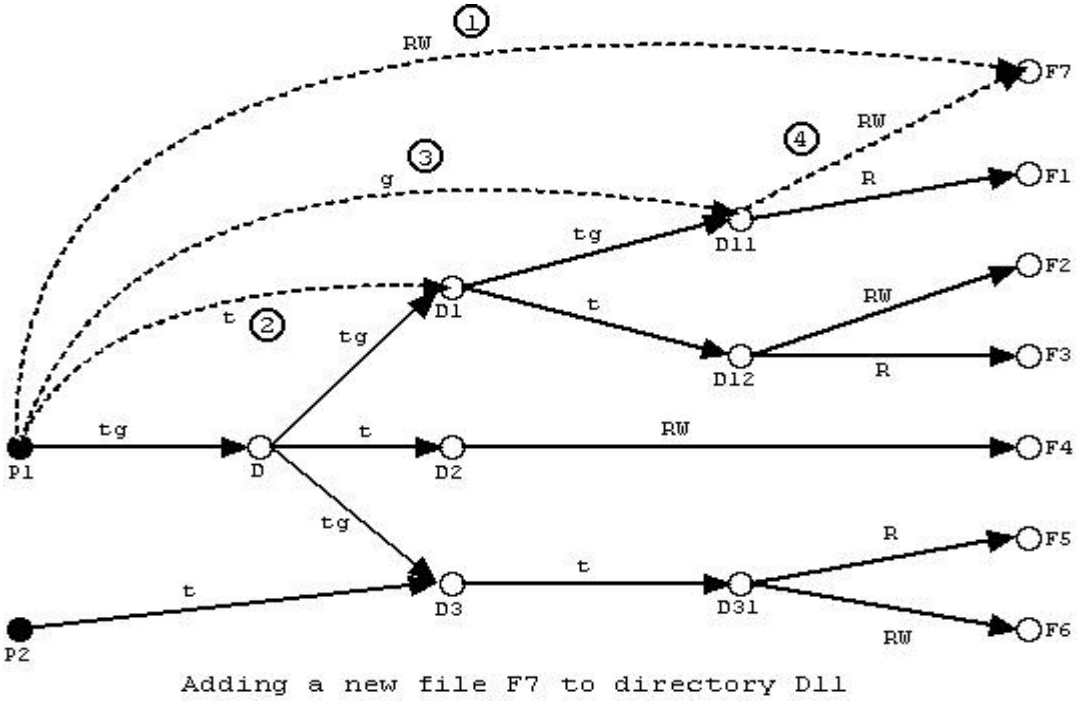


figure 2

Example 3 Describe the rules that must be applied in the TG model in figure 3 such that s_1 to have the right r over the object z. Build the access matrix model.

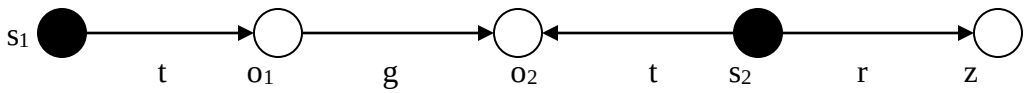


figure 3