# A Linear Time Algorithm for Deciding Subject Security

R. J. Lipton and L. Snyder

Research Report #72

June 1976

This work was supported in part by the Office of Naval Research grant N00014-75-C-0752 and in part by NSF grant DCR74-24193.

## I. Introduction

The theoretical analysis of systems for protecting the security of information should be of interest to the practitioner as well as the theoretician. The practitioner must convince users that the integrity of their programs and files are maintained, i.e. he must convince them that the operating system and its mechanisms will correctly protect these programs and files. Vague or informal arguments are unacceptable since they are often wrong. Indeed the folklore is replete with stories of "secure" systems being compromised in a matter of hours.

A primary reason for the abundance of these incidents is that even a small set of apparently simple protection primitives can often lead to complex systems that can be exploited, and therefore compromised, by some adversary. But it is precisely this fact, simple primitives with complex behavior, that lures the theoretician. Our purpose here is to present a concrete example of a protection system and then to completely analyze its behavior.

Our motivation for doing this analysis is twofold. The protection system that we will study is not one we invented, rather it is one that has been defined, discussed, and studied by those in operating systems (Denning and Graham [2], Cohen [1], Jones [4]). This point is most important, for the space of possible protection systems is exceedingly rich and it is trivial to think up arbitrary systems to study. We are not interested in arbitrary systems, but in systems that have practical application.

The above motivation is necessary but not sufficient for us to

establish that these questions should interest the theoretician. Our second reason for studying these problems is that in a natural way they can be viewed as "generalizations of transitive closure." Roughly these protection questions can be modeled as:

*Given:* A directed labeled graph G and a set of rewriting rules $\mathcal{R}$.

*Determine:* Whether or not there is a sequence of graphs $G_1$, $G_2$, ..., $G_n$ such that $G = G_1$, $G_n$ has property X, and $G_{i+1}$ follows from $G_i$ by some rule in $\mathcal{R}$.

Here property X encodes that there is a protection violation in $G_n$. Our goal then is to show that it is impossible to reach such a $G_n$, i.e. that a protection violation is impossible.

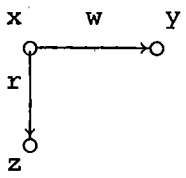
Property X is frequently stated as

X: there is an edge from vertex p to q with label $\alpha$.

For these properties our protection questions do indeed look very much like transitive closure questions. Indeed if the rules $\mathcal{R}$ only allowed the addition of edges, then these problems would be easily solved by known methods. They are not so simple. The rules of interest to those in protection, and the particular rules we will study, allow new vertices to be added. This simple change of allowing graphs to "grow new vertices" make these problems challenging. Indeed the particular one we will study is no longer even obviously decidable.

Let us now make the above concrete by introducing the particular protection system we will study. We consider directed graphs whose arcs

are labeled with an r or a w or a c. While we will manipulate these graphs as formal objects it is helpful to keep in mind the following informal semantics: A vertex corresponds to a "user", r = "read", w = "write", c = "call". If there is a directed arc from x to y with label r (respectively w,c), then x can read y (respectively write, call). For example, in the graph

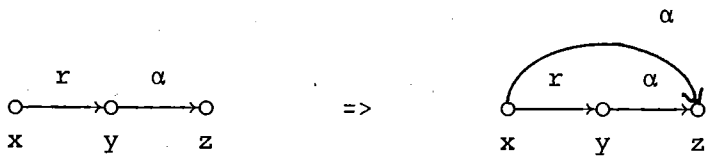



x can write y, x can read z, but y cannot write z since this edge is missing. More formally, a protection graph is a finite, directed graph with each arc labeled by a nonempty subset of {r,w,c}. We interpret the case where an edge is labeled with other than a single element to mean that multiple "rights" are allowed.

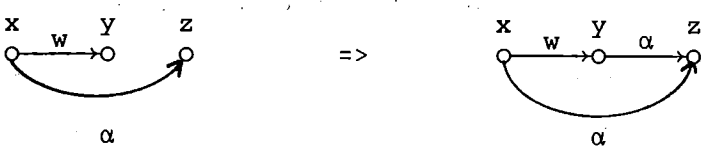
This protection model, called the take and grant system, is now completed by presenting five rewriting rules.

1. *Take:* Let x, y, and z be three distinct vertices in a protection graph and let there be an arc from x to y with label $\gamma$ such that $r \in \gamma$ and an arc from y to z with some label $\alpha \subseteq \{r,w,c\}$. Then the take rule allows one to add the arc from x to z with label $\alpha$ yielding a new graph G'. Intuitively x takes the ability to do $\alpha$ to z from y. We will represent* this rule by

*Here and in later diagrams we abuse notation by writing an explicit right as as arc label ($x \xrightarrow{r} y$) to mean the arc label contains that right (i.e., $x \xrightarrow{\gamma} y$ such that $r \in \gamma$).



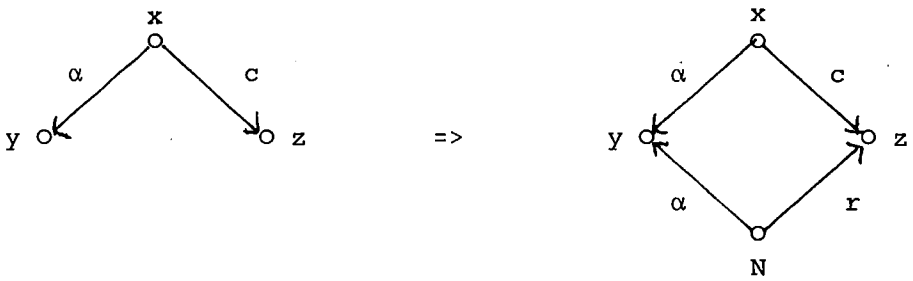

2. *Grant:* Let x, y and z be distinct vertices in a protection graph G and let there be an arc from x to y with label $\gamma$ such that $w \in \gamma$ and an arc from x to z with label $\gamma \subseteq \{r,w,c\}$. Then the grant rule allows one to add an arc from y to z with label $\alpha$ yielding a new graph G'. Intuitively x grants y the ability to do $\alpha$ to z. In our representation grant is given by:



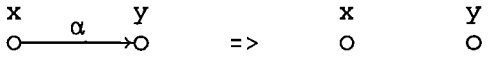

3. *Create:* Let x be any vertex in a protection graph, then create allows one to add a new vertex N and an arc from x to N with label {r,w,c} yielding a new graph G'. Intuitively x creates a new user that it can read, write and call. In our representation


4. *Call:* Let x, y and z be distinct vertices in a protection graph G and let $\alpha \subseteq \{r,w,c\}$ be an arc from z to y and $\gamma$ an arc from x to z such that $c \in \gamma$. Then the call rule allows one to add a new

vertex N, an arc from N to y with label α, and an arc from N to z with label r yielding a new graph G'. Intuitively x is calling a program z and passing parameters y. The N "process" is created to effect the call: N can read the program z and can α the parameters. In our representation




5. *Remove:* Let x and y be distinct vertices in a protection graph G with an arc from x to y with label α. Then the remove rule allows one to remove the arc from x to y yielding a new graph G'. Intuitively x removes its rights to y. In our representation,



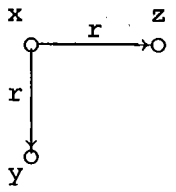

The remove rule is defined mainly for completeness, since protection systems tend to have such a rule. Moreover, we expect to study properties of protection systems other then protection violations which will use remove in a crucial way. But, for the present, remove may be ignored.

The operation of applying one of the rules to a protection graph G yielding a new protection graph G' is writen $G \vdash G'$. As usual $G \vdash^{*} G'$ denotes the reflexive, transitive closure.

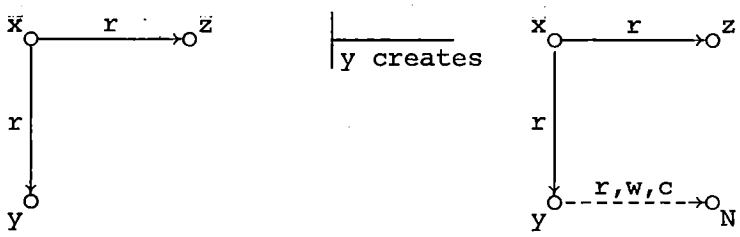
An important technical point is this system is monotone in the sense

that if a rule can be applied, then adding arcs cannot change this. This property is crucial later.

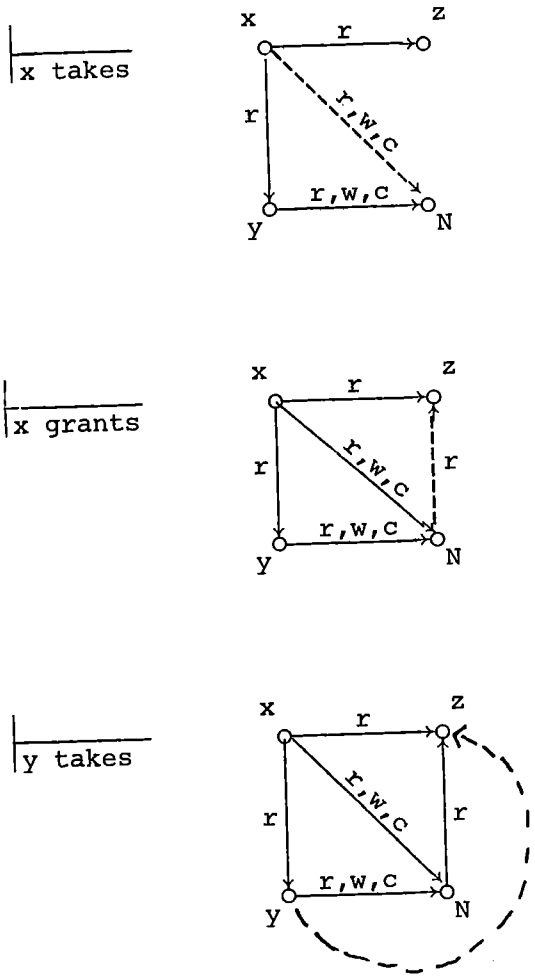
Now that we have seen the rules, let us look at their behavior. We will start with a simple question: in the graph




is it possible for y to r z? The answer is obviously no since there is no r arc from y to z. But we are really asking: is there a sequence of rule applications that leads to a graph with an r arc from y to z? More generally, say p can $\alpha$ q if there is a series of rules that leads to a graph with an arc from p to q. Then to state our question more precisely, we ask: is it true that y can r z? Clearly, without create, the answer is no since none of the operations take, grant or call can apply. The following sequence of applicatons of the rules* shows that by using create the answer is yes:




*In the diagrams, dashed lines are used only as a visual aid to set off the added arcs of the current operation.




This example demonstrates the kind of clean graph-type problems we will be studying. Our main theorem is state in the next section. This theorem presents a complete answer to the question: is is true that p can $\alpha$ q? Indeed this theorem leads easily to a linear time algorithm for answering the question.

A final word about how this theorem contributes to our understanding of protection. Each user of a protection system needs to know:

what information of mine can be accessed by others;

what information of others can be accessed by me?

The question is vague is general, but here it is rendered in the simple question: is it true that p can $\alpha$ q?

The types of protection models studied here have received considerable attention recently. Our approach is related closely to the interesting work of Harrison, Ruzzo, and Ullman [3]. They show that what can be called the "uniform safety problem" is undecidable. Interpreted as a graph model, their result says that given an arbitrary set of rules (similar in spirit to take, grant, etc.) and an initial graph, it is undecidable whether or not there will ever be an arc from p to q with label $\alpha$. This is a uniform problem in the sense that the rules are arbitrary. Even when the rules have to satisfy certain additional constraints the results of [3] and the results of Lipton and Snyder [5] show that protection is impractically complex.

Our view here is that since the uniform protection problem is so difficult and since operating systems usually require only one fixed set of protection rules, then the nonuniform problem should be studied. As stated before we chose the take and grant system by studying the protection literature.

## II. Basic Results

Our objective is to show that there are two simple conditions that are necessary and sufficient to determine if vertex p can $\alpha$ vertex q. Let G be a protection graph and $\alpha \in \{r,w,c\}$. Call p and q connected if there exists a path between p and q independent of the directionality or labels of the arcs. Define the predicates:

*Condition 1:* p and q are connected in G.

*Condition 2:* there exists a vertex x in G and an arc form x to q with label $\alpha$ such that

$\alpha = r$ implies $\{r,c\} \cap \beta \neq \emptyset$, or

$\alpha = w$ implies $w \in \beta$, or

$\alpha = c$ implies $c \in \beta$.

Informally, these conditions will state that p can $\alpha$ q if and only if there is an undirected path between p and q (condition 1) and some vertex x $\alpha$'s q (condition 2).

The first step is to demonstrate the necessity of conditions (1) and (2).

*Lemma 1:* Let G be a protection graph with vertices p and q and let $\alpha$ be a label. Then p can $\alpha$ q is true implies conditions (1) and (2) hold.

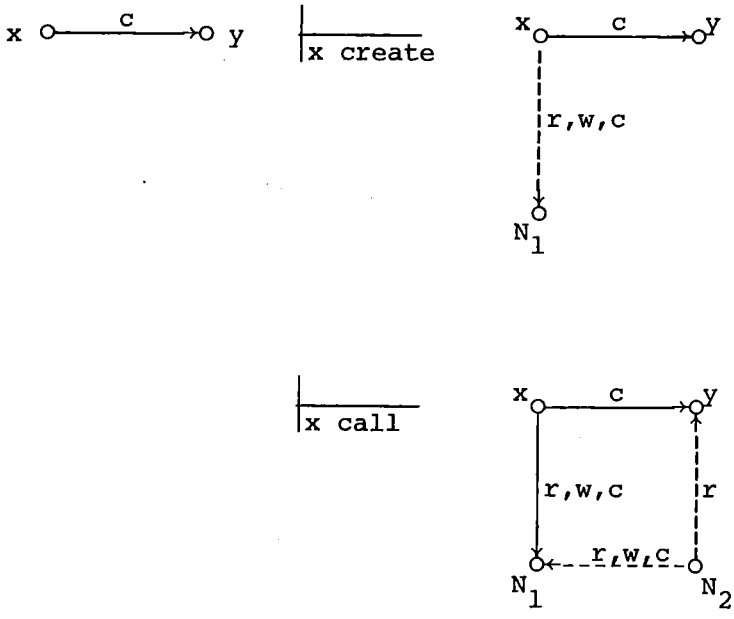
*Proof:* If p can $\alpha$ q in G then (1) and (2) are satisfied, so suppose p cannot $\alpha$ q in G and $G_1,\ldots,G_n$ is a sequence such that p can $\alpha$ q in $G_n$. If (1) is not satisfied in $G_i$ then it is not satisfied in $G_{i+1}$ since no rule

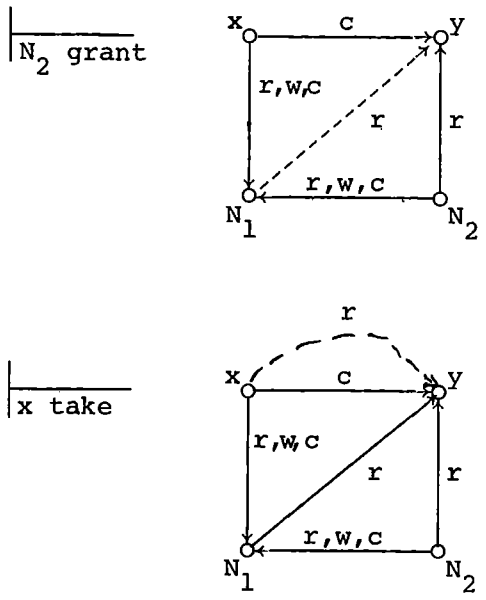
application connects vertices not already connected. If (2) is not satisfied in G, let $G_i$ be the first graph satisfying (2) and $G_{i-1} \vdash G_i$. If p is take or grant, the choice of $G_i$ is violated. Create cannot place an incoming arc to q, so p must be call. But regardless of what $\alpha$ is, p = call violates our choice of $G_i$. □

To simplify matters later and to clear up an apparent anomaly in condition (2), we next show that if a user is allowed to call another user then he is allowed to read him as well. It is this fact that allows us to write $\{r,c\} \cap \beta \neq \emptyset$ in condition (2) rather than just $r \in \beta$.

*Lemma 2:* In a protection graph G, x $\xrightarrow{c}$ y implies x $\xrightarrow{r,c}$ y.

*Proof:* Apply the following rules:



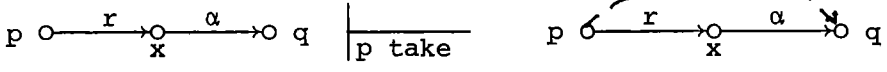



□

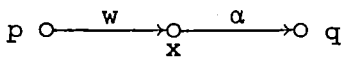
We next prove a key lemma that shows that the directionality and labels along a connected path are unimportant. Call vertices p and q of a protection graph directly connected if there is an arc between them independent of the directionality.

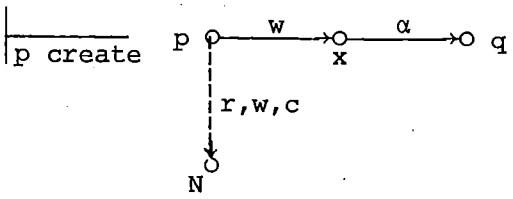
*Lemma 3:* Let p, q and x be distinct vertices in a protection graph, let there be an arc from x to q with label α and let p and x be directly connected. Then p and α q.

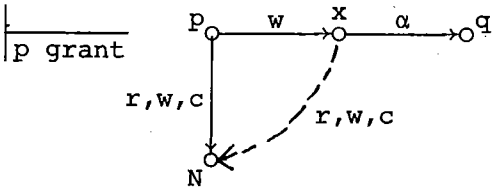
*Proof:* By monotonicity, there are only six distinct cases.

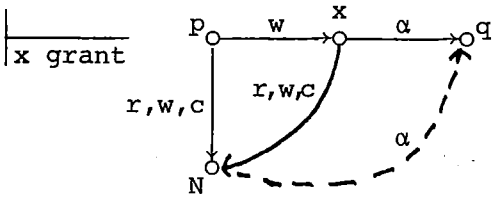
Case 1: 

Case 2:

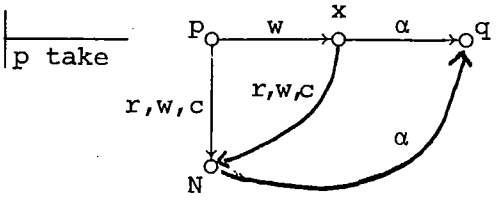


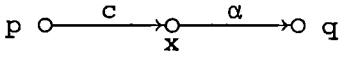



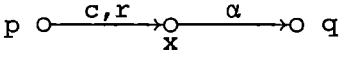

α




Case 3:

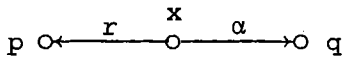


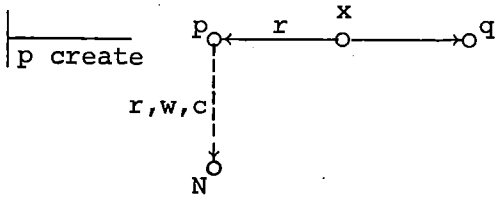

By lemma 2 this can be written as

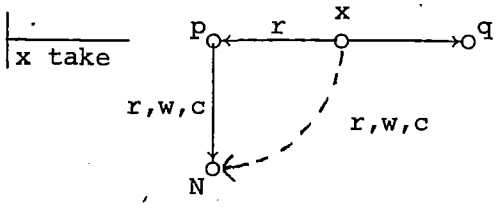


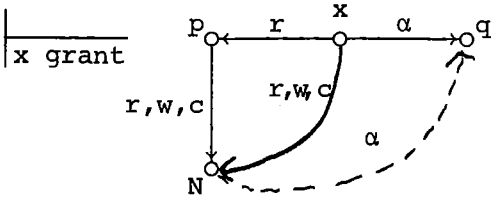

and we can appeal to case 1.

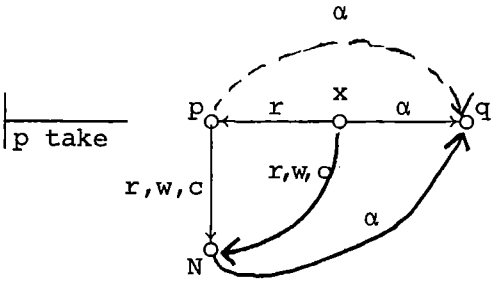
Case 4:



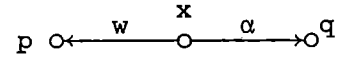



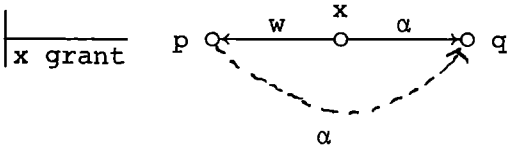



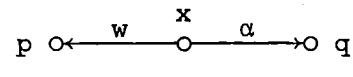




Case 5:

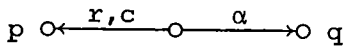



Case 6:




By lemma 2 this can be written as

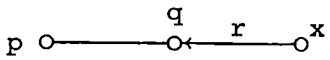


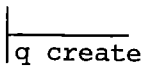

and we can apply case 4. □

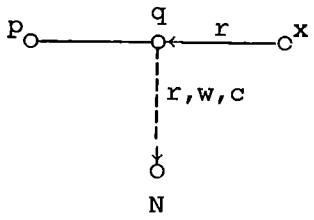
We now use lemma 3 to prove three additional lemmas to be used in the basis of our later induction.

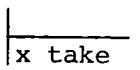
*Lemma 4:* Let p, q and x be distinct vertices in a protection graph such that p is directly connected to q and there is an arc from x to q with label $\gamma$ such that $\{r,c\} \cap \gamma \neq \emptyset$. Then p can r q.

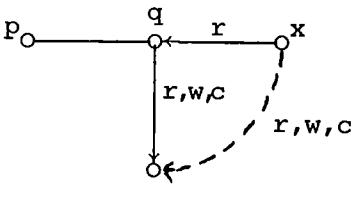
*Proof:* By lemma 2 we can assume that $\gamma = r$. Then we apply the following rules:

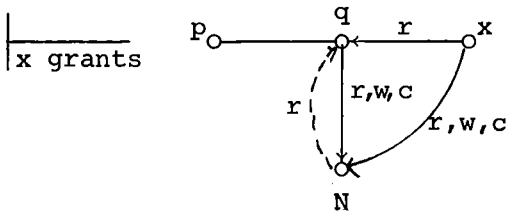


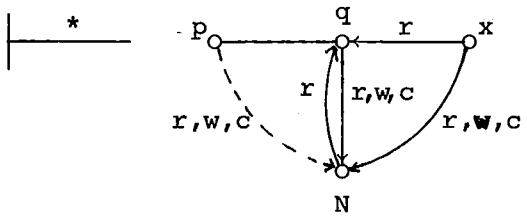



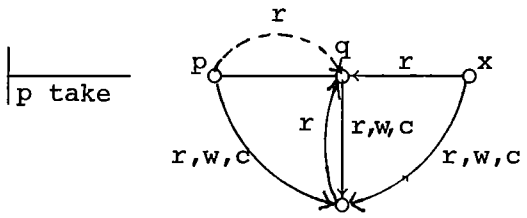






By lemma 3 we can realize

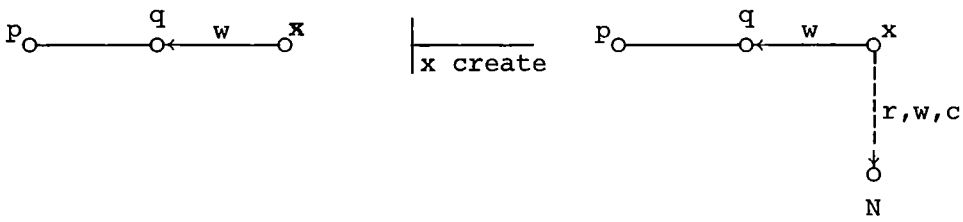


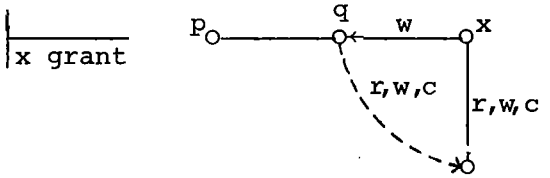



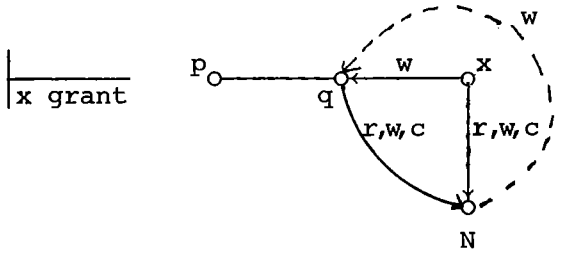

□

*Lemma 5:* Let p, q and x be distinct vertices in a protection graph such that p is directly connected to q and there is an arc from x to q with label $\gamma$ such that $w \in \gamma$. Then p can w q.

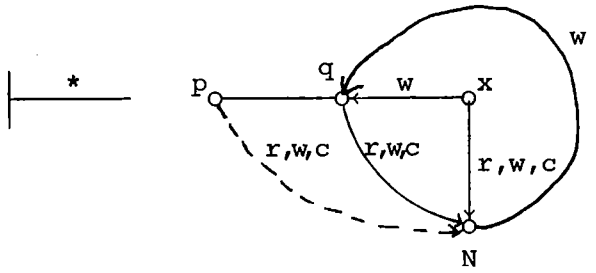
*Proof:* We apply the following rules:

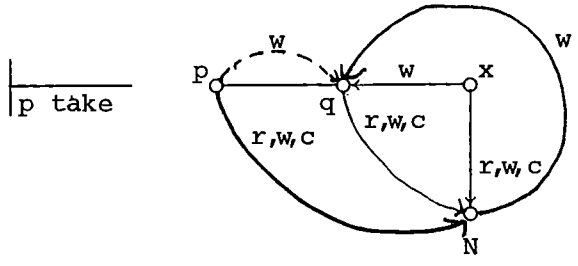





By lemma 3 we can realize

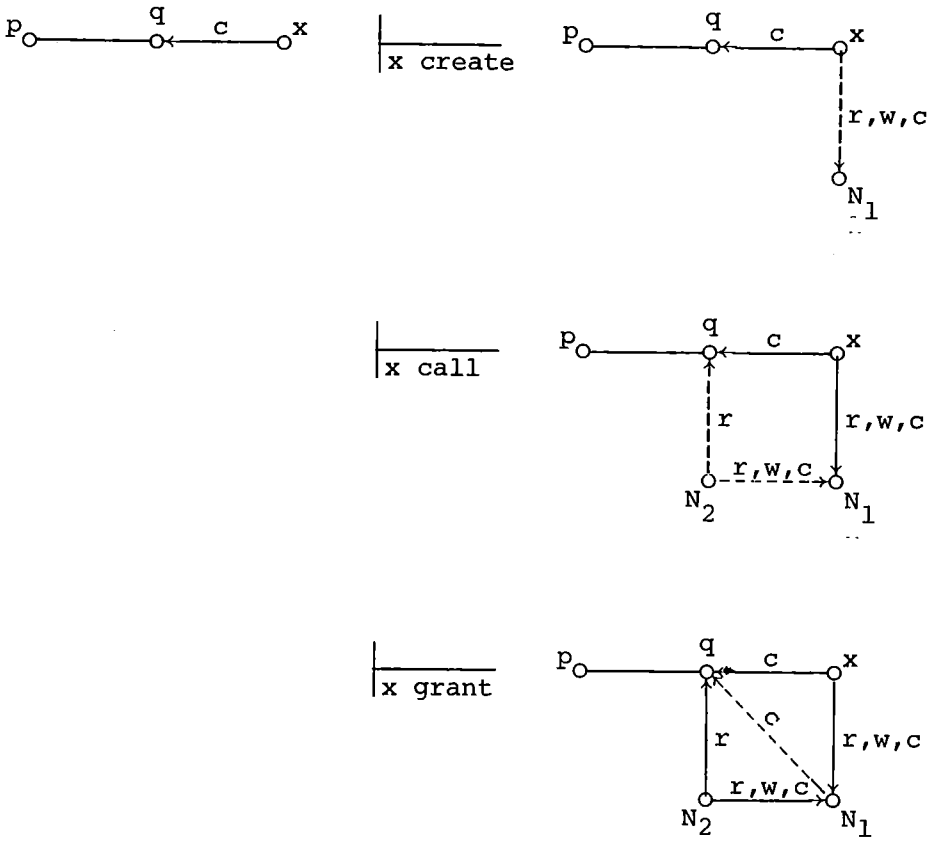


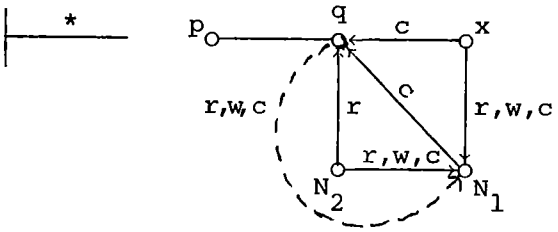

□

*Lemma 6:* Let p, q and x be distinct vertices in a protection graph such that p is directly connected to q and there is an arc from x to q with label $\gamma$ such that $c \in \gamma$. Then p can c q.

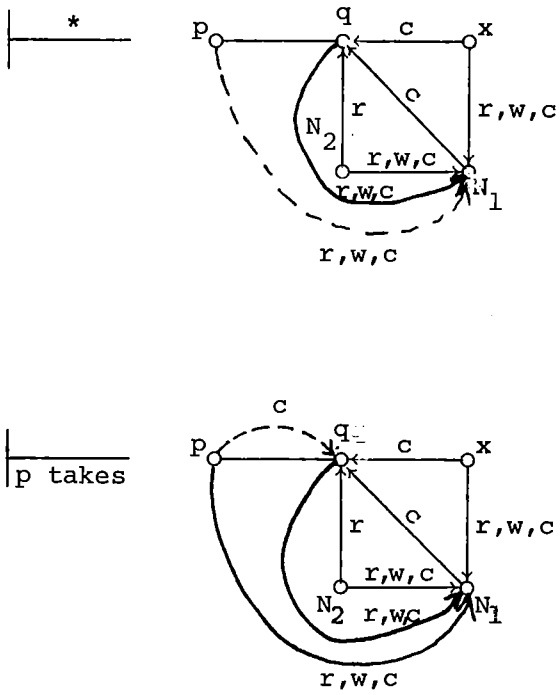
*Proof:* Apply the following rules:




By lemma 3 we can realize




By a second application of lemma 3 we get




□

*Theorem:* Let p and q be distinct vertices in a protection graph and α a label. Conditions (1) and (2) are necessary and sufficient to imply p can α q.

*Proof:* Lemma 1 demonstrates necessity so we proceed by induction to show sufficiency. Let $p = x_n, x_{n-1}, \ldots, x_1, x_0 = q$ be the vertices on a connected path.

(Basis) For n = 1, there are two possibilities. The x guaranteed by condition (2) either coincides with $x_1 = p$ in which case the sufficiency is immediately true or else x and $x_1$ are distinct. By lemmas 4, 5 and 6, p can α q.

(Induction) Suppose the theorem is true for $n \geq 1$ and $p = x_{n+1}$ and $x_{n+1}$ is

directly connected to $x_n$. By hypothesis $x_n$ can $\alpha$ q, and by lemma 3 this implies $x_{n+1}$ can $\alpha$ q. □

*Corollary 1:* There is an algorithm for deciding if p can $\alpha$ q that operates in linear time in the size of the protection graph.

*Proof:* To verify condition (1) apply Tarjan [6]. Verifying condition (2) requires no more time than scanning the in arcs to vertex q.

An obvious consequence of the constructions of this section is that it is simple to acquire the right to a given object if it can be acquired.

*Corollary 2:* If p can $\alpha$ q then there is an algorithm to add an arc from p to q with label $\alpha$ that is linear in the length of the path between p and q.

## III. Discussion

The consequence of our main theorem is that we can precisely state the protection policy for this take-grant system.

*Policy:* If p can read (write) (call) q then any user in the connected component containing p and q can attain the right to read (write) (read and call) q.

This policy may appear to be more undiscriminating than one might have expected. A primary reason for this is that our take-grant system treats all elements of the system the same where as most protection models [3] recognize two different entities: subjects and objects. If we dichotomize the vertices of our model into subject and object sets and require (as is usually the case) that only subjects can initiate the application of our rules,* then the system becomes much more difficult to analyze. Such an analysis is not yet complete, but preliminary indications are that there exists protection graphs satisfying conditions (1) and (2) for subject vertices p and q such that p can $\alpha$ q is false.

In addition to completing the subject/object analysis, there are other problems to be studied. For example, the take-grant system discussed here is only representative of existing protection schemes. Others have been proposed in the literature and they should be studied since they may provide more powerful security policies. One could also consider modifications

*The restriction that only subjects can initiate protection rules is enforced by requiring the x vertex in our rule definitions to be a subject and all other vertices may be either subjects or objects.

to our rules to make the take-grant system more discriminating among connected users. Other rights, such as ownership, could be added to the model.

References

1. E. Cohen.
Ph.D. Thesis (in progress), Carnegie-Mellon University, 1976.

2. P. J. Denning and G. S. Graham.
Protection principles and practice.
AFIPS Conference Proceedings 40:417-429, 1972.

3. M. A. Harrison, W. L. Ruzzo, and J. D. Ullman.
On protection in operating systems.
Proceedings of the 5th annual SIGOPS Conference, 1975.

4. A. K. Jones.
Protection in programmed systems.
Ph.D. Thesis, Carnegie-Mellon University, 1973.

5. R. J. Lipton and L. Snyder.
Synchronization and security.
In preparation, 1976.

6. R. E. Tarjan.
Depth first search and linear graph algorithms.
SIAM J. Computing 1:146-160, 1972.