THE SHARING OF RIGHTS AND INFORMATION IN A CAPABILITY-BASED PROTECTION SYSTEM

AMES GRANT

IN-32-CR

Matt Bishop

224140

Technical Report PCS-TR88-136

P36

The Sharing of Rights and Information in a Capability-Based Protection System

Matt Bishop

Department of Mathematics and Computer Science
Dartmouth College
Hanover, NH 03755

ABSTRACT

This paper examines the question of sharing of rights and information in the Take-Grant Protection Model by concentrating on the similarities between the two; in order to do this, we state and prove new theorems for each that specifically show the similarities. The proof for one of the original theorems is also provided. These statements of necessary and sufficient conditions are contrasted to illustrate the proposition that transferring rights and transferring information are fundamentally the same, as one would expect in a capability-based system. We then discuss directions for future research in light of these results.

1. Introduction

Capability-based protection systems control access to objects by means of a *ticket* or *capability*. This piece of information, typically an ordered pair consisting of an address and a set of rights, grants to the holder unconditional access to the object at the given address in the manner indicated by the set of rights. If the ticket is copied to another process, that process has access to the object commensurate with the associated ticket. For this reason, capabilities are normally made inaccessible to ordinary processes.

In addition to being a grant of rights, a capability is also information. When an

This work was supported by grant NAG 2-480 from the National Aeronautics and Space Administration (NASA) to Dartmouth College, and a Burke Award from Dartmouth College.

object possesses a capability, it knows the location (address) of some other object and how it may access that other object (set of rights). No action of any kind is necessary to implement the use of a capability; the possessor need not be checked against an access control list, or provide a special key; just knowing what the ticket contains means the object has those rights to that object. In essence, then, a capability is permissions encoded as information.

In this paper we explore how the transfer of rights and the transfer of information are reflected in a capability-based protection model. We shall use the Take-Grant Protection Model, introduced in [5] and expanded upon in [1,2,6-8]. because both sharing of rights and sharing of information in that model has been analyzed at some length. This model also has some interesting theoretical properties: specifically, whether or not a right (or information) can be transferred may be determined in time linearly proportional to the size of the graph even if the number of objects which can be created is unbounded, This is a direct consequence of the graph rewriting rules, which will be explained in the second and fourth sections. These rules make the model mono-operational; indeed, for any such system, there is an algorithm that decides whether that system and a given initial state is safe for a generic right. (With the more general system, of course, the question is undecidable [4].)

The next section discusses the sharing of rights; we present a framework for stating the standard theorem, and do so. We then state and prove a slightly different (albeit equivalent) theorem; this theorem parallels a result presented in a later section, after we have discussed the sharing of information in the model; in that section, another theorem presents necessary and sufficient conditions for information transfer to occur. Following this, we state and prove a version of this theorem which parallels the standard theorem for sharing rights. We conclude by looking at the idea of ''meta-theorems'' and how the dual nature of rights in a capability system demonstrates their utility.

2. Transfers of Authority

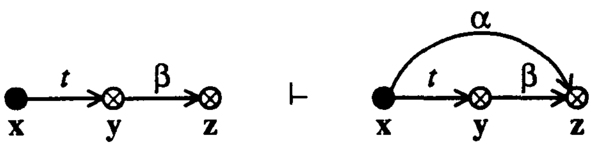
Let a finite, directed graph called a *protection graph* represent a system to be modelled. A protection graph has two distinct kinds of vertices, called *subjects* and *objects*. Subjects are the active vertices, and (for example) can represent users; they can pass information and authority by invoking *graph rewriting rules*. Objects, on the other hand, are completely passive; they can (for example) represent files, and do nothing.

In protection graphs, the subjects are represented by ● and objects by ○. Vertices which may be either subjects or objects are represented by ⊗ Pictures are very often used to show the effects of applying a graph rewriting rule on the graph; the symbol $\vdash$ is used to mean that the graph following it is produced by the action of the graph rewriting rule on the graph preceding it. The rewriting rule itself is often written after the derived graph. The symbol $\vdash^*$ represents several rule applications. The term *witness* means a sequence of graph rewriting rules which produce the predicate or condition being witnessed, and a witness is often demonstrated by listing the graph rewriting rules that make up the witness (usually with pictures.)

The edges of a protection graph are labelled with subsets of a finite set R of rights. Suppose that $\{r,w,t,g\} \subseteq R$, where r, w, t, and g represent *r*ead, *w*rite, *t*ake, and *g*rant rights, respectively. When written as labels on a graph, the set braces are normally omitted.

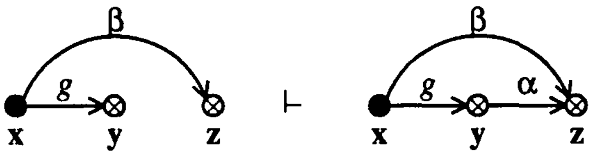
The Take–Grant Model permits users with certain rights to transfer rights from one vertex to another. The rules governing the transfer of rights are called *de jure* rules and are as follows :

take: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** be a subject. Let there be an edge from **x** to **y** labelled γ with $t \in \gamma$, an edge from **y** to **z** labelled β, and $\alpha \subseteq \beta$. Then the *take* rule defines a new graph G_1 by adding an edge to the protection graph from **x** to **z** labelled α. Graphically,



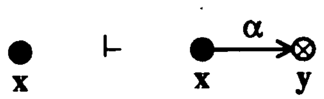
The rule is written: ‘‘**x** takes (α to **z**) from **y**.’’

grant: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** be a subject. Let there be an edge from **x** to **y** labelled γ with $g \in \gamma$, an edge from **x** to **z** labelled β, and $\alpha \subseteq \beta$. Then the *grant* rule defines a new graph G_1 by adding an edge to the protection graph from **y** to **z** labelled α. Graphically,



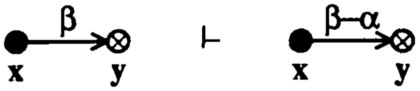
The rule is written: ‘‘**x** grants (α to **z**) to **y**.’’

create: Let **x** be any subject in a protection graph G_0 and let α be a subset of R. *Create* defines a new graph G_1 by adding a new vertex **y** to the graph and an edge from **x** to **y** labelled α. Graphically,



The rule is written: "**x** creates (α to new vertex) **y**."

remove: Let **x** and **y** be any distinct vertices in a protection graph G_1 such that **x** is a subject. Let there be an explicit edge from **x** to **y** labelled β, and let α be any subset of R. Then *remove* defines a new graph G_1 by deleting the α labels from β. If β becomes empty as a result, the edge itself is deleted. Graphically,

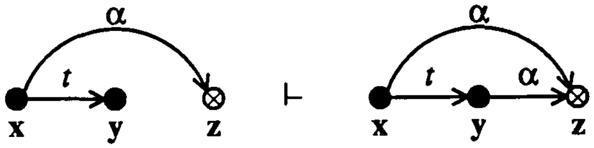


The rule is written: "**x** removes (α to) **y**."

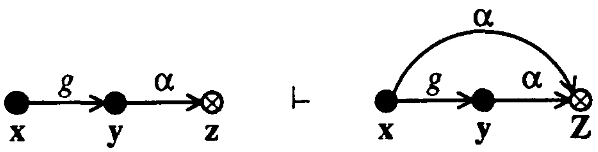
The edges which appear in the above graphs are called *explicit* because they represent authority known to the protection system.

Note that there is a duality between the take and grant rules when the edge labelled *t* or *g* is between two subjects. Specifically, with the cooperation of both subjects, rights can be transmitted backwards along the edges. The following two lemmas (from [5]) demonstrate this:

Lemma 1:



Lemma 2:



As a result, when considering the transfer of authority between subjects, neither direction nor label of the edge is important, so long as the label is in the set $\{t,g\}$.

The first question that comes to mind is under what conditions can rights be shared? To answer this question, we first need to examine some characteristics of take-grant graphs.

Definition: A *tg–path* is a nonempty sequence $v_0, \ldots, v_k$ of distinct vertices such that for all i, $0 \le i < k$, v_i is connected to v_{i+1} by an edge (in either direction) with a label containing t or g.

Note that the vertices in a tg–path may be either subjects or objects.

Definition: Vertices are *tg–connected* if there is a tg–path between them.

Definition: An *island* is a maximal tg–connected subject–only subgraph.

Any right that one vertex in an island has can be obtained by any other vertex in that island. In other words, an island is a maximal set of subject–only vertices which possess common rights.

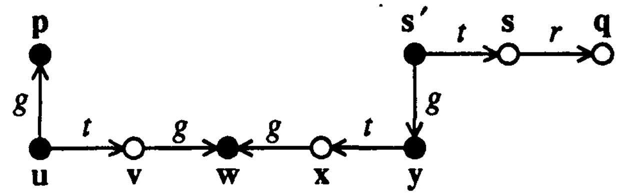
With each tg–path, associate one or more words over the alphabet $\{ \vec{t}, \overleftarrow{t}, \vec{g}, \overleftarrow{g} \}$ in the obvious way. If the path has length 0, then the associated word is the null word ν.

Definition: A vertex v_0 *initially spans* to v_k if v_0 is a subject and there is a tg–path between v_0 and v_k with associated word in $\{ \vec{t}^{*} \vec{g} \} \cup \{\nu\}$.

Definition: A vertex v_0 *terminally spans* to v_k if v_0 is a subject and there is a tg–path between v_0 and v_k with associated word in $\{ \vec{t}^{*} \}$.

Definition: A *bridge* is a tg–path with v_0 and v_k both subjects and the path's associated word in { $\overrightarrow{t}{}^{*}$, $\overleftarrow{t}{}^{*}$, $\overrightarrow{t}{}^{*}\overset{\times}{\overline{g t}}{}^{*}$, $\overrightarrow{t}{}^{*}\overleftrightarrow{g t}{}^{*}$ }.

An initial span is a tg–path along which the first vertex in the path can transmit authority; a terminal span is a tg–path along which the first vertex in the path can acquire authority. A bridge is an edge along which a right can be passed, possibly by using lemma 1 and 2 as well as the *de jure* rules. As a note, a bridge is said to be *directed away from* v_0. The following diagram illustrates these terms:



islands:	I_1={p ,u },I_2={w },I_3={y ,s′}
bridges:	u ,v ,w and w ,x ,y
initial span:	p with associated word: v
terminal span:	s′,s with associated word: $\overrightarrow{t}$

The following predicate formally defines the notion of *transferring authority*:

Definition: The predicate *can•share(*α, x, y, G_0) is true for a right α and two vertices x and y if and only if there exist protection graphs $G_1, \ldots, G_n$ such that $G_0 \vdash^* G_n$ using only *de jure* rules, and in G_n there is an edge from x to y labelled α.

In short, if x can acquire α rights to y, then *can•share*(α, x, y, G_0) is true. The theorem which establishes necessary and sufficient conditions for this predicate to hold is [5]:

Theorem 3: The predicate *can•share*(α, x, y, G_0) is true if and only if there is an edge from x to y in G_0 labelled α, or if the following hold simultaneously:

(i) there is a vertex s ∈ G_0 with an s-to-y edge labelled α;

(ii) there exists a subject vertex $\mathbf{p}'$ such that $\mathbf{p}'=\mathbf{p}$ or $\mathbf{p}'$ initially spans to $\mathbf{x}$;

(iii) there exists a subject vertex $\mathbf{s}'$ such that $\mathbf{s}'=\mathbf{s}$ or $\mathbf{s}'$ terminally spans to $\mathbf{s}$; and

(iv) there exist islands $I_1, \ldots, I_v$ such that $\mathbf{p}'$ is in I_1, $\mathbf{s}'$ is in I_v, and there is a bridge from I_j to I_{j+1} $(1 \le j < v)$.

This statement of the necessary and sufficient conditions can be much simpler, and that is the topic of the next section.

3. Alternate Necessary and Sufficient Conditions for *can•share*

For reasons that will become clear in the next section, we can state conditions necessary and sufficient for *can•share* to be true as follows:

Theorem 4: The predicate *can•share*(α, $\mathbf{x}$, $\mathbf{y}$, G_0) is true if and only if there is a vertex $\mathbf{s}$ with an edge to $\mathbf{y}$ labelled α, and a sequence of subjects $\mathbf{u}_1, \ldots, \mathbf{u}_m$ such that all of the following conditions hold simultaneously:

(i) $\mathbf{u}_1 = \mathbf{x}$ or $\mathbf{u}_1$ initially spans to $\mathbf{x}$;

(ii) $\mathbf{u}_m = \mathbf{s}$ or $\mathbf{u}_m$ terminally spans to $\mathbf{s}$; and

(iii) for all i $(1 \le i < m)$, there is a *tg*-path between $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ with an associated word in B.

Proof: ($\Leftarrow$) We prove this by induction on m.

BASIS: $m = 1$. First, note that if $\mathbf{u}_1 = \mathbf{s}$, $\mathbf{u}_1$ has α rights to $\mathbf{y}$; but if if $\mathbf{u}_1 \ne \mathbf{s}$, then by applying the *take* rule (repeatedly if necessary), $\mathbf{u}_1$ can acquire α rights to $\mathbf{y}$. If $\mathbf{u}_1 = \mathbf{x}$, we are done. If not, by using the take rule (possibly repeatedly), $\mathbf{u}_1$ can acquire *grant* rights to $\mathbf{x}$, and can then use the grant rule to give $\mathbf{x}$ α rights over $\mathbf{y}$. Since only the take and grant (*de jure*) rules were used, *can•share*(α, **x**, **y**, G_0) is true by definition.

INDUCTION HYPOTHESIS: For all $m \leq k$, if the three conditions in the theorem are true, then *can•share*(α, **x**, **y**, G_n) is true.

INDUCTION STEP: Let $m = k+1$. By the induction hypothesis, clearly *can•share*(α, $\mathbf{u}_2$, **y**, G_n) is true. It suffices to consider whether $\mathbf{u}_1$ can get α rights to **y**. For if $\mathbf{u}_1 = \mathbf{x}$, we are done, and if not, then $\mathbf{u}_1$ initially spans to **x**; so $\mathbf{u}_1$ may use a (possibly null) series of take rules to acquire *grant* rights over **x**, and then use the grant rule to pass its α rights to **x**.

Consider now whether or not $\mathbf{u}_1$ may acquire α rights to **y**. Note that $\mathbf{u}_1$ and $\mathbf{u}_2$ are distinct, else $m = k$, not $k+1$. By condition (iii), the word associated with the *tg*-path between $\mathbf{u}_1$ and $\mathbf{u}_2$ is in B. So let us consider those one at a time. In these cases, by the take tule and the definition of *bridge*, it is necessary to consider only those cases where bridges are of length 2 or less.

Case 1. The associated word is $\vec{t}$.

In this case $\mathbf{u}_1$ simply applies the take rule to acquire α rights to **y**.

Case 2. The associated word is $\vec{g}$.

The result is true by lemma 2; take $\mathbf{x} = \mathbf{u}_1$, $\mathbf{y} = \mathbf{u}_2$, and $\mathbf{z} = \mathbf{y}$ in that lemma.

Case 3. The associated word is $\vec{t}$.

The result is true by lemma 1; take $\mathbf{x} = \mathbf{u}_2$, $\mathbf{y} = \mathbf{u}_1$, and $\mathbf{z} = \mathbf{y}$ in that lemma.

Case 4. The associated word is $\overleftarrow{g}$.

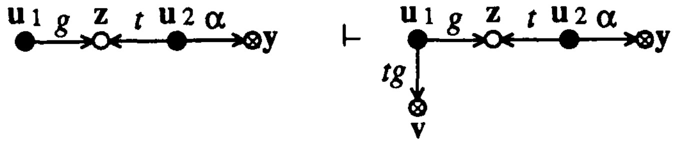
In this case $\mathbf{u}_2$ simply applies the grant rule to give α rights to **y**.

Case 5. The associated word is $\overset{*}{\overline{tg}}$.

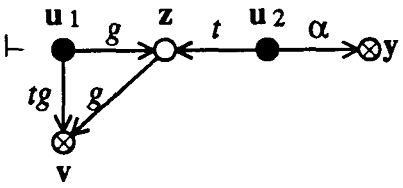
Let **z** be the intermediate vertex; note that it must be an object. Then **u**2 simply applies the grant rule to give α rights to **z**, and **u**1 simply uses the take rule to acquire them.

Case 6. The associated word is $\overset{*}{\overline{gt}}$.

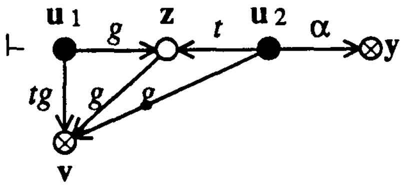
The following construction suffices. First, **u**1 creates (*tg* to new) **v**:



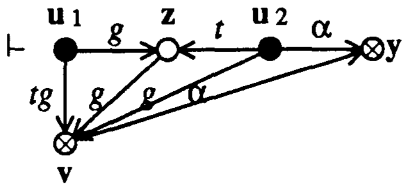
Then, **u**1 grants (*g* to **v**) to **z**:



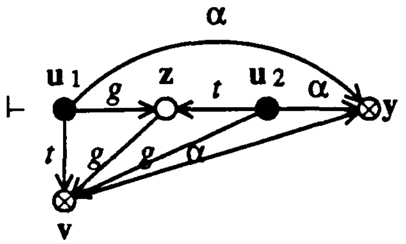
Next, **u**2 takes (*g* to **v**) from **z**:



After that, **u**2 grants (α to **y**) to **v**:



Finally, **u**2 grants (α to **y**) to **v**:



In all cases, u_1 can acquire α rights to y using only the *de jure* rules; so by the definition of *can•share*, *can•share*(α, x, y, G_n) is true. This proves the induction step and hence the claim.

(⇒) Assume now *can•share*(α, x, y, G_n) is true. By inspection of the *de jure* rules, none adds an incoming edge with right α to a vertex y unless there is already an edge with that right and with y as target; hence, there must exist a vertex s with an α edge to y in G_0.

Now consider the *de jure* rule applications $\rho_1, \ldots, \rho_n$ needed to produce a witness to the predicate in G_n. Without loss of generality, we may eliminate all instances of the remove rule, since that rule never adds edges, and without loss of generality we may reorder the rule applications so all create rules come at the beginning. We induct on the number n to prove the claim.

BASIS: If $n=1$, then either the take or grant rule was used, since the create rule would not add a new edge to an existing vertex. The initiator must have been a subject by the nature of the rules.

If the take rule was used, then, x initiated it and took from s α rights to y; in this case, if s is not a subject, choose $m=1$ and $u_1=x$ in the theorem. Conditions (i) and (ii) clearly hold, and condition (iii) trivially holds. On the other hand, if s is a subject, choose $m=2$, $u_1=x$, and $u_2=y$. Then all three conditions hold.

Now suppose the grant rule was used. Then **s** initiated it and granted to **x** α rights to **y**. In this case, if **x** is not a subject, choose $m=1$ and $\mathbf{u}_1=\mathbf{s}$ in the theorem. Conditions (i) and (ii) clearly hold, and condition (iii) trivially holds. On the other hand, if **x** is a subject, choose $m=2$, $\mathbf{u}_1=\mathbf{x}$, and $\mathbf{u}_2=\mathbf{y}$. Then all three conditions hold.

In either case, *can•share*(α, x, y, G_0)'s being true implies that all three conditions hold, as claimed.

INDUCTION HYPOTHESIS: For $n=1, \ldots, k$, if *can•share*(α, x, y, G_0) is true, then all three conditions hold.

INDUCTION STEP: Let $n=k+1$ and consider the rule ρ_n. If it is a create rule, then there is an edge from **x** to **y** labelled α in G_{n-1}, since the create rule cannot add a new incoming edge to an existing vertex; in this case the induction hypothesis assures us that the three conditions hold.

The proof that the claim holds for the take and the grant rules are very similar, so we present the one for take here. Suppose the rule ρ_{k+1} is a take rule. For this rule to be applied to G_k, **x** must be a subject, there must be an **x**-to-**z** edge labelled *t*, and a **z**-to-**y** edge labelled α for some vertex **z** in G_k. So *can•share*(*t*, x, z, G_k) and *can•share*(α, z, y, G_k) both hold.

If **z** exists in G_0, it is clear that the predicates *can•share*(*t*, x, z, G_0) and *can•share*(α, z, y, G_0) both hold. Condition (i) follows immediately from the first of these and the induction hypothesis, and condition (ii) from the second and the induction hypothesis. So, consider the first of those two predicates. By the induction hypothesis, there is a subject vertex $\mathbf{z}'$ such that either $\mathbf{z}'=\mathbf{z}$ or $\mathbf{z}'$ terminally spans to

$\mathbf{z}$ (condition (ii).) By the second, there is a subject vertex $\mathbf{z}''$ such that either $\mathbf{z}''=\mathbf{z}$ or $\mathbf{z}''$ initially spans to $\mathbf{z}$ (condition (i).) Note that by the definition of *bridge*, the path between $\mathbf{z}'$ to $\mathbf{z}''$ is a bridge unless $\mathbf{z}'=\mathbf{z}''$. In either case, the union of the sets of $\mathbf{u}_i$s of the two predicates provides the $\mathbf{u}_i$s for *can•share*(α, $\mathbf{x}$, $\mathbf{z}$, G_0) between each is a tg-path with associated word in B. This proves condition (iii).

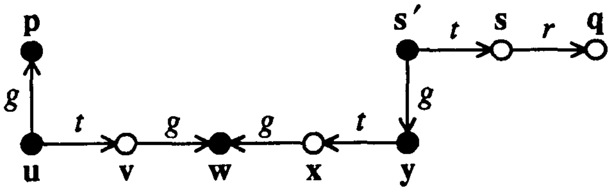
If, on the other hand, $\mathbf{z}$ does not exist in G_0 (and therefore was created by one of the create rules), consider the graph G_c that exists after all create rules have been applied but before any other *de jure* rules have been used. The vertex $\mathbf{z}$ must lie in a subgraph connected to the original graph G_0 by one edge. Call the vertex lying on this edge and existing in G_0, $\mathbf{z}_0$. Thus, there is a path between $\mathbf{z}_0$ and $\mathbf{z}$; moreover, by the nature of the create rule, this path is directed out of $\mathbf{z}_0$ and towards $\mathbf{z}$. Let $\mathbf{z}_0, \ldots, \mathbf{z}_m = \mathbf{z}$ be the vertices on the path. All but the last of them must be subjects (because each invokes the create rule to create the next one.) Further, the edge between $\mathbf{z}_{m-1}$ and $\mathbf{z}$ must contain a t in its label, since no *de jure* rule adds an incoming edge with a right unless there is already an incoming edge with that right; so, if the edge in question did not have a t in its label, *can•share*(t, $\mathbf{x}$, $\mathbf{z}$, G_c) would be false. Also, since rights may only be transferred using the take and grant rules, the label of each edge in the path from $\mathbf{z}_0$ to $\mathbf{z}$ must contain either a t or a g. By the induction hypothesis, then, *can•share*(t, $\mathbf{z}_0$, $\mathbf{z}$, G_c) is true. This means that either *can•share*(g, $\mathbf{z}_0$, $\mathbf{x}$, G_c) is true or *can•share*(t, $\mathbf{x}$, $\mathbf{z}_0$, G_c) is true.

To show that *can•share*(α, $\mathbf{z}_0$, $\mathbf{y}$, G_0) is true consider a witness to *can•share*(α, $\mathbf{z}$, $\mathbf{y}$, G_0). Append to that witness a take rule of the form "$\mathbf{z}_0$ takes (α to $\mathbf{y}$) from $\mathbf{z}$." These rules produce a series of protection graphs generated by a sequence of *de jure* rule applications, and in the final one there is an edge labelled α from $\mathbf{z}_0$ to $\mathbf{y}$. But by definition, this means *can•share*(α, $\mathbf{z}_0$, $\mathbf{y}$, G_0) is true.

Now, if *can•share*(t, $\mathbf{x}$, $\mathbf{z}_0$, G_c) is true, as $\mathbf{z}_0$ exists in G_0 and *can•share*(α, $\mathbf{z}_0$, $\mathbf{y}$, G_0) is true, we have already shown that all three conditions hold. A similar argument when *can•share*(g, $\mathbf{z}_0$, $\mathbf{x}$, G_c) is true also shows that all three conditions hold. This proves the induction hypothesis for $n = k + 1$.

Since the induction step is successful, the claim is proven. □

Return now to the picture discussed in the earlier section:



In this picture, we can easily see that *can•share*(α, $\mathbf{p}$, $\mathbf{q}$, G_0) is true; take $\mathbf{u}_1$=$\mathbf{p}$, $\mathbf{u}_2$=$\mathbf{u}$, $\mathbf{u}_3$=$\mathbf{w}$, $\mathbf{u}_4$=$\mathbf{y}$, and $\mathbf{u}_5$=$\mathbf{s}'$. Condition (i) holds as $\mathbf{u}_1$=$\mathbf{p}$; condition (ii) holds as $\mathbf{u}_5$ terminally spans to s; and condition (iii) holds as all words associated with the paths betweenm the subjects are bridges.

It should be clear why this theorem holds. Since an island is a set of subjects connected by edges labelled *take* and *grant*, the vertices of the island involved in the sharing of the right are all connected by elements of B. Between each island is a bridge, and of course the associated words are in B. So the theorem in this section merely focuses attention on these vertices and ignores the others in the island. Similarly, the conditions in theorem 3 simply merge all subsequences of the $\mathbf{u}_i$s that have no objects as part of the tg-path between them into an island. This reasoning can be made more formal to show:

Corollary 5: The conditions in theorem 3 hold if and only if the conditions in theorem 4 hold.

Of course, this theorem also preserves the following property:

Corollary 6: Truth or falsity of the predicate *can•share* may be determined in linear time in the size of the initial graph.

Let us now consider the transfer of information, and re-examine the proof of the predicate governing that type of transfer. or connection.

4. Transfers of Information

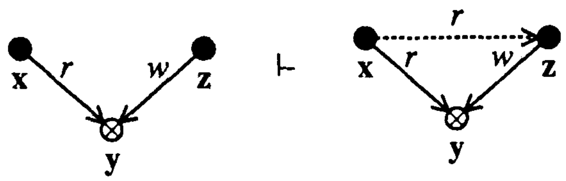
The *de jure* rules control the transfer of authority only; they say nothing about the transfer of information. The two are clearly different; for example, if a user is shown a document containing information which he does not have authority to read, the information has been transfered to the user. The *de jure* rules do not model cases like this. Instead, we use a different set of rules, called *de facto* rules, to derive paths along which information may flow.

In order to describe transfers of information, we cannot use explicit edges, because no change in authority occurs. Still, some indication of the paths along which information can be passed is necessary. Hence, we use a dashed line, labelled by r, to represent the path of a potential *de facto* transfer. Such an edge is called an *implicit* edge. Notice that implicit edges cannot be manipulated by the *de jure* rules, since the *de jure* rules can affect only authorities recorded in the protection system, and implicit edges do not represent such authority.

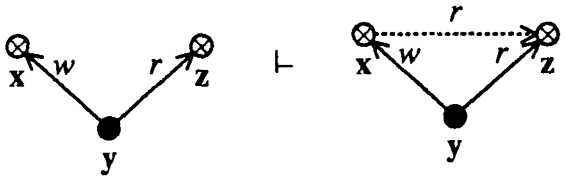
A protection graph records all authorities as explicit edges, so when a *de jure* rule is used to add a new edge, an actual transfer of authority has taken place. But when a *de facto* rule is used, a path along which information can be transferred is exhibited; the actual transfer may, or may not, have occurred. It is impossible to tell this from the graph, because the graph records authorities and *not* information. For the purposes of this model, however, we shall assume that if it is possible for information to be transferred from one vertex to another, such a transfer has in fact occurred.

One set of proposed *de facto* rules was introduced in [1] to model the transfer of information. Although these are not the only rules possible, their effects have been explored, and so we shall use them.

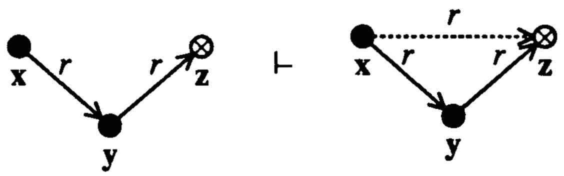
post: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0 and let **x** and **z** be subjects. Let there be an edge from **x** to **y** labelled α, where $r \in \alpha$, and an edge from **z** to **y** labelled β, where $w \in \beta$. Then the *post* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled $\{r\}$. Graphically,



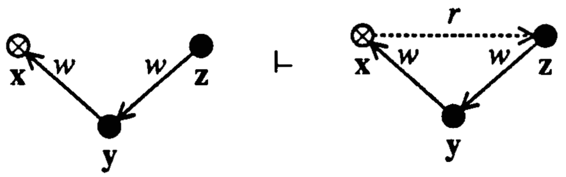
pass: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **y** be a subject. Let there be an edge from **y** to **x** labelled α, where $w \in \alpha$, and an edge from **y** to **z** labelled β, where $r \in \beta$. Then the *pass* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled $\{r\}$. Graphically,



spy: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **x** and **y** be subjects. Let there be an edge from **x** to **y** labelled α, where $r \in \alpha$, and an edge from **y** to **z** labelled β, where $r \in \beta$. Then the *spy* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled $\{r\}$. Graphically,



find: Let **x**, **y**, and **z** be three distinct vertices in a protection graph G_0, and let **y** and **z** be subjects. Let there be an edge from **y** to **x** labelled α, where $w \in \alpha$, and an edge from **z** to **y** labelled β, where $w \in \beta$. Then the *find* rule defines a new graph G_1 with an implicit edge from **x** to **z** labelled $\{r\}$. Graphically,



Note that these rules add implicit and not explicit edges. Further, as these rules model information flow, they *can* be used when either (or both) of the edges between **x** and **y**, or **y** and **z**, are implicit.

4.1. Information Flow in a Graph with Static Rights

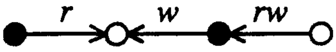
Now, consider the conditions necessary for a potential *de facto* transfer to exist in a graph.

Definition: The predicate *can•know•f(*x, y, G_0) is true if and only if there exists a sequence of graphs $G_1, \ldots, G_n$ $(0 \le n)$, such that $G_i \vdash^* G_{i+1}$ $(0 \le i < n)$ by one of the *de facto* rules and in G_n either a **x**-to-**y** edge labelled r exists or a **y**-to-**x** edge labelled *w* exists and if the edge is explicit, its source is a subject.

Intuitively, *can•know•f*(**x**, **y**, G_0) is true if and only if **x** has the authority to read **y**, **y** has the authority to write to **x**, or an implicit edge from **x** to **y** can be added by means of the *de facto* rules. Note the duality of read and write. If **x** can write to **y**, then **y** effectively can read **x**. All **x** has to do is write to **y** any information that **y** wants to see. This duality will play an important role in later results.

Definition: An *rw–path* is a nonempty sequence $\mathbf{v}_0, \ldots, \mathbf{v}_k$ of distinct vertices such that for all i, $0 \le i < k$, $\mathbf{v}_i$ is connected to $\mathbf{v}_{i+1}$ by an edge (in either direction) with a label containing an *r* or a *w*.

With each rw–path, associate one or more words over the alphabet { $\vec{r}$, $\overleftarrow{r}$, $\vec{w}$, $\overleftarrow{w}$ } in the obvious way; for instance, the protection graph



has associated $\vec{r}\overleftarrow{w}\overleftarrow{r}$ and $\vec{r}\overleftarrow{w}\overleftarrow{w}$. If the path has length 0, then the associated word is the null word ν.

Definition: An rw–path $\mathbf{v}_0, \ldots, \mathbf{v}_k$, $k \ge 1$, is an *admissible rw–path* if and only if it has an associated word $a_1 a_2 \cdots a_k$ in the regular language $(\vec{r} \cup \overleftarrow{w})^*$, and if $a_i = \vec{r}$ then $\mathbf{v}_{i-1}$ is a subject and if $a_i = \overleftarrow{w}$ then $\mathbf{v}_i$ is a subject.

Note that there cannot be two consecutive objects on an rw–admissible path. Given these definitions,

Theorem 7: Let **x** and **y** be vertices in a protection graph G_0. Then *can•know•f*(**x**, **y**, G_0) is true if and only if there is an admissible rw–path between **x** and **y**.

4.2. Information Flow in a Graph with Changing Rights

These results can be extended to include both *de jure* and *de facto* rules. To do so, we must define terms combining characteristics of those used in both the *de jure* and *de facto* developments.

Definition: The predicate *can•know*(**x**, **y**, G_0) is true if and only if there is a sequence of protection graphs $G_1, \ldots, G_n$ such that $G_0 \vdash^* G_n$ and in G_n either a **x**-to-**y** edge labelled r exists, or a **y**-to-**x** edge labelled w exists and, if the edge is explicit, its source is a subject.

This is merely *can•know•f*(**x**, **y**, G_0) without the restriction on the types of rules used.

Definition: An *rwtg–path* is a nonempty sequence $\mathbf{v}_0, \ldots, \mathbf{v}_k$ of distinct vertices such that for all i, $0 \le i < k$, $\mathbf{v}_i$ is connected to $\mathbf{v}_{i+1}$ by an edge (in either direction) with a label containing a t, g, r, or a w.

With each rwtg–path, associate one or more words over the alphabet $\{ \overrightarrow{t}, \overleftarrow{t}, \overrightarrow{g}, \overleftarrow{g}, \overrightarrow{r}, \overleftarrow{r}, \overrightarrow{w}, \overleftarrow{w} \}$ in the obvious way.

Definition: The vertex v_0 *rw–initially spans* to $\mathbf{v}_k$ if $\mathbf{v}_0$ is a subject and there is an rwtg–path between $\mathbf{v}_0$ and $\mathbf{v}_k$ with associated word in $\{ \overrightarrow{t}^* \overrightarrow{w} \}$.

Definition: A vertex $\mathbf{v}_0$ *rw–terminally spans* to $\mathbf{v}_k$ if $\mathbf{v}_0$ is a subject and there is an rwtg–path between $\mathbf{v}_0$ and $\mathbf{v}_k$ with associated word in $\{ \overrightarrow{t}^* \overrightarrow{r} \}$.

Definition: A *bridge* is an rwtg–path with associated word in the regular language

$$B = \{ \overrightarrow{t}^* \cup \overleftarrow{t}^* \cup \overrightarrow{t}^* \overset{\times}{\overline{g\,t}}{}^* \cup \overrightarrow{t}^* \overleftrightarrow{g\,t}{}^* \}$$

(Note that this is the same as the definition given earlier in this section.) A *connection* is an rwtg–path with associated word in the regular language

$$C = \{ \overrightarrow{t}^{*}\overrightarrow{r} \cup \overleftarrow{w}\overleftarrow{t}^{*} \cup \overrightarrow{t}^{*}\overrightarrow{r}\overleftarrow{w}\overleftarrow{t}^{*} \}$$

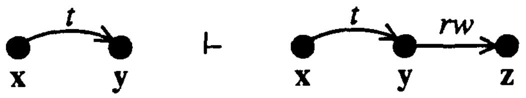
We can characterize the set of graphs for which *can•know* is true:

In order to appreciate these results, let us now look at some examples of the uses of the rules; these will be useful in deriving our later results. These two results are quite basic, and state that there is a bridge or connection between two subjects, either can (with the co-operation of the other) read information from the other. More formally:

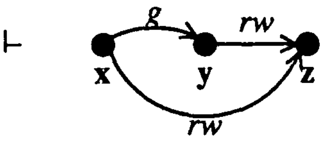
Lemma 8: If there is a bridge from **x** to **y**, then **x** can obtain an implicit read edge to **y**.

Proof: By the take rule and the definition of *bridge*, it suffices to prove the lemma for bridges of length 2 or less. Six cases arise.

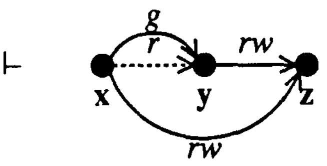
Case 1: First, **y** creates (*rw* to new vertex) **z**:



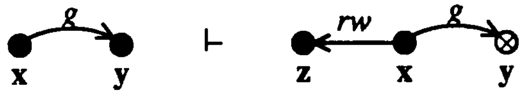
Then, **x** takes (*rw* to **z**) from **y**:



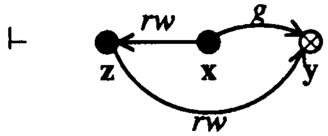
Finally, **x** and **y** use the post rule:



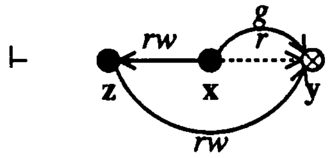
Case 2: First, **x** creates (*rw* to new vertex) **z**:



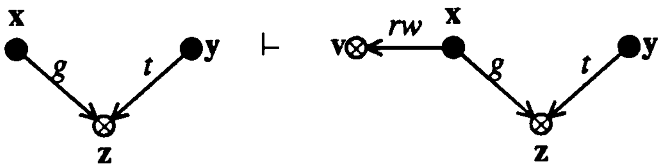
Then, **x** grants (*rw* to **z**) to **y**:



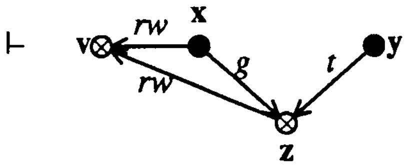
Finally, **x** and **y** use the post rule:



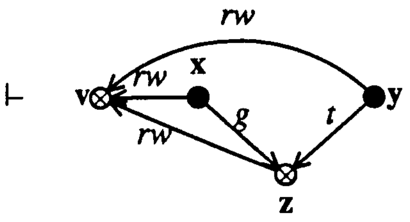
Case 3: First, **x** creates (*rw* to new vertex) **v**:



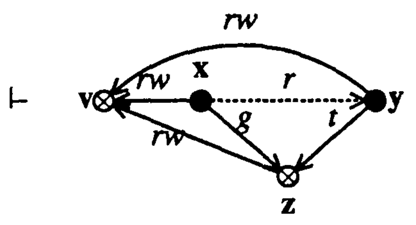
Next, **x** grants (*rw* to **v**) to **z**:



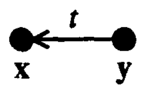
Then **y** takes (*rw* to **v**) from **z**:



Then **y** takes (*rw* to **v**) from **z**:

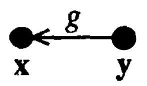


Case 4:



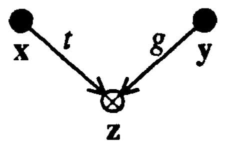
see case 1

Case 5:



see case 2

Case 6:



see case 3

In all cases, **x** gets implicit read rights over **y**, proving the claim. □

Lemma 9: Let **x** and **y** be subjects with a bridge or connection between them. Then at least one of the following is true:

(i) an explicit read edge from **x** to **y** exists or may be added;

(ii) an implicit read edge from **x** to **y** may be added; or

(iii) an explicit write edge from **y** to **x** exists or may be added.

Proof: If there is a bridge between **x** and **y**, case (i) holds by lemma 8. So, suppose there is a connection from **x** to **y**. If the associated word is in $\vec{t}^*\vec{r}$, by using the take rule, **x** can obtain an explicit read edge to **y**, establishing case (i). If the associated word is in $\overleftarrow{w}\overleftarrow{t}^{*}$, by using the take rule, **y** can obtain an explicit write edge to **x**, establishing case (iii). If the associated word is in $\vec{t}^{*}\vec{r}\,\overleftarrow{w}\overleftarrow{t}^{*}$, **x** and **y** can each apply the take rule until **x** obtains a read edge to a vertex to which **y** has a write edge (or vice versa); then **x** and **y** can use the post rule to add an implicit read edge from **x** to **y** (case (ii).)

□

5. Statement and Proof of *can•know*

The following theorem was presented in [1]; the proof has been expanded upon here.

Theorem 10: The predicate *can•know*(**x**, **y**, G_0) is true if and only if there is a sequence of subjects $\mathbf{u}_1, \ldots, \mathbf{u}_m$ such that all of the following conditions hold simultaneously:

(i) $\mathbf{u}_1 = \mathbf{x}$ or $\mathbf{u}_1$ rw-initially spans to **x**;

(ii) $\mathbf{u}_m = \mathbf{y}$ or $\mathbf{u}_m$ rw-terminally spans to **y**; and

(iii) for all i $(1 \le i < m)$, there is an *rwtg*-path between $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ with an associated word in $B \cup C$.

Proof: (⇐) Assume the three conditions hold. By lemma 9, if **p** and **q** are subjects joined by an *rwtg*-path with associated word in $B \cup C$, then one can add an implicit read edge between them. Once this is done, there is an *rw*-admissible path from **x** to **y**; by theorem 7, this is a necessary and sufficient condition for *can•know*(**x**, **y**, G_0) to be true.

(⇒) Assume *can•know*(**x**, **y**, G_n) is true. If a witness can be found by applying *de jure* rules only, then *can•share*(r, **x**, **y**, G_0) is true. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the sequence of subjects in theorem 4 instantiating that predicate, and let **s** be the vertex with an r edge to **q**. Taking $m=n+1$, $\mathbf{u}_i=\mathbf{v}_i$ for $1 \le i \le n$, and $\mathbf{u}_{n+1}=\mathbf{s}$ (if **s** is a subject) or $m=n$ and $\mathbf{u}_i=\mathbf{v}_i$ for $1 \le i \le n$ (if **s** is an object), conditions (i), (ii), and (iii) follow immediately.

So suppose at least one *de facto* rule application had to be used. Since *de jure* rules do not manipulate implicit edges, we may without loss of generality reorder the rule applications so that all *de jure* rules precede all *de facto* rules. It follows by theorem 7 that there exists an rw-admissible path in the graph resulting from the application of the *de jure* rules but before any *de facto* rules are applied. Let $\mathbf{x}=\mathbf{v}_0, \ldots, \mathbf{v}_l=\mathbf{y}$ be the vertices on that path. We now show by induction on the length l of the path that the desired conditions hold.

BASIS: Let $l=1$. Then by the definition of an rw-admissible path, either **x** is a subject and rw-terminally spans to **y**, or **y** is a subject and rw-initially spans to **x**. In the first case, take $n=1$ and $\mathbf{x}=\mathbf{u}_1$ in the claim to see that conditions (i) and (ii) are true and condition (iii) is trivially true; in the other case, take $n=1$ and $\mathbf{y}=\mathbf{u}_1$ in the claim to see once again that conditions (i) and (ii) are true and condition (iii) is trivially true.

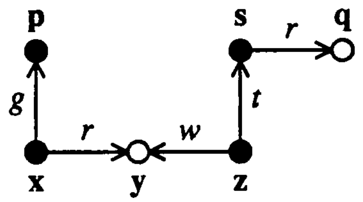
INDUCTION HYPOTHESIS: For $l=1, \ldots, k$, if *can•know*(**x**, **y**, G_0), then the three conditions hold.

INDUCTION STEP: Let $l=k+1$ and consider which *de facto* rule was applied to produce the witness. What follows assumes that the rule application ρ_l wis a *spy* rule; the proofs for the other rules are similar.

If **z** exists in G_0, that the spy rule was used means there is an (implicit or explicit) edge from **x** to **z**, and an (implicit or explicit) edge from **z** to **y**. If the edge from **x** to **z** is explicit, theorem 4 gives us condition (i) (take $\mathbf{u}_1=\mathbf{x}$ in that theorem.) If it is implicit, since **z** and **y** are distinct (by definition of the spy rule), we may apply the induction hypothesis to obtain condition (i). Similarly, regardless of whether the edge is implicit or explicit, either theorem 4 or the induction hypothesis gives us condition (ii). Finally, condition (iii) follows from either the induction hypothesis or theorem 4 (remember, condition (iii) requires the edge labels to be in a subset of what condition (iii) requires) or both.

Now suppose **z** does not exist in G_0. Then by the nature of the create rule, **z** must be in a subgraph connected to the vertices in G_0 by a single edge; moreover, there is a path $\mathbf{z}_0, \ldots, \mathbf{z}_m=\mathbf{z}$ with edges from $\mathbf{z}_i$ to $\mathbf{z}_{i+1}$, and for $0 \le i < m$, $\mathbf{z}_i$ is a subject. If the edge from **x** to **z** is explicit, this means that by theorem 4, as $\mathbf{z}_0$ lies on all possible paths from **x** to **z**, one of *can•share*(g, $\mathbf{z}_0$, **x**, G_0) and *can•share*(t, **x**, $\mathbf{z}_0$, G_0) is true. In either case, as **x** and $\mathbf{z}_0$ are subjects, by lemma 8, *can•know*(**x**, $\mathbf{z}_0$, G_0) is true. On the other hand, if the edge from **x** to **z** is implicit, since **z** and **y** are distinct (by definition of the spy rule), we may apply the induction hypothesis to show *can•know*(**x**, $\mathbf{z}_0$, G_0) is true. The three conditions follow immediately. □

An example will perhaps clarify what the theorem says. Consider the graph:



Take $\mathbf{u}_1=\mathbf{p}$, $\mathbf{u}_2=\mathbf{x}$, $\mathbf{u}_3=\mathbf{z}$, and $\mathbf{u}_4=\mathbf{s}$. **s** indeed *rw*-terminally spans to **q**, and between each pair of $\mathbf{u}_i$s there is a bridge (between **x** and **q**, and between **s** and **z**) or connection (between **x** and **z**). In this case, therefore, *can•know*(**p**, **q**, G_0) is true. In fact, the following witnesses the predicate:

(1) **z** takes (r to **q**) from **s**;

(2) **x** grants (r to **y**) to **p**;

(3) **p** and **z** use the post rule to add an implicit edge labelled r from **p** to **z**;

(4) **p** and **z** use the spy rule to add an implicit edge labelled r from **p** to **q**.

However, note that *can•share(r*, **p**, **q**, G_0) is false because there is no bridge between **x** and **z**.

6. An Alternate Statement of *can•know*

Earlier, we gave a statement of the necessary and sufficient conditions for *can•share* to be true that is quite similar to the conditions for *can•know* to be true. In this section, we prove a symmetric result, namely that the conditions for *can•know* to be true may be stated in a form quite similar to the original statement of the theorem for *can•know*. The theorem relies upon lemma 9, which states a set of conditions that must be met before *can•know* is true. Hence, the theorem given next really only tries to establish when there will be a series of bridges or connections from a vertex **q** to another vertex **p** along which information can be sent.

Theorem 11: Let **p** and **q** be vertices in a protection graph G_0. Then *can•know(***p**, **q**, G_0) is true if, and only if, *can•share(r*, **p**, **q**, G_0) is true or all of the following conditions hold:

(i) There is a subject **p**′ such that **p**′ = **p** or **p**′ is *rw*-initially connected to *p*;

(ii) There is a subject **q**′ such that **q**′ = **q** or **q**′ is *rw*-terminally connected to *q*;

(iii) There is a sequence of islands $\{I_j \mid 1 \le j \le m\}$ such that there is a bridge or connection from I_j to I_{j+1}, $1 \le j \le m$, and $\mathbf{p}' \in I_1$, and $\mathbf{q}' \in I_m$.

Informal argument: To prove the "if" part, note that if *can•share*(r, **p**, **q**, G_0) is true then by definition *can•know* is true. If not, condition (i) says that **p**′ can send any information it gets to **p**, and part (ii) says that **q**′ can obtain any information it needs from **q**. Condition (iii) simply says that **q**′ can send the information to a vertex $\mathbf{z}_1$ in I_{m-1}, which can in turn send it to a vertex $\mathbf{z}_2$ in I_{m-2}, and so on, until a vertex $\mathbf{z}_{m-1}$ in I_1 gets the information. By the properties of an island, this means that **p**′ can get the information. Putting all this together, *can•know* is true.

Going the other way involves considering the rule applications needed to produce a witness. We can require all *de facto* rules to be applied last, and examine the conditions needed for them to be applied. For example, in the pass rule, there is a vertex **y** for which *can•share*(w, **y**, **p**, G_n) and *can•share*(r, **y**, **p**, G_n) are true. From the conditions required for both *can•share* rules to hold, the three condition of the theorem must be true. (In the formal proof, we will show this for the post and spy rules; the formal proof for the pass and find rules are left as an exercise for the reader.) And if no *de facto* rules are used to generate a witness to *can•know*, obviously *can•share*(r, **p**, **q**, G_0) is true.

Proof: ($\Leftarrow$) Consider condition (i). As **p**′ is *rw*-initially connected to **p**, it either has or can obtain (through the take rule) a write edge to **p**. Similarly, by condition (i), if **q**′ is *rw*-terminally connected to **q**, it either has or can obtain (using the take rule) a read edge to **q**. Thus, it suffices to show *can•know*(**p**′, **q**′, G_0) is true by condition (iii); merely apply the spy rule (if $\mathbf{q}' \ne \mathbf{q}$), and then the post rule (if $\mathbf{p}' \ne \mathbf{p}$), to obtain *can•know*(**p**, **q**, G_0).

We show condition (iii) implies *can•know*(**p**′, **q**′, G_0) by inducting on m, the number of islands in that condition.

BASIS: Let $m = 1$. Then **p**′ and **q**′ are in the same island, whence by cases 1, 2, 4, and 5 of lemma 8, *can•know*(**p**′, **q**′, G_0) is true.

INDUCTION HYPOTHESIS: For $m = 1, \ldots, k$, if condition (iii) holds, *can•know*(**p**′, **q**′, G_0) also holds.

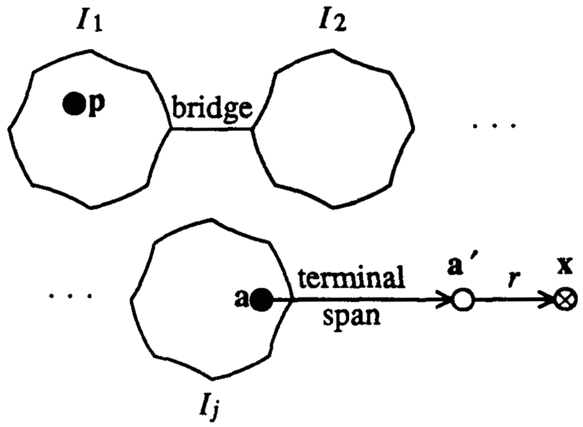
INDUCTION STEP: Let $m = k + 1$. Let $\mathbf{z}_k$ be the subject in I_k that bounds the bridge or connection between I_k and I_{k+1}; let $\mathbf{z}_{z+1}$ be $\mathbf{z}_k$'s counterpart in I_{k+1}. By lemma 9, *can•know*($\mathbf{z}_{k+1}$, **q**′, G_0) holds; by lemma 8, this means *can•know*($\mathbf{z}_k$, **q**′, G_0) holds; and by the induction hypothesis, *can•know*(**p**′, $\mathbf{z}_k$, G_0) holds; whence by the spy rule, *can•know*(**p**′, **q**′, G_0) is true. This proves the induction hypothesis, and hence the "if" part of the theorem.

(⇒) Now assume *can•know*(**p**, **q**, G_0) is true, and consider a minimal set of rule applications ρ_i needed to produce a witness. Without loss of generality, we may reorder the ρ_i's so that all *de jure* rule applications precede any *de facto* rule applications, since *de facto* rule applications do not change the state of the protection graph. If no *de facto* rules are applied, the witness will end with an explicit read edge from **p** to **q**, in which case *can•share(r*, **p**, **q**, G_0) holds; the three conditions then follow from theorem 3. So suppose that at least one *de facto* rule application is needed.

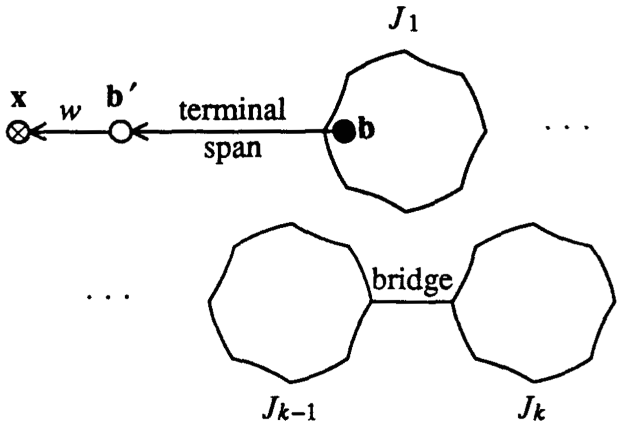
Induct on the number m of such *de facto* rule applications.

BASIS: Let $m = 1$. Each of the *de facto* rules must be considered. We will give the proof for the post rule; the other rules are treated similarly.

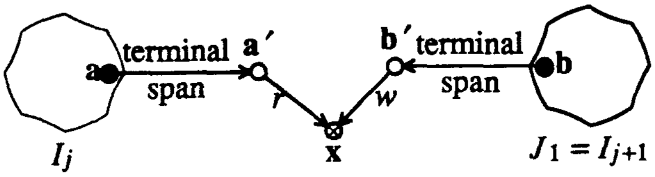
Consider the post rule. In order to apply this rule, there must be a vertex **x** such that *can•share*(r, **p**, **x**, G_0) and *can•share*(w, **q**, **y**, G_0) are true. By theorem 3, this means that there is a sequence of islands $I_1, \ldots, I_j$ with $\mathbf{p} \in I_1$, and a vertex $\mathbf{a} \in I_j$ which terminally spans to another vertex $\mathbf{a}'$, which has a read edge to **x**:



Similarly, there is a sequence of islands $J_1, \ldots, J_k$, with $\mathbf{q} \in J_k$, and a vertex $\mathbf{b} \in J_1$ which terminally spans to another vertex $\mathbf{b}'$, which has a write edge to **x**:



Now, combining these two facts, relabel the islands $J_1, \ldots, J_k$ as $I_{j+1}, \ldots, I_{j+k}$. Note that, as a terminal span has associated word in $\overrightarrow{t}^*$, we have a connection $\overrightarrow{t}^*\overrightarrow{r}\overleftarrow{w}\overleftarrow{t}^*$ from **a** to **b**:

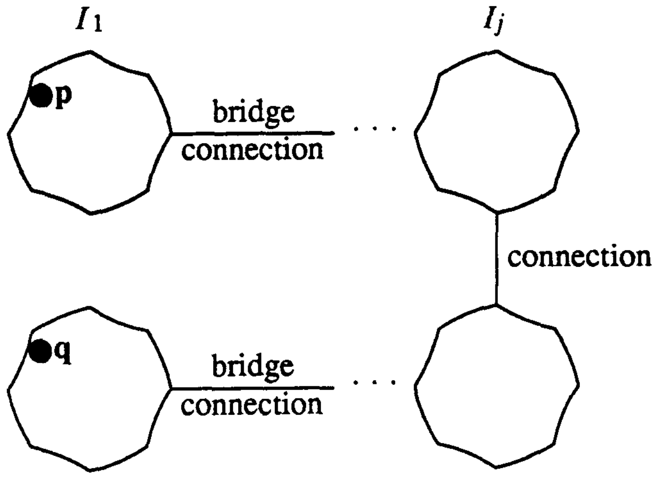


This is condition (iii). Taking $\mathbf{p}' = \mathbf{p}$ and $\mathbf{q}' = \mathbf{q}$, conditions (i) and (ii) hold.

INDUCTION HYPOTHESIS: Let $m = 1, \ldots, k$. Then after n *de facto* rule applications, if *can•know*(**p**, **q**, G_0) is true, conditions (i), (ii), and (iii) hold.

INDUCTION STEP: Let $n = k + 1$, and assume the k+1st rule applied is a spy rule (proofs for the other three rules are similar.)

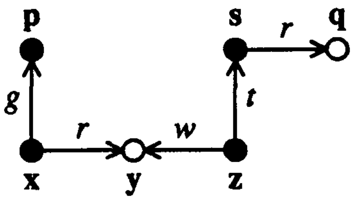
As the spy rule is used, **p** is a subject, and there is a subject vertex **x** such that *can•know*(**p**, **x**, G_0) and *can•know*(**x**, **q**, G_0) are true. By the induction hypothesis, the first *can•know* ensures that there is a subject $\mathbf{p}'$ such that $\mathbf{p}' = \mathbf{p}$ or $\mathbf{p}'$ *rw*-initially spans to **p**, giving condition (i); the second *can•know* ensures that there is a subject $\mathbf{q}'$ such that $\mathbf{q}' = \mathbf{q}$ or $\mathbf{q}'$ *rw*-terminally spans to **q**, giving condition (ii). By the induction hypothesis, condition (iii) is assumed to hold for both *can•know*(**p**, **x**, G_0) and *can•know*(**x**, **q**, G_0). So, let $I_1, \ldots, I_j$ and $J_1, \ldots, J_k$ be the sets of islands for *can•know*(**p**, **x**, G_0) and *can•know*(**x**, **q**, G_0), respectively. Thus, the configuration is:



Again, relabel $J_1, \ldots, J_k$ to be $I_{j+1}, \ldots, I_{j+k}$; also, recall that an *rw*-terminal span between subjects is a connection. This establishes condition (iii), proving the induction hypothesis and the claim.

Hence, theorem 11 has been proven. □

Let us return to the example of the previous section.



Take $\mathbf{p}'=\mathbf{p}$, $\mathbf{q}'=\mathbf{s}$, $I_1=\{\ \mathbf{p},\mathbf{x}\ \}$, and $I_2=\{\ \mathbf{s},\mathbf{a}\ \}$. As **s** indeed *rw*-terminally spans to **q**, and there is a connection between I_1 and I_2, *can•know*(**p**, **q**, G_0) is true, agreeing with our previous result.

It should be clear that this theorem and the one presented in the previous section are equivalent. As cases 1, 2, 4, and 5 of lemma 8 show, if there is an edge between two subjects labelled *t* or *g* an implicit edge labelled *r* can be added between those vertices. Hence in addition to being a subgraph in which rights may be freely transferred, an island is also a subgraph in which information may be freely transferred assuming all subjects co-operate. (If not, other conditions apply; see [2].) Hence, the islands connected by bridges and connections ensures that provides a sequence of subjects connected by bridges and connections along which information can flow, as condition (v) requires. Hence the conditions in theorem 11 being true implies the conditions in theorem 10 are true. Similarly, the sequence of subjects in theorem 10 may be grouped into islands (if need be, the islands may contain only one vertex) to obtain the conditions for theorem 11. This reasoning can be made more formal to show:

Corollary 12: The conditions in theorem 10 hold if and only if the conditions in theorem 11 hold.

Of course, this theorem also preserves the following property:

Corollary 13: Truth or falsity of the predicate *can•know* may be determined in linear time in the size of the initial graph. or connection.

7. Discussion and Future Directions

One of the more strinking feature of these theorems is the similarity betweem theorems 3 and 11, and between theorems 4 and 10. Let us look at the first two in some detail.

Take the capability represented by the **s**-to-**x** edge in theorem 3 to be a piece of information. Then the only difference between the two theorems lies in the nature of the paths along which the information is transmitted; if the information is a capability, the class of paths is much smaller than if it were ordinary information. This makes sense, because one of the characteristics of capabilities is that ''... no ordinary program can manufacture or modify the bit pattern with which a capability is represented'' [3]. Capabilities are very special pieces of information, and cannot be handed around like the contents of a file. So, in the Take-Grant Protection Model, the *de jure* rules manipulate the flow of capability information, and the *de facto* rules implement the flow of all other types of information.

This suggests two things. In [6], the notion of classifying subjects into *groups* was explored in the context of ''groups of users.'' One could extend the classification to quantities to be controlled, and provide several different sets of rules, each set describing the flow of that quantity throughout the graph. One would then have to discuss the interactions of the different rules and the quantities as the rules were applied. Interestingly enough, only the flow of capabilities would change the explicit edges in the graph, since the graph is designed to abstract only the capabilities; in some instances, a new graph might have to be defined. (This has been done with the *de facto* rules and the implicit edges; when one deals with implicit edges, one uses a ''pseudo-graph'' with dashed arrows representing edges that ''really aren't there.'')

Continuing on in our speculations, since the characterization of the transfer of capabilities and information is so similar, the idea of a ''meta-theorem'' that captures the generalized notion of ''sharing'' immediately suggests itself. This theorem, and a corresponding one for the predicates describing the theft of information and rights, would be broad enough to incorporate the extensions to the model that deal with information flow of all kinds, classes of information and subjects, and so forth. Such theorems would be a step forward in making Take-Grant type protection models practical and useful.

References

1. Bishop, M. and Snyder, L., ''The Transfer of Information and Authority in a Protection System'', *Proceedings of the Seventh Symposium on Operating System Principles*, 45-54 (December 1979).
2. Bishop,, M., *Practical Take-Grant Systems: Do They Exist?*, Ph. D. Thesis, Purdue University, May 1984.
3. Fabry, R., ''Capability-Based Addressing'', *Communications of the ACM, 17*, 7 (July 1974) 403-412.
4. Harrison, M., Ruzzo, W. and Ullman, J., ''Protection in Operating Systems'', *Communications of the ACM, 19*, 8 (August 1976) 461-471.
5. Jones, A., Lipton, R. and Snyder, L., ''A Linear Time Algorithm for Deciding Security'', *Proceedings of the Seventeenth Annual Symposium on Foundations of Computer Science*, (1976).

6. Lipton, R. and Snyder, L., ''A Linear Time Algorithm for Deciding Subject Security'', *Journal of the ACM, 24*, 3 (July 1977) 455-464.
7. Snyder, L., ''Theft and Conspiracy in the Take-Grant Protection Model'', *Journal of Computer and System Sciences, 23*, 3 (December 1981) 333-347.
8. Snyder, L., ''Formal Models of Capability-Based Protection Systems'', *IEEE Transactions on Computers, C-30*, 3 (March 1981) 172-181.