Tạp chí Tin học và Điều khiển học, T.22, S.2 (2006), 164–179

SỰ PHÂN LY TRÁCH NHIỆM TRONG MÔ HÌNH KIỂM SOÁT TRUY NHẬP DỰA TRÊN VAI VỚI RÀNG BUỘC THỜI GIAN

LÊ THANH[1], NGUYỄN VĂN NGỌC[2], NGUYỄN THÚC HẢI[3]

[1] *Trường Đại học Sư phạm Thể dục Thể thao Hà Tây*
[2] *Cục B12, Tổng cục 5, Bộ Công an*
[3] *Khoa Công nghệ thông tin, Trường Đại học Bách khoa Hà Nội*

Abstract. The role-based access control models are interested by many researchers analysing and modeling theoretically as well as designing the security infrastructure for an organization' s resource management system. Generalized Temporal Role Based Access Control model (GTRBAC) that captures an comprehensive set of temporal constraints need for access control has recently been proposed. Its language structures allow one to specify various temporal constraints on role, user-role assignments and permission-role assignments. Here, we present the separation of duty constraints (SoD) of the temporal constraint role-based access control model. Associating the control flow dependency constraints with such ones allows specification of dynamically changing access control requirements that are typical in today's large systems. In addition to allowing specification of time, the constraints introduced here also allow expressing access control policies at a finer granularity. We also present the relationships beetwen these separation of duty constraints and demonstrate its correctness. Thereby it allows to construct a minimum set of constraints in practical implementation.

Tóm tắt. Các mô hình kiểm soát truy nhập dựa trên vai đang là mối quan tâm của nhiều nhà nghiên cứu trong việc phân tích và lập mô hình về mặt lý thuyết cũng như trong việc thiết kế cơ sở hạ tầng an ninh, an toàn cho hệ thống quản lý tài nguyên của một tổ chức. Mô hình kiểm soát truy nhập dựa trên vai theo thời gian tổng quát (GTRBAC) với một tập toàn diện các ràng buộc thời gian cần cho kiểm soát truy nhập đã được đề xuất mới đây. Các cấu trúc ngôn ngữ của mô hình này cho phép người ta đặc tả các ràng buộc thời gian trên các vai, trong việc gán người dùng cho vai và gán giấy phép cho vai. Ở đây chúng tôi trình bày các ràng buộc phân ly trách nhiệm của mô hình kiểm soát truy nhập dựa trên vai với ràng buộc thời gian. Các ràng buộc loại này cùng với các ràng buộc phụ thuộc kiểm soát luồng cho phép đặc tả các yêu cầu kiểm soát truy nhập thay đổi động thường thấy trong các hệ thống lớn ngày nay. Ngoài việc cho phép đặc tả thời gian, các ràng buộc được đưa ra ở đây cũng cho phép biểu diễn các chính sách kiểm soát truy nhập ở mức mịn hơn. Chúng tôi cũng trình bày các mối quan hệ tương đương giữa các ràng buộc phân ly trách nhiệm này và chứng minh tính đúng đắn của chúng. Điều này cho phép xây dựng một tập tối thiểu các ràng buộc trong cài đặt thực tế.

1. MỞ ĐẦU

Kiểm soát truy nhập dựa trên vai (Role-based access control - RBAC) đã nổi lên như một lựa chọn đầy hứa hẹn thay thế các mô hình kiểm soát truy nhập tùy ý và kiểm soát truy nhập

bắt buộc truyền thống [6, 7], nhưng chúng có một số hạn chế về đặc tính kế thừa. Một số đặc tính có lợi như chính sách trung tính, trợ giúp đặc quyền ít nhất, quản lý kiểm soát truy nhập hiệu quả được kết hợp với các mô hình RBAC [7]. Một trong những mặt quan trọng của kiểm soát truy nhập đó là ràng buộc thời gian các kiểm soát truy nhập để hạn chế việc sử dụng tài nguyên. Đề cập về các yêu cầu kiểm soát truy nhập dựa trên thời gian, Bertino và cộng sự [2] đề xuất một mô hình RBAC theo thời gian (Temporal RBAC - TRBAC), mà mới đây đã được Joshi và cộng sự [3] tổng quát hoá. Các ràng buộc số lượng và các ràng buộc phân ly trách nhiệm (separation of duty - SoD) đóng vai trò quyết định trong việc đảm bảo an toàn cho một số ứng dụng trong môi trường thương mại. Một số nhà nghiên cứu đã làm nổi bật tầm quan trọng và việc sử dụng các ràng buộc số lượng, các ràng buộc SoD trong các mô hình RBAC. Tuy nhiên có rất ít người, trong đó đặc biệt có Joshi và cộng sự đề cập đến các ràng buộc số lượng và các ràng buộc SoD dựa trên thời gian. Việc sử dụng một ràng buộc cụ thể cho một chu kỳ thời gian hoặc một độ dài thời gian là quan trọng đối với các ứng dụng thịnh hành hiện nay vì là các yêu cầu truy nhập thường xuyên thay đổi theo thời gian. Trong bài báo này, chúng tôi tập trung vào các ràng buộc SoD trong khung làm của mô hình GTRBAC [3]. Chúng tôi sử dụng khung làm việc tổng quát do Joshi và cộng sự đưa vào trong [5] cho phép biểu diễn một miền rộng lớn các ràng buộc số lượng dựa trên thời gian với sự trợ giúp của các vị từ trạng thái GTRBAC, một hàm liệt kê miền trị các vị từ này và một toán tử chiếu dùng để trừu xuất một tập hợp các phần tử từ miền trị được liệt kê. Chúng tôi đưa ra một tập các ràng buộc SoD có thể có khi sử dụng các vị từ trạng thái GTRBAC. Các SoD này đem lại khả năng mô hình hoá mịn hơn nhiều. Bài báo được tổ chức như sau. Mục 2 nêu vắn tắt các ràng buộc của GTRBAC, các vị từ trạng thái đối với một hệ thống GTRBAC và các ràng buộc số lượng. Mục 3 trình bày các ràng buộc SoD hạn chế thời gian, các mối quan hệ tương đương giữa một số ràng buộc SoD này và chứng minh tính đúng đắn của chúng. Mục 4 trình bày một số kết luận.

2. KIỂM SOÁT TRUY NHẬP DỰA TRÊN VAI VỚI RÀNG BUỘC THỜI GIAN

2.1. Mô hình GTRBAC

Mô hình GTRBAC cung cấp một khung làm việc thời gian để đặc tả một tập hợp mở rộng các ràng buộc thời gian [3]. Mô hình này là một sự mở rộng của mô hình TRBAC [2] và sử dụng một khung làm việc dựa trên ngôn ngữ. GTRBAC cho phép nhiều loại ràng buộc thời gian khác nhau như là các ràng buộc thời gian trong việc tạo khả năng cho vai / làm mất khả năng của vai, các ràng buộc thời gian trong việc gán người dùng cho vai và trong việc gán giấy phép cho vai, các ràng buộc thời gian kích hoạt vai, v.v.. Các sự kiện run-time quản trị của GTRBAC cho phép nhà quản trị khởi tạo động các sự kiện. Một tập các sự kiện run-time khác cho phép người dùng tạo ra các yêu cầu kích hoạt tới hệ thống. Hơn nữa, các biểu thức tạo khả năng cho ràng buộc bao gồm các sự kiện mà tạo khả năng hoặc làm mất khả năng các ràng buộc độ dài thời gian và các ràng buộc kích hoạt vai. Các *trigger* GTRBAC cho phép biểu diễn sự phụ thuộc giữa những sự kiện của GTRBAC và có được các sự kiện quá khứ. GTRBAC có thể có được các nhu cầu kiểm soát truy nhập thay đổi động của một hệ thống [3, 4]. Các biểu thức chu kỳ được viết là (I, P), trong đó I là một

khoảng thời gian và P là một tập vô hạn các khoảng con của I. (I, P) biểu diễn tập tất cả các khoảng P được chứa trong I. Chẳng hạn $(I, P) = ([1/1/2005, 12/31/2005], Mondays)$ xét tất cả các ngày *Thứ Hai* của năm 2005. Các ràng buộc thời gian được biểu diễn theo dạng tổng quát (I, P, E) với (I, P) là biểu thức chu kỳ hoặc theo dạng một ràng buộc độ dài thời gian $c = ([I, P|D], D_x, E)$, trong đó D_x đặc tả độ dài thời gian mà sự kiện E là đúng và tùy chọn D hoặc (I, P) đặc tả độ dài/khoảng thời gian mà ràng buộc độ dài thời gian c là đúng. Các biểu thức chu kỳ (I, P) dùng trong biểu thức ràng buộc dựa trên các biểu thức chu kỳ ở [2, 3]. Xét các thời điểm bắt đầu và kết thúc được biểu thị tương ứng bằng begin và end. Tập các khoảng thời gian được biểu thị bằng ([begin, end], P) được xác định thông qua việc sử dụng hàm $Sol()$ được định nghĩa một cách hình thức qua hàm $\Pi(P)$ và việc biểu diễn hình thức của P được đề cập chi tiết trong [3].

Định nghĩa 2.1. Cho t là một thời điểm, P là một biểu thức chu kỳ, begin và end là hai biểu thức ngày tháng. Xác định $t \in Sol([\text{begin, end}], P)$ nếu và chỉ nếu tồn tại $\tau \in \Pi(P)$ sao cho $t \in \tau, t_b \leqslant t \leqslant t_e$, trong đó t_b và t_e là các thời điểm được biểu thị tương ứng bằng begin và end.

2.2. Các vị từ trạng thái

Trong [4], Joshi và cộng sự đã đưa ra một số vị từ trạng thái mà chúng tôi đã xét đến trong [8] và có bổ sung một số vị từ trạng thái mới, được sử dụng trong Mục 3 để phân loại các ràng buộc SoD. Ta có **U, R, P, S** tương ứng biểu diễn tập hợp người dùng, tập hợp các vai, tập hợp các giấy phép và tập hợp các phiên, **T** là tập các thời điểm $(0, \infty)$; $u \in \mathbf{U}, r \in \mathbf{R}, p \in \mathbf{P}, s \in \mathbf{S}, t \in \mathbf{T}$.

$enabled(r, t)$: r có khả năng tại thời điểm t
$disabled(r, t)$: r không có khả năng tại thời điểm t
$u\_assigned(u, r, t)$: u được gán vào r tại thời điểm t
$p\_assigned(p, r, t)$: p được gán vào r tại thời điểm t
$active(u, r, t)$: r ở trạng thái kích hoạt trong phiên (các phiên) của u tại thời điểm t
$s\_active(u, r, s, t)$: r ở trạng thái kích hoạt trong phiên s của u tại thời điểm t
$can\_activate(u, r, t)$: u có khả năng kích hoạt r tại thời điểm t
$s\_can\_activate(u, r, s, t)$: u có khả năng kích hoạt r trong phiên s tại thời điểm t
$can\_acquire(u, p, t)$: u có khả năng có được p tại thời điểm t
$r\_can\_acquire(u, p, r, t)$: u có khả năng có được p thông qua r tại thời điểm t
$can\_be\_acquired(p, r, t)$: p có thể có được thông qua r tại thời điểm t
$acquires(u, p, t)$: u có được p tại thời điểm t
$r\_acquires(u, p, r, t)$: u có được p thông qua r tại thời điểm t
$s\_acquires(u, p, s, t)$: u có được p trong phiên s tại thời điểm t
$rs\_acquires(u, p, r, s, t)$: u có được p thông qua r trong phiên s tại thời điểm t

Hệ tiên đề sau nêu các quan hệ chủ yếu giữa các vị từ trên, làm cơ sở để nhận biết chính xác sự có được giấy phép và sự kích hoạt vai có khả năng hoặc đang xảy ra trong một hệ thống RBAC.

Hệ tiên đề. *Với* $\forall r \in \mathbf{R}, \forall u \in \mathbf{U}, \forall p \in \mathbf{P}, \forall s \in \mathbf{S}$ *và* $\forall t \in \mathbf{T}$, *các phép kéo theo sau là đúng:*

1. $p\_assigned(p, r, t) \rightarrow can\_be\_acquired(p, r, t)$.

2. $u\_assigned(u, r, t) \rightarrow can\_activate(u, r, t)$.
3. $can\_activate(u, r, t) \wedge can\_be\_acquired(p, r, t) \rightarrow can\_acquire(u, p, t)$.
4. $s\_active(u, r, s, t) \wedge can\_be\_acquired(p, r, t) \rightarrow s\_acquires(u, p, s, t)$.

Sau đây chúng ta định nghĩa một hàm liệt kê miền trị vị từ *list* trên các vị từ trạng thái và một toán tử chiếu $\Pi k_1, k_2, ..., k_m$ trên miền trị được liệt kê của một vị từ như sau.

Định nghĩa 2.2. Cho $status(a_1, ..., a_n)$ là một vị từ trạng thái trong đó $(a_1, ..., a_n)$ là một danh sách đối số tương ứng có miền trị $D_1, ..., D_n, (\forall k \in \{1, ..., n\}, D_k \in \{\mathbf{U}, \mathbf{R}, \mathbf{P}, \mathbf{S}, \mathbf{T}\})$. Nếu DOM là miền trị của vị từ trạng thái $status(a_1, ..., a_n)$ thì chúng ta định nghĩa hàm liệt kê miền trị *list* và toán tử chiếu $\Pi_{k_1,k_2,...,k_m}$ như sau:

- $list(status(a_1, ..., a_n)) = \{(x_1, ..., x_n) | ((x_1, ..., x_n) \in DOM) \wedge status(x_1, ..., x_n)\}$
- $\Pi_{(k_1,k_2,...,k_m\ list\ (status(a_1,...,a_n))} =$
  $\big\{(x_{k_1}, x_{k_2}, ..., x_{k_m}) \,|\, \exists (x_1, x_2, ..., x_n) \in list\ (status(a_1, ..., a_n)), x_{k_i} \in \{x_1, x_2, ..., x_n\},$
  $\forall i \in \{1, 2, ..., m\}; \forall (x_1, x_2, ..., x_n), (y_1, y_2, ..., y_n) \in list(status(a_1, ..., a_n))$ thì $x_j = y_j,$
  $\forall j \in \{1, 2, ..., n\} \backslash \{k_1, k_2, ..., k_m\}\big\}$.

Hàm liệt kê miền trị vị từ *list* trả về tập con của miền trị tương ứng với vị từ mà nó đánh giá. Chẳng hạn, *list* $(enabled(r, t))$ là tập con của miền $(\mathbf{R} \times \mathbf{T})$. Toán tử chiếu $\Pi_{k_1,k_2,...,k_m}$ cho phép chúng ta chiếu hàm liệt kê miền trị của một vị từ trên một đối số cụ thể được chỉ mục bởi i. Chẳng hạn $\Pi_1 list\ (enabled(r, t))$ trả về tập hợp tất cả các vai mà có khả năng tại thời điểm t. Tương tự, $\Pi_2 list\ (enabled(r, t))$ trả về tập hợp tất cả các thời điểm mà tại đó vai r có khả năng. Ta ký hiệu tập tất cả các hàm chiếu trên các vị từ được xác định ở trên là Π. Chú ý rằng, cũng có thể có hàm phủ định của các vị từ này, chẳng hạn $\Pi_1 list\ (disabled(r, t))$ hay $\Pi_1 list\ (\neg enabled(r, t))$. Ta ký hiệu Π^{-1} là tập tất cả các toán tử chiếu trên các vị từ phủ định. Dựa trên các toán tử chiếu này và tập hợp các phần tử tập hợp $\{\mathbf{U}, \mathbf{R}, \mathbf{P}, \mathbf{S}, \mathbf{T}\}$, Joshi và cộng sự [5] đã xây dựng một khung làm việc để biểu diễn một tập toàn diện các ràng buộc số lượng. Cho $\mathrm{OP} \in \{\cup, \cap, \backslash\}$ là một phép toán tập hợp, chúng ta có một hàm tập hợp tổng quát f như sau:

1. $f \in (\Pi \cup \Pi^{-1})$.
2. $f = (f\,\mathrm{OP}\,X)$, trong đó $X \subseteq E \in \{\mathbf{U}, \mathbf{R}, \mathbf{P}, \mathbf{S}, \mathbf{T}\}$.
3. $f = (f_1\,\mathrm{OP}\,f_2)$, trong đó f_1 và f_2 là các hàm tập hợp tổng quát.

Chúng ta có thể biểu diễn một ràng buộc số lượng như là $(|f|\,\mathrm{cop}\,n)$, trong đó $|f|$ là số phần tử trong tập hợp f, $\mathrm{cop} \in \{=, \neq, <, >, \geqslant, \leqslant\}$ là một toán tử so sánh và n là một số nguyên dương. Các ràng buộc chu kỳ và độ dài thời gian trên một ràng buộc số lượng $C = (|f|\,\mathrm{cop}\,n)$ có thể được xác định một cách đơn giản khi dùng khung làm việc thời gian của GTRBAC như là (I, P, C) chỉ ra rằng ràng buộc số lượng là đúng đối với mỗi thời điểm trong các khoảng thời gian được xác định bởi (I, P) và như là $([I, P, |D|], D_x, C)$ với D_x chỉ độ dài thời gian trong đó ràng buộc số lượng là đúng. Chúng ta chú ý rằng một số ràng buộc số lượng có dạng $C = (|\Pi_{k_1,k_2,...,k_m} list\ (status(a_1, ..., a_n)|\mathrm{cop}\,n)$ có thể không có ứng dụng trực tiếp trong khung làm việc GRBAC. Ví dụ. $\Pi_1 list\ (s\_active(u, r, s, t))$ (tập các người dùng đã kích hoạt vai r trong phiên s ở thời điểm t kết hợp nhiều người dùng với cùng một phiên. Các trường hợp như thế có thể hữu ích nếu ta xét một hệ thống cộng tác trong đó một phiên được tạo ra cho nhiều người dùng kích hoạt.

3. PHÂN LOẠI CÁC RÀNG BUỘC PHÂN LY TRÁCH NHIỆM

Các chính sách phân ly trách nhiệm (SoD) đã được nhận thấy là rất quan trọng để đảm bảo an toàn cho các ứng dụng thương mại. Các hệ thống dựa trên vai đặc biệt rất hữu dụng trong việc biểu diễn và thực thi các chính sách như vậy. Các SoD khác nhau đã được nói đến trong nhiều tài liệu. Tuy nhiên tất cả các nghiên cứu trước đây tập trung vào các SoD trong một môi trường phi thời gian. Joshi và cộng sự đề cập nhiều đến các SoD có tính đến thời gian trong [5]. Trong phần này chúng ta xác định các loại ràng buộc SoD thời gian đối với các vị từ trạng thái GTRBAC đã được đưa vào trong Mục 2.2. Trong mục này, chúng ta sử dụng ký hiệu ::= để định nghĩa một biểu thức ràng buộc. Không làm mất tính tổng quát, chúng ta xét: $\forall u \in \mathbf{U}, \forall r \in \mathbf{R}, \forall p \in \mathbf{P}, \forall s \in \mathbf{S}, \forall t \in Sol(I, P)$.

3.1. Các ràng buộc SoD thời gian trong việc tạo khả năng / làm mất khả năng của vai

1) Không có hai vai nào của $\mathbf{R}$ có thể đồng thời có khả năng trong khoảng thời gian (I, P). Biểu thức: EN-SoD $= (I, P, \text{EN}, \mathbf{R})$. Trong đó EN $::= |\Pi_1 list\ (enabled(r, t))| \leqslant 1$.

2) Không có hai vai nào của $\mathbf{R}$ có thể mất khả năng đồng thời trong khoảng thời gian (I, P). Biểu thức: DIS-SoD $= (I, P, \text{DIS}, \mathbf{R})$. Trong đó DIS $::= |\Pi_1 list\ (disabled(r, t))| \leqslant 1$.

3.2. Các ràng buộc SoD thời gian trong các phép gán/thôi gán người dùng cho vai

1) Không có hai vai nào của $\mathbf{R}$ có thể đồng thời được gán cho một người dùng của $\mathbf{U}$ trong khoảng thời gian (I, P). Biểu thức: UAS_1-SoD $= (I, P, \text{UAS}_1, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{UAS}_1 ::= |\Pi_2 list\ (u\_assigned(u, r, t))| \leqslant 1.$$

2) Không có hai người dùng nào thuộc $\mathbf{U}$ có thể đồng thời được gán vào một vai của $\mathbf{R}$ trong khoảng thời gian (I, P). Biểu thức: UAS_2-SoD $= (I, P, \text{UAS}_2, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{UAS}_2 ::= |\Pi_1 list\ (u\_assigned(u, r, t))| \leqslant 1.$$

3) Những người dùng khác nhau của $\mathbf{U}$ không thể đồng thời được gán vào các vai khác nhau của $\mathbf{R}$ trong khoảng thời gian (I, P). Biểu thức: UAS_3-SoD $= (I, P, \text{UAS}_3, \mathbf{U}, \mathbf{R})$, trong đó:

$$\begin{aligned}\text{UAS}_3 ::=&((|\Pi_1 list\ (u\_assigned(u, r, t))| \geqslant 1) \wedge (|\Pi_2 list\ (u\_assigned(u, r, t))| = 1))\\ &\vee ((|\Pi_2 list\ (u\_assigned(u, r, t)) \geqslant 1) \wedge (|\Pi_1 list\ (u\_assigned(u, r, t))| = 1)).\end{aligned}$$

4) Các vai của $\mathbf{R}$ chỉ có thể đồng thời được gán vào một người dùng của $\mathbf{U}$ trong khoảng thời gian (I, P). Biểu thức: UAS_4-SoD $= (I, P, \text{UAS}_4, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{UAS}_4 ::= (|\Pi_2 list\ (u\_assigned(u, r, t))| \geqslant 1) \wedge (|\Pi_1 list\ (u\_assigned(u, r, t))| = 1).$$

5) Các người dùng của $\mathbf{U}$ chỉ có thể đồng thời được gán vào một vai của $\mathbf{R}$ trong khoảng thời gian (I, P). Biểu thức: UAS_5-SoD $= (I, P, \text{UAS}_5, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{UAS}_5 ::= (|\Pi_1 list\ (u\_assigned(u, r, t))| \geqslant 1) \wedge (|\Pi_2 list\ (u\_assigned(u, r, t))| = 1).$$

6) Một vai của $\mathbf{R}$ chỉ có thể được gán vào một người dùng của $\mathbf{U}$ (và ngược lại) tại một thời điểm trong khoảng thời gian (I, P). Biểu thức: UAS_6-SoD $= (I, P, \text{UAS}_6, \mathbf{U}, \mathbf{R})$, trong

đó:

$$\text{UAS}_6 ::= (|\Pi_2 list\ (u\_assigned(u,r,t))| \leqslant 1) \wedge (|\Pi_1 list\ (u\_assigned(u,r,t))| \leqslant 1).$$

Ví dụ. Trong một tổ chức, ràng buộc $(I, P, \text{UAS}_3, \mathbf{U}, \mathbf{R})$ không cho phép hai người dùng có quan hệ họ hàng (cha-con, vợ-chồng, anh-em...) được gán vào hai vai khác nhau có khả năng tạo ra gian lận làm phương hại đến tổ chức, như : vai kế toán trưởng và vai thủ quỹ, vai thủ trưởng ký mua hàng và vai nhân viên đi mua hàng.

Định lý 3.2. *Các ràng buộc SoD thời gian gán người dùng cho vai sau là tương đương:*

1) UAS_4-SoD $\Leftrightarrow$ UAS_2-SoD $\wedge$ UAS_3-SoD.

2) UAS_5-SoD $\Leftrightarrow$ UAS_1-SoD $\wedge$ UAS_3-SoD.

3) UAS_6-SoD $\Leftrightarrow$ UAS_1-SoD $\wedge$ UAS_2-SoD.

Chứng minh:

1) Trước hết ta chứng minh rằng: $\text{UAS}_4 \Leftrightarrow \text{UAS}_2 \wedge \text{UAS}_3$.

Thật vậy ta có: $\text{UAS}_2 \wedge \text{UAS}_3$

$\Leftrightarrow (|\Pi_1 list\ (u\_assigned(u,r,t))| \leqslant 1) \wedge$

$(((|\Pi_1 list\ (u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list\ (u\_assigned(u,r,t))| = 1)) \vee$

$((|\Pi_2 list\ (u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list\ (u\_assigned(u,r,t))| = 1)))$

$\Leftrightarrow ((|\Pi_1 list\ (u\_assigned(u,r,t))| \leqslant 1) \wedge$

$(|\Pi_1 list\ (u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list\ (u\_assigned(u,r,t))| = 1)) \vee$

$((|\Pi_1 list\ (u\_assigned(u,r,t))| \leqslant 1) \wedge$

$(|\Pi_2 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| = 1))$

$\Leftrightarrow ((|\Pi_2 list(u\_assigned(u,r,t))| = 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| = 1)) \vee$

$((|\Pi_2 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| = 1))$

$\Leftrightarrow (|\Pi_2 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| = 1)) \Leftrightarrow UAS_4.$

Suy ra:

$(I, P, \text{UAS}_4, \mathbf{U}, \mathbf{R}) \Leftrightarrow (I, P, \text{UAS}_2 \wedge \text{UAS}_3, \mathbf{U}, \mathbf{R}) \Leftrightarrow (I, P, \text{UAS}_2, \mathbf{U}, \mathbf{R}) \wedge (I, P, \text{UAS}_3, \mathbf{U}, \mathbf{R})$.

Vậy ta được: UAS_4-SoD $\Leftrightarrow$ UAS_2-SoD $\wedge$ UAS_3-SoD.

2) Tương tự 1), để chứng minh: UAS_5-SoD $\Leftrightarrow$ UAS_1-SoD $\wedge$ UAS_3-SoD, ta chỉ cần chứng tỏ rằng: $\text{UAS}_5 \Leftrightarrow \text{UAS}_1 \wedge \text{UAS}_3$.

Ta có: $\text{UAS}_1 \wedge \text{UAS}_3$

$\Leftrightarrow (|\Pi_2 list(u\_assigned(u,r,t))| \leqslant 1) \wedge$

$(((|\Pi_1 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(u\_assigned(u,r,t))| = 1)) \vee$

$((|\Pi_2 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| = 1)))$

$\Leftrightarrow ((|\Pi_2 list(u\_assigned(u,r,t))| \leqslant 1) \wedge$

$(|\Pi_1 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(u\_assigned(u,r,t))| = 1)) \vee$

$((|\Pi_2 list(u\_assigned(u,r,t))| \leqslant 1) \wedge$

$(|\Pi_2 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| = 1))$

$\Leftrightarrow ((|\Pi_1 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(u\_assigned(u,r,t))| = 1)) \vee$

$((|\Pi_1 list(u\_assigned(u,r,t))| = 1) \wedge (|\Pi_2 list(u\_assigned(u,r,t))| = 1))$

$\Leftrightarrow (|\Pi_1 list(u\_assigned(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(u\_assigned(u,r,t))| = 1)) \Leftrightarrow UAS_5.$

Vậy ta được: $\text{UAS}_5 \Leftrightarrow \text{UAS}_1 \wedge \text{UAS}_3$.

3) Tương tự 1), để chứng minh: UAS_6-SoD $\Leftrightarrow$ UAS_1-SoD $\wedge$ UAS_2-SoD, ta chỉ cần chứng tỏ rằng: $\text{UAS}_6 \Leftrightarrow \text{UAS}_1 \wedge \text{UAS}_2$. Nhưng điều này là hiển nhiên vì:

$$\text{UAS}_1 \wedge \text{UAS}_2 \Leftrightarrow (|\Pi_2 list(u\_assigned(u,r,t))| \leqslant 1) \wedge (|\Pi_1 list(u\_assigned(u,r,t))| \leqslant 1)$$
$$\Leftrightarrow \text{UAS}_6.$$

3.3. Các ràng buộc SoD thời gian trong các phép gán / thôi gán giấy phép cho vai

1) Không có hai vai nào của $\mathbf{R}$ có thể đồng thời được gán một giấy phép thuộc $\mathbf{P}$ trong khoảng thời gian (I, P). Biểu thức: PAS_1-SoD $= (I, P, \text{PAS}_1, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{PAS}_1 ::= (|\Pi_2 list(p\_assigned(p,r,t))| \leqslant 1.$$

2) Không có hai giấy phép nào thuộc $\mathbf{P}$ có thể đồng thời được gán cho một vai của $\mathbf{R}$ trong khoảng thời gian (I, P). Biểu thức: PAS_2-SoD $= (I, P, \text{PAS}_2, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{PAS}_2 ::= (|\Pi_1 list(p\_assigned(p,r,t))| \leqslant 1.$$

3) Các giấy phép khác nhau thuộc $\mathbf{P}$ không thể đồng thời được gán cho các vai khác nhau của R trong khoảng thời gian (I, P). Biểu thức: PAS_3-SoD $= (I, P, \text{PAS}_3, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{PAS}_3 ::= ((|\Pi_1 list(p\_assigned(p,r,t))| \geqslant 1) \wedge (|\Pi_2 list(p\_assigned(p,r,t))| = 1))$$
$$\vee ((|\Pi_2 list(p\_assigned(p,r,t))| \geqslant 1 \wedge (|\Pi_1 list(p\_assigned(p,r,t))| = 1)).$$

4) Các vai của $\mathbf{R}$ chỉ có thể đồng thời được gán một giấy phép thuộc $\mathbf{P}$ trong khoảng thời gian (I, P). Biểu thức: PAS_4-SoD $= (I, P, \text{PAS}_4, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{PAS}_4 ::= (|\Pi_2 list(p\_assigned(p,r,t))| \geqslant 1) \wedge (|\Pi_1 list(p\_assigned(p,r,t))| = 1).$$

5) Các giấy phép thuộc $\mathbf{P}$ chỉ có thể đồng thời được gán vào một vai của $\mathbf{R}$ trong khoảng thời gian (I, P). Biểu thức: PAS_5-SoD $= (I, P, \text{PAS}_5, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{PAS}_5 ::= (|\Pi_1 list(p\_assigned(p,r,t))| \geqslant 1) \wedge (|\Pi_2 list(p\_assigned(p,r,t))| = 1).$$

6) Một giấy phép thuộc P chỉ có thể được gán vào một vai của R (và ngược lại) tại một thời điểm trong khoảng thời gian (I, P). Biểu thức: PAS_6-SoD $= (I, P, \text{PAS}_6, P, R)$, trong đó:

$$\text{PAS}_6 ::= (|\Pi_2 list(p\_assigned(p,r,t))| \leqslant 1) \wedge (|\Pi_1 list(p\_assigned(p,r,t))| \leqslant 1).$$

Định lý 3.3. *Các ràng buộc SoD thời gian gán giấy phép cho vai sau là tương đương:*

1) PAS_4-SoD $\Leftrightarrow$ PAS_2-SoD $\wedge$ PAS_3-SoD.

2) PAS_5-SoD $\Leftrightarrow$ PAS_1-SoD $\wedge$ PAS_3-SoD.

3) PAS_6-SoD $\Leftrightarrow$ PAS_1-SoD $\wedge$ PAS_2-SoD.

Chúng ta có nhận xét là giữa việc gán người dùng cho vai và việc gán giấy phép cho vai có sự đối ngẫu, nên việc chứng minh Định lý 3.3 thì tương tự như chứng minh Định lý 3.2.

3.4. Các ràng buộc SoD thời gian kích hoạt vai

1) Không có hai vai nào của $\mathbf{R}$ có thể đồng thời ở trạng thái kích hoạt trong một phiên (các phiên) của một người dùng thuộc $\mathbf{U}$ trong khoảng thời gian (I, P). Biểu thức: ACT_1-SoD $=$

$(I, P, \mathrm{ACT}_1, \mathbf{U}, \mathbf{R})$, trong đó:

$$\mathrm{ACT}_1 ::= (|\Pi_2 list(active(u, r, t))| \leqslant 1.$$

2) Không có hai người dùng nào thuộc $\mathbf{U}$ có thể đồng thời có một vai của $\mathbf{R}$ ở trạng thái kích hoạt trong khoảng thời gian (I, P). Biểu thức: ACT_2-SoD $= (I, P, \mathrm{ACT}_2, \mathbf{U}, \mathbf{R})$, trong đó:

$$\mathrm{ACT}_2 ::= (|\Pi_1 list(active(u, r, t))| \leqslant 1.$$

3) Không có hai người dùng nào thuộc $\mathbf{U}$ có thể đồng thời có hai vai khác nhau của $\mathbf{R}$ ở trạng thái kích hoạt trong khoảng thời gian (I, P). ACT_3-SoD $= (I, P, \mathrm{ACT}_3, \mathbf{U}, \mathbf{R})$, trong đó:

$$\begin{aligned}\mathrm{ACT}_3 ::=&(|\Pi_1 list(active(u, r, t))| \geqslant 1) \wedge (|\Pi_2 list(active(u, r, t))| = 1))\\ &\vee (|\Pi_2 list(active(u, r, t))| \geqslant 1) \wedge (|\Pi_1 list(active(u, r, t))| = 1)).\end{aligned}$$

4) Các vai khác nhau của $\mathbf{R}$ có thể đồng thời ở trạng thái kích hoạt trong một phiên (các phiên) chỉ của một người dùng thuộc $\mathbf{U}$ trong khoảng thời gian (I, P). Biểu thức: ACT_4-SoD $= (I, P, \mathrm{ACT}_4, \mathbf{U}, \mathbf{R})$, trong đó:

$$\mathrm{ACT}_4 ::= (|\Pi_2 list(active(u, r, t))| \geqslant 1) \wedge (|\Pi_1 list(active(u, r, t))| = 1).$$

5) Các người dùng thuộc $\mathbf{U}$ chỉ có thể đồng thời có một vai của $\mathbf{R}$ ở trạng thái kích hoạt trong trong khoảng thời gian (I, P). Biểu thức: ACT_5-SoD $= (I, P, \mathrm{ACT}_5, \mathbf{U}, \mathbf{R})$, trong đó:

$$\mathrm{ACT}_5 ::= (|\Pi_1 list(active(u, r, t))| \geqslant 1) \wedge (|\Pi_2 list(active(u, r, t))| = 1).$$

6) Một vai của $\mathbf{R}$ chỉ có thể ở trạng thái kích hoạt trong một phiên (các phiên) của một người dùng thuộc U và ngược lại một người dùng thuộc $\mathbf{U}$ chỉ có thể có một vai của $\mathbf{R}$ ở trạng thái kích hoạt trong một phiên (các phiên) của mình tại một thời điểm trong khoảng thời gian (I, P). Biểu thức: ACT_6-SoD $= (I, P, \mathrm{ACT}_6, \mathbf{U}, \mathbf{R})$, trong đó:

$$\mathrm{ACT}_6 ::= (|\Pi_2 list(active(u, r, t))| \leqslant 1) \wedge (|\Pi_1 list(active(u, r, t))| \leqslant 1).$$

7) Một người dùng thuộc $\mathbf{U}$ có thể có một vai của $\mathbf{R}$ ở trạng thái kích hoạt chỉ trong một phiên đơn của mình tại một thời điểm trong khoảng thời gian (I, P). Biểu thức: ACT_7-SoD $= (I, P, \mathrm{ACT}_7, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\mathrm{ACT}_7(|\Pi_3 list(s\_active(u, r, t))| \leqslant 1.$$

8) Hai vai của $\mathbf{R}$ không thể đồng thời ở trạng thái kích hoạt trong một phiên đơn của một người dùng thuộc $\mathbf{U}$ trong khoảng thời gian (I, P). Biểu thức: ACT_8-SoD $= (I, P, \mathrm{ACT}_8, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\mathrm{ACT}_8 ::= (|\Pi_2 list(s\_active(u, r, s, t))| \leqslant 1.$$

9) Không có hai phiên nào của một người dùng thuộc $\mathbf{U}$ có thể có hai vai của $\mathbf{R}$ đồng thời ở trạng thái kích hoạt trong khoảng thời gian (I, P). Biểu thức: ACT_9-SoD $= (I, P, \mathrm{ACT}_9, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\begin{aligned}\text{ACT}_9 ::=&((|\Pi_2 list(s\_active(u,r,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,t))| = 1))\\ &\vee ((|\Pi_3 list(s\_active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,t))| = 1)).\end{aligned}$$

10) Một người dùng thuộc **U** chỉ có thể có một vai của **R** ở trạng thái kích hoạt trong các phiên đơn của mình tại một thời điểm trong khoảng thời gian (I, P). Biểu thức: ACT_{10}-SoD $= (I, P, \text{ACT}_{10}, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{ACT}_{10} ::= (|\Pi_3 list(s\_active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,t))| = 1).$$

11) Các vai của **R** chỉ có thể đồng thời ở trạng thái kích hoạt trong một phiên đơn của một người dùng thuộc **U** trong khoảng thời gian (I, P). Biểu thức: ACT_{11}-SoD $= (I, P, \text{ACT}_{11}, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{ACT}_{11} ::= (|\Pi_2 list(s\_active(u,r,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,t))| = 1).$$

12) Các vai của **R** có thể đồng thời ở trạng thái kích hoạt trong một phiên đơn chỉ của một người dùng thuộc **U** trong khoảng thời gian (I, P). Biểu thức: ACT_{12}-SoD $= (I, P, \text{ACT}_{12}, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{ACT}_{12} ::= (|\Pi_2 list(s\_active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(s\_active(u,r,t))| = 1).$$

Định lý 3.4. *Các ràng buộc SoD thời gian trong việc kích hoạt vai sau là tương đương:*

1) ACT_4-SoD $\Leftrightarrow$ ACT_2-SoD $\wedge$$\text{ACT}_3$-SoD.
2) ACT_5-SoD $\Leftrightarrow$ ACT_1-SoD $\wedge$$\text{ACT}_3$-SoD.
3) ACT_6-SoD $\Leftrightarrow$ ACT_1-SoD $\wedge$$\text{ACT}_2$-SoD.
4) ACT_{10}-SoD $\Leftrightarrow$ ACT_8-SoD $\wedge$$\text{ACT}_9$-SoD.
5) ACT_{11}-SoD $\Leftrightarrow$ ACT_7-SoD $\wedge$$\text{ACT}_9$-SoD.

Chứng minh:

1) Trước hết ta chứng minh rằng: $\text{ACT}_4 \Leftrightarrow \text{ACT}_2 \wedge \text{ACT}_3$. Thật vậy ta có: $\text{ACT}_2 \wedge \text{ACT}_3$

$$\begin{aligned}&\Leftrightarrow (|\Pi_1 list(active(u,r,t))| \leqslant 1)\wedge\\ &\quad(((|\Pi_1 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1))\vee\\ &\quad((|\Pi_2 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1)))\\ &\Leftrightarrow ((|\Pi_1 list(active(u,r,t))| \leqslant 1)\wedge\\ &\quad(|\Pi_1 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1))\vee\\ &\quad((|\Pi_1 list(active(u,r,t))| \leqslant 1)\wedge\\ &\quad(|\Pi_2 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1))\\ &\Leftrightarrow ((|\Pi_2 list(active(u,r,t))| = 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1))\vee\\ &((|\Pi_2 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1))\\ &\Leftrightarrow (|\Pi_2 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1)) \Leftrightarrow \text{ACT}_4.\end{aligned}$$

Suy ra: $(I, P, \text{ACT}_4, \mathbf{U}, \mathbf{R}) \Leftrightarrow (I, P, \text{ACT}_2 \wedge \text{ACT}_3, \mathbf{U}, \mathbf{R}) \Leftrightarrow (I, P, \text{ACT}_2, \mathbf{U}, \mathbf{R}) \wedge (I, P, \text{ACT}_3, \mathbf{U}, \mathbf{R})$.

Vậy ta được: ACT_4-SoD $\Leftrightarrow$ ACT_2-SoD $\wedge$$\text{ACT}_3$-SoD.

2) Theo phần 1), để chứng minh: ACT_5SoD $LRa$$\text{ACT}_1$SoD $wedge$$\text{ACT}_3$-SoD, ta chỉ cần chứng tỏ rằng:

$\text{ACT}_5 \Leftrightarrow \text{ACT}_1 \wedge \text{ACT}_3$. Ta có: $\text{ACT}_1 \wedge \text{ACT}_3 \Leftrightarrow$

$\Leftrightarrow (|\Pi_2 list(active(u,r,t))| \leqslant 1) \wedge$
$(((|\Pi_1 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1)) \vee$
$((|\Pi_2 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1)))$
$\Leftrightarrow ((|\Pi_2 list(active(u,r,t))| \leqslant 1) \wedge$
$(|\Pi_1 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1)) \vee$
$((|\Pi_2 list(active(u,r,t))| \leqslant 1) \wedge$
$(|\Pi_2 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_1 list(active(u,r,t))| = 1))$
$\Leftrightarrow ((|\Pi_1 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1)) \vee$
$((|\Pi_1 list(active(u,r,t))| = 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1))$
$\Leftrightarrow (|\Pi_1 list(active(u,r,t))| \geqslant 1) \wedge (|\Pi_2 list(active(u,r,t))| = 1)) \Leftrightarrow \mathrm{ACT}_5.$
Vậy ta được: $\mathrm{ACT}_5 \Leftrightarrow \mathrm{ACT}_1 \wedge \mathrm{ACT}_3$.

3) Theo phần 1), để chứng minh: ACT_6-SoD $\Leftrightarrow$ ACT_1-SoD $\wedge$$\mathrm{ACT}_2$-SoD, ta chỉ cần chứng tỏ rằng: $\mathrm{ACT}_6 \Leftrightarrow \mathrm{ACT}_1 \wedge \mathrm{ACT}_2$. Nhưng điều này là hiển nhiên vì:

$$\mathrm{ACT}_1 \wedge \mathrm{ACT}_2 \Leftrightarrow (|\Pi_2 list(active(u,r,t))| = 1) \wedge (|\Pi_1 list(active(u,r,t))| \leqslant 1) \Leftrightarrow \mathrm{ACT}_6.$$

4) Theo phần 1), để chứng minh: ACT_{10}-SoD $\Leftrightarrow$ ACT_8-SoD $\wedge$$\mathrm{ACT}_9$-SoD, ta chỉ cần chứng tỏ rằng: $\mathrm{ACT}_{10} \Leftrightarrow \mathrm{ACT}_8 \wedge \mathrm{ACT}_9$.

Ta có: $\mathrm{ACT}_8 \wedge \mathrm{ACT}_9 \Leftrightarrow$
$\Leftrightarrow (|\Pi_2 list(s\_active(u,r,s,t))| \leqslant 1) \wedge$
$(((|\Pi_2 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1)) \vee$
$((|\Pi_3 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1)))$
$\Leftrightarrow ((|\Pi_2 list(s\_active(u,r,s,t))| \leqslant 1) \wedge$
$(|\Pi_2 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1)) \vee$
$((|\Pi_2 list(s\_active(u,r,s,t))| \leqslant 1) \wedge$
$(|\Pi_3 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1))$
$\Leftrightarrow ((|\Pi_3 list(s\_active(u,r,s,t))| = 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1)) \vee$
$((|\Pi_3 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1))$
$\Leftrightarrow (|\Pi_3 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1)) \Leftrightarrow \mathrm{ACT}_{10}.$
Vậy ta được: $\mathrm{ACT}_{10} \Leftrightarrow \mathrm{ACT}_8 \wedge \mathrm{ACT}_9$.

5) Theo phần 1), để chứng minh: ACT_{11}-SoD $\Leftrightarrow$ ACT_7-SoD $\wedge$$\mathrm{ACT}_9$-SoD, ta chỉ cần chứng tỏ rằng: $\mathrm{ACT}_{11} \Leftrightarrow \mathrm{ACT}_7 \wedge \mathrm{ACT}_9$. Ta có: $\mathrm{ACT}_7 \wedge \mathrm{ACT}_9 \Leftrightarrow$
$\Leftrightarrow (|\Pi_3 list(s\_active(u,r,s,t))| \leqslant 1) \wedge$
$(((|\Pi_2 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1)) \vee$
$((|\Pi_3 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1)))$
$\Leftrightarrow ((|\Pi_3 list(s\_active(u,r,s,t))| \leqslant 1) \wedge$
$(|\Pi_2 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1)) \vee$
$((|\Pi_3 list(s\_active(u,r,s,t))| \leqslant 1) \wedge$
$(|\Pi_3 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_active(u,r,s,t))| = 1))$
$\Leftrightarrow ((|\Pi_2 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1)) \vee$
$((|\Pi_2 list(s\_active(u,r,s,t))| = 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1))$
$\Leftrightarrow (|\Pi_2 list(s\_active(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_active(u,r,s,t))| = 1)) \Leftrightarrow \mathrm{ACT}_{11}.$
Vậy ta được: $\mathrm{ACT}_{11} \Leftrightarrow \mathrm{ACT}_7 \wedge \mathrm{ACT}_9$.

3.5. Các ràng buộc SoD thời gian về việc có khả năng kích hoạt vai

1) Một người dùng thuộc **U** không có khả năng kích hoạt đồng thời hai vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACT_1-SoD $= (I, P, \text{CACT}_1, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{CACT}_1 ::= |\Pi_2 list(can\_activate(u, r, t))| \leqslant 1.$$

2) Không có hai người dùng nào thuộc **U** có khả năng kích hoạt đồng thời một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACT_2-SoD $= (I, P, \text{CACT}_2, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{CACT}_2 ::= |\Pi_1 list(can\_activate(u, r, t))| \leqslant 1.$$

3) Không có hai người dùng nào thuộc **U** có khả năng kích hoạt đồng thời hai vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACT_3-SoD $= (I, P, \text{CACT}_3, \mathbf{U}, \mathbf{R})$, trong đó:

$$\begin{aligned}\text{CACT}_3 ::=&((|\Pi_1 list(can\_activate(u, r, t))| \leqslant 1) \wedge ((|\Pi_2 list(can\_activate(u, r, t))| = 1))\\ &\vee ((|\Pi_2 list(can\_activate(u, r, t))| \leqslant 1) \wedge ((|\Pi_1 list(can\_activate(u, r, t))| = 1)).\end{aligned}$$

4) Một người dùng thuộc **U** có khả năng kích hoạt đồng thời hai vai khác nhau của **R** trong một phiên (các phiên) của mình trong khoảng thời gian (I, P).

Biểu thức: CACT_4-SoD $= (I, P, \text{CACT}_4, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{CACT}_4 ::= ((|\Pi_2 list(can\_activate(u, r, t))| \geqslant 1) \wedge (|\Pi_1 list(can\_activate(u, r, t))| = 1).$$

5) Các người dùng của **U** chỉ có khả năng kích hoạt đồng thời một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACT_5-SoD $= (I, P, \text{CACT}_5, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{CACT}_5 ::= (|\Pi_1 list(can\_activate(u, r, t))| \geqslant 1) \wedge (|\Pi_2 list(can\_activate(u, r, t))| = 1).$$

6) Một người dùng của **U** chỉ có khả năng kích hoạt một vai của **R** (và ngược lại) tại một thời điểm trong khoảng thời gian (I, P). Biểu thức: CACT_6-SoD $= (I, P, \text{CACT}_6, \mathbf{U}, \mathbf{R})$, trong đó:

$$\text{CACT}_6 ::= (|\Pi_2 list(can\_activate(u, r, t))| \leqslant 1) \wedge (|\Pi_1 list(can\_activate(u, r, t))| \leqslant 1).$$

7) Một người dùng thuộc **U** không có khả năng kích hoạt một vai của **R** đồng thời trong các phiên khác nhau của mình trong khoảng thời gian (I, P). Biểu thức: CACT_7-SoD $= (I, P, \text{CACT}_7, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{CACT}_7 ::= |\Pi_3 list(s\_can\_activate(u, r, s, t))| \leqslant 1).$$

8) Một người dùng thuộc **U** không có khả năng kích hoạt đồng thời hai vai khác nhau của **R** trong một phiên đơn s của mình trong khoảng thời gian (I, P). Biểu thức:CACT_8-SoD $= (I, P, \text{CACT}_8, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{CACT}_8 ::= |\Pi_2 list(s\_can\_activate(u, r, s, t))| \leqslant 1).$$

9) Một người dùng thuộc **U** không có khả năng kích hoạt đồng thời hai vai khác nhau của **R** trong các phiên khác nhau của mình trong khoảng thời gian (I, P).

Biểu thức: $\text{CACT}_9\text{SoD} = I, P, \text{CACT}_9, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{CACT}_9 ::= ((|\Pi_2 list(s\_can\_activate(u,r,s,t))| \leqslant 1) \wedge ((|\Pi_3 list(s\_can\_activate(u,r,s,t))| = 1)) \\ \vee ((|\Pi_3 list(s\_can\_activate(u,r,s,t))| \leqslant 1) \wedge (|\Pi_2 list(s\_can\_activate(u,r,s,t))| = 1)).$$

10) Một người dùng thuộc **U** chỉ có khả năng kích hoạt một vai của **R** trong các phiên của mình tại một thời điểm trong khoảng thời gian (I, P).
Biểu thức: $\text{CACT}_{10}\text{SoD} = I, P, \text{CACT}_{10}, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$\text{CACT}_{10} ::= (|\Pi_3 list(s\_can\_activate(u,r,s,t))| \geqslant 1) \wedge (|\Pi_2 list(s\_can\_activate(u,r,s,t))| = 1)$.

11) Một người dùng thuộc **U** có khả năng kích hoạt đồng thời hai vai khác nhau của **R** chỉ trong một phiên đơn của mình trong khoảng thời gian (I, P). Biểu thức: CACT_{11}-SoD $= (I, P, \text{CACT}_{11}, \mathbf{U}, \mathbf{R}, \mathbf{S})$, trong đó:

$$\text{CACT}_{11} ::= (|\Pi_2 list(s\_can\_activate(u,r,s,t))| \geqslant 1) \wedge (|\Pi_3 list(s\_can\_activate(u,r,s,t))| = 1).$$

Định lý 3.5. *Các ràng buộc SoD thời gian có khả năng kích hoạt vai sau là tương đương:*
1) CACT_4-SoD $\Leftrightarrow \text{CACT}_2\text{SoD} \wedge \text{CACT}_3$-SoD.
2) CACT_5-SoD $\Leftrightarrow \text{CACT}_1$-SoD $\wedge \text{CACT}_3$-SoD.
3) CACT_6-SoD $\Leftrightarrow \text{CACT}_1$-SoD $\wedge \text{CACT}_2$-SoD.
4) CACT_{10}-SoD $\Leftrightarrow \text{CACT}_8$-SoD $\wedge \text{CACT}_9$-SoD.
5) CACT_{11}-SoD $\Leftrightarrow \text{CACT}_7$-SoD $\wedge \text{CACT}_9$-SoD.
Việc chứng minh Định lý 3.5 hoàn toàn tương tự như chứng minh Định lý 3.4.

3.6. Các ràng buộc SoD thời gian về khả năng có được giấy phép cho người dùng

3.6.1. Các ràng buộc SoD về khả năng người dùng có được giấy phép

1) Một người dùng thuộc **U** không có khả năng đồng thời có được các giấy phép khác nhau của **P** trong khoảng thời gian (I, P). Biểu thức: CACQ_1-SoD $= (I, P, \text{CACQ}_1, \mathbf{U}, \mathbf{P})$, trong đó:

$$\text{CACQ}_1 ::= (|\Pi_2 list(can\_acquire(u,p,t))| \leqslant 1.$$

2) Không có hai người dùng nào thuộc **U** có khả năng đồng thời có được một giấy phép của **P** trong khoảng thời gian (I, P). Biểu thức: CACQ_2-SoD $= (I, P, \text{CACQ}_2, \mathbf{U}, \mathbf{P})$, trong đó:

$$\text{CACQ}_2 ::= (|\Pi_1 list(can\_acquire(u,p,t))| \leqslant 1.$$

3) Không có hai người dùng nào thuộc **U** có khả năng đồng thời có được các giấy phép khác nhau của **P** trong khoảng thời gian (I, P). Biểu thức: CACQ_3-SoD $= (I, P, \text{CACQ}_3, \mathbf{U}, \mathbf{P})$, trong đó:

$$\text{CACQ}_3 ::= ((|\Pi_1 list(can\_activate(u,p,t))| \leqslant 1) \wedge (|\Pi_2 list(can\_activate(u,p,t))| = 1)) \\ \vee ((|\Pi_2 list(can\_activate(u,p,t))| \leqslant 1) \wedge (|\Pi_1 list(can\_activate(u,p,t))| = 1)).$$

4) Các người dùng thuộc **U** chỉ có khả năng đồng thời có được một giấy phép của **P** trong khoảng thời gian (I, P). Biểu thức: CACQ_4-SoD $= (I, P, \text{CACQ}_4, \mathbf{U}, \mathbf{P})$, trong đó:

$$\text{CACQ}_4 ::= (|\Pi_1 list(can\_activate(u,p,t))| \geqslant 1) \wedge (|\Pi_2 list(can\_activate(u,p,t))| = 1).$$

3.6.2. Các ràng buộc SoD về khả năng nhận được giấy phép thông qua vai

5) Không có một giấy phép nào của **P** có thể đồng thời nhận được thông qua hai vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_5-SoD $= (I, P, \text{CACQ}_5, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_5 ::= |\Pi_2 list(can\_be\_acquired(p,r,t))| \leqslant 1.$$

6) Không có hai giấy phép nào của **P** có thể đồng thời nhận được thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_6-SoD $= (I, P, \text{CACQ}_6, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_6 ::= |\Pi_1 list(can\_be\_acquired(p,r,t))| \leqslant 1.$$

7) Các giấy phép khác nhau của **P** không thể đồng thời nhận được thông qua các vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_7-SoD $= (I, P, \text{CACQ}_7, \mathbf{P}, \mathbf{R})$, trong đó:

$$\begin{aligned}\text{CACQ}_7 ::= &((|\Pi_1 list(can\_be\_acquired(p,r,t))| \geqslant 1) \wedge (|\Pi_2 list(can\_be\_acquired(p,r,t))| = 1)) \\ &\vee ((|\Pi_2 list(can\_be\_acquired(p,r,t))| \geqslant 1) \wedge (|\Pi_1 list(can\_be\_acquired(p,r,t))| = 1)).\end{aligned}$$

8) Một giấy phép của **P** chỉ có thể đồng thời nhận được thông qua các vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_8-SoD $= (I, P, \text{CACQ}_8, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_8 ::= (|\Pi_2 list(can\_be\_acquired(p,r,t))| \geqslant 1) \wedge (|\Pi_1 list(can\_be\_acquired(p,r,t))| = 1).$$

9) Các giấy phép khác nhau của **P** chỉ có thể đồng thời nhận được thông qua cùng một vai của **R** trong khoảng thời gian(I, P). Biểu thức: CACQ_9-SoD $= (I, P, \text{CACQ}_9, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_9 ::= (|\Pi_1 list(can\_be\_acquired(p,r,t))| \geqslant 1) \wedge (|\Pi_2 list(can\_be\_acquired(p,r,t))| = 1).$$

3.6.3. Các ràng buộc SoD về khả năng người dùng có được giấy phép thông qua vai

10) Một người dùng thuộc **U** không có khả năng đồng thời nhận được các giấy phép khác nhau của **P** thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{10}-SoD $= (I, P, \text{CACQ}_{10}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{10} ::= |\Pi_2 list(r\_can\_acquire(u,p,r,t))| \leqslant 1.$$

11) Các người dùng khác nhau thuộc **U** không có khả năng đồng thời có được một giấy phép của **P** thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{11}-SoD $= (I, P, \text{CACQ}_{11}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{11} ::= |\Pi_1 list(r\_can\_acquire(u,p,r,t))| \leqslant 1.$$

12) Các người dùng khác nhau thuộc **U** không có khả năng đồng thời có được các giấy phép khác nhau của **P** thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{12}-SoD $= (I, P, \text{CACQ}_{12}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{12} ::= ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| = 1)) \vee ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| = 1)).$$

13) Các người dùng khác nhau thuộc **U** chỉ có khả năng đồng thời có được một giấy phép của **P** thông qua cùng một vai của **R** trong khoảng thời gian(I, P). Biểu thức: CACQ_{13}-SoD $= (I, P, \text{CACQ}_{13}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{13} ::= (|\Pi_1 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| = 1).$$

14) Một người dùng thuộc **U** không có khả năng có được một giấy phép của **P** đồng thời thông qua các vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{14}-SoD $= (I, P, \text{CACQ}_{14}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{14} ::= |\Pi_3 list(r\_can\_acquire(u,p,r,t))| \leqslant 1.$$

15) Các người dùng khác nhau thuộc **U** không có khả năng đồng thời có được một giấy phép của **P** thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{15}-SoD $= (I, P, \text{CACQ}_{15}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{15} ::= |\Pi_1 list(r\_can\_acquire(u,p,r,t))| \leqslant 1.$$

16) Không có hai người dùng nào thuộc **U** có khả năng đồng thời có được một giấy phép của **P** thông qua các vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{16}-SoD $= (I, P, \text{CACQ}_{16}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{16} ::= ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| = 1)) \vee ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| = 1)).$$

17) Các người dùng khác nhau thuộc **U** có khả năng đồng thời có được một giấy phép của **P** chỉ thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{17}-SoD $= (I, P, \text{CACQ}_{17}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{17} ::= (|\Pi_1 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge (|\Pi_3 list(r\_can\_acquire(u,p,r,t))| = 1).$$

18) Một người dùng thuộc **U** không có khả năng đồng thời có được hai giấy phép khác nhau của **P** thông qua các vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{18}-SoD $= (I, P, \text{CACQ}_{18}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{18} ::= ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| = 1)) \vee ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge (|\Pi_2 list(r\_can\_acquire(u,p,r,t))| = 1)).$$

19) Một người dùng thuộc **U** chỉ có khả năng đồng thời nhận được các giấy phép khác nhau của **P** thông qua cùng một vai của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ_{19}-SoD $= (I, P, \text{CACQ}_1 9, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{19} ::= (|\Pi_2 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| = 1).$$

20) Một người dùng thuộc **U** có khả năng đồng thời nhận được chỉ một giấy phép của **P** thông qua các vai khác nhau của **R** trong khoảng thời gian(I, P). Biểu thức: CACQ$_{20}$-SoD $= (I, P, CACQ_{20}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\text{CACQ}_{20} ::= (|\Pi_3 list(r\_can\_acquire(u,p,r,t))| \geqslant 1 \wedge ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| = 1).$$

21) Không có hai người dùng nào thuộc **U** có khả năng đồng thời có được các giấy phép khác nhau của **P** thông qua hai vai khác nhau của **R** trong khoảng thời gian (I, P). Biểu thức: CACQ$_{21}$-SoD $= (I, P, \text{CACQ}_{21}, \mathbf{U}, \mathbf{P}, \mathbf{R})$, trong đó:

$$\begin{aligned}\text{CACQ}_{21} ::= ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| \geqslant 1) \wedge ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| = 1)\\ \vee (|\Pi_3 list(r\_can\_acquire(u,p,r,t))| = 1))) \vee ((|\Pi_2 list(r\_can\_acquire(u,p,r,t))| \geqslant 1)\\ \wedge ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| = 1) \vee ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| = 1)))\\ \vee ((|\Pi_3 list(r\_can\_acquire(u,p,r,t))| \geqslant 1) \wedge ((|\Pi_1 list(r\_can\_acquire(u,p,r,t))| = 1)\\ \vee (|\Pi_2 list(r\_can\_acquire(u,p,r,t))| = 1))).\end{aligned}$$

Ta có thể chứng tỏ các ràng buộc SoD về khả năng có được giấy phép sau là tương đương:

1) CACQ$_4$-SoD $\Leftrightarrow$ CACQ$_1$-SoD $\wedge$ CACQ$_3$-SoD;
2) CACQ$_8$-SoD $\Leftrightarrow$ CACQ$_6$-SoD $\wedge$ CACQ$_7$-SoD;
3) CACQ$_9$-SoD $\Leftrightarrow$ CACQ$_5$-SoD $\wedge$ CACQ$_7$-SoD;
4) CACQ$_{13}$-SoD $\Leftrightarrow$ CACQ$_{10}$-SoD $\wedge$ CACQ$_{12}$-SoD;
5) CACQ$_{17}$-SoD $\Leftrightarrow$ CACQ$_{14}$-SoD $\wedge$ CACQ$_{16}$-SoD;
6) CACQ$_{19}$-SoD $\Leftrightarrow$ CACQ$_{14}$-SoD $\wedge$ CACQ$_{18}$-SoD;
7) CACQ$_{20}$-SoD $\Leftrightarrow$ CACQ$_{10}$-SoD $\wedge$ CACQ$_{18}$-SoD.

Do khuôn khổ bài báo, chúng tôi không đưa ra chứng minh ở đây.

4. KẾT LUẬN

Chúng tôi đã trình bày các ràng buộc phân ly trách nhiệm đối với mô hình GTRBAC khi sử dụng một hàm liệt kê miền trị và một toán tử chiếu kết hợp với một tập các vị từ trạng thái GTRBAC để xây dựng một khung làm việc chi tiết (được Joshi và cộng sự đề xuất trong [5]) cho việc biểu diễn các ràng buộc số lượng nói chung và các ràng buộc phân ly trách nhiệm nói riêng. Các ràng buộc phân ly trách nhiệm dựa trên thực tế là khái niệm mâu thuẫn giữa các phần tử trong một tập hợp thì thường được kết hợp với tập hợp khác. Chúng tôi cũng chứng minh mối quan hệ tương đương giữa một số ràng buộc này. Điều đó cho phép xây dựng các tập ràng buộc tối thiểu để cài đặt trong thực tế kiểm soát truy nhập dựa trên vai thời gian.

TÀI LIỆU THAM KHẢO

[1] Gail-Joon Ahn and Ravi Sandhu, Role-based authorization constraints specification, *ACM Transactions on Information and System Security* (TISSEC) **3** (4) (2000).

[2] E. Bertino, P. A. Bonatti, E. Ferrari, TRBAC: A temporal role-based access control model, *ACM Transactions on Information and System Security* **4** (2001) 191–233.

[3] J. B. D. Joshi, E. Bertino, U. Latif, A. Ghafoor, Generalized temporal role based access control model (GTRBAC) (Part I)- specification and modeling, *CERIAS TR 2001-47*, Purdue University, USA (2001).

[4] J. B. D. Joshi, E. Bertino, A. Ghafoor, Temporal hierarchy and inheritance semantics for GTRBAC, *7th ACM Symposium on Access Control Models and Technologies*, Monterey, CA, June 3-4, 2002.

[5] J. B. D. Joshi, E. Bertino, B. Shafiq, A. Ghafoor, Dependencies and separation of duty constraints in GTRBAC, *SACMAT' 03*, June 1-4, 2003.

[6] S. Osborn, R. Sandhu, Q. Munawer, Configuring role-based access control to enforce mandatory and discretionary access control policies, *ACM Transactions on Information and System Security* **3** (2) (2000) 85–106.

[7] R. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, Role-based access control models, *IEEE Computer* **29** (2) (1996) 38–47.

Nhận bài ngày 16 - 3 - 2006