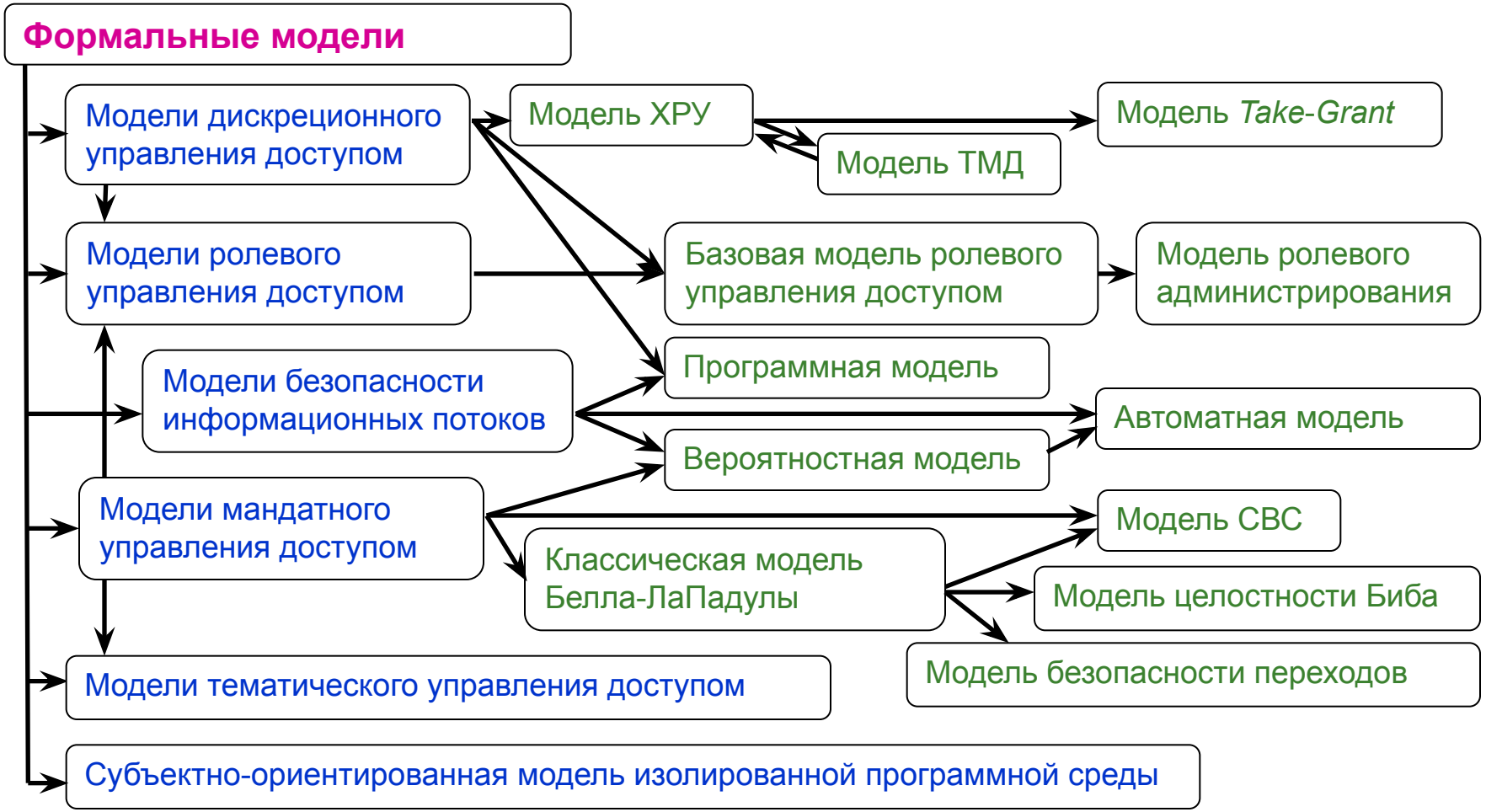
АНАЛИЗ БЕЗОПАСНОСТИ УПРАВЛЕНИЯ ДОСТУПОМ И ИНФОРМАЦИОННЫМИ ПОТОКАМИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

к.т.н., доцент Девянин П.Н.
ИКСИ, г. Москва
peter_devyanin@hotmail.com

Основные формальные модели управления доступом и информационными потоками



Основные существенные особенности функционирования современных КС

- Возможность реализации в КС доверенных и недоверенных субъектов с различными условиями функционирования;
- Различие в условиях реализации в КС информационных потоков по памяти и по времени;
- Наличие в современных КС иерархической структуры сущностей и возможность ее использования при реализации информационных потоков по времени;
- Возможность кооперации или, наоборот, противодействия субъектов друг другу при передаче прав доступа и реализации информационных потоков.
- Возможность изменения функциональности субъекта при реализации информационного потока по памяти на функционально ассоциированные с ним сущности.

Критический анализ основных формальных моделей

Особенности функционирования современных КС	Модель *Take-Grant*	Модель Белла-ЛаПадула	Модель СВС	Модель ИПС
Различие в условиях реализации информационных потоков по памяти и по времени	—	—	—	—
Наличие иерархической структуры сущностей и возможность ее использования при реализации информационных потоков по времени	—	—	+	—
Возможность кооперации части субъектов при передаче прав доступа или реализации информационных потоков	+	—	—	—
Возможность реализации доверенных и недоверенных субъектов с различными условиями функционирования	—	+	—	+
Возможность противодействия доверенными субъектами передаче прав доступа или реализации информационных потоков недоверенными субъектами	+	—	—	—
Возможность изменения функциональности субъекта при реализации информационного потока по памяти на функционально ассоциированные с ним сущности	—	—	—	+
Необходимость определения различных правил управления доступом и информационными потоками для распределенных компонент КС	—	—	—	—

Основные предположения

Предположение 1. Все действия в КС, в том числе выполнение операций над сущностями, порождение информационных потоков, изменение параметров и настроек системы защиты КС, порождение новых субъектов, могут быть инициированы только субъектами КС с использованием доступов к сущностям КС.

Предположение 2. Все информационные потоки в КС порождены доступами субъектов к сущностям. Иные информационные потоки, например по побочному электромагнитному излучению, не рассматриваются.

Предположение 3. В начальном состоянии КС отсутствуют доступы субъектов к объектам и информационные потоки.

Предположение 4. При любых запросе субъекта на доступ к сущности или доступе субъекта к сущности реализуется информационный поток по времени.

Предположение 5. При реализации информационного потока по памяти от объекта-источника к объекту-приемнику в том же направлении реализуется информационный поток по времени.

Базовая ДП-модель. Иерархия сущностей

$E = O \cup C$ —множество сущностей, где O —множество объектов, C — множество контейнеров; $S \subseteq E$ —множество субъектов;

$R_r = \{read_r, write_r, append_r, execute_r, own_r\}$ — множество видов прав доступа;

$R_a = \{read_a, write_a, append_a\}$ — множество видов доступа;

$R_f = \{write_m, write_t\}$ —множество видов информационных потоков;

$R_{raf} = R_r \cup R_a \cup R_f$ —множество видов прав доступа, видов доступа и видов информационных потоков.

Определение. Определим H: $E \to 2^E$ —функцию иерархии сущностей, сопоставляющую каждой сущности $c \in E$ множество сущностей $H(c) \subset E$ и удовлетворяющую условиям:

Условие 1. Если сущность $e \in H(c)$, то $e < c$ и не существует сущности-контейнера $d \in C$, такой, что $e < d$, $d < c$.

Условие 2. Для любых сущностей $e_1, e_2 \in E$, $e_1 \neq e_2$, по определению выполняются равенство $H(e_1) \cap H(e_2) = \varnothing$ и условия:

- если $o \in O$, то выполняется равенство $H(o) = \varnothing$;
- если $e_1 < e_2$, то или $e_1, e_2 \in E \setminus S$, или $e_1, e_2 \in S$;
- если $e \in E \setminus S$, то $H(e) \subset E \setminus S$;
- если $s \in S$, то $H(s) \subset S$.

Определение системы

Определение. Пусть определены множества S, E, $R \subseteq S \times E \times R_r$, $A \subseteq S \times E \times R_a$, $F \subseteq E \times E \times R_f$ и функция иерархии сущностей H. Определим $G = (S, E, R \cup A \cup F, H)$ — конечный помеченный ориентированный граф без петель, где элементы множеств S, E являются вершинами графа, элементы множества $R \cup A \cup F$ — ребрами графа. Назовем $G = (S, E, R \cup A \cup F, H)$ графом доступов.

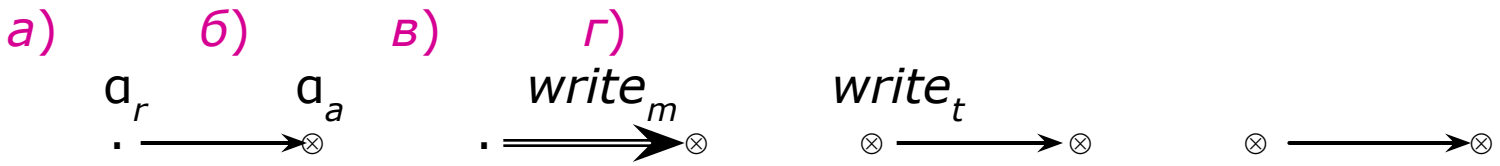


Рис. 1. Обозначения ребер графа доступов

$\Sigma(G^*, OP)$ — система, при этом:

- каждое состояние системы представляется графом доступов;
- G^* — множество всех возможных состояний;
- OP — множество правил преобразования состояний;

$G \vdash op\ G'$ — переход системы $\Sigma(G^*, OP)$ из состояния G в состояние G'

Правила преобразования состояний

В базовой ДП-модели определены 16 монотонных и немонотонных правил преобразования состояний: *take_right*(α_r, *x*, *y*, *z*), *grant_right*(α_r, *x*, *y*, *z*), *remove_right*(α_r, *x*, *y*, *z*), *own_take*(α_r, *x*, *y*), *own_remove*(α_r, *x*, *y*), *create_entity*(*x*, *y*, *z*), *create_subject*(*x*, *y*, *z*), *delete_entity*(*x*, *y*, *z*), *rename_entity*(*x*, *y*, *z*), *access_read*(*x*, *y*), *access_write*(*x*, *y*), *access_append*(*x*, *y*), *flow*(*x*, *y*, *y'*, *z*), *find*(*x*, *y*, *z*), *post*(*x*, *y*, *z*), *pass*(*x*, *y*, *z*)

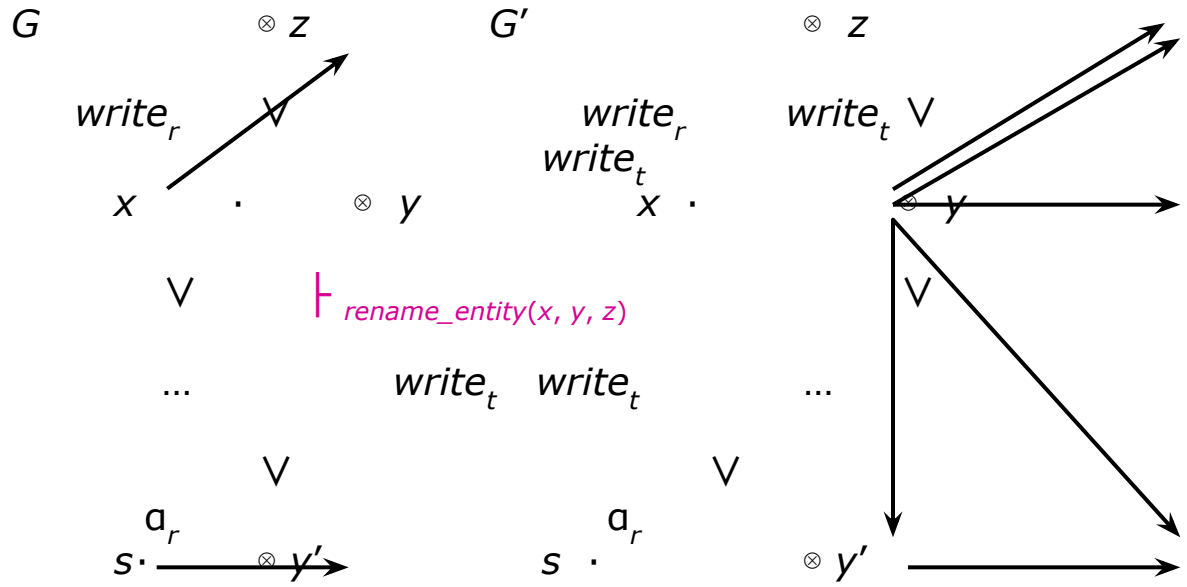


Рис. 2. Реализация информационных потоков по времени при применении правила *rename_entity*(*x*, *y*, *z*) , где $\alpha_r \in R_r$

Примеры правил преобразования состояний базовой ДП-модели

Правило	Исходное состояние $G = (S, E, R \cup A \cup F, H)$	Результирующее состояние $G' = (S', E', R' \cup A' \cup F', H')$
$take\_right(\alpha_r, x, y, z)$	$x, y \in S, z \in E, x \neq z, \alpha_r \in R_r,$ $(x, y, own_r) \in R, (y, z, \alpha_r) \in R$	$S' = S, E' = E, A' = A, H' = H, R' = R \cup \{(x, z, \alpha_r)\},$ $F' = F \cup \{(y, x, write_t)\}$
$grant\_right(\alpha_r, x, y, z)$	$x, y \in S, z \in E, y \neq z, \alpha_r \in R_r,$ $(x, y, own_r) \in R, (x, z, \alpha_r) \in R$	$S' = S, E' = E, A' = A, H' = H, R' = R \cup \{(y, z, \alpha_r)\},$ $F' = F \cup \{(x, y, write_t)\}$
$own\_take(\alpha_r, x, y)$	$x \in S, y \in E, \alpha_r \in R_r, (x, y, own_r) \in R$	$S' = S, E' = E, A' = A, F' = F, H' = H, R' = R \cup \{(x, y, \alpha_r)\}$
$create\_entity(x, y, z)$	$x \in S, y \notin E, z \in E \setminus S, (x, z, \alpha_r) \in R,$ где $\alpha_r \in \{write_r, append_r\}$	$S' = S, E' = E \cup \{y\}, A' = A, H'(z) = H(z) \cup \{y\}, H'(y) = \varnothing,$ для $e \in E \setminus \{z\}$ выполняется равенство $H'(e) = H(e), R' = R \cup \{(x, y, own_r)\}, F' = F \cup \{(x, e, write_t): e \in E$ и $y \leq e\}$
$rename\_entity(x, y, z)$	$x \in S, y, z \in E, y \in H(z), (x, z, write_r) \in R$	$S' = S, E' = E, R' = R, A' = A, H' = H,$ $F' = F \cup \{(x, z, write_t)\} \cup \{(x, e, write_t): e \in E, x \neq e$ и $e \leq y\} \cup \{(x, s,$ $write_t): s \in S, x \neq s$ и $(s, e, \alpha_r) \in R$, где $e \in E, e \leq y$ и $\alpha_r \in R_r\}$
$access\_read(x, y)$	$x \in S, (x, y, read_r) \in R$	$S' = S, E' = E, R' = R, H' = H, A' = A \cup \{(x, y, read_a)\},$ $F' = F \cup \{(y, x, write_m)\} \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\}$
$access\_write(x, y)$	$x \in S, (x, y, write_r) \in R$	$S' = S, E' = E, R' = R, H' = H, A' = A \cup \{(x, y, write_a)\},$ $F' = F \cup \{(x, y, write_m)\} \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\}$
$flow(x, y, y', z)$	$x, z \in S, y, y' \in E, x \neq z,$ $\{(x, y, \alpha_r), (z, y', \beta_r)\} \subset R$, где $y \leq y', \alpha_r, \beta_r \in R_r$	$S' = S, E' = E, R' = R, A' = A, H' = H,$ $F' = F \cup \{(x, z, write_t), (z, x, write_t)\}$
$post(x, y, z)$	$x, z \in S, y \in E, x \neq z, \{(x, y, \alpha), (z, y, read_r)\} \subset R \cup F$, где $\alpha \in \{write_r, append_r, write_m, write_t\}$	$S' = S, E' = E, R' = R, A' = A, H' = H,$ если $\alpha \in \{write_r, append_r, write_m\},$ то $F' = F \cup \{(x, z, write_m)\}$, если $\alpha = write_t$, то $F' = F \cup \{(x, z, write_t)\}$

Условия передачи прав доступа или реализации информационных потоков

Определение. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ —состояние системы $\Sigma(G^*, OP)$ и субъект $x \in S_0$, сущность $y \in E_0$, где $x \neq y$, и пусть $\alpha \in R_r$ — некоторое право доступа. Определим предикат $can\_share(\alpha, x, y, G_0)$, который будет истинным тогда, и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$ такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(x, y, \alpha) \in R_N$, где $N \geq 0$.

Определение. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ —состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, где $x \neq y$. Определим предикат $can\_write\_memory(x, y, G_0)$, который будет истинным тогда, и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$ такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(x, y, write_m) \in F_N$, где $N \geq 0$.

Определение. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ —состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, где $x \neq y$. Определим предикат $can\_write\_time(x, y, G_0)$, который будет истинным тогда, и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$ такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(x, y, \alpha_f) \in F_N$, где $\alpha_f \in \{write_m, write_t\}$ и $N \geq 0$.

Условия реализации информационного потока по времени

Теорема. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ —состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, $x \neq y$. Предикат $can\_write\_time(x, y, G_0)$ истинен тогда, и только тогда, когда существует последовательность сущностей $e_1, \ldots, e_m \in E_0$, где $e_1 = x$, $e_m = y$ и $m \geq 2$, таких, что выполняется одно из условий:

Условие 1. $m = 2$ и $(x, y, \alpha_f) \in F_0$, где $\alpha_f \in \{write_m, write_t\}$.

Условие 2. Для каждого $i = 1, \ldots, m - 1$ выполняется одно из условий:

- $e_i \in S_0$ и $(e_i, e_{i+1}, write_t) \in F_0$;
- $e_i \in S_0$ и истинен предикат $can\_write\_memory(e_i, e_{i+1}, G_0)$;
- $e_i \in S_0$ и существует сущность $e'_{i+1} \in E_0$, такая, что $e'_{i+1} \leq e_{i+1}$ и истинен предикат $can\_share(\alpha_r, e_i, e'_{i+1}, G_0)$, $\alpha_r \in R_r$;
- $e_{i+1} \in S_0$ и истинен предикат $can\_share(read_r, e_{i+1}, e_i, G_0)$;
- $e_i, e_{i+1} \in S_0$ и существуют сущности $e'_i, e'_{i+1} \in E_0$: или $e'_i \leq e'_{i+1}$, или $e'_{i+1} \leq e'_i$, и истинны предикаты $can\_share(\alpha_r, e_i, e'_i, G_0)$ и $can\_share(\beta_r, e_{i+1}, e'_{i+1}, G_0)$, $\alpha_r, \beta_r \in R_r$.

Фрагменты доказательства теоремы

Пример задачи на преобразование графа доступов расширенной модели *Take-Grant*

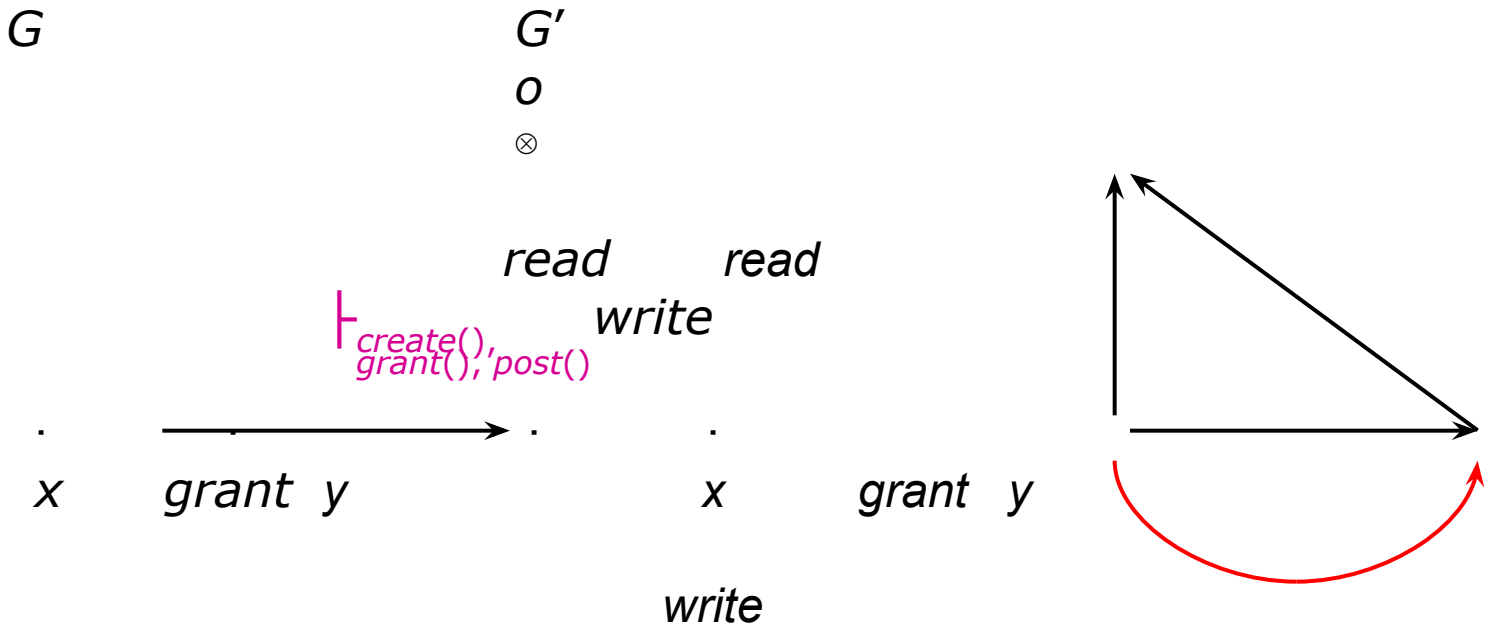


Рис. 3. Реализация информационного потока на запись

Фрагменты доказательства теоремы

Докажем достаточность выполнения условия теоремы для истинности предиката *can_write_time*(x, y, G_0).

…

Пусть выполняется условие 2 теоремы, тогда существует последовательность сущностей e_1, …, $e_m \in E_0$, где $e_1 = x$, $e_m = y$ и $m \geq 2$. Выполним доказательство индукцией по длине m последовательности сущностей.

Пусть $m = 2$. Возможны пять случаев.

…

Рассмотрим четвертый случай: $x \in S_0$ и существует сущность $y' \in E_0$, такая, что $y' \leq y$ и истинен предикат *can_share*(α_r, x, y', G_0), где $\alpha_r \in R_r$. Так как истинен предикат *can_share*(α_r, x, y', G_0), то существуют состояния G_1, …, $G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний op_1, …, op_N такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \dots \vdash_{opN} G_N$ и $(x, y', \alpha_r) \in R_N$, где $N \geq 0$.

Если $\alpha_r \in \{write_r, append_r, read_r\}$, то пусть $op_{N+1} = access\_\alpha(x, y')$ и $G_N \vdash_{op(N+1)} G_{N+1}$, где $G_{N+1} = (S_{N+1}, E_{N+1}, R_{N+1} \cup A_{N+1} \cup F_{N+1}, H_{N+1})$. Тогда $(x, y, write_t) \in F_{N+1}$ и предикат *can_write_time*(x, y, G_0) истинен.

Если $\alpha_r = own_r$, то пусть $op_{N+1} = own\_take(write_r, x, y')$, $op_{N+2} = access\_write(x, y')$ и $G_N \vdash_{op(N+1)} G_{N+1} \vdash_{op(N+2)} G_{N+2}$, где $G_{N+2} = (S_{N+2}, E_{N+2}, R_{N+2} \cup A_{N+2} \cup F_{N+2}, H_{N+2})$. Тогда $(x, y, write_t) \in F_{N+2}$ и предикат *can_write_time*(x, y, G_0) истинен.

Если $\alpha_r = execute_r$, то пусть $op_{N+1} = create\_subject(x, y', z)$ и $G_N \vdash_{op(N+1)} G_{N+1}$, где $G_{N+1} = (S_{N+1}, E_{N+1}, R_{N+1} \cup A_{N+1} \cup F_{N+1}, H_{N+1})$ и $z \notin S_N$. Тогда $(x, y, write_t) \in F_{N+1}$ и предикат *can_write_time*(x, y, G_0) истинен.

Фрагменты доказательства теоремы

Докажем необходимость выполнения условия теоремы для истинности предиката *can_write_time*(x, y, G_0).

Пусть истинен предикат *can_write_time*(x, y, G_0), при этом по определению существуют состояния G_1, …, $G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний op_1, …, op_N такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(x, y, \alpha_f) \in F_N$, где $\alpha_f \in \{write_m, write_t\}$ и $N \geq 0$. Среди всех таких последовательностей выберем ту, у которой длина N является минимальной. В этом случае $(x, y, \alpha_f) \notin F_{N-1}$. Проведем доказательство индукцией по длине N последовательности преобразований.

Пусть $N = 1$, тогда $(x, y, \alpha_f) \notin F_0$ и существует правило преобразования состояний op_1, такое, что $G_0 \vdash_{op1} G_1$ и $(x, y, \alpha_f) \in F_1$. Из определения правил преобразования состояний следует, что возможны девять случаев:

- $x, y \in S_0$, $(x, y, own_r) \in R_0$;
- $x, y \in S_0$, $(y, x, own_r) \in R_0$;
- $x \in S_0$, $y \in E_0$ и существует сущность $y' \in E_0$, такая, что $y' \leq y$ и $(x, y', \alpha_r) \in R_0$, при этом $\alpha_r \in \{read_r, write_r, append_r\}$;
- $x, y \in S_0$ и существуют сущности x', $y' \in E_0$, такие, что или $x' \leq y'$, или $y' \leq x'$ и $\{(x, x', \alpha_r), \{(y, y', \beta_r)\} \subset R_0$, где $\alpha_r, \beta_r \in R_r$;
- $y \in S_0$, $x \in E_0$, $(y, x, read_r) \in R_0$;
- $x \in S_0$, $y \in E_0$, $(x, y, \alpha_r) \in R_0$, где $\alpha_r \in \{write_r, append_r\}$;
- $x \in S_0$, $y \in E_0$ и существует сущность $e \in E_0$, такая, что $\{(x, e, \alpha), (e, y, \beta)\} \subset R_0 \cup F_0$, где $\alpha, \beta \in \{write_r, append_r, write_m, write_t\}$;
- $x, y \in S_0$ и существуют сущность $e \in E_0$, такая, что $\{(x, e, \alpha), (y, e, read_r)\} \subset R_0 \cup F_0$, где $\alpha \in \{write_r, append_r, write_m, write_t\}$;
- $x, y \in E_0$ и существует субъект $e \in S_0$, такой, что $\{(e, x, read_r), (e, y, \alpha)\} \subset R_0 \cup F_0$, где $\alpha \in \{write_r, append_r, write_m, write_t\}$.

ДП-модель без кооперации доверенных и недоверенных субъектов

Определение. Назовем траекторию функционирования системы Σ(*G**, *OP*) траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа, если при ее реализации используются монотонные правила преобразования состояний, и доверенные субъекты:

- не дают недоверенным субъектам права доступа к сущностям;
- не берут у недоверенных субъектов права доступа к сущностям.

Определение. Назовем траекторию функционирования системы Σ(G*, OP) траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков, если она является траекторий без кооперации доверенных и недоверенных субъектов для передачи прав доступа и при ее реализации используются правила преобразования состояний:

- *take_right*(α_r, *x*, *y*, *z*), *grant_right*(α_r, *x*, *y*, *z*), *own_take*(α_r, *x*, *y*, *z*) с условиями и результатами применения определенными в рамках базовой ДП-модели;
- *create_entity*(*x*, *y*, *z*), *create_subject*(*x*, *y*, *z*), *rename_entity*(*x*, *y*, *z*), *access_read*(*x*, *y*), *access_write*(*x*, *y*), access_append(*x*, *y*), *flow*(*x*, *y*, *y'*, *z*), *find*(*x*, *y*, *z*), *post*(*x, y, z*), *pass*(*x, y, z*) с условиями и результатами применения определенными в рамках ДП-модели без кооперации доверенных и недоверенных субъектов.

Правила преобразования состояний

Предположение. Доверенные субъекты системы $\Sigma(G^*, OP)$ не участвуют в реализации информационных потоков по времени.

Базовая ДП-модель		
$find(x, y, z)$	$x, y \in S, z \in E, x \neq z,$ $\{(x, y, \alpha), (y, z, \beta)\} \subset R \cup F,$ где $\alpha, \beta \in \{write_r, append_r, write_m, write_t\}$	$S' = S, E' = E, R' = R, A' = A, H' = H,$ если $\alpha, \beta \in \{write_r, append_r, write_m\},$ то $F' = F \cup \{(x, z, write_m)\},$ если $write_t \in \{\alpha, \beta\},$ то $F' = F \cup \{(x, z, write_t)\}$
ДП-модель без кооперации доверенных и недоверенных субъектов		
$find(x, y, z)$	$x, y \in S, z \in E, x \neq z,$ $\{(x, y, \alpha), (y, z, \beta)\} \subset R \cup F,$ где $\alpha, \beta \in \{write_r, append_r, write_m, write_t\}$	$S' = S, E' = E, R' = R, A' = A, H' = H,$ если $\alpha, \beta \in \{write_r, append_r, write_m\},$ то $F' = F \cup \{(x, z, write_m)\},$ если $write_t \in \{\alpha, \beta\}$ и $x, y \in N_S \cap S,$ то $F' = F \cup \{(x, z, write_t)\},$ если $write_t \in \{\alpha, \beta\}$ и $\{x, y\} \cap (L_S \cap S) \neq \varnothing,$ то $F' = F$

ДП-модель с блокирующими доступами доверенных субъектов

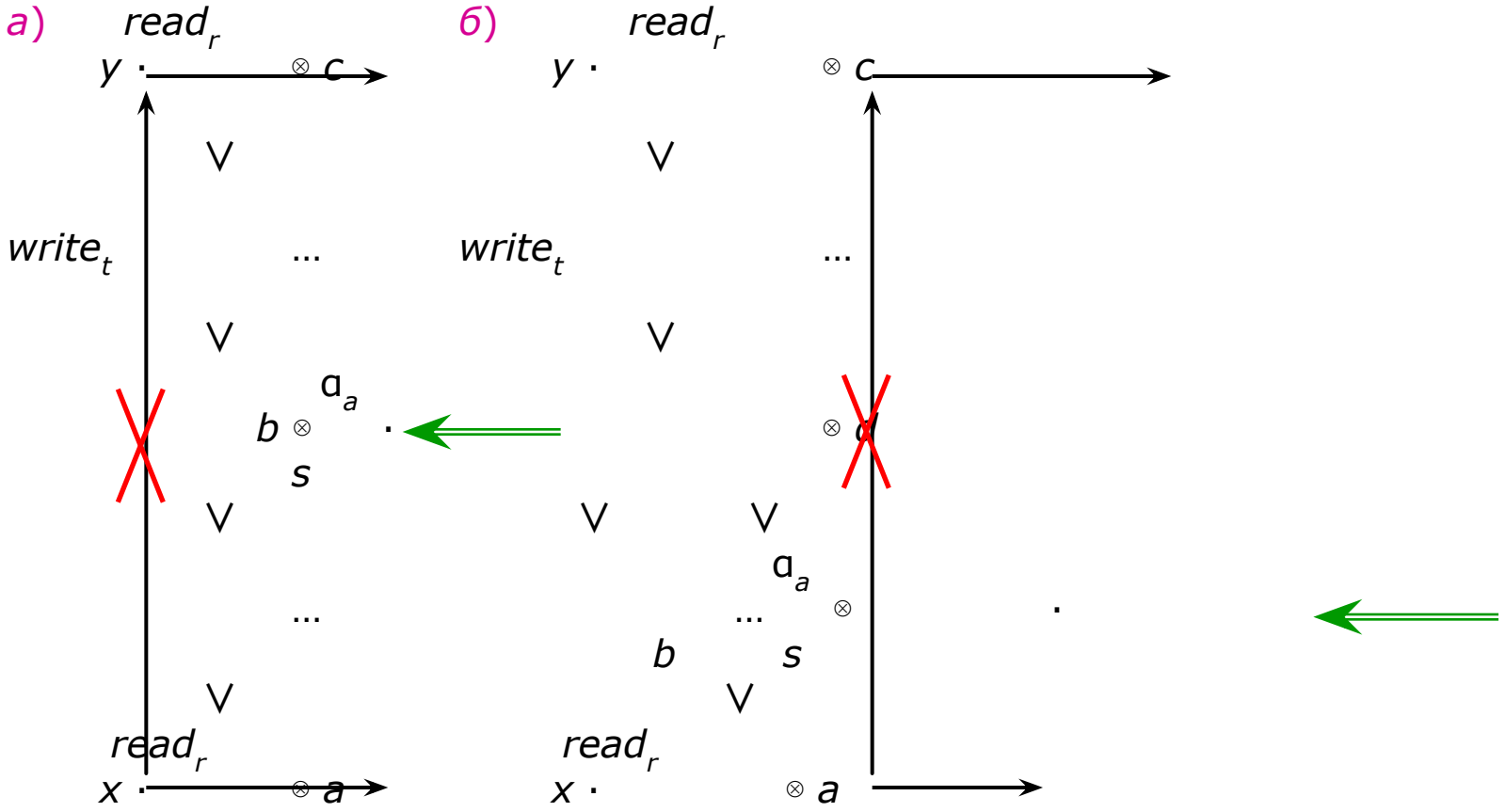


Рис. 4. Примеры, когда невозможна реализация информационного потока по времени с использованием отношения иерархии сущностей

ДП-модель с блокирующими доступами доверенных субъектов

Предположение. Блокирующие доступы доверенных субъектов к сущностям системы $\Sigma(G^*, OP)$ препятствуют реализации информационных потоков по времени с использованием данных сущностей и иерархии сущностей, в которую входит данная сущность, за исключением случая, когда сущность является субъектом и участвует в реализации данного информационного потока по времени.

Определение. Иерархией сущностей с учетом блокирующих доступов доверенных субъектов называется заданное на множестве сущностей E отношение частичного порядка «$\le_B$», удовлетворяющее условиям:

- для двух сущностей $e_1, e_2 \in E$ если выполняется неравенство $e_1 \le_B e_2$, то выполняется неравенство $e_1 \le e_2$;
- если сущность $e_1 \in E_B$, то выполняется равенство $\{e \in E: e \le_B e_1 \text{ или } e_1 \le_B e\} = \{e_1\}$ (каждая сущность из E_B сравнима только сама с собой);
- если для сущности $e_1 \in E$ существует сущность $e_2 \in E_B$, такая, что выполняется условие $e_2 \le e_1$, то выполняется равенство $\{e \in E: e \le_B e_1 \text{ или } e_1 \le_B e\} = \{e_1\}$;
- если для сущности $e \in E$ существуют сущности $e_1, e_2 \in E$, такие, что $e \le_B e_2$, $e \le_B e_1$, то или $e_1 \le_B e_2$, или $e_2 \le_B e_1$.
- В случае, когда для двух сущностей $e_1, e_2 \in E$ выполняются условия $e_1, e_2 \in E \setminus E_B$, $e_1 \le_B e_2$ и $e_1 \ne e_2$, будем использовать обозначение: $e_1 <_B e_2$.

Определение. Определим $H_B: E \to 2^E$ —функцию иерархии сущностей с учетом блокирующих доступов доверенных субъектов, сопоставляющую каждой сущности $c \in E$ множество сущностей $H_B(c) \subset H(c)$, удовлетворяющих условию $H_B(c) = \{e \in H(c): e <_B c\}$.

Условия реализации информационного потока по времени

Теорема. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_{B0})$ —состояние системы $\Sigma(G^*, OP)$ и сущности $x, y \in E_0$, $x \neq y$. Пусть также в G_0 отсутствуют информационные потоки по времени, исходящие из доверенных субъектов: $F_0 \cap \{(s, e, write_t): s \in L_S \cap S_0, e \in E_0\} = \varnothing$. Предикат $can\_write\_time\_block(x, y, G_0, L_S)$ истинен тогда, и только тогда, когда выполняется одно из условий:

Условие 1. В состоянии G_0 реализован информационный поток $(x, y, \alpha_f) \in F_0$, где $\alpha_f \in \{write_m, write_t\}$.

Условие 2. Существует последовательность сущностей $e_1, \ldots, e_m \in E_0$, где $e_1 = x$, $e_m = y$ и $m \geq 2$, таких, что для каждого $i = 1, \ldots, m - 1$ выполняется одно из условий:

- $e_i \in N_S \cap S_0$ и или $(e_i, e_{i+1}, \alpha_f) \in F_0$, где $\alpha_f \in \{write_m, write_t\}$, или истинен предикат $can\_share(\alpha_r, e_i, e_{i+1}, G_0, L_S)$, где $\alpha_r \in \{write_r, append_r\}$;
- $e_i \in N_S \cap S_0$, $e_{i+1} \in E_0 \setminus E_B$ и существует сущность $e'_{i+1} \in E_0 \setminus E_B$, такая, что $e'_{i+1} \leq_B e_{i+1}$, и истинен предикат $can\_share(\alpha_r, e_i, e'_{i+1}, G_0, L_S)$, где $\alpha_r \in R_r$;
- $e_{i+1} \in N_S \cap S_0$ и истинен предикат $can\_share(read_r, e_{i+1}, e_i, G_0, L_S)$;
- $e_i, e_{i+1} \in N_S \cap S_0$ и существуют сущности $e'_i, e'_{i+1} \in E_0 \setminus E_B$, такие, что или $e'_i \leq_B e'_{i+1}$, или $e'_{i+1} \leq_B e'_i$, и истинны предикаты $can\_share(\alpha_r, e_i, e'_i, G_0, L_S)$ и $can\_share(\beta_r, e_{i+1}, e'_{i+1}, G_0, L_S)$, где $\alpha_r, \beta_r \in R_r$;
- $e_i, e_{i+1} \in N_S \cap S_0$ и или истинен предикат $can\_share(own_r, e_i, e_{i+1}, G_0, L_S)$, или истинен предикат $can\_share(own_r, e_{i+1}, e_i, G_0, L_S)$.

Метод предотвращения запрещенных информационных потоков по времени

Метод 1. Условие применения метода. Пусть определена система $\Sigma(G^*, OP, G_0)$ с начальным состоянием $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ и определены множества доверенных субъектов L_S, недоверенных субъектов N_S и запрещенных информационных потоков N_f.

Шаг 1. Создать в системе доверенный субъект $u \in L_S \cap S_0$, который будет реализовывать блокирующие доступы.

Шаг 2. Для каждого запрещенного информационного потока по времени $(x, y, write_t)$, где x и y — недоверенные субъекты, использовать алгоритм проверки истинности предиката $can\_write\_time\_block(x, y, G_0, L_S)$. Если предикат $can\_write\_time\_block(x, y, G_0, L_S)$ является истинным, рассмотреть все пути (пролеты моста) в G_0, с использованием которых возможна реализация информационного потока. Если для всех запрещенных информационных потоков из N_f по времени предикат $can\_write\_time\_block(x, y, G_0, L_S)$ является ложным, то закончить применение метода.

Шаг 3. Для каждого пути, определенного на шаге 2, определить его вид. Возможны два вида путей. Если путь первого вида, то с его использованием реализуется информационный поток по памяти. Если путь второго вида, то с его использованием может быть реализован только информационный поток по времени.

Шаг 4. Если пролет моста первого вида, то удалить любое ребро пути (удалить у недоверенного субъекта право доступа к сущности). Перейти на шаг 2.

Шаг 5. Если путь второго вида реализовать блокирующий доступ доверенного субъекта u к любой сущности, не являющейся субъектом и принадлежащей пути. Перейти на шаг 2.

Теорема. Пусть $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ —начальное состояние системы $\Sigma(G^*, OP, G_0)$. Пусть определены множества доверенных субъектов L_S, недоверенных субъектов N_S и запрещенных информационных потоков N_f. Тогда в результате применения в системе метода 1 для каждого информационного потока $(x, y, write_t) \in N_f$, где x и y —недоверенные субъекты, предикат $can\_write\_time\_block(x, y, G_0, L_S)$ является ложным.

ДП-модель с функционально ассоциированными с субъектами сущностями

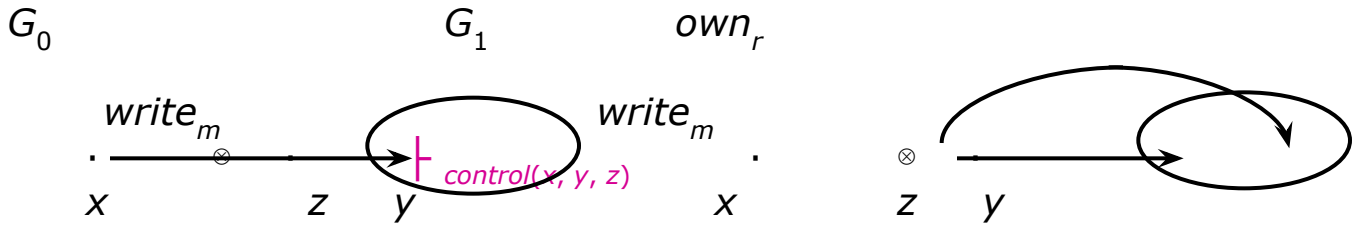


Рис. 5. Пример применения правила *control*(*x*, *y*, *z*)

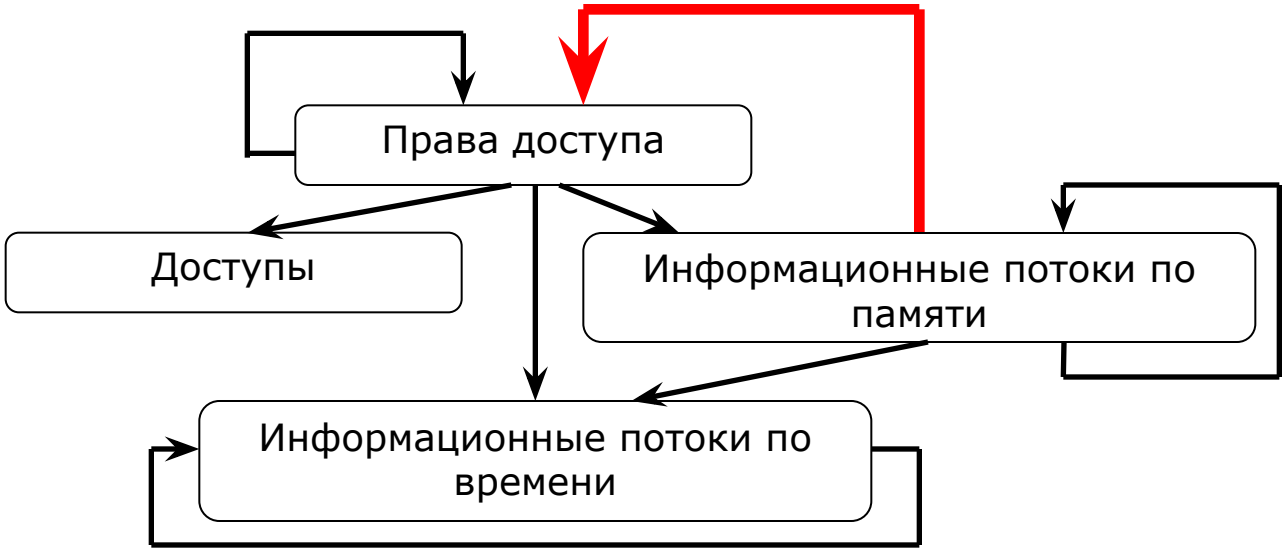


Рис. 6. Связь условий и результатов применения правил преобразования состояний

ДП-модель с функционально ассоциированными с субъектами сущностями

Определение. Доверенного субъекта *y* назовем функционально корректным, если во множество функционально ассоциированным с ним сущностей [*y*] не входят недоверенные субъекты.

Определение. Доверенного субъекта *y* назовем корректным относительно сущности *e*, не являющейся доверенным субъектом, если субъект *y* не реализует информационный поток по памяти от сущности *e* к сущности *e'*, функционально ассоциированной с некоторым доверенным субъектом *y'*. При этом по определению каждый доверенный субъект корректен относительно другого доверенного субъекта.

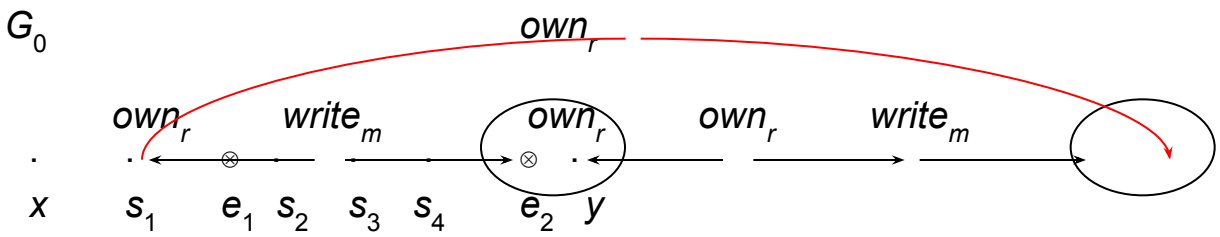


Рис. 7. Пример состояния, в котором возможно получение недоверенным субъектом *x* права доступа владения к доверенному субъекту *y*

ДП-модель для политики безопасного администрирования

Определение. В политике безопасного администрирования рассматриваются две угрозы безопасности КС.

Угроза 1. Наличие у пользователя возможности разместить ресурсы на компьютере (запустить процесс или разместить на компьютере файлы) несет угрозу получения им всех прав доступа и привилегий на данном компьютере.

Угроза 2. Обращение пользователя к компьютеру (с целью получить данные файлов локально или по сетевым коммуникационным каналам) несет угрозу захвата его прав доступа и привилегий пользователем, разместившим свои ресурсы на этом компьютере.

Предположение. Будем считать, что нарушитель имеет два уровня возможностей:

- если нарушитель может разместить на компьютере свой ресурс, то он имеет возможность управления функциями КС (нарушитель третьего уровня);
- если нарушитель может только удаленно обращаться к компьютеру, то он имеет возможность ведения диалога с КС (нарушитель первого уровня).

Определение. Будем говорить, что в КС реализована политика безопасного администрирования тогда, и только тогда, когда в КС выполняется следующее условие: получение нарушителем полного контроля (всех привилегий и прав доступа к сущностям, размещенным на компьютере) над компьютером КС не должно приводить к захвату нарушителем полного контроля над другими компьютерами КС посредством реализации нарушителем угроз 1 и 2.

ДП-модель для политики абсолютного разделения административных и пользовательских полномочий

Определение. Для системы $\Sigma(G^*, OP)$ рассматриваются два возможных начальных состояния системы и два вида траекторий:

- G_0 — начальное состояние для пользовательских траекторий функционирования системы $P(G_0)$, на которых недоверенные пользователи выполняют свои функции в системе;
- GA_0 — начальное состояние для административных траекторий функционирования системы $P(GA_0)$, предназначенных только для администрирования системы.

Предположение. На всех траекториях системы из множеств $P(G_0)$ и $P(GA_0)$ определены доверенные пользователи, обладающие правом доступа владения к каждой сущности, размещенной на компьютере. Кроме того, на траекториях системы из множества $P(G_0)$:

- определен доверенный пользователь $kernel_c \in S$ — пользователь-«ядро ОС», при этом всегда выполняется условие $(kernel_c, own_r) \in RC(c)$;
- всегда найдется недоверенный пользователь, активизировавший процесс от своего имени на компьютере c.

Предположение. Доверенный пользователь $kernel_c$ реализован функционально корректным и корректным относительно всех сущностей $ECE(c)$ компьютера c. На траекториях $P(GA_0)$ активизируют процессы только доверенные пользователи, которые являются корректными относительно всех сущностей $ECE(c)$ компьютера c. Если в системе $\Sigma(G^*, OP)$ существует доверенный пользователь $s_1 \in S \setminus \{kernel_c\}$, который в состоянии $G = (S, E, R \cup A \cup F, H)$ системы активизировал процесс от своего имени на компьютере $c \in EC$, то выполняются условия $(s_1, c, write_r) \in R$, $ECE(c) \subset [s_1]$. При этом, если недоверенный пользователь $s_2 \in S$ в состоянии G системы активизировал процесс от своего имени на компьютере c, то выполняется условие $s_2 \in [s_1]$.

Методы предотвращения возможности получения права доступа владения

- Метод предотвращения возможности получения права доступа владения недоверенным субъектом к доверенному субъекту с использованием реализации информационного потока по памяти к сущности, функционально ассоциированной с доверенным субъектом;
- Метод реализации политики безопасного администрирования (для распределенных КС);
- Метод реализации политики абсолютного разделения административных и пользовательских полномочий (для рабочих станций пользователей).

Мандатная ДП-модель. Виды запрещенных информационных потоков

Определение. Определим как запрещенные в КС с мандатным управлением доступом следующие четыре вида информационных потоков (запрещенные информационные потоки из множества N_f).

1. Информационные потоки по памяти и по времени между сущностями одного уровня конфиденциальности (определяются в соответствии с априорно заданной политикой управления доступом и информационными потоками).
2. Информационные потоки по памяти от сущностей с большим уровнем конфиденциальности к сущностям с меньшим уровнем конфиденциальности информации.
3. Информационные потоки по времени от сущностей с большим уровнем конфиденциальности к сущностям с меньшим уровнем конфиденциальности информации.
4. Информационные потоки по памяти от субъектов с низким уровнем доступа к субъектам с высоким уровнем доступа или к сущностям, функционально ассоциированным с субъектами с высоким уровнем доступа.

Мандатная ДП-модель. Основные определения

$(L, \leq)$ —решетка линейно упорядоченных уровней доступа и конфиденциальности;

$ES \subset E \setminus S$ —множество сущностей, которые могут быть применены для создания новых субъектов.

Определим функции:

$f_s: S \to L$ —функция, определяющая уровень доступа каждого субъекта системы $\Sigma(G^*, OP)$;

$f_e: E \setminus S \to L$ —функция, определяющая уровень конфиденциальности каждой сущности системы, не являющейся субъектом, при этом, если для двух сущностей $e_1, e_2 \in E$ выполняется неравенство $e_1 \leq e_2$ (сущность e_1 содержится в контейнере e_2), то по определению выполняется условие $f_e(e_1) \leq f_e(e_2)$;

$CCR: E \setminus S \to \{true, false\}$ —функция, определяющая способ доступа к сущностям, не являющимся субъектами, внутри контейнеров.

Определение. В состоянии $G = (S, E, R \cup A \cup F, H, (f_s, f_e), CCR)$ системы $\Sigma(G^*, OP)$ доступ $(s, e, \alpha) \in A$, где субъект $s \in S$, сущность $e \in E \setminus S$, вид доступа $\alpha \in R_a$, обладает *ss*-свойством, если выполняются условия:

- $f_s(s) \geq f_e(e)$;
- для каждой сущности-контейнера $e' \in E \setminus S$, такой, что $e < e'$ и $CCR(e') = true$, выполняется неравенство $f_s(s) \geq f_e(e')$.

Определение. В состоянии $G = (S, E, R \cup A \cup F, H, (f_s, f_e), CCR)$ системы $\Sigma(G^*, OP)$ доступы $(s, e_1, read_a), (s, e_2, \alpha) \in A$, где субъект $s \in S$, сущности $e_1, e_2 \in E \setminus S$, вид доступа $\alpha \in \{write_a, append_a\}$, обладают *-свойством, если выполняется условие $f_e(e_1) \leq f_e(e_2)$.

Мандатная ДП-модель. Правила преобразования состояний

Правило	Исходное состояние $G = (S, E, R \cup A \cup F, H, (f_s, f_e), CCR)$	Результирующее состояние $G' = (S', E', R' \cup A' \cup F', H', (f_s', f_e'), CCR')$
rename_entity(x, y, z)	$x \in S, y, z \in E \setminus S, y \in H(z)$, $(x, z, write_a) \in A$, не существует сущности $e \in E \setminus S$, такой, что $f_e(y) < f_e(e)$ и $(x, e, read_a) \in A$	$S' = S, E' = E, R' = R, H' = H, f_s' = f_s, f_e' = f_e, CCR' = CCR, A' = A \cup \{(x, y, write_a)\}$, если $x \in N_S \cap S$, то $F' = F \cup \{(x, z, write_t)\} \cup \{(x, e, write_t): e \in E, x \neq e$ и $e \le y\} \cup \{(x, s, write_t): s \in S, x \neq s$ и $(s, e, \alpha_r) \in R$, где $e \in E, e \le y$ и $\alpha_r \in R_r\}$, если $x \in L_S \cap S$, то $F' = F$
access_write(x, y)	$x \in S, y \in E \setminus S, f_s(x) \ge f_e(y)$, для каждой сущности-контейнера $y' \in E \setminus S$, такой, что $y < y'$ и $CCR(y') = true$, выполняется неравенство $f_s(x) \ge f_e(y')$, не существует сущности $z \in E \setminus S$, такой, что $f_e(y) < f_e(z)$ и $(x, z, read_a) \in A$	$S' = S, E' = E, R' = R, H' = H, f_s' = f_s, f_e' = f_e, CCR' = CCR, A' = A \cup \{(x, y, write_a)\}$, если $x \in N_S \cap S$, то $F' = F \cup \{(x, y, write_m)\} \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \le e\}$, если $x \in L_S \cap S$, то $F' = F \cup \{(x, y, write_m)\}$
flow(x, y, y', z)	$x, z \in S, y, y' \in E, x \neq z$, $\{(x, y, \alpha_a), (z, y', \beta_a)\} \subset A$, где $y \le y'$, $\alpha_a, \beta_a \in R_a$	$S' = S, E' = E, R' = R, A' = A, H' = H, f_s' = f_s, f_e' = f_e, CCR' = CCR$, если $x, z \in N_S \cap S$, то $F' = F \cup \{(x, z, write_t), (z, x, write_t)\}$, если $\{x, z\} \cap (L_S \cap S) \neq \varnothing$, то $F' = F$
find(x, y, z)	$x, y \in S, z \in E, x \neq z$, $\{(x, y, \alpha), (y, z, \beta)\} \subset A \cup F$, где $\alpha, \beta \in \{write_a, append_a, write_m, write_t\}$	$S' = S, E' = E, R' = R, A' = A, H' = H, f_s' = f_s, f_e' = f_e, CCR' = CCR$, если $\alpha, \beta \in \{write_a, append_a, write_m\}$, то $F' = F \cup \{(x, z, write_m)\}$, если $write_t \in \{\alpha, \beta\}$ и $x, y \in N_S \cap S$, то $F' = F \cup \{(x, z, write_t)\}$, если $write_t \in \{\alpha, \beta\}$ и $\{x, y\} \cap (L_S \cap S) \neq \varnothing$, то $F' = F$
control(x, y, z)	$x, y \in S, z \in E, z \in [y], f_s(x) < f_s(y)$ и или $x = z$, или $(x, z, write_m) \in F$	$S' = S, E' = E, R' = R, A' = A, H' = H, F' = F, f_e' = f_e, CCR' = CCR, f_s'(x) = f_s(y)$, для $s \in S \setminus \{x\}$ выполняется равенство $f_s'(s) = f_s(s)$

Условия несанкционированного повышения уровня доступа субъекта

Теорема. Пусть уровень доступа $l \in L$ и $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0, (f_{s0}, f_{e0}), CCR_0)$ —состояние системы $\Sigma(G^*, OP)$, которое безопасно в смысле Белла-ЛаПадула и в котором существует недоверенный субъект $x \in N_S \cap S_0$, и $A_0 = F_0 = \varnothing$. Предикат *can_increase_level*(x, G_0) является истинным тогда, и только тогда, когда для $l = f_{s0}(x)$ существует субъект $y \in S_0 \setminus N_S(l)$, такой, что $f_{s0}(y) > l$ и выполняется одно из условий:

Условие 1. Субъект y функционально некорректен относительно субъекта x.

Условие 2. Существует субъект $y' \in S_0$ такой, что $f_{s0}(y') > l$, и субъект y некорректен относительно сущности x и субъекта y'.

Условие 3. Существует сущность $e \in E_0$ такая, что $e \in [y]$, $f_{e0}(e) \le l$, и для каждой сущности-контейнера $e' \in E_0$ такой, что $e < e'$ и $CCR_0(e') = true$, выполняется неравенство $f_{e0}(e') \le l$.

Условие 4. Существуют сущность $e \in E_0$ и субъект $y' \in S_0$ такие, что $f_{e0}(e) \le l$, $f_{s0}(y') > l$, и для каждой сущности-контейнера $e' \in E_0$ такой, что $e < e'$ и $CCR_0(e') = true$, выполняется неравенство $f_{e0}(e') \le l$, и субъект y некорректен относительно сущности e и субъекта y'.

Информационные потоки по времени

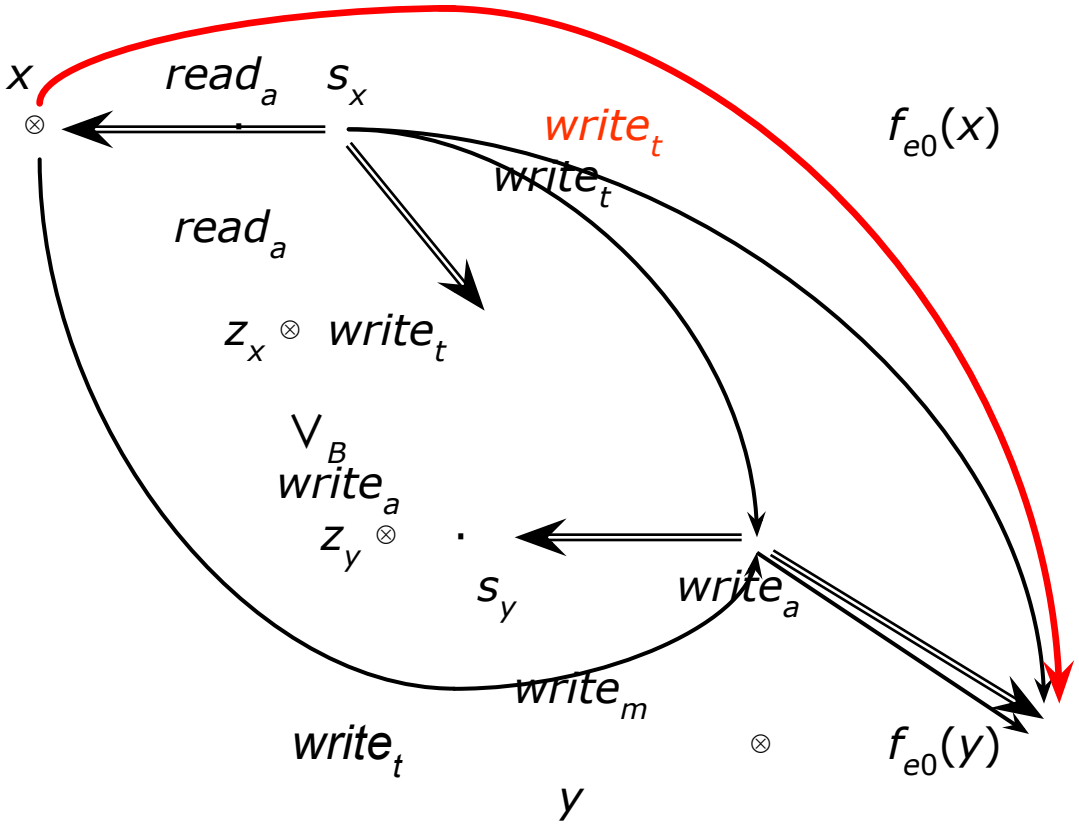


Рис. 8. Пример доступов и информационных потоков в системе с мандатным управлением доступом

Мандатные ДП-модели

- Мандатная ДП-модель с блокирующими доступами доверенных субъектов;
- ДП-модель с отождествлением порожденных субъектов (в системе существует только один недоверенный субъект, от имени которого порождены все другие недоверенные субъекты, и доступы всех недоверенных субъектов рассматриваются как доступы одного субъекта);
- ДП-модель для политики строгого мандатного управления доступом (в системе могут существовать несколько недоверенных субъектов, каждый из которых реализует доступы к сущностям только одного уровня конфиденциальности).

Метод предотвращения возможности реализации запрещенных информационных потоков по времени

Метод. *Условие применения метода.* Пусть определена система $\Sigma(G^*, OP, G_0)$ с начальным состоянием $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_{B0}, (f_{s0}, f_{e0}), CCR_0)$ и $A_0 = A_B$, $F_0 = \varnothing$.

Шаг 1. Выделить в системе $\Sigma(G^*, OP, G_0)$ компьютеры $EC1$, функциональность которых ограничена.

Шаг 2. Для компьютеров из $EC1$ обеспечить доверенность всех активизируемых на них субъектов. Реализовать систему управления доступом системы таким образом, чтобы обеспечивалась безопасность в смысле Белла-ЛаПадула доступов доверенных субъектов.

Шаг 3. Выделить компьютеры $EC2$, на каждом из которых должна быть обеспечена возможность активизации субъектов от имени только одного недоверенного субъекта-пользователя.

Шаг 4. Если это необходимо для реализации используемой в моделируемой реальной КС технологии обработки информации, то для каждого недоверенного субъекта-пользователя компьютера из $EC2$, во множество доступных ему сущностей включить сущности, размещенные на компьютерах из $EC1$. При этом реализовать иерархию подчиненности сущностей системы, обеспечивающую невозможность доступов недоверенных субъектов к сущностям в одной иерархии.

Шаг 5. Реализовать систему управления доступом системы $\Sigma(G^*, OP, G0)$ таким образом, чтобы обеспечивалась безопасность доступов доверенных субъектов компьютеров из $EC2$ в смысле Белла-ЛаПадула, доступов недоверенных субъектов, активизированных от имени субъектов-пользователей компьютеров из $EC2$, в смысле Белла-ЛаПадула с отождествлением порожденных субъектов.

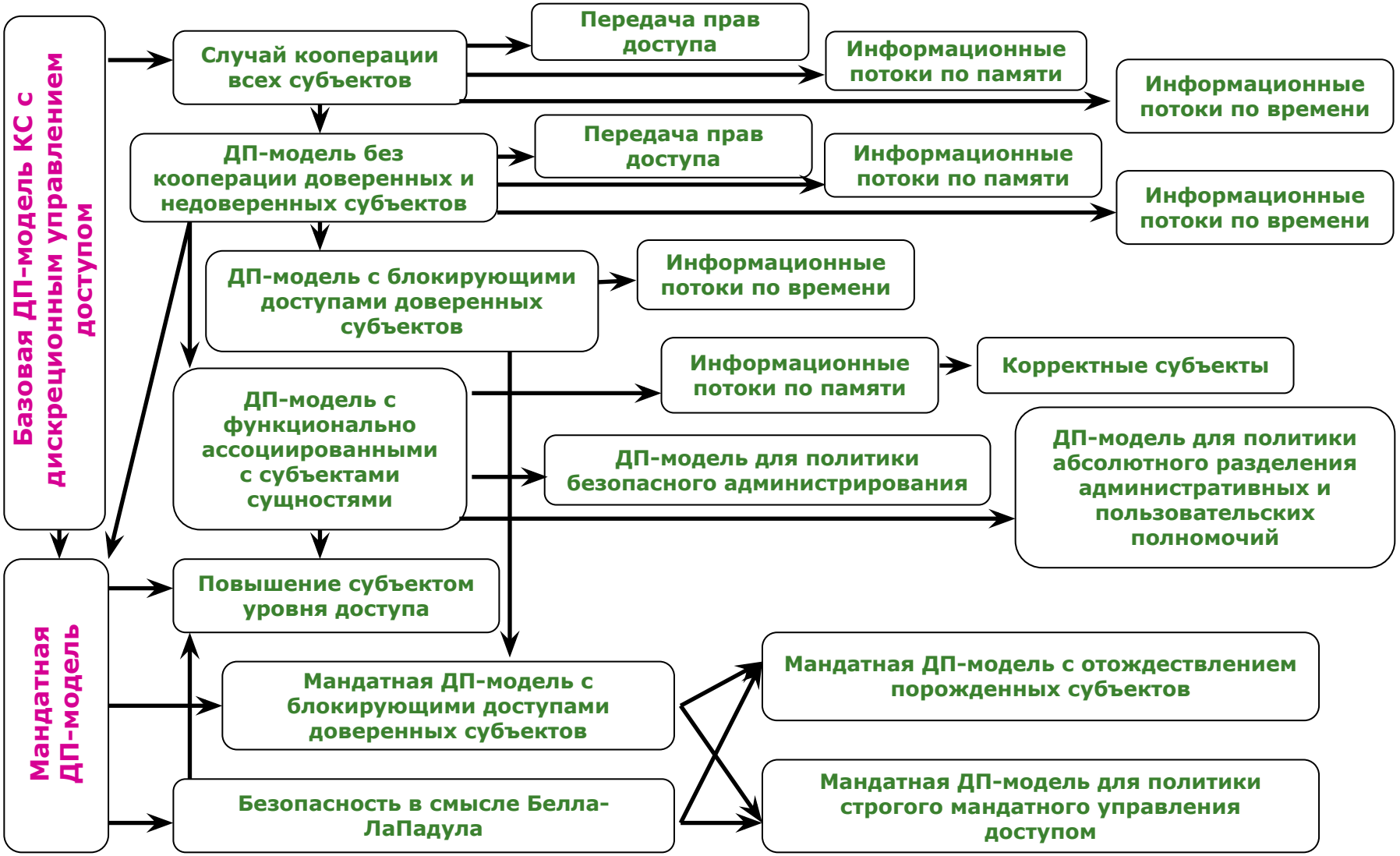
Шаг 6. Выделить компьютеры $EC3$, на каждом из которых должна быть обеспечена возможность активизации субъектов несколькими недоверенными субъектами-пользователями. При этом реализовать иерархию подчиненности компьютеров системы, удовлетворяющую условию: для любых $c_1 \in EC2$, $c_2 \in EC3$ не выполняется одно из сравнений: $c_1 < c_2$ или $c_2 < c_1$.

Шаг 7. Если это необходимо для реализации используемой в моделируемой реальной КС технологии обработки информации, то для каждого недоверенного субъекта-пользователя $user \in N_S$, активизирующего субъектов на компьютерах из $EC3$, во множество сущностей $EU(user)$ включить сущности, размещенные на компьютерах из $EC1$. При этом реализовать иерархию подчиненности сущностей системы, обеспечивающую невозможность доступов недоверенных субъектов к сущностям в одной иерархии.

Шаг 8. Реализовать систему управления доступом системы $\Sigma(G^*, OP, G_0)$ таким образом, чтобы обеспечивалась безопасность доступов доверенных субъектов компьютеров из $EC3$ в смысле Белла-ЛаПадула, недоверенных субъектов-пользователей компьютеров из $EC3$ в смысле строгого мандатного управления доступом.

Шаг 9. Реализовать всех доверенных субъектов системы $\Sigma(G^*, OP, G_0)$ таким образом, чтобы на всех траекториях функционирования системы доверенные субъекты не создавали недоверенных субъектов.

Семейство ДП-моделей



Сравнительный анализ возможностей применения моделей безопасности в современных КС

Особенности функционирования современных КС	Модель *Take-Grant*	Модель Белла-ЛаПадула	Модель СВС	Модель ИПС	ДП-модели
Различие в условиях реализации информационных потоков по памяти и по времени	—	—	—	—	+
Наличие иерархической структуры сущностей и возможность ее использования при реализации информационных потоков по времени	—	—	+	—	+
Возможность кооперации части субъектов при передаче прав доступа или реализации информационных потоков	+	—	—	—	+
Возможность реализации доверенных и недоверенных субъектов с различными условиями функционирования	—	+	—	+	+
Возможность противодействия доверенными субъектами передаче прав доступа или реализации информационных потоков недоверенными субъектами	+	—	—	—	+
Возможность изменения функциональности субъекта при реализации информационного потока по памяти на функционально ассоциированные с ним сущности	—	—	—	+	+
Необходимость определения различных правил управления доступом и информационными потоками для распределенных компонент КС	—	—	—	—	+