Partie II Cours 2 : Politique de sécurité

Odile PAPINI

ESIL
Université de la méditerranée
Odile.Papini@esil.univ-mrs.fr
http://odile.papini.perso.esil.univmed.fr/sources/SSI.html

Plan du cours

1. Introduction
2. Définition d'une politique de sécurité
3. Mise en oeuvre d'une politique de sécurité
4. Validation d'une politique de sécurité
5. Gestion de la continuité d'activité
6. Formalisation de politiques de sécurité

Définition d'une politique de sécurité

politique de sécurité :

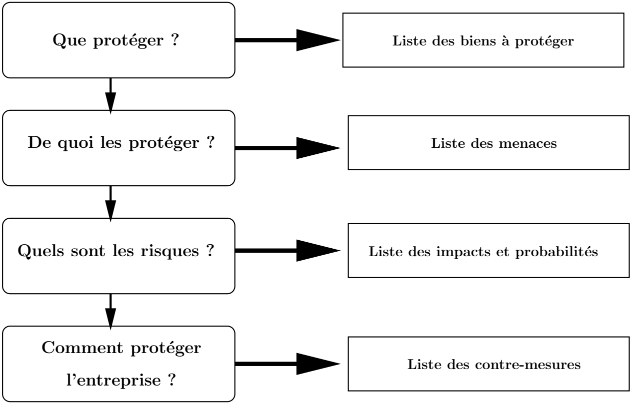
ensemble de lois, de règles et de pratiques qui régissent la façon dont l'information sensible et les autres ressources sont :

- **gérées**
- **protégées**
- **distribuées**

à l'intérieur d'un système d'informations

Définition d'une politique de sécurité

Sécurité



Définition d'une politique de sécurité

objectifs d'une politique de sécurité

protéger le SI contre les menaces identifiées par l'analyse de risques

plus précisémment définir :

- **les objectifs de sécurité** décrivant les propriétés de :
  - confidentialité
  - intégrité
  - disponibilité
- **l'état du système** où ces propriétés sont vérifées
- **des règles de sécurité** décrivant les moyens de modifier l'état de sécurité du système

Définition d'une politique de sécurité

les politiques de sécurité sont classées en 3 catégories

- les politiques de sécurité **internes**
- les politiques de sécurité **techniques**
- les politiques de sécurité **système**

Définition d'une politique de sécurité

les politiques de sécurité internes

aspects organisationnels

procédures sur la :

- **répartition** des tâches et responsabilités entre utilisateurs
- **limitation** du cumul de pouvoir
- **séparation** de pouvoir dans une organisation

Définition d'une politique de sécurité

les politiques de sécurité techniques

aspects matériels et logiciels

procédures sur :

- **le vol**
- **les catastrophes naturelles**
- **le feu, · · ·**

Définition d'une politique de sécurité

les politiques de sécurité système

spécifient l'ensemble des lois, des règlements et pratiques qui régissent la façon de :

- **gérer**
- **protéger**
- **diffuser**

les informations et les autres ressources sensibles au sein du SI

Définition d'une politique de sécurité

les politiques de sécurité système s'appuie sur

- **politique d'identification :** identifier de manière unique chaque utilisateur
- **politique d'authentification :** permet à l'utilisateur de prouver son identité
- **politique d'autorisation :** détermine les opérations légitimes qu'un utilisateur peut réaliser

Définition d'une politique de sécurité

Une politique de sécurité doit être **bien définie**

- **cohérente**
- **complète**

Une politique de sécurité doit être **mise en application :**

- **sensibilisation**
- **simplicité (définition et implantation)**

Définition d'une politique de sécurité

Une politique de sécurité doit être **un document de référence** adopté par tous

Différentes méthodes :

- **MARION** (Méthodologie d'Analyse de Risques Informatiques Orientée par Niveaux)
  https ://www.clusif.asso.fr/fr/production/mehari/
- **MEHARI** (Méthodologie Harmonisée d'Analyse de RIsques)
  https ://www.clusif.asso.fr/fr/production/mehari/
- **EBIOS** (Expression des Besoins et Identification des Objectifs de Sécurité)
  http ://www.ssi.gouv.fr/fr/confiance/ebios.html

Mise en oeuvre d'une politique de sécurité

choix des mécanismes les plus **simples possibles** permettant de protéger les ressources, de manière la plus **efficace** avec un **coût acceptable**

- système d'authentification (biométrie, serveur d'authentification , · · ·)
- chiffrement (PKI, mécanismes intégrés à des protocoles de communication (IPsec), · · ·)
- pare feux (firewall)
- système anti-virus
- outil de détection de failles de sécurité
- système de détection d'intrusions
- système d'exploitation sécurisé
- · · ·

Mise en oeuvre d'une politique de sécurité

validation d'une politique de sécurité

audit de sécurité par un **tiers de confiance**

- **valide** les moyens de protection mis en oeuvre par rapport à la politique de sécurité
- **vérifie que**
  - chaque règle de sécurité est **correctement appliquée**
  - l'ensemble des dispositions forme un tout **cohérent** et **sûr**

Validation d'une politique de sécurité

test d'une politique de sécurité

test d'intrusion : éprouver les moyens de protection d'un SI en essayant de **s'introduire** dans le système en situation réelle

deux méthodes :

- **black box**

  s'introduire dans le système sans aucune connaissance préalable de celui-ci (situation réelle)

- **white box**

  s'introduire dans le système en ayant connaissance de l'ensemble du système (éprouver au maximum le système)

Validation d'une politique de sécurité

test d'une politique de sécurité

- se fait avec **l'accord** de la hierarchie
  - le propriétaire du système doit donner une autorisation
  - dégâts possibles sur le système
- permet de **sensibiliser** le personnel
- ne permet pas de garantir la sécurité du système

Validation d'une politique de sécurité

gestion des incidents

que faire après une attaque ?

- obtention de l'adresse du pirate et riposte ?
- extinction de l'alimentation de la machine ?
- débranchement de la machine du réseau ?
- réinstallation du système ?

plan de continuité de l'activité

Gestion de la continuité d'activité

- **définition des responsabilités** (à l'avance)
- constitution de **preuves sur l'attaque** (en cas d'enquête judiciaire)
- **datation** de l'intrusion (degré de compromission de la machine)
- **confinement** de la compromission (éviter la propagation)
- **sauvegarde** (comparaison des données du système avec la sauvegarde)
- **mise en place d'un plan de repli** (continuité de service)

plan de continuité de l'activité

Gestion de la continuité d'activité

politique de sauvegarde

- **définition des parties du SI à sauvegarder**
- **disponibilité** des sauvegardes
- **politique des supports** de sauvegarde
- **organisation** des sauvegardes
- **replication** sur un site distant
- outil de simulation (ex : http ://www.distrilogie.com)

Formalisation de politiques de sécurité

une **politique de sécurité** spécifie l'ensemble :

$$\left.\begin{array}{l}\textbf{des lois}\\[2em] \textbf{des règlements}\\[2em] \textbf{des pratiques}\end{array}\right\} \text{afin de} \left\{\begin{array}{l}\textbf{gérer}\\[2em] \textbf{protéger}\\[2em] \textbf{diffuser}\end{array}\right\} \text{les informations}$$

Formalisation de politiques de sécurité

Règlement de sécurité a pour objectif de définir les actions que les agents ont :

$$\left.\begin{array}{l} \textit{la permission} \\ \\ \textit{l'obligation} \\ \\ \textit{l'interdiction} \end{array}\right\} \textbf{de réaliser}$$

2 types de contraintes

- **contraintes fonctionnelles :** s'appliquent aux agents lorsqu'ils effectuent des actions portant sur des objets
- **contraintes organisationnelles :** s'appliquent aux agents lorsqu'ils interagissent avec d'autres agents

Formalisation de politiques de sécurité

Contrainte fonctionnelle : règle de la forme :

$$\text{Un agent a} \left\{ \begin{array}{l} \textit{la permission} \\ \textit{l'interdiction} \\ \textit{l'obligation} \end{array} \right\} \text{de} \left\{ \begin{array}{l} \textit{faire telle action} \\ \textit{connaître telle information} \end{array} \right\}$$

nécessité de modéliser les concepts :

- **de permission**
- **d'obligation**
- **d'interdiction**

Formalisation de politiques de sécurité

Exemple de règles de politique de sécurité :

- **R1 :** Any agent playing the role User is permitted to read any public file
- **R2 :** Any agent playing the role User is permitted to write his own public file
- **R3 :** Any agent playing the role User is forbidden to downgrade a file

Formalisation de politiques de sécurité

Représentation formelle de politiques de sésurité : approches logiques :

représentation formelle : ensemble de formules

raisonnement : vérification de cohérence, inférence

- logique classique (logique des prédicats $\mathcal{L}_{Pr}$)
  - mise en oeuvre du raisonnement : résolution
    implantation : programmation logique
- logiques modales
  - mise en oeuvre du raisonnement : tableaux sémantiques
    implantation : systèmes dédiés (FaCT, $\cdots$)

Logique des prédicats (rappel)

Le langage de logique des prédicats

Vocabulaire

- un ensemble infini dénombrable de symboles de prédicats ou **prédicats** (prédicats d'arité 0 : propositions)
- un ensemble infini dénombrable de symboles **fonctionnels** ou fonctions (fonctions d'arité 0 : constantes 0 ou F ou $\perp$ pour Faux et 1 ou V ou $\perp$ pour Vrai)
- un ensemble infini dénombrables de **variables**
- les connecteurs : $\neg$, $,\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
- les **quantificateurs** $\forall$, $\exists$

Logique des prédicats (rappel)

Définitions

terme

- x une variable , f un symbole fonctionnel est un terme
- si $t_1, \cdots, t_n$ sont des termes alors $f(t_1, \cdots, t_n)$ est un terme

atome

- si $t_1, \cdots, t_n$ sont des termes et P est un prédicat alors $P(t_1, \cdots, t_n)$ est un atome

formules bien formées de la logique des prédicats :

• un atome est une formule

• si A et B sont des formules alors $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ sont des formules

• si A est une formule et x une variable alors $\forall x A$, $\exists x A$ sont des formules

Logique des prédicats (rappel)

Système formel de la logique des prédicats

les axiomes

soit A, B, C des formules de $\mathcal{L}_{Pr}$, x une variable et t un terme, D une formule n'ayant pas x pour variable libre

A1) $(A \rightarrow (B \rightarrow A))$

A2) $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$

A3) $((\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A))$

A4) $(\forall x\ A(x) \rightarrow A(t))$

A5) $((D \rightarrow B) \rightarrow (D \rightarrow \forall x\ B))$

Logique des prédicats (rappel)

règles de déduction

substitutions

modus ponens

$$\frac{\Gamma \vdash A, \;\; \Delta \vdash A \rightarrow B}{\Gamma, \Delta \vdash B}$$

généralisation

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x \; A}$$

Logique des prédicats (rappel)

Déduction

Soit B une formule de $\mathcal{L}_{Pr}$ et $H_1, \cdots, H_n$ **des hypothèses**
Une **déduction** de B à partir des hypothèses $H_1, \cdots, H_n$

$$H_1, \cdots, H_n \vdash B$$

est une suite de formules $F_1, \cdots, F_i, F_n$ telle que :

$F_n = B$

et $F_i, \quad 1 \leq i < n$ est :

- soit une des hypothèses $H_1, \cdots, H_n$
- soit un axiome
- soit obtenue par l'application de règles de déduction à partir de formules $F_j, \quad j < i$

Logique des prédicats (rappel)

sémantique de la logique des prédicats

interprétation : $I = (D, I_c, I_v)$ où

- D ensemble non vide, domaine d'interprétation
- I_c la fonction :

$$\begin{array}{ccc} D^n & \rightarrow & D \\ f & \rightarrow & I_c(f) \end{array} \qquad \begin{array}{ccc} D^m & \rightarrow & \{0,1\} \\ P & \rightarrow & I_c(P) \end{array}$$

- I_v la fonction :

$$\begin{array}{ccc} Var & \rightarrow & D \\ x & \rightarrow & I_v(x) \end{array}$$

Logique des prédicats (rappel)

interprétation d'une formule de la logique des prédicats

A une formule de $\mathcal{L}_{Pr}$, association d'une valeur de vérité $I(A)$ à A

- si x est une variable libre alors $I(x) = I_v(x)$
- $I(f(t_1, \cdots, t_n)) = (I_c(f))(I(t_1), \cdots, I(t_n))$
- $I(P(t_1, \cdots, t_m)) = (I_c(P))(I(t_1), \cdots, I(t_m))$
- si A et B sont des formules alors $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ s'interprètent comme dans la logique propositionnelle
- si A est une formule et x une variable alors $I(\forall x\ A) = 1$ si $I_{x/d}(A) = 1$ pour tout élément $d \in D$
- si A est une formule et x une variable alors $I(\exists x\ A) = 1$ si $I_{x/d}(A) = 1$ pour au moins un élément $d \in D$

Logique des prédicats (rappel)

quelques définitions

Soient $A \in \mathcal{L}_{Pr}$ $\quad B \in \mathcal{L}_{Pr}$ $\quad$ et $\quad \mathcal{F} \subset \mathcal{L}_{Pr}$

A est une **tautologie** , $\models A$, si pour toute interprétation I, $I(A) = 1$

B est une **conséquence** de A si pour toute interprétation I, $I(A) = 1$ alors $I(B) = 1$, on écrit $A \models B$

B est une **conséquence** de $\mathcal{F}$ si pour toute interprétation I, tq $\forall A \in \mathcal{F}$, $I(A) = 1$ alors $I(B) = 1$, on écrit $\mathcal{F} \models B$

Logique des prédicats (rappel)

quelques définitions (suite)

Soient $A \in \mathcal{L}_{Pr}$ $\quad B \in \mathcal{L}_{Pr}$ $\quad$ et $\quad \mathcal{F} \subset \mathcal{L}_{Pr}$

A est **satisfaisable** s'il existe une interprétation I tq $I(A) = 1$

$\mathcal{F}$ est **satisfaisable** s'il existe une interprétation I tq $\forall A \in \mathcal{F}$, $I(A) = 1$

A est **insatisfaisable** ou **incohérente** si pour toute interprétation I, $I(A) = 0$

$\mathcal{F}$ est **insatisfaisable** si pour toute interprétation I, $\exists A \in \mathcal{F}$ tq $I(A) = 0$

Formalisation de politiques de sécurité

traduction de l'ensemble des règles en $\mathcal{F}$ un ensemble de formules de $\mathcal{L}_{Pr}$

- détermination des prédicats
- choix des prédicats pour les permissions, obligations, interdictions
- représentation des contraintes d'intégrité

test de la cohérence

- traduction de $\mathcal{F}$ en un programme logique $\mathcal{P}$
- test de la cohérence de $\mathcal{P}$ avec un moteur d'inférence (par exemple gnu-prolog)

Formalisation de politiques de sécurité

exemple : règle R1

Any agent playing the role User is permitted to read any public file

- prédicats unaires : File, Public
- prédicats binaires : Play, Permitted_read
- constante : User

Formule de $\mathcal{L}_{Pr}$

$\forall f \, \forall A \; File(f) \wedge Public(f) \wedge Play(A, User) \rightarrow Permitted\_read(A, f)$

Exemple de contrainte d'intégrité :

$\forall f \, \forall A \; Permitted\_read(A, f) \leftrightarrow \neg Forbidden\_read(A, f)$

Formalisation de politiques de sécurité

Logiques modales : introduction des modalités $\Box$ et $\Diamond$

lectures de $\Box A$	**lectures de** $\Diamond A$
il est nécessaire que A	il est possible que A
il sera toujours vrai que A	il sera parfois vrai que A
A est obligatoire	A est permis
A est su	l'inverse de A n'est pas su
A est connu	l'inverse de A n'est pas connu
A est cru	l'inverse de A n'est pas cru
toute exécution du programme produit A	il y a une exécution du programme qui produit A

Formalisation de politiques de sécurité

Le langage de logique modale propositionnelle

Vocabulaire

un ensemble infini dénombrable de **propositions**

les constantes : 0 (Faux) et 1 (Vrai)

les connecteurs : $\neg$, $, \wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

les **modalités** : $\Box$, $\Diamond$

Formalisation de politiques de sécurité

Définitions

formules bien formées de la logique modale propositionnelle :

- 0 et 1 sont des formules
- une variable propositionnelle est une formule
- si A et B sont des formules alors
  $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ sont des formules
- si A est une formule alors $\Box A$, $\Diamond A$ sont des formules
- si A est une formule alors $\Diamond A =_{def} \neg \Box \neg A$

Formalisation de politiques de sécurité

Système formel de la logique modale propositionnelle (système K)

les axiomes

soit A, B, C des formules de la logique modale propositionnelle

A1) $(A \rightarrow (B \rightarrow A))$

A2) $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$

A3) $((\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A))$

K) $(\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B))$

Formalisation de politiques de sécurité

règles de déduction

modus ponens

$$\frac{\Gamma \vdash A, \;\; \Delta \vdash A \rightarrow B}{\Gamma, \Delta \vdash B}$$

nécéssité (règle N)

$$\frac{\Gamma \vdash A}{\Gamma \vdash \Box A}$$

Formalisation de politiques de sécurité

K-**dérivation (déduction)**

F : une formule modale, Γ : un ensemble de formules modales

une K-**dérivation** de F à partir de Γ est une séquence de formules se terminant par F, dont chaque formule est :

- soit une axiome
- soit un membre de Γ
- soit obtenu par l'application des règles de substitution, de modus ponens ou de nécessité

une K-**preuve** de F est une K-**dérivation** de F à partir de $\emptyset$: $\vdash F$

Formalisation de politiques de sécurité

règles dérivées :

régularité pour $\Box$ (règle R)

$$\frac{\Gamma \vdash A \rightarrow B}{\Gamma \vdash \Box A \rightarrow \Box B}$$

régularité généralisée pour $\Box$

$$\frac{\Gamma \vdash (A_1 \wedge \cdots \wedge A_n) \rightarrow B}{\Gamma \vdash (\Box A_1 \wedge \cdots \wedge \Box A_n) \rightarrow \Box B}$$

régularité pour $\Diamond$

$$\frac{\Gamma \vdash A \rightarrow B}{\Gamma \vdash \Diamond A \rightarrow \Diamond B}$$

Formalisation de politiques de sécurité

Système formel K : formé à partir des axiomes, A1, A2, A3, K

Système formel KT : formé à partir des axiomes, A1, A2, A3, K et de l'axiome de la **connaisance T) :** $\Box A \rightarrow A$

Système formel KT4 ou S4 : formé à partir des axiomes, A1, A2, A3, K, T et de l'axiome **d'introspection positive 4) :**
$\Box A \rightarrow \Box \Box A$

Système formel KT45 ou S5 : formé à partir des axiomes, A1, A2, A3, K, T, 4 et de l'axiome **d'introspection négative 5) :**
$\diamond A \rightarrow \Box \diamond A$

Formalisation de politiques de sécurité

sémantique de la logique modale propositionnelle

sémantique des "mondes possibles"

une formule modale évaluée dans un "univers" de **mondes possibles**
une **relation d'accessibilité** lie les mondes possibles entre eux

$\Box A$ est vraie dans un monde possible ω si A est vraie dans **tous les mondes possibles** accessibles à partir de ω

$\diamond A$ est vraie dans un monde possible ω si A est vraie dans **au moins un monde possible** accessible à partir de ω

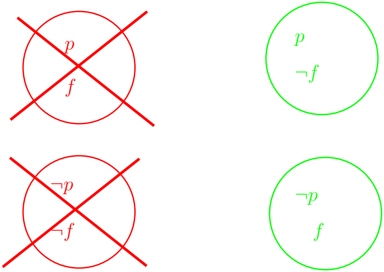
Exemple 1 : on lance une pièce de monnaie

p : "on obtient PILE" f : "on obtient FACE"

p **est possible** f **est possible**

$p \vee f$ **est nécessairement** VRAI

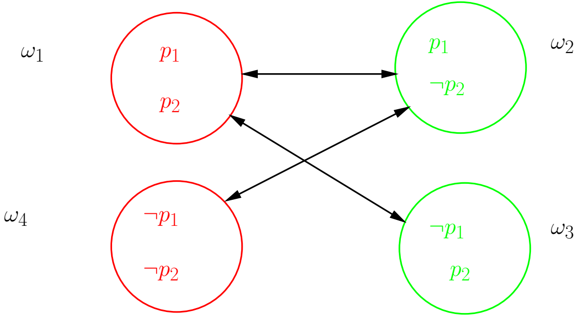
$p \wedge f$ est **impossible** : **nécessairement** FAUX



mondes possibles

Exemple 2 : on lance deux pièces de monnaie 1 et 2

p_i : "on obtient PILE pour i" $\neg p_i$: "on obtient FACE pour i"



Relation d'accèssibilité

$$\omega_i \, \mathcal{R} \, \omega_j \text{ ssi } d_{HAMMING}(\omega_i, \omega_j) = 1$$

Formalisation de politiques de sécurité

sémantique : définitions

système : paire $(\mathcal{W}, \mathcal{R})$ où $\mathcal{W}$ est l'ensemble des interprétations du langage et $\mathcal{R}$ est une relation binaire sur $\mathcal{W}$
$\omega,\ \omega' \in \mathcal{W}, \quad \omega\mathcal{R}\omega' \quad : \quad \omega'$ est accessible à partir de ω

valuation : application v de $\mathcal{W} \times \mathcal{P}$ dans $\{0,\ 1\}$ qui associe une valeur de vérité $v(\omega, p)$ à la variable p dans l'interprétation ω

modèle : triplet $\mathcal{M} = (\mathcal{W}, \mathcal{R}, v)$ où $(\mathcal{W}, \mathcal{R})$ est un système et v une valuation

notation : $\mathcal{M}, \omega \models F$: F est vraie dans le monde possible ω pour le modèle $\mathcal{M}$

Formalisation de politiques de sécurité

définitions

Soit $\mathcal{M} = (\mathcal{W}, \mathcal{R}, v)$ un modèle,
la relation de conséquence est définie par :
$\mathcal{M}, \omega \models p$ ssi $v(\omega, p) = 1$
$\mathcal{M}, \omega \models \top$
$\mathcal{M}, \omega \not\models \bot$
$\mathcal{M}, \omega \models \neg A$ ssi $\mathcal{M}, \omega \not\models A$
$\mathcal{M}, \omega \models A \rightarrow B$ ssi $\mathcal{M}, \omega \not\models A$ ou $\mathcal{M}, \omega \models B$
$\mathcal{M}, \omega \models A \wedge B$ ssi $\mathcal{M}, \omega \models A$ et $\mathcal{M}, \omega \models B$
$\mathcal{M}, \omega \models \Box A$ ssi $\mathcal{M}, \omega' \models A$ pour tout ω' tq $\omega \mathcal{R} \omega'$
$\mathcal{M}, \omega \models \diamond A$ ssi $\mathcal{M}, \omega' \models A$ pour au moins un modèle ω' tq $\omega \mathcal{R} \omega'$

Formalisation de politiques de sécurité

une formule A est valide dans un modèle $\mathcal{M} = (\mathcal{W}, \mathcal{R}, v)$ ssi A est vraie dans tous les mondes possibles du modèle

$$\text{notation} \quad \mathcal{M} \models A$$

une formule A est valide dans un système $(\mathcal{W}, \mathcal{R})$ ssi A est vraie dans tout modèle $\mathcal{M} = (\mathcal{W}, \mathcal{R}, v)$

$$\text{notation} \quad (\mathcal{W}, \mathcal{R}) \models A$$

une formule A est valide (ou est une **tautologie**), ssi A est vraie dans tout système $(\mathcal{W}, \mathcal{R})$

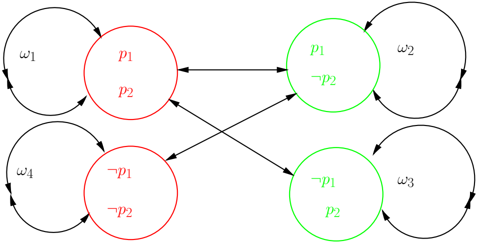
$$\text{notation} \quad \models A$$

Exemple 2 : on lance deux pièces de monnaie 1 et 2

$$\mathcal{M}, \omega_1 \models p_1 \qquad \mathcal{M}, \omega_2 \models p_1 \qquad \mathcal{M}, \omega_3 \not\models p_1$$

$$\mathcal{M}, \omega_1 \models \Box(p_1 \vee p_2) \qquad \mathcal{M}, \omega_2 \not\models \Box(p_1 \vee p_2)$$

$$\mathcal{M}, \omega_2 \models \Diamond(p_1 \vee p_2)$$



Relation d'accèssibilité

$$\omega_i \, \mathcal{R} \, \omega_j \quad \text{ssi} \quad d_{HAMMING}(\omega_i, \omega_j) \leq 1$$

Formalisation de politiques de sécurité

il y a une infinité de logiques modales qui se comportent plus ou moins bien :

- K : logique modale la plus faible, formule caractéristique :

$$\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$$

- T : systèmes réflexifs, $\mathcal{R}$ est réflexive, formule caractéristique :

$$\Box A \rightarrow A$$

- $K4$: systèmes transitifs, $\mathcal{R}$ est transitive, formule caractéristique :

$$\Box A \rightarrow \Box\Box A$$

Formalisation de politiques de sécurité

- $S4$: systèmes réflexifs et transitifs, $\mathcal{R}$ est réflexive et transitive
- KB : systèmes symétriques, $\mathcal{R}$ est symétrique, formule caractéristique :

$$\Box A \rightarrow \Box \diamond A$$

- B : systèmes réflexifs et symétriques, $\mathcal{R}$ est réflexive et symétrique
- $S5$: systèmes réflexifs, symétriques et transitifs $\mathcal{R}$ est une relation d'équivalence, formule caractéristique :

$$\neg\Box A \rightarrow \Box\neg\Box A$$

Formalisation de politiques de sécurité

résultats de correction et de complétude

le système formel K

théorème (de correction) : soit A une formule modale, si $\vdash A$ alors $\models A$
(les formules qui sont des théorèmes du système K sont des tautologies pour la classe K)

théorème (de complétude) : soit A une formule modale, si $\models A$ alors $\vdash A$

Formalisation de politiques de sécurité

résultats de correction et de complétude

système formel T

théorème : $\Box A \rightarrow A$ est une tautologie ssi $\mathcal{R}$ est réflexive

système formel $S4$

théorème : $\Box A \rightarrow \Box\Box A$ est une tautologie ssi $\mathcal{R}$ est transitive

système formel $S5$

théorème : $\neg\Box A \rightarrow \Box\neg\Box A$ est une tautologie ssi $\mathcal{R}$ est euclidienne

Formalisation de politiques de sécurité

quelques résultats de décidabilité

définition : une logique modale $\mathcal{L}$ possède la propriété de **modèle fini** si pour toute formule A qui n'est pas valide dans $\mathcal{L}$, il existe un modèle fini dans lequel A est falsifié

proposition : si une logique modale $\mathcal{L}$ possède une procédure de preuve et la propriété de **modèle fini** alors $\mathcal{L}$ est **décidable**

théorème : K, T, $S4$, $S5$ **sont décidables**

Formalisation de politiques de sécurité

Logique Déontique Standard (SDL)

G von Wright 1951 : introduction des modalités O, P et F

- O : obligation
- P : permission
- F : interdiction

définitions : soit A une formule propositionnelle

$$F\ A =_{def} O\ \neg A$$
$$P\ A =_{def} \neg F\ A$$
$$P\ A =_{def} \neg O\ \neg A$$

Formalisation de politiques de sécurité

Logique Déontique Standard (SDL)

Logique modale de type KD :

Axiome K : $(O(A \rightarrow B) \rightarrow (O\ A \rightarrow O\ B))$

Axiome D : $O\ A \rightarrow \neg O\ \neg A$

Logique Déontique Standard est décidable

Formalisation de politiques de sécurité

Logique Déontique Standard (SDL)

- représentation formelle d'une politique de sécurité
- raisonnement
  - vérification de cohérence d'une politique de sécurité
  - inférence
  - interopérabilité de politiques de sécurité

Des systèmes performants basés sur la **méthode des tableaux sémantiques**

- LoTREC
- TWB
- KSAT
- FaCT

Formalisation de politiques de sécurité

Méthode des tableaux sémantiques

formules modales : préfixées par des étiquettes qui "nomment" le monde dans lequel les formules sont supposées satisfaites.

formule préfixée : σ F (σ est le préfixe)

- $\sigma_0.\sigma_1$: préfixe obtenu par concaténation des préfixes σ_0 et σ_1
- $\sigma.n$: la suite σ suivie de n (n entier)
- $\sigma.n$ nomme un monde accessible par l'un des mondes nommé par σ.

Formalisation de politiques de sécurité

construction d'une preuve de F : construction d'un arbre dont

- **la racine** est étiquetée par la formule préfixée $1\ \neg F$
- les **noeuds** sont étiquetés par des formules préfixées
- les **descendants de noeuds** sont produits soit par des **règles d'expansion** (prolongation, ramification, double négation, nécessité, possibilité)

Formalisation de politiques de sécurité

règles de prolongation

représente la satisfaction de la formule $\sigma\ \alpha = \sigma\ \alpha_1\ \wedge\ \sigma\ \alpha_2$

$$\frac{\sigma\ \alpha}{\begin{array}{c}\sigma\ \alpha_1\\ \sigma\ \alpha_2\end{array}} \qquad \frac{\sigma\ x\wedge y}{\begin{array}{c}\sigma\ x\\ \sigma\ y\end{array}} \qquad \frac{\sigma\ \neg(x\vee y)}{\begin{array}{c}\sigma\ \neg x\\ \sigma\ \neg y\end{array}} \qquad \frac{\sigma\ \neg(x\rightarrow y)}{\begin{array}{c}\sigma\ x\\ \sigma\ \neg y\end{array}} \qquad \frac{\sigma\ x\leftrightarrow y}{\begin{array}{c}\sigma\ x\rightarrow y\\ \sigma\ x\rightarrow y\end{array}}$$

Formalisation de politiques de sécurité

règles de ramification

représente la satisfaction de la formule $\sigma\ \beta = \sigma\ \beta_1 \ \vee\ \sigma\ \beta_2$

$$\frac{\sigma\ \beta}{\sigma\ \beta_1 \quad | \quad \sigma\ \beta_2} \qquad \frac{\sigma\ x \vee y}{\sigma\ x \quad | \quad \sigma\ y} \qquad \frac{\sigma\ \neg(x \wedge y)}{\sigma\ \neg x \quad | \quad \sigma\ \neg y}$$

$$\frac{\sigma\ \neg(x \rightarrow y)}{\sigma\ \neg x \quad | \quad \sigma\ y} \qquad \frac{\sigma\ \neg(x \leftrightarrow y)}{\sigma\ \neg(x \rightarrow y) \quad | \quad \sigma\ \neg(y \rightarrow x)}$$

Formalisation de politiques de sécurité

règle de double négation

représente la satisfaction de la formule $\sigma \neg\neg\alpha = \sigma \alpha$

$$\frac{\sigma \neg \neg\alpha}{\sigma \alpha}$$

Formalisation de politiques de sécurité

règle de possibilité

le monde nommé par σ satisfait $\Diamond\alpha$ et il existe au moins un monde accéssible à partir du monde dont le nom est σ qui satisfait α

$$\frac{\sigma \ \Diamond\alpha}{\sigma.n \ \alpha} \qquad \frac{\sigma \ \neg \ \Box\alpha}{\sigma.n \ \neg\alpha}$$

Formalisation de politiques de sécurité

règle de nécessité

le monde nommé par σ satisfait $\Box\alpha$ et tous les mondes accéssibles à partir du monde dont le nom est σ satisfont α

$$\frac{\sigma\ \Box\alpha}{\sigma.n\ \alpha} \qquad \frac{\sigma\ \neg\ \Diamond\alpha}{\sigma.n\ \neg\alpha}$$

Formalisation de politiques de sécurité

F : une formule de la logique modale.

F a une preuve par tableaux sémantiques si **le tableau sémantique pour** $1\ \neg F$ **est fermé**

- un tableau sémantique est fermé (ou clos) si toutes ses branches sont fermées.
- une branche d'un tableau est fermée (ou close) si les formules préfixées $\sigma\ F$ et $\sigma\ \neg F$ apparaissent dans la branche

Formalisation de politiques de sécurité

$$1\ \neg(\Box(p \wedge q) \rightarrow (\Box p \wedge \Box q)) \quad (1)$$

$$1\ \Box(p \wedge q) \quad (2)$$

$$1\ \neg(\Box p \wedge \Box q) \quad (3)$$

$1\ \neg\Box p \quad (4)$	$1\ \neg\Box q \quad (9)$
$1.1\ \neg p \quad (5)$	$1.1\ \neg q \quad (10)$
$1.1\ p \wedge q \quad (6)$	$1.1\ p \wedge q \quad (11)$
$1.1\ p \quad (7)$	$1.1\ p \quad (12)$
$1.1\ q \quad (8)$	$1.1\ q \quad (13)$

$$F = \Box(p \wedge q) \rightarrow (\Box p \wedge \Box q)$$

Formalisation de politiques de sécurité

Exemple de formalisation en logique déontique SDL.

- **: prédicats unaire :** File, Public, Secret, Old_Passwd, Acces_System, Change_Passwd
- **: prédicats binaire :** Play, Owner, Passwd, Cleared, Login, Read, Write, Downgrade

Formalisation de politiques de sécurité

Exemple de formalisation en logique déontique SDL.

- **R1 :** Any agent playing the role User is permitted to read any public file

- **formule SDL :**

$$\forall f, \forall A \; File(f) \wedge Public(f) \wedge Play(A, User) \rightarrow P \; Read(A, f)$$

Formalisation de politiques de sécurité

Exemple de formalisation en logique déontique SDL.

- **R2 :** Any agent playing the role User is permitted to write his own public file
- **formule SDL :**

  $\forall f, \forall A\ File(f) \wedge Public(f) \wedge Owner(f, A) \wedge Play(A, User) \rightarrow P\ Write(A, f)$

Formalisation de politiques de sécurité

Exemple de formalisation en logique déontique SDL.

- **R3 :** Any agent playing the role User is forbidden to downgrade a file
- **formule SDL :**

$$\forall f, \forall A \; File(f) \land Public(f) \land Play(A, User) \rightarrow F \; Downgrade(A, f)$$

Formalisation de politiques de sécurité

Exemple de formalisation en logique déontique SDL.

- **R4 :** Any agent playing the role User is obliged to change his password if it is more than one month old
- **formule SDL :**

$$\forall A, \forall pass\ Password(A, pass) \wedge Old\_Passwd(pass) \wedge Play(A, User) \rightarrow O\ Change\_Passwd(A)$$