CFA Properties

SOUNDNESS

The analysis is semantically correct. If $(\rho, \kappa) \models P$ and $P \rightarrow Q$ then $(\rho, \kappa) \models Q$

EXISTENCE

- The set of proposed solutions can be partially ordered ($(\rho, \kappa) \subseteq (\rho', \kappa')$ iff ordered pointwise).
- The set $\{(\rho, \kappa) \mid (\rho, \kappa) \models P\}$ is a Moore family*
- Therefore there always exists a least solution (ρ, κ).

(*) A set $\mathcal{I}$ of proposed estimates is a *Moore family* if and only if it contains $\sqcap \mathcal{J}$ for all $\mathcal{J} \subseteq \mathcal{I}$.

CFA Properties (cont.)

c'è una procedura costruttiva per ottenere la soluzione migliore e questo è stato studiato che si può fare con una complessità polinomiale molto bassa, quindi questo è un motivo per cui vi dovrebbe essere chiaro che questa è un'analisi in cui pago davvero l'approssimazione perché spendo molto meno perché ho la possibilità di avere una complessità polinomiale anche bassa,

CONSTRUCTION

There is a constructive procedure for obtaining the least solution of low polynomial complexity.

quindi per capire, ma solo l'idea della costruzione della soluzione, vi potete immaginare di rileggere queste queste clausole (slide 23 CFAforPiCalculusparte1.pdf) in termini di vincoli, quindi potete fare un'operazione per cui il vincolo che viene messo riquadrato sostanzialmente è un modo che guida l'analisi, che è un po' quello che abbiamo fatto quando abbiamo fatto l'esempio, cioè se torniamo a questo esempio qua (slide 20 CFAforPiCalculusparte1.pdf) quello che succede è che io comincio analizzare gli output e gli input che sono a top level e comincia a vedere laddove appunto ho tutti nomi liberi, posso cominciare a dire che siccome c'è un output di d su a , sicuramente k di a deve includere d e poi lo stesso posso dire che k di a include b, a questo punto posso andare a vedere l'analisi degli input a di w e sapendo che k di a può includere sia b che d posso dedurre che w possa essere legato ad entrambi e via così. Quindi le stesse regole che vedete qua in forma di controllo le potete immaginare trasformate sotto forma di vincolo.

CFA pattern (2)

- Choose those values of interest for the language and define estimates and clauses.
- Prove that all estimates are semantically correct.
- Prove that least estimate exist.
- Derive a *constructive* procedure that builds estimates.
- Select a specific dynamic security property and define a static check on estimates.

This may require to refine estimates.

ora, a questo punto abbiamo visto il pattern per capire come costruire l'analisi e l'analisi in questo momento è un'analisi generica, cioè mi dice più o meno come si comportano i miei processi rispetto alla comunicazione che poi sono le primitive principali che mi rappresentano l'interazione tra processi concorrenti. A questo punto manca un solo pezzettino ed è quello di andare a vedere ora che ho la possibilità di predire il comportamento con approssimazione, ma predire il comportamento dinamico di un sistema, vado a studiare le proprietà di sicurezza. Si possono anche studiare altre proprietà ovviamente, ma noi pensiamo alla sicurezza.

CFA pattern (2)

- Choose those values of interest for the language and define estimates and clauses.
- Prove that all estimates are semantically correct.
- Prove that least estimate exist.
- Derive a *constructive* procedure that builds estimates.
- Select a specific dynamic security property and define a static check on estimates.

This may require to refine estimates.

quindi a questo punto mi manca di stabilire qual è la la proprietà dinamica di sicurezza che mi interessa e capire come posso vederla in anticipo sul controllo statico questo quindi richiede di raffinare le mie soluzioni e le mie stime.

Now comes Security: a simple **Secrecy** property

Dynamic notion: Carefulness

No secret datum flows on a public channel.

Static notion: Confinement

1. partition names in $\begin{cases} \mathcal{S} & \text{Secret names} \\ \mathcal{P} & \text{Public names} \end{cases}$
2. compute $(\rho, \kappa) \models P$
3. check that $\kappa(a \in \mathcal{P}) \subseteq \mathcal{P}$

Theorem If a system S is confined then S is careful.

Example

vediamo qualche esempio ritorniamo alla nostra all' esempio della rana dalla bocca larga e supponiamo, ma è una supposizione solo a uso e consumo dell'esempio, perché non ha senso la partizione, supponiamo di avere una partizione di verdi e di rossi che è quella che vedete e che naturalmente non corrisponde a quello che uno vorrebbe in questo caso. Quello che succede è che se la partizione fosse questa io andrei a vedere la mia soluzione che già abbiamo, la colorerei di conseguenza e mi accorgerei che questo processo non rispetta la mia proprietà statica perché vedrei che c'è almeno un caso in cui la proprietà k di canale pubblico incluso in pubblico non è rispettata. Perché abbiamo l'ultimo caso della componente k di C_AB che contiene m che è rosso. Succede è che che ho creato la mia analisi ho deciso qual era la mia proprietà di interesse ho preso la nozione dinamicha e ho provato a vederla dal punto di vista statico naturalmente quello che mi serve è poi tecnicamente formalmente dimostrare che c'è una corrispondenza tra le due, quindi io posso dimostrare che il sistema è confinato, che è il nome che abbiamo dato alla nozione statica allora, se il sistema è confinato, allora è anche careful che è invece la nozione corrispondente dinamica.

$$
\begin{aligned}
P &= (\nu c_{AS})(\nu c_{BS})(\,(A|B)\,|S) \\
A &= (\nu c_{AB})(\overline{c_{AS}}\langle c_{AB}\rangle.\overline{c_{AB}}\langle M\rangle) \\
S &= c_{AS}(x).\overline{c_{BS}}\langle x\rangle \\
B &= c_{BS}(w).w(y)
\end{aligned}
$$

$$
\begin{array}{llll}
\rho(x) \supseteq \{c_{AB}\} & & \kappa(c_{AS}) \supseteq \{c_{AB}\} & \\
\rho(w) \supseteq \{c_{AB}\} & & \kappa(c_{BS}) \supseteq \{c_{AB}\} & \\
\rho(y) \supseteq \{M\} & & \kappa(c_{AB}) \supseteq \{M\} & \leftarrow
\end{array}
$$

If $\mathcal{S} = \{M, c_{AS}, c_{BS}\}$ and $\mathcal{P} = \{c_{AB}\}$, then P has leaks.

$$\mathcal{S} \ni M \in \kappa(c_{AB} \in \mathcal{P}) \not\subseteq \mathcal{P}$$

The process would have no leaks if c_{AB} and M, were secret or public, respectively.

Other example

$$P \mid Q \mid R = (\underbrace{\bar{a}d.P' + \bar{a}b.P''}_{P} \mid R \mid \underbrace{a(w).\bar{c}w.Q'}_{Q})$$

Suppose that $\mathcal{S} = \{a, d\}$ and that $\mathcal{P} = \{b, c\}$.

	ρ
a	$\{a\}$
b	$\{b\}$
c	$\{c\}$
d	$\{d\}$
w	$\{b, d\}$

	κ
a	$\{b, d\}$
b	$\emptyset$
c	$\{b, d\}$
d	$\emptyset$

The system does not satisfy the property: $\boxed{\kappa(c) = \{b, d\} \not\subseteq \mathcal{P}}$

Another Property: No Read Up/No Write Down (NRU/NWD)

Processes are given levels of security clearance
NRU/NWD: the sender has a clearance level lower than the level of the receiver.

- Syntax: $S ::= \langle P \rangle^{l} \mid (\nu x)S \mid S|S \mid !S$,with l level label;

Supponiamo di avere un'altra proprietà che è un'altra proprietà tipica della sicurezza e ha a che fare con i livelli di clearance quindi supponiamo che i processi siano classificati o di livello basso o di livello alto ora per semplicità utilizziamo solo due livelli, ma potete immaginarne un reticolo come avete visto anche in tutta la parte sull information flow e quindi supponiamo che anche e qui serve per forza un'estensione, supponiamo che in qualche modo anche la sintassi del pi calcolo che non è pensata per fare questo lo preveda, quindi supponiamo che il singolo processo, quindi la singola entità possa essere etichettata con il suo livello di clearance, quindi l'unica modifica che devo fare è mettere tra parentesi angolate il processo p e accompagnarlo con un'etichetta che gli sta sopra, che indica il livello di clearance.

Ora naturalmente questo mi comporta che anche a livello di semantica debba tenere traccia del livello di clearance e quindi ci possiamo immaginare una semantica in cui se il processo così com'è fa un'azione mu e diventa Q, anche il processo incluso in questo contesto di livello l, fa lo stesso passo arriva in Q sempre con lo stesso livello e mi ricordo dell'azione con mu, ma anche del livello con l mettendolo sull'etichetta della transizione. Ora a questo punto posso provare a fare un'analisi tipo quella che avevamo però devo arricchirla per tenere conto dei vari ingredienti, quindi anche del livello. Tornando indietro alla slide 18 di CFAparte1.pdf vi chiedo quali modifiche posso fare all' analisi che abbiamo visto, cioè quindi quello che mi interessa, ne ho parlato dando per scontato, No read up e no write down, la proprietà che voglio è che non ci siano mai comunicazioni di output dall'alto verso il basso e corrispondentemente non ci siano letture dal basso verso l'alto, quindi chiedo a voi quali sono gli oggetti che posso tracciare e quali modifiche necessarie devo fare per tracciare questo tipo di comportamento? Mi servirà ancora di avere RHO di x che mi dice come vanno avanti i miei legami, però forse dovrò strutturare meglio le mie informazioni sull output e sull input, quindi su questo chiedo il vostro aiuto. Dobbiamo avere k w e k r per definire le due operazioni, cioè a me serve distinguere che cosa passa sui canali, ma a questo punto che cosa passa in output e che cosa passa in input in modo da capire se c'è un passaggio sbagliato, cioè un passaggio non previsto dalla politica no read up e no write down. Quindi se tengo conto dell' output e dell'input dovrei riuscire a vedere la direzione. Poi l'altra cosa che mi serve è che l'informazione di livello è sostanzialmente spezzettata all'interno del mio sistema, cioè ho processi di alto livello e processi di basso. Quindi anche su questo devo fare qualcosa perché se io k, come abbiamo visto è globale come informazione quindi k mi dice in tutto il processo che cosa viene spedito su a e quindi anche l'ipotesi che è stata fatta è corretta, però ancora va perfezionata perché se io ho k in output, k w lo ho comunque su tutto il sistema invece qui ho bisogno di vedere chi fa output e chi fa input all'interno del sistema perché abbiamo visto che la mia sintassi mi dice che sostanzialmente i sistemi sono una composizione parallela di processi che possono avere livelli diversi di clearance quindi come mi posso muovere sapendo che io vedo che i processi hanno un'etichetta come potrei fare? Quindi io passo da kappa(a) a kappa_w(a) e kappa_r(a) e questo è un primo passo, però è ancora globale, come faccio a localizzare queste informazioni di k(w) e k(r)? Se devo localizzare questa informazione, non mi basta più dire qual è il k in uscita o in entrata relativo a un canale, mentre prima abbiamo visto che partizionavo i canali in alto e basso, adesso sono i processi che sono alti e bassi, avete visto che la sintassi è quella che dice: <P>^l dove quella l può essere h o l maiuscolo, no? Quindi, per ancorare le informazioni di entrata ed uscita, io lo posso fare legando il k all' etichetta l, quindi scriverò, ora poi noi abbiamo usato un'altra notazione, ma l'idea è questa, è del tutto omomorfa, io dovrò dire kappa_w(a,l) e kappa_r(a,l) cioè io lego l'informazione di cosa può uscire e cosa può entrare al punto in cui lo faccio.

- Syntax: $S ::= \langle P \rangle^{l} \mid (\nu x)S \mid S|S \mid !S$,with l level label;
- Semantics: $$\frac{P \xrightarrow{\mu} Q}{\langle P \rangle^{l} \xrightarrow{\mu,l} \langle Q \rangle^{l}}$$

Prevedendo cosa può uscire su quel canale, in tutti i processi di livello basso e cosa può entrare da quel canale in tutti i processi di un'altro livello, posso andare a mettere insieme le informazioni e controllare la mia proprietà che non vuole che ci sia un passaggio dall'alto in basso in uscita. Quindi questa è l'idea, quindi vi potete immaginare la proprietà sia statica che dinamica, a questo punto l'analisi è molto simile a quella che abbiamo visto, quindi abbiamo le regole fatte così più o meno (slide Estimate Validation CFAparte1.pdf) però dovrò aggiungerci qualche dettaglio tecnico in modo da renderle adatte al nuovo tipo di soluzione, quindi in particolare dovrò quando analizza un'output o un input, ricordarmi nel contesto di quale etichetta lo sto facendo, quindi se l'azione è di tipo alto o di tipo basso, cioè se il processo è di tipo alto o di tipo basso, dovrò ricordarmelo in modo da scrivere non k(a) ma in questo caso k di output di a e di l. Quindi sostanzialmente ho bisogno di ricordarmi nel contesto di quale processo sono quindi mi serve detto, in termini delle analisi, un modo per poter dire che sto analizzando un processo di livello l, quindi devo capire qual'è la regola per ricordarmi la l e quindi io devo analizzare tutti i casi sintattici quand'è quindi un'analisi è valida per un processo etichettato l? (rho,kappa) |= <P>^l IFF (rho,kappa) |= _l P Allora il sistema che si usa è quello di farlo portandomi dietro un nuovo judgement che vuol dire che quando andrò ad analizzare i pezzi di P avrò nel judgment il ricordo di qual è il livello del mio processo e quindi potrò fare l'operazione che dicevamo prima sul lucido, cioè potrò ricordarmi non solo che rho(y) sta in k(a) ma sta in k(a,l).

Another Property: No Read Up/No Write Down (NRU/NWD)

Processes are given levels of security clearance
NRU/NWD: the sender has a clearance level lower than the level of the receiver.

- Syntax: $S ::= \langle P \rangle^{l} \mid (\nu x)S \mid S|S \mid !S$,with l level label;
- Semantics: $$\frac{P \xrightarrow{\mu} Q}{\langle P \rangle^{l} \xrightarrow{\mu,l} \langle Q \rangle^{l}}$$
- Analysis: $(\rho, \kappa, \sigma) \models_{l} P$, with $\sigma = \langle \sigma_{in}, \sigma_{out} \rangle$, where
  - $\sigma_{in}(l)$: set of the channels that can be received by an input within a sub-process with level l.
  - $\sigma_{out}(l)$: set of the channels that can be sent by an output within a sub-process with level l.

L'analisi diventa più complessa quindi, invece di cambiare k posso pensare di aggiungere una componente sigma, sigma_in e sigma_out, quindi sigma_in(l) è l'insieme di canali che possono essere ricevuti da un input dentro un sotto processo di livello l e sigma_out può essere il corrispondente al livello di output.

NRU/NWD: CFA rules

quindi quello che succede è che ho una delle regole leggermente diverse, per cui mi ricordo RHO(y) sta in k(a) esattamente come prima, ma adesso la nuova componente che è il k_w k_r che diceva il vostro compagno e mi ricordo che se ho un output in un processo di livello l vado a scrivere che RHO(y) appartiene anche a sigma_out(l)(a), nella stessa maniera vado a fare l'input e quindi mi ricordo che k(a) apparterrà a sigma_in(l)(x) e l'ultima regola è quella che vi dicevo è quella che mi consente che una volta che individua un processo complessivamente etichettato con l di portarmi avanti nell'analisi il fatto cioè di ricordarmi quando analizzo P che il suo livello è l. Quindi l'idea è semplicemente che io posso cambiare leggermente la mia analisi per osservare una nuova proprietà. Vi ricordo che qui sono costretta a cambiarla la mia analisi perché ho cambiato anche la sintassi e quindi c'è un modo per mettere in relazione, sempre a livello di soundness, quello che mi dice l'analisi con quello che mi fa vedere la semantica a tempo di esecuzione.

- $(\rho, \kappa, \sigma) \models^{l} \overline{x}y.P$ iff $(\rho, \kappa, \sigma) \models^{l} P \wedge$
  $\forall a \in \rho(x) :$
  $\begin{pmatrix} \rho(y) \subseteq \kappa(a) \wedge \\ \rho(y) \subseteq \sigma_{out}(l)(a) \end{pmatrix}$
- $(\rho, \kappa, \sigma) \models^{l} x(y).P$ iff $(\rho, \kappa, \sigma) \models^{l} P \wedge$
  $\forall a \in \rho(x) :$
  $\begin{pmatrix} \kappa(a) \subseteq \rho(y) \wedge \\ \kappa(a) \subseteq \sigma_{in}(l)(x) \end{pmatrix}$
- $(\rho, \kappa, \sigma) \models \langle P \rangle^{l}$ iff $(\rho, \kappa, \sigma) \models^{l} P$

NRU/NWD

Dynamic Property: **no read-up/no write-down**

A high level process cannot write any value to a process at low level; symmetrically a process at low level cannot read data from one of a high level.

Static Property: **discreet**

Each channel cannot be used for sending an object from a process with high level l to a process with low level l'.

$$\forall l', l \text{ with } l' \text{ below } l : \forall a : \sigma_{out}(l)(a) \cap \sigma_{in}(l')(a) = \emptyset.$$

Theorem If a system S is discreet then S is No Read Up/No

Infatti vedete qui di nuovo l'associazione tra la proprietà dinamica e la corrispondente proprietà statica, quindi la proprietà dinamica è quella classica NRU e NWD che mi dice che un processo di livello alto non può scrivere un processo di livello basso e quindi simmetricamente il processo di livello alto non può leggere dati che provengono da un processo a livello più alto e noi qui abbiamo trovato la formulazione in termini statici, quindi io prendo tutte le coppie di livelli e vado a controllare se per ciascun canale l'uso è quello corretto, quindi non ci sia mai un caso in cui ci sia un passaggio di informazione dall'alto al basso e questo esattamente la proprietà vista in termini statici, di nuovo possiamo verificare e provare formalmente che se un sistema gode della proprietà di discretezza, in questo caso, come abbiamo chiamato qui la proprietà, allora sarà di tipo NRU e NWD. Le analisi sono descrittive quindi mi dicono che cosa succede eventualmente mi dicono anche se c'è una violazione. Cosa posso dire dal punto di vista della sovrappassimazione? (slide 5) cosa succede se io trovo che k(a) per ciascuna pubblico sta dentro P cosa posso dedurre su quello che succede a livello dinamico? Ci sarà una violazione a tempo dinamico? Non dovrebbe perché questa è la parte giusta, cioè se una cosa non è prevista allora non accadrà cosa succede invece se, almeno in un caso, io trovo che un k(a) mi fa passare almeno un elemento segreto? Cosa mi fa pensare questo? Cosa succede a livello di tempo dinamico? Non possiamo dedurlo, quindi non è detto, cioè se io trovo che c'è una violazione, cioè se prevedo a tempo statico una violazione, potrebbe non verificarsi a tempo dinamico, ugualmente laddove naturalmente i sistemi siano un po' più complessi di quelli che abbiamo visto, è indicativo anche la violazione statica, perché potrebbe essere invece un caso da guardare con maggiore attenzione, cioè laddove l'analisi statica mi mi predice una possibile violazione, adesso poi lo vedremo nel contesto dei protocolli, abbiamo una sorta di suggerimento ad andare a controllare se tutto è a posto a livello dinamico, quindi rispetto ad un'analisi dinamica che mi va a vedere qualsiasi tipo di di evoluzione e in tutti i punti possibili del mio sistema di transizione, l'analisi statica laddove mi predice una violazione è un po come se mi suggerisse di andare a vedere in qualche punto particolare e in generale questo potrebbe voler dire che ci sono dei punti in cui a livello dinamico vado a rischiare, questo rimettendo insieme altri concetti visti durante il corso, può suggerirmi, ad esempio, di istrumentare il codice, e nei punti dove c'è il rischio potrei a livello semantico avere una sorta di vigile, che è il reference monitor, che mi dice passa solo se la semantica mi garantisce che qui non ci sono problemi. L'idea è che la mia analisi statica mi potrebbe mettere in guardia perché alcuni punti del mio programma potrebbero essere a rischio, a questo punto potrei istruire un reference monitor che mi controlla che non si creino queste situazioni di rischio proprio a programma. Quindi io se mi trovo di fronte a una cosa rischiosa controllo, questa volta a tempo dinamico, che non si creino le condizioni di rischio e blocco eventuali operazioni, se la mia analisi statica mi dice che è possibile che ci sia una comunicazione dall'alto al basso sbagliata, io vado a vedere ogni volta che c'è una comunicazione dall'alto, se viene indirizzata, per esempio, a un processo che sta più in basso, non lo vado a guardare sempre lo vado a guardare sul canale che la mia analisi mi ha indicata quindi l'analisi statica guida l' ottimizzazione dell'uso, per esempio di un reference monitor.