Abbiamo visto come la CFA si può applicare ad algebre di processo che rappresentano sistemi distribuiti concorrenti e serve essenzialmente in prima battuta per predire in qualche maniera il comportamento a run time di questi sistemi, senza appunto seguirne l'evoluzione dinamica, ma semplicemente guardandone la specifica con tutta l'approssimazione del caso, quindi è proprio una sorta di baratto, si baratta la precisione con il costo inferiore. Abbiamo visto che sulla base delle informazioni raccolte, che si chiamano stime soluzioni o estimates, noi riusciamo a prevedere, con appunto un po di approssimazione, l'evoluzione in termini di che cosa? Principalmente in termini di comunicazioni, quindi, sincronizzazione di questi processi tra di loro e scambio di valori e quindi vediamo anche quali sono, per ogni variabile, i valori che verranno possibilmente assunti a run time. Questo consente poi su questa base di investigare proprietà, proprietà che in questo caso quelle che abbiamo visto sono principalmente legate alla sicurezza, abbiamo visto una proprietà di secrecy per quanto riguarda la CFA del PI calculus e quindi una volta che si sono raccolte queste stime va a partizionare quindi proprio a valle di questo procedimento va a partizionare i nomi in pubblici e segreti e va a controllare se si può verificare a run time l' eventualità che un nome segreto passi in chiaro, cioè su un canale pubblico e quindi questo lo possiamo vedere sia in termini statici che dinamici. Naturalmente tutto ciò ha un senso se la proprietà vista in termini statici ha come conseguenza la proprietà in termini dinamici, quindi andremo a vedere se ci sono possibili violazioni oppure no. Stesso analogo ragionamento viene compiuto su LySa che è l'estensione del PI calculus e dello SPI calculus, che è stato adottato in questo caso e che di nuovo produce un'analisi che mi dice più o meno come si evolvono le comunicazioni nel tempo e a questo viene aggiunto, quindi l'analisi è corredata da un'ulteriore componente rispetto al rho e k che abbiamo già visto anche per il PI calculus, una componente di errore psi che non fa altro che raccogliere le coppie di punti di rispettivamente origine e destinazione dei test cifrati che vengono scambiati dentro un protocollo. Questo abbiamo visto ci consente di determinare una proprietà che ci porta a mettere l'attenzione su effetti collaterali di possibili attacchi. Quindi abbiamo visto che spesso la interferenza sostanzialmente nel protocollo, la presenza di un nemico che può intercettare quello che ti viene passato in rete ma allo stesso tempo può immettere in rete messaggi costruiti da lui in modo da ingannare i partecipanti onesti, fa sì che alcune encryption pensate per essere codificate in un punto e decodificato in un'altro, invece, seguano un percorso diverso e abbiamo visto la volta scorsa che spesso questo tipo di possibilità in realtà punta a una possibile vulnerabilità del nostro protocollo, quindi, anche se di nuovo trattiamo una sovrapposizione del comportamento e quindi ormai abbiamo capito che l'unica certezza che abbiamo è in negativo, cioè se io non trovo una violazione di questo tipo, allora questa violazione non sorgerà nemmeno a tempo di esecuzione, in questo caso specifico vale la pena anche andare a controllare laddove invece una violazione è ipotizzata e questo significa andare a controllare il protocollo e vedere se esiste una sequenza di attacco che conduca a quel tipo di problema e in effetti è stato trovato che i falsi positivi non sono moltissimi. Quello che naturalmente però si può dire invece per certo è che se io non riesco a trovare nessuna coppia all'interno della componente di errore dell'analisi, allora posso essere sicura che a run time quella violazione non si presenterà e questo vuol dire, dal punto di vista di poi come vado a implementare la semantica, che posso in questo caso fare a meno della cosiddetta reference monitor semantics, che è quella che a run time si mette a controllare, eventualmente bloccando tutto, se l' encryption e decryption seguono il percorso stabilito a priori e quindi in questo caso, se so che l'analisi mi da una componente vuota posso semplicemente spegnere la reference monitor semantics perché sono sicura che quel controllo non sarà utile, quindi sarebbe solo un aggravio rispetto all' implementazione, ma posso farne a meno. Abbiamo visto quindi come applicare la CFA a una verifica di proprietà di sicurezza legate alla modellazione attraverso LySa di protocolli crittografici di quelli standard, quindi quello che riusciamo a vedere sono proprietà di tipo origine destinazione dei messaggi cifrati. Naturalmente uno potrebbe utilizzare la stessa analisi per fare indagini su altre proprietà, per esempio potrebbe riformulare la proprietà di secrecy che abbiamo visto per la CFA del PI calculus in maniera molto simile o comunque fare altri ragionamenti. Chiaramente abbiamo mostrato un tipo di analisi, così come abbiamo mostrato un tipo di algebra di processo, ma come potete immaginare, lo stesso tipo di analisi può essere applicata ad altri contesti, ad altri linguaggi, così come lo stesso linguaggio può essere in qualche maniera analizzato utilizzando altre tecniche statiche, per esempio, potrebbe farci un typesystem. Oggi ci concentreremo, come annunciato la volta scorsa, su un contesto, uno scenario nuovo di applicazione di queste tecniche e anche di algebra di processo che è quello dell' internet of things. Quindi abbiamo fatto un'estensione di LySa in termini IoT per vedere se era possibile studiare sempre proprietà, cioè comportamento e proprietà di sicurezza di sistemi di tipo internet of things.

IoT Challenges

Designing & implementing IoT systems is hard

- Smart objects are heterogeneous
  - Many hw, sw, protocols, etc.
- Cyber-physical and dynamic systems
- Security & privacy
  - Highly critical but difficult to achieve

Ora naturalmente l'internet of things è un nuovo trend che si sta sviluppando sempre più velocemente e presenta tutta una serie di problematiche diverse dagli scenari precedenti anche se non si inventa completamente nulla, però la spinta verso questo tipo di situazioni comporta tutta una serie di problemi che diventano di difficile scalabilità perché cosa succede? L'internet of things vuol dire che adesso in rete non dobbiamo più pensare soltanto ai computer in quanto tali, ma anche a tutti i sistemi di calcolo cablati all'interno degli oggetti e naturalmente, proprio per la natura degli oggetti, questi oggetti sono eterogenei, quindi è difficile anche pensare come ben combinarne il funzionamento, quindi si parla di oggetti smart.

Our Contribution

Quello che quindi andiamo cercando è un framework con un contesto unificante che è basato di nuovo sui formal methods, quindi l'idea è voglio vedere se riesco a specificare i sistemi IoT e quindi ragionare in astratto sulle proprietà di sicurezza di correttezza in primis e anche di costi. Quindi qual è l'idea? Chiaramente in questo filotto di lezioni è proseguire sul percorso tracciato da LySa provandolo a estenderlo con caratteristiche che modellino le peculiarità dei sistemi internet of things.

The process algebra IoT-LySa extends LySa

- Network of nodes
- Sensors & actuators
- Group communication
- Local communication à la Linda (a coordination language)

Tracking data analysis to predict

- Interaction among nodes
- How data flow in the network and are manipulated, also related to security issues

Tracking data analysis

Utilizzeremo una CFA per approssimare in maniera safe di nuovo le interazioni, quindi le comunicazioni questa volta non tra i processi, ma tra i processi collocati all'interno dei nodi, quindi le comunicazioni tra i nodi e seguiremo in particolare come i dati percorrono il loro cammino, quindi partendo dai dati che vengono raccolti sul terreno dai sensori fino alle lavorazioni che su questi dati vengono fatti nei livelli più alti della gerarchia e quindi in particolare vedremo come questi dati vengono raccolti, aggregati attraverso funzioni che vi potete immaginare, ad esempio,se io raccolgo temperature in varie zone di un edificio o di un posto all'aperto, poi avrò bisogno di aggregarli per calcolare una media o altre cose del genere. Poi vedremo un verifica basata sui risultati delle analisi che può essere utilizzata per vari scopi, ad esempio vedere se ci sono troppe comunicazioni che non sono utili oppure facendo taint analysis, vedere se ci sono delle decisioni e di attuazione che dipendono da dati che non sono affidabili perché provengono da nodi che non lo sono e quindi potrebbero influenzare in maniera errata e malintenzionata le decisioni di attuazione.

A Control Flow Analysis (CFA) to safely approximate

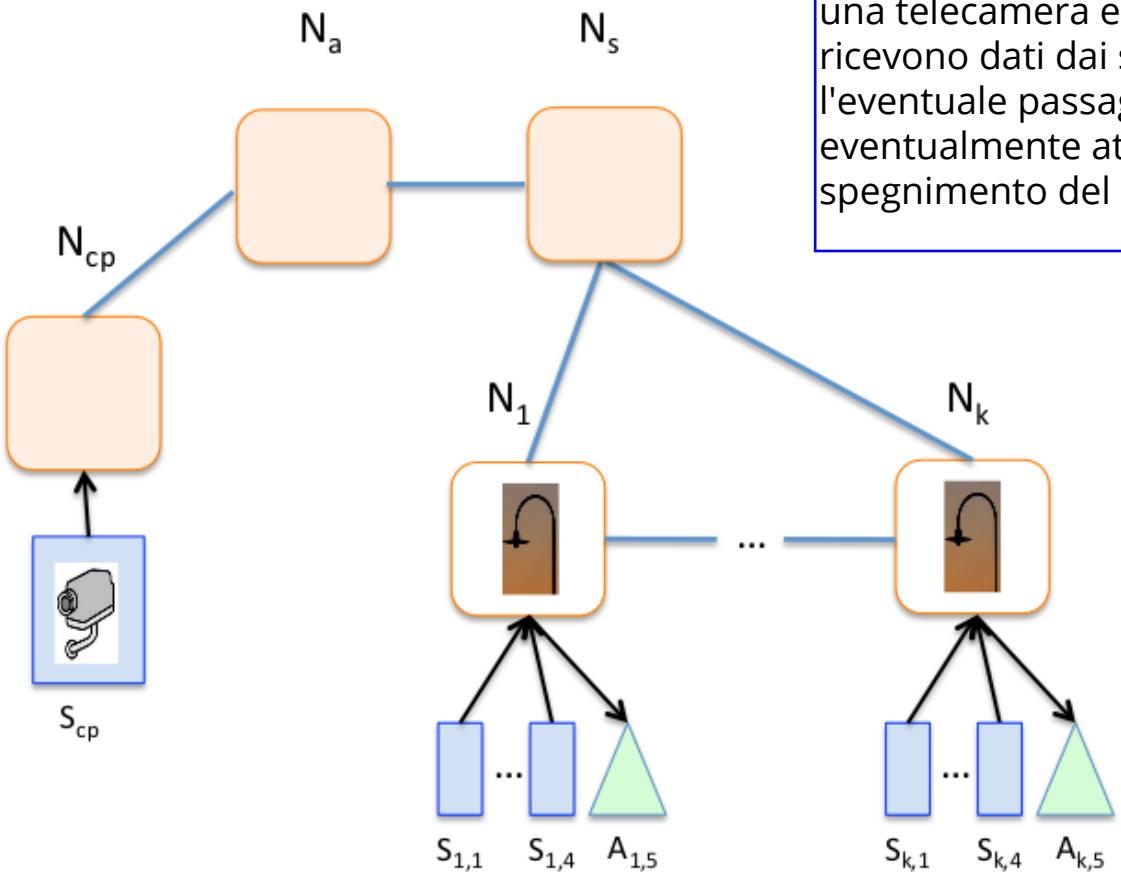
- Interactions among nodes
- How data spread from sensors to network
- How data are manipulated

Verification based on the analysis results

- Checking whether classified data reach untrusted nodes
- Checking whether data come from untrusted sources
- ...

A smart street light control system

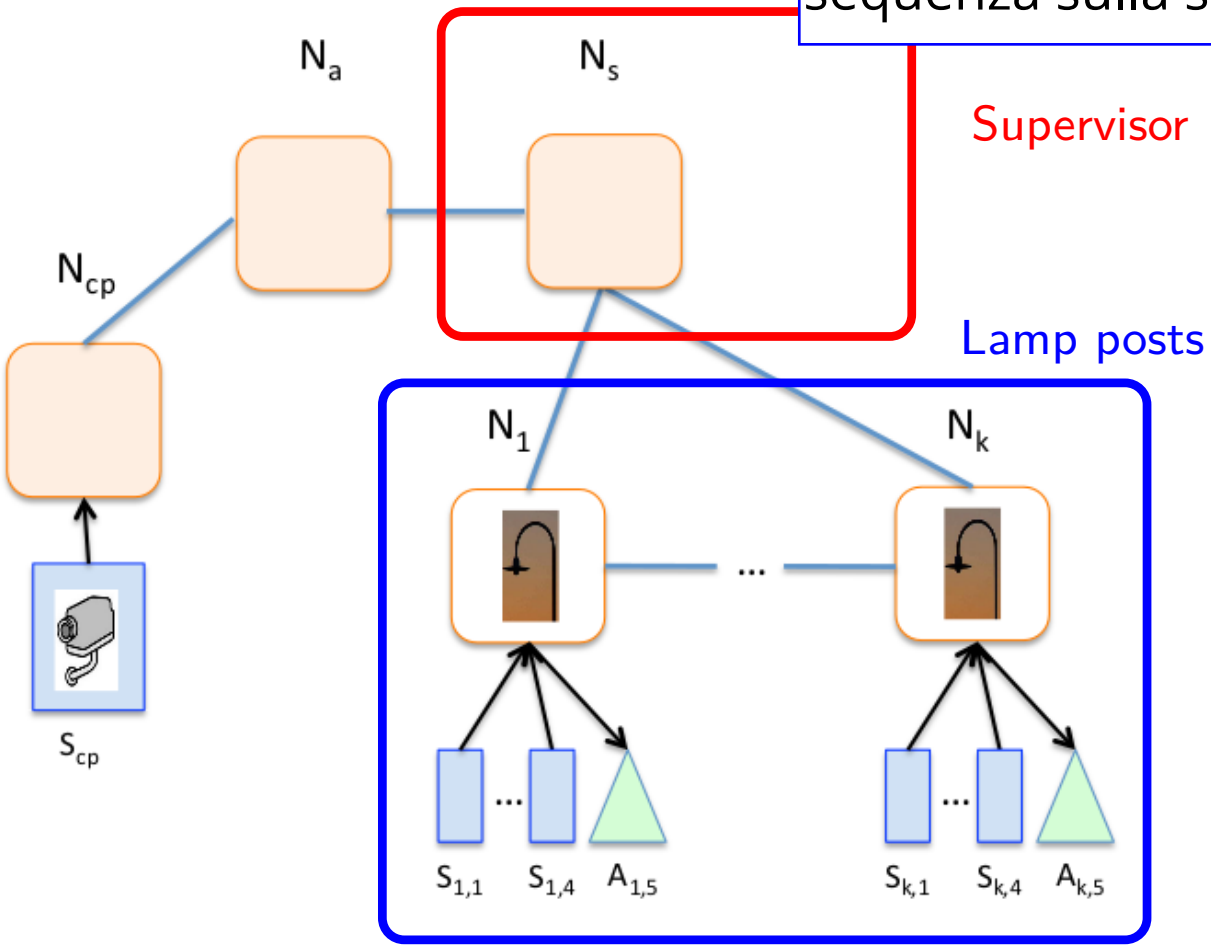
Allora vediamo un esempio di modello di questo tipo di sistema e vediamo come si può rendere proprio con il nuovo tipo di algebra di processo, cioè con IoT LySa. Quello che stiamo modellando è un sistema di controllo che controlla i lampioni in una strada e lo fa di nuovo in maniera smart. Possiamo ipotizzare di avere varie componenti e vari nodi, ci sono vari nodi che controllano e in particolare abbiamo un nodo che riceve dati da una telecamera e poi abbiamo nodi che invece ricevono dati dai sensori che controllano l'eventuale passaggio di persone o macchine ed eventualmente attuano l'accensione o lo spegnimento del lampione.



$$N = N_{cp} \mid N_a \mid N_s \mid N_1 \mid \cdots \mid N_k$$

A smart street light control system

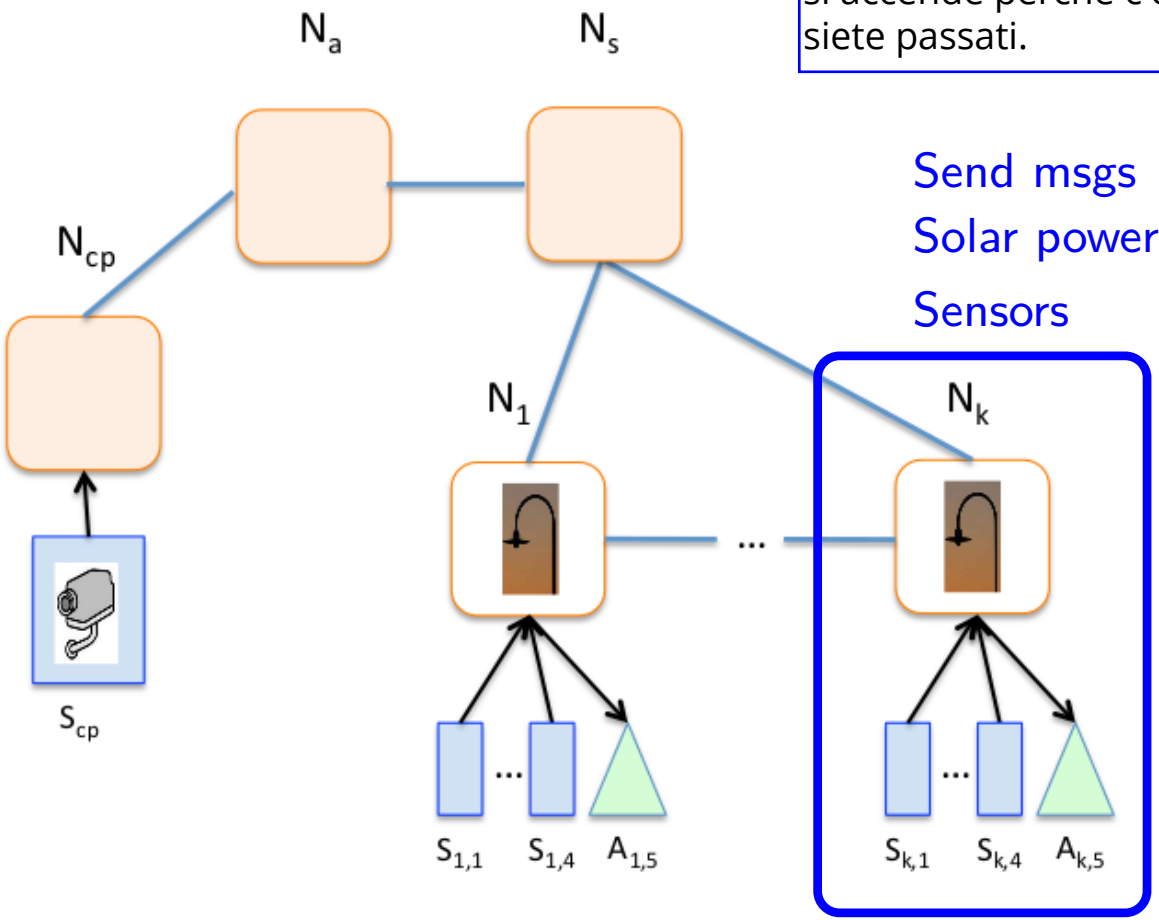
Quindi abbiamo una parte di supervisione e la parte centrale del nostro sistema che è quella che controlla proprio i singoli lampioni, supponete che siano in una sequenza sulla strada.



First part = lamp posts + lamp post supervisor

A smart street light control system

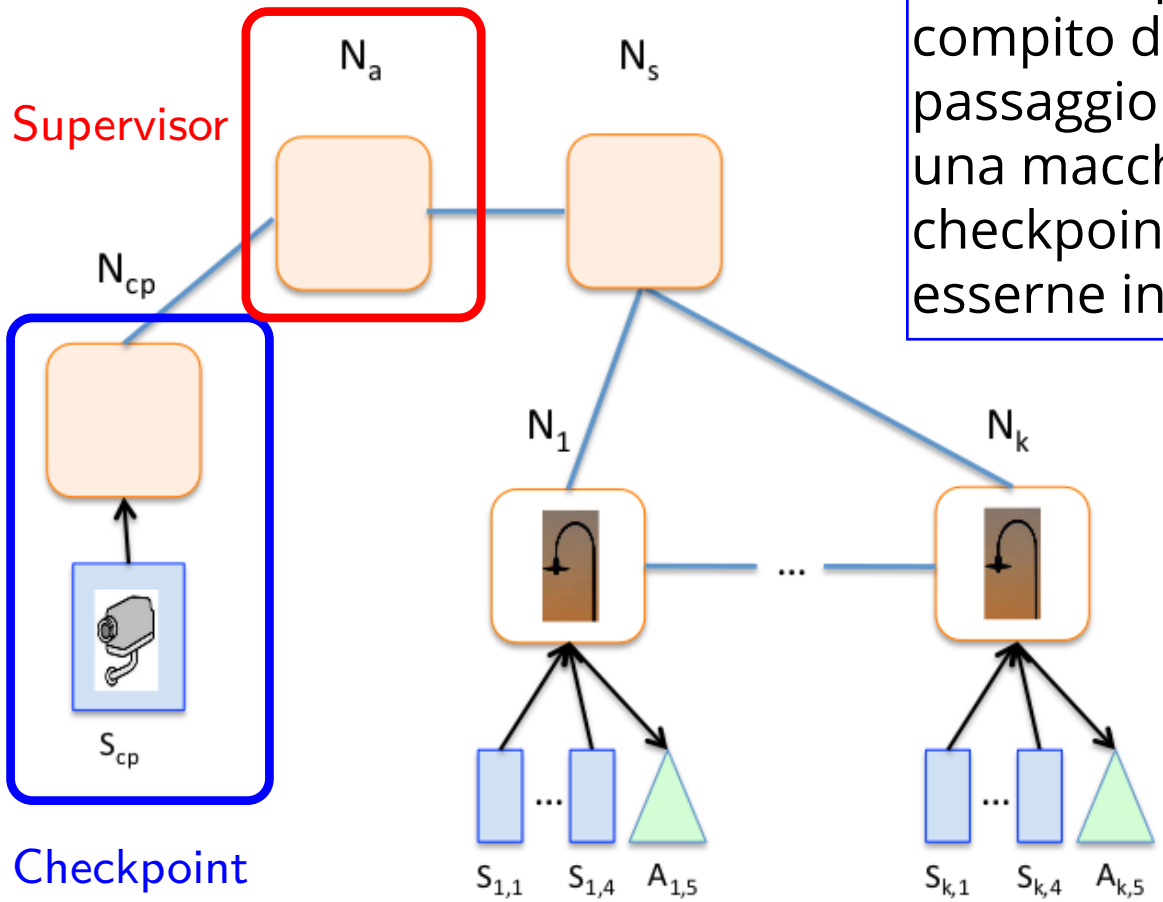
In particolare, il nodo che controlla il singolo lampione può mandare messaggi perché naturalmente deve scambiarsi messaggi con sensori e attuatori dal basso e con l'altro si dovrà collegare al sistema che fa il controllo e quello che succede è che se passa un pedone allora deve accendersi, quindi questo è un tipico sistema che c'è anche in molti edifici dove voi entrate e senza bisogno di premere l'accensione della luce, la luce si accende perché c'è un sensore che rivela che siete passati.



If there is a pedestrian and not enough the light switch on

A smart street light control system

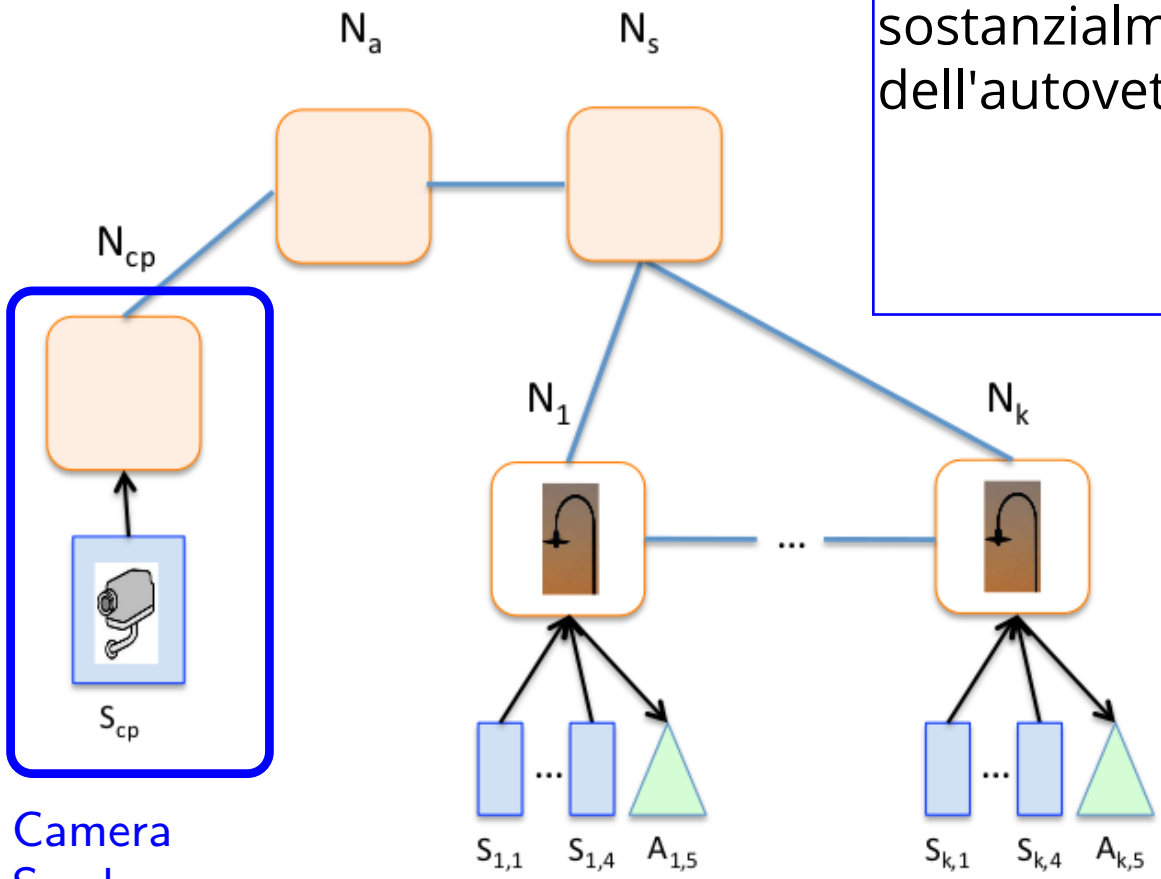
Dall'altra parte abbiamo un checkpoint per le automobili con un suo supervisore e questo ha il compito di controllare il passaggio delle auto, quindi se una macchina attraversa il checkpoint, il supervisore deve esserne informato.



Second part = car checkpoint + supervisor

A smart street light control system

Come fa a vederlo? C'è una telecamera che quindi può mandare i messaggi con la foto sostanzialmente dell'autovettura.



Camera
Send msgs

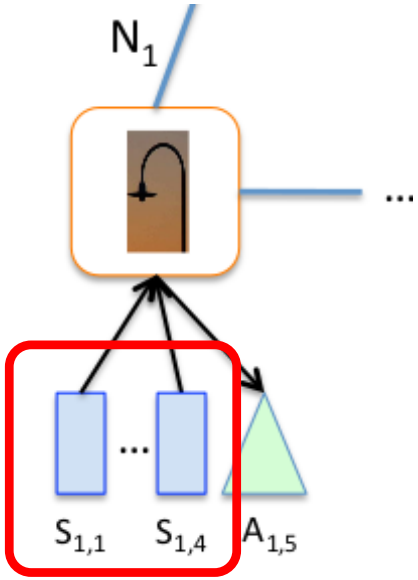
If a car crosses the checkpoint, the supervisor is informed

Come si possono rappresentare i singoli modelli dei sensori? Qui vediamo un esempio, abbiamo un costrutto che mi dice sostanzialmente che iterativamente, quindi mu h è il costrutto per l'iterazione, quindi vuol dire che costantemente ripete lo stesso comportamento, il sensore che cosa fa? Va a fare il sensore, quindi va a rivelare e a raccogliere il valore in quel momento e quindi lo mette in una variabile che può essere chiamata i, ha proprio una sua locazione riservata dove va a mettere il valore, dopodiché fa un'iterazione con l'ambiente che è come al solito un'azione tau e ricomincia. Quindi ogni volta il sensore ,in questo tipo di algebra di processo, non c'è un concetto di tempo e quindi sostanzialmente voi potete immaginare che ripetutamente lui fa il sensing dell'ambiente, salva il valore rilevato e poi prosegue ricostruendo di nuovo il percorso.

Sensors

The 4 sensors sense: (1) the environment light, (2) the solar light, (3) the battery level and (4) the presence of a pedestrian

$$S_{1,i} = \mu\, h.\, (i := v).\, \tau.\, h$$



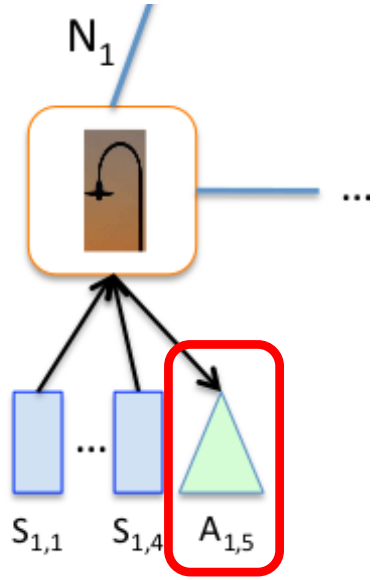
- Iteration: $\mu\, h$
- Interaction with the environment: silent action
- Sensing a value v
  - Communication with processes through a shared store
  - Each sensor has a reserved location (also sensor id)

Actuators

Invece che cosa fa l' attuatore? L' attuatore si aspetta di avere un comando e il comando che può arrivare dall'alto, cioè dalla parte intelligente del lampione, può essere accenditi oppure spegniti, e quindi una sorta di entità più passiva che si limita ad eseguire degli ordini, anche lui ragiona in termini iterativi, ha un identificatore, in questo caso 5, su cui riceve la richiesta e quindi saprà qual è l'azione che deve svolgere.

The actuator can turn on or turn off the light

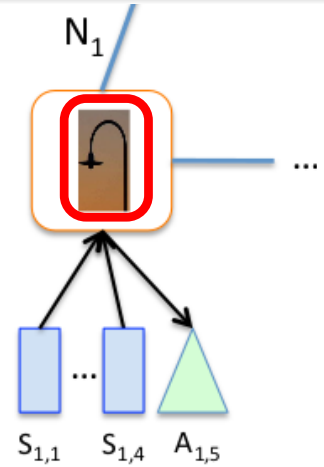
$$A_{1,5} = \mu\, h.\, (\!|5, \{\mathsf{turnon}, \mathsf{turnoff}\}|\!).\ h$$



- A passive entity: it only executes orders
- Iteration: $\mu\, h$
- Identifier
- Actions it can carry out

Control processes: lamp post

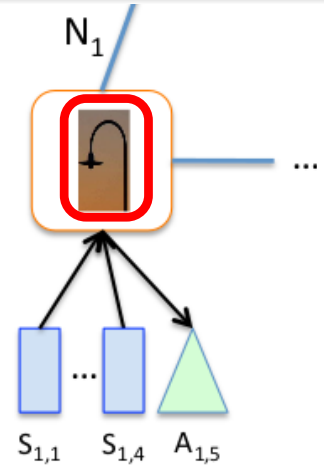
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables

Control processes: lamp post

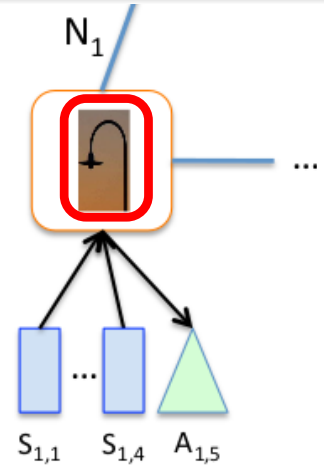
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian

Control processes: lamp post

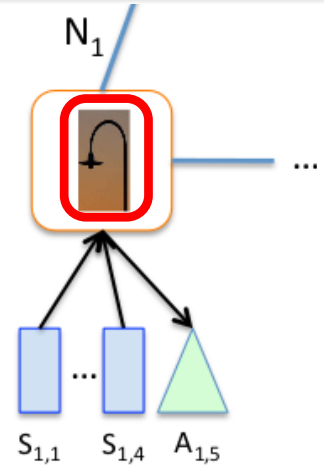
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian

Control processes: lamp post

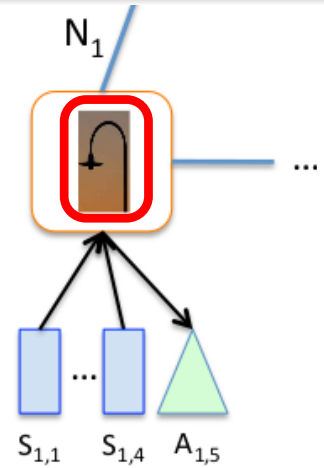
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds

Control processes: lamp post

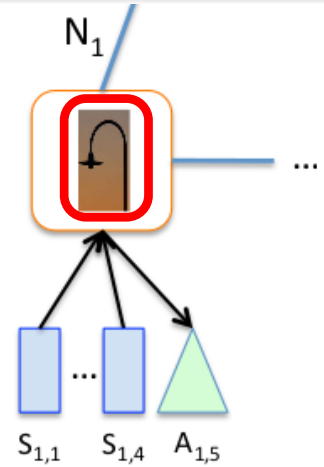
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds & enough battery:

Control processes: lamp post

$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$

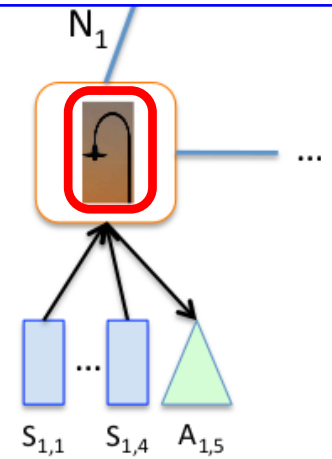


1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds & enough battery: turn light on

Control processes: lamp post

...altra cosa ci sarà una sorta di processo a catena per cui N1 avvertirà N2 che avvertirà N3 quindi piano piano al passaggio del pedone si accenderanno uno dopo l'altro tutti i lampioni...

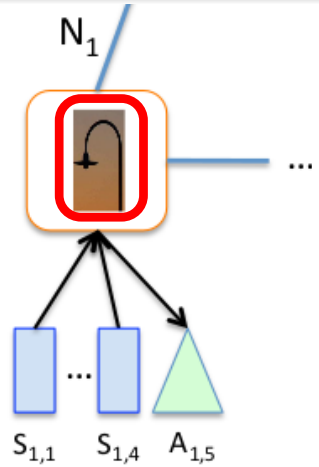
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \mathrm{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \mathrm{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \mathrm{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds & enough battery: turn light on & inform neighbours (nodes with labels in L_P)

Control processes: lamp post

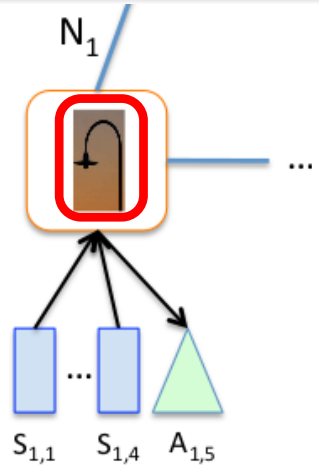
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \text{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \text{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \text{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds & enough battery: turn light on & inform neighbours (nodes with labels in L_P)
3. Not enough battery:

Control processes: lamp post

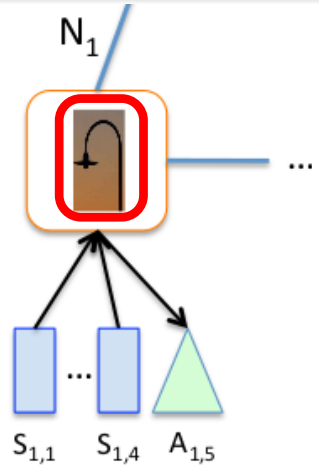
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \text{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \text{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \text{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds & enough battery: turn light on & inform neighbours (nodes with labels in L_P)
3. Not enough battery:

Control processes: lamp post

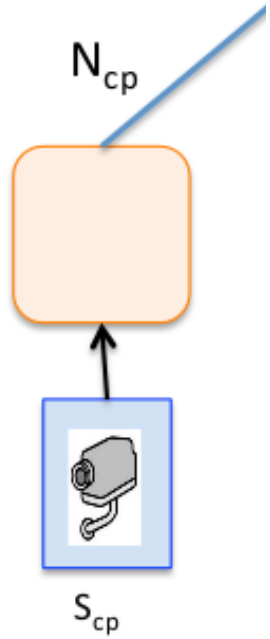
$$P_{1,1} = \mu h.(x_1 := 1.\, x_2 := 2.\, x_3 := 3.\, x_4 := 4).$$
$$(x_4 = true)\ ?$$
$$(x_1 \leq th_1 \wedge x_2 \leq th_2)\ ?$$
$$(x_3 \geq th_3)\ ?\ \langle 5, \text{turnon}\rangle.\, \langle\!\langle x_4 \rangle\!\rangle \triangleright L_p.\ h$$
$$:\ \langle\!\langle \text{err}, \ell_p \rangle\!\rangle \triangleright \{\ell_s\}.\ h$$
$$: h$$
$$:\ \langle 5, \text{turnoff}\rangle.\, h$$



1. Store current sensor value into local variables
2. If pedestrian & street and solar light under given thresholds & enough battery: turn light on & inform neighbours (nodes with labels in L_P)
3. Not enough battery: inform supervisor (N_s)

Control processes: car checkpoint

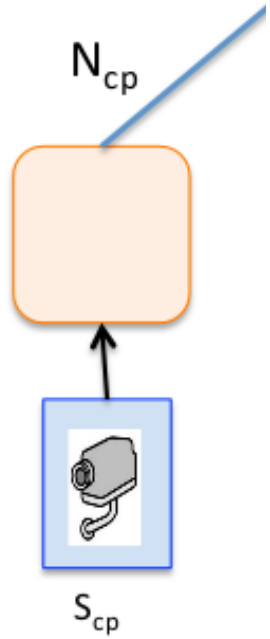
$$P_{cp} = \mu h.(z := 1).(z' := noiseRed(z)).\langle\langle z' \rangle\rangle \triangleright \{\ell_a\}.\ h$$



1. Store current sensor value into local variables: the picture

Control processes: car checkpoint

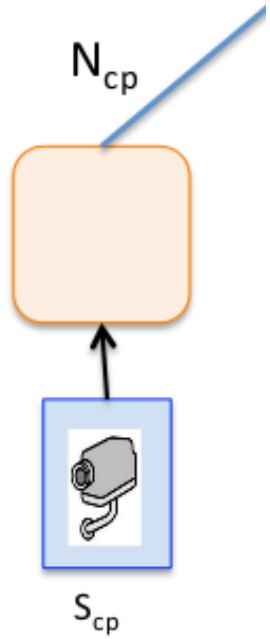
$$P_{cp} = \mu h.(z := 1).(z' := noiseRed(z)).\langle\langle z' \rangle\rangle \triangleright \{\ell_a\}.\ h$$



1. Store current sensor value into local variables: the picture

Control processes: car checkpoint

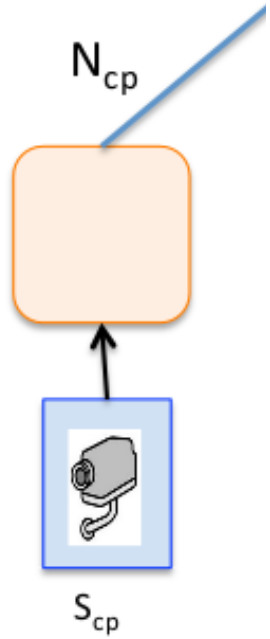
$$P_{cp} = \mu h.(z := 1).(z' := noiseRed(z)).\langle\langle z' \rangle\rangle \triangleright \{\ell_a\}.\ h$$



1. Store current sensor value into local variables: the picture
2. Enhance the taken picture

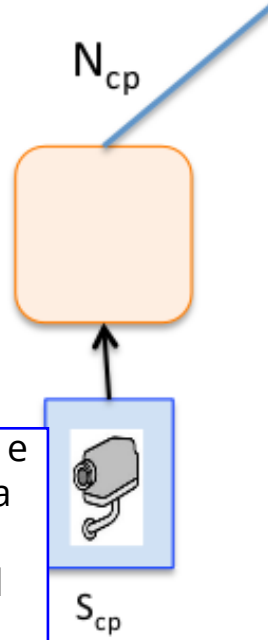
Control processes: car checkpoint

$$P_{cp} = \mu h.(z := 1).(z' := noiseRed(z)).\langle\langle z' \rangle\rangle \triangleright \{\ell_a\}.\ h$$



1. Store current sensor value into local variables: the picture
2. Enhance the taken picture

Control processes: car checkpoint



$$P_{cp} = \mu h.(z := 1).(z' := noiseRed(z)).\langle\langle z' \rangle\rangle \triangleright \{\ell_a\}.\ h$$

Dall'altra parte invece abbiamo il check point che di nuovo rileva le foto scattate dalla telecamera e va ad'applicarci una funzione che qui chiamiamo noise reduction per renderla più leggibile e poi la spedisce sopra, al supervisore degli accessi delle auto che sarà in grado di dire se quell'auto può passare oppure no, quindi il meccanismo è lo stesso, si migliora la fotografia e poi la si spedisce al supervisore che eventualmente passa al suo supervisore diretto

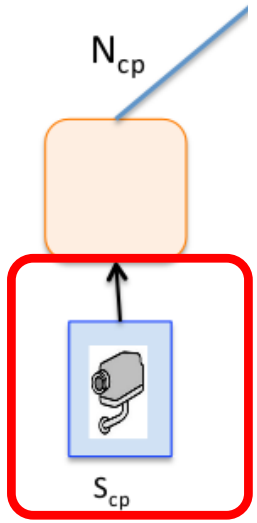
1. Store current sensor value into local variables: the picture
2. Enhance the taken picture
3. Send it to the supervisor of car access (N_a) that send it the supervisor (N_s)

L'analisi risponderà a domande del tipo: come vengono manipolate le foto, le immagini che la telecamera, nella parte sinistra del processo del sistema e scatta e manda più in alto e allo stesso tempo come queste immagini fluiscano nel sistema di comunicazione del sistema e vedremo che quindi si può andare a vedere la manipolazione dei valori, quindi in particolare chiameremo 1^cp il valore che mi rappresenta la foto che viene presa dalla telecamera e vedremo che a un certo punto il processo che sta sopra metterà in campo una tecnica di riduzione del rumore sulla foto e quindi applicherà una funzione a questo valore e poi vedremo i flussi, vedremo che sarà il supervisore che riceve le immagini dell'automobile e anche vedremo che cosa fanno i lampioni.

Analysis

Question

1. How are car pictures manipulated?
2. Where do car pictures flow in the system?



Value manipulation (1)

- 1^{cp} pictures taken by camera S_{cp}
- $noiseRed^{cp}(1^{cp})$ pictures from S_{cp} elaborated inside N_{cp}

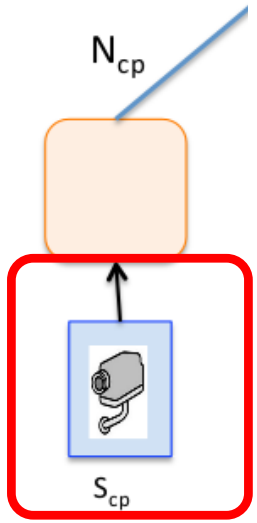
Flows (2)

- Supervisors receive car pictures and propagate them to lamp posts
- Lamp post

Analysis

Question

1. How are car pictures manipulated?
2. Where do car pictures flow in the system?



Value manipulation (1)

- 1^{cp} pictures taken by camera S_{cp}
- $noiseRed^{cp}(1^{cp})$ pictures from S_{cp} elaborated inside N_{cp}

Flows (2)

- Supervisors receive car pictures and propagate them to lamp posts

Quindi vedremo che ci può essere una violazione se queste immagini passano e invece devono essere considerate private.

- Lamp post (security violation if pictures are private)

Syntax of IoT-LySA (1)

Nodes

N	::=	0	nil node
		$\ell : [B]$	single node
		$N_1 \mid N_2$	composition

Andiamo a vedere adesso la semantica dell'IoT LySa, quindi vediamo come LySa viene alterato e modificato in modo da catturare questo tipo nuovo di informazioni. Allora vediamo intanto che, come dicevamo, c'è una struttura gerarchica, quindi in particolare abbiamo dei nodi che possono essere fatti, e questo ormai non è più una grande sorpresa, come nodo che non fa nulla, composizione parallela di nodi n1 | n2 e poi nodi etichettati, quindi in particolare ogni nodo verrà fuori con un'etichetta l e starà dentro una parentesi, il che vorrà dire che all'interno della parentesi ci sarà una nuova struttura sintattica ...

Syntax of IoT-LySA (1)

Nodes

N	$::=$	0	nil node
		$\ell : [B]$	single node
		$N_1 \mid N_2$	composition

Node body

B	$::=$	Σ_ℓ	store
		S	sensors
		A	actuators
		P	processes
		$B \parallel B$	composition

...e infatti abbiamo il body, il corpo del nodo, che è quello che contiene sostanzialmente il resto delle cose funzionali che mi servono, cioè processi che sovraintenderanno alla logica di tutto il nodo, i sensori, gli attuatori e poi, naturalmente questi processi potranno essere in parallelo, quindi io posso avere più processi, più attuatori, più sensori e infine ci sarà un'altro possibile elemento del corpo che è lo store, che non è altro che una sorta di memoria locale, quindi in questo modello esiste un concetto di memoria ed è un concetto che si applica localmente, cioè ogni nodo ha la sua personale memoria dove va a mettere le variabili sostanzialmente che raccoglie con i dati che raccoglie sia dalle segnalazioni dei sensori che gli stanno sotto sia dalle comunicazioni che ti fanno altri processi e altri nodi nel sistema. La struttura è gerarchica perché voi avete nodi in parallelo al cui interno i corpi sono articolati di nuovo in parallelo contenendo una parte di processi in parallelo e poi sensori, attuatori e la memoria. A questo punto vediamo come sono fatti i singoli sensori e gli attuatori.

Syntax of IoT-LySA (1)

Nodes

N	::=	0	nil node
		$\ell : [B]$	single node
		$N_1 \mid N_2$	composition

Node body

B	::=	Σ_ℓ	store
		S	sensors
		A	actuators
		P	processes
		$B \parallel B$	composition

Sensors

S	::=	0	nil
		$\tau.S$	internal action
		$i := v.S$	storing v
		h	iteration var.
		$\mu h.S$	iteration

Il sensore, ma l'abbiamo già visto nell'esempio, può essere un sensore che non fa niente oppure un sensore che comincia con un'azione interna cui come al solito non ci dobbiamo occupare oppure ha una variabile che è proprio sua identificata, dove può andare a fare la memorizzazione dei valori che raccoglie proprio in quanto sensore e poi naturalmente c'è il costrutto che mi dice che il suo comportamento si può iterare.

Syntax of IoT-LySA (1)

Nodes

In maniera molto simile abbiamo la sintassi degli attuatori, ho di nuovo l' attuatore che non fa niente, quello che fa l'azione vuota e poi ho una cosa che naturalmente il sensore non ha e che è che può ricevere un comando ed è un comando che appartiene all' insieme gamma quindi per esempio, nel caso del lampione, abbiamo visto che il comando poteva essere sia turnon che turnoff, quindi lui riceve un' indicazione che deve stare per forza nel range delle azioni che lui può compiere, quindi si può stabilire in ogni punto quali sono le azioni che utilmente il mio attuatore può eseguire. Infine c'è proprio l'attuazione vera e propria, che è gamma.A, quindi l' attuatore, a un certo punto riceve il comando di fare una certa cosa e poi la può attuare.

infatti, come abbiamo visto, i sensori sono in grado di non fare niente oppure di fare un un'azione interna, andare avanti, ma soprattutto di fare lo storing dei valori che mano a mano rilevano dall'esterno e poi naturalmente può avere un costrutto per iterare analogamente l' attuatore che in particolare può ricevere indicazioni dall'alto e di eseguire un certo comando, basta che sia un comando dentro gamma oppure può eseguire direttamente l'azione gamma minuscolo.

$\mu h.S$ iteration

Actuators

$A ::=$	0	nil
	$\tau.A$	internal action
	$(\lfloor j, \Gamma \rfloor).A$	command
	h	iteration var.
	$\mu h.A$	iteration
	$\gamma.A$	action

Quindi rispetto al LySa cambia che abbiamo una struttura, quindi abbiamo una sovrastruttura di nodi per cui quello che LySa poteva essere prima solo processi naturalmente abbiamo 3 nuove componenti che sono i sensori, gli attuatori e lo store.

Syntax of IoT-LySa (1)

Nodes

N	::=	0	nil node
		$\ell : [B]$	single node
		$N_1 \mid N_2$	composition

Node body

B	::=	Σ_ℓ	store
		S	sensors
		A	actuators
		P	processes
		$B \parallel B$	composition

Sensors

S	::=	0	nil
		$\tau.\, S$	internal action
		$i := v.\, S$	storing v
		h	iteration var.
		$\mu h.S$	iteration

Actuators

A	::=	0	nil
		$\tau.\, A$	internal action
		$(\!\lvert j, \Gamma \rvert\!).\, A$	command
		h	iteration var.
		$\mu h.\, A$	iteration
		$\gamma.\, A$	action

Syntax of IoT-LySA (2)

Processes

$P ::=$	$\langle\langle E_1, \cdots, E_k \rangle\rangle \triangleright L.\, P$	multi-output
	$(E_1, \cdots, E_j;\, x_{j+1}, \cdots, x_k).\, P$	input (with matching)
	$x := E.\, P$	assignment
	$\langle j, \gamma \rangle.\, P$	command to actuator j
	$\ldots$	nil, conditional, iteration

I processi sono abbastanza simili a quelli di LySa, la differenza è che ancora è un calcolo poliadrico, ancora ho il pattern matching all'interno degli input, ma c'è una piccola novità, cioè l' output ha una sintassi un po particolare, abbiamo la doppia parentesi angolate, vuol dire che io sto spedendo quella tupla e1...ek e la sto spedendo (vedete il triangolino rovesciato L) a tutte le etichette l minuscolo che fanno parte dell'insieme L, quindi io in realtà sto indicando che quell'output è un multioutput, cioè lo voglio mandare potenzialmente a più di un processo, quindi sto avendo una comunicazione del tipo uno a tanti ed è vedremo asincrona, cioè vedremo che questa è una comunicazione che avviene in due passi. Abbiamo la possibilità di mandare la stessa tupla a più di un processo, cioè a più di un nodo, l sono le etichette dei nodi e possono essere raggruppate naturalmente in insiemi. L'input è esattamente quello che vi potete immaginare, analogo in tutto e per tutto all' input con pattern matching di LySa e poi abbiamo l' assegnamento, quindi l' assegnamento è una caratteristica che in LySa non c'era, ricorda un pochino più il linguaggio imperativi e non a caso vedete nel corpo del nodo, che ho una componente che si chiama store quindi memoria quindi naturalmente io avrò delle variabili dello store che possono essere istanziate con certe espressioni E e poi continuare. L'altra cosa, naturalmente, che fa il processo è comunicare con gli altri processi di input e output, andarsi a copiare dei valori all'interno della sua memoria e poi comunicare con l' attuatore per chiedere eventuali comandi, poi le altre cose che può fare un processo sono quelle che uno si aspetta, quindi avrò un caso nhil per il processo che non fa niente, avrò un comando condizionale in cui può fare delle cose a seconda del valore booleano di una variabile e avrò la possibilità di iterare.

Syntax of IoT-LySA (2)

Di nuovo come in LySa e come in SPI calculus ho termini possibilmente strutturati, quindi in particolare io posso avere dei valori semplici delle variabili e fin qui non c'è niente di nuovo, però poi ho anche tra i termini possibili i famosi identificatori usati all'interno dei sensori e servono sostanzialmente per acquisire i valori raccolti dal sensore. Nel processo che controlla il lampione (slide 21), prima di decidere vado a raccogliere dal sensore 1, 2, 3, 4 i valori in quel momento rilevati. Vuol dire che posso inserire, per esempio nell assegnamento che vedete qua sopra, x prende i, l'ultimo caso della categoria delle espressioni è quello delle funzioni sui dati e queste funzioni le potete pensare, ad esempio come funzioni di aggregazione, quindi raccolgo temperature in vari punti e poi avrò una funzione che mi calcola la media oppure mi calcola se queste temperature mediamente sono sopra una certa soglia, quindi qui vi potete immaginare funzioni di aggregazione di ogni tipo.

Terms

E	$::=$	v	values
		i	sensor location
		x	variables
		$f(E_1, \cdots, E_n)$	function on data

la semantica è a riduzione non è etichettata semplicemente vediamo l'evoluzione a due livelli sia dei nodi che dei processi abbiamo ancora un concetto di congruenza strutturale sui sistemi e sugli elementi del sistema, che sono quelli che ci possiamo attendere, cioè anche i nodi sono monoidi comunicativi, abbiamo l' iterazione, poi, come abbiamo visto la volta scorsa, possiamo assimilare l' output verso un'insieme vuoto di etichette di nodi come direttamente il processo nhil, le valutazioni delle singole espressioni viene data in maniera simile alla semantica denotazionale.

The semantics of IoT-LySA

A 2-level reduction semantics $N \to N'$ and $P \to P'$ based on

- Structural congruence on system elements
  - standard axioms (e.g. nodes are commutative monoids)
  - iteration
  - multi-output with no receivers

$$\langle\!\langle E_1, \cdots, E_k \rangle\!\rangle \triangleright \emptyset.\ 0 \equiv 0$$

- Standard denotational interpretation $[\![E]\!]_\Sigma$ for terms

La semantica di IoT LySa è un semantica a riduzione che è di nuovo basata su una congruenza strutturale, quindi di nuovo abbiamo degli assiomi standard per esempio vi potete immaginare che la struttura parallela sia dei nodi che degli elementi del corpo del nodo abbiano tutte le varie proprietà monoidali, quindi sono commutativa associativa eccetera, elemento neutro è nhil, e poi potete immaginare che ci sono altri assiomi che potete aggiungere, in particolare potete vedere che laddove io voglio fare un multi output di E1...Ek verso un'insieme vuoto di etichette e poi proseguire come zero, vi potete immaginare che il comportamento di questo processo è assimilabile in tutto e per tutto al processo che non fa nulla proprio perché sto cercando di mandare un output all' insieme vuoto di processi. L'altra cosa che va presa come premessa è che non andiamo a dettagliare troppo, e qui non lo vedremo, come si analizzano i termini semanticamente, quindi supponiamo che ci sia una sorta di interpretazione denotazionale per ciascun termine che mi va a calcolare a che cosa corrisponde, quindi banalmente la denotazione di v sara v, la denotazione di x andrà a prenderne il contenuto e la valutazione di f(E1...En) andrà a calcolare f su tutti i pezzettini delle espressioni.

A glimpse of reduction rules

Node

$$\frac{B \rightarrow B'}{\ell : [B] \rightarrow \ell : [B']} \text{(Node)}$$

Body components reductions propagate to nodes

Vedete la semantica è a due livelli, proprio perché se la componente di un nodo può avanzare di un passo, quindi se da b può evolversi in b' allora la regola del nodo che sta sopra, quindi il secondo livello semplicemente conserva questo comportamento e semplicemente lo rinchiude sempre dentro il contesto etichettato con l, quindi le riduzioni dei componenti del corpo si propagano attraverso il livello superiore, che è quello dei nodi

A glimpse of reduction rules

a questo punto abbiamo visto la volta scorsa invece come si fa a memorizzare le informazioni che provengono dai sensori da parte di tutto il nodo che poi ci deve lavorare per fare le proprie valutazioni. In particolare se il sensore fa un assegnamento di un nuovo valore appena rilevato, la semantica fa sì che questo valore venga memorizzato dentro la memoria condivisa

Sensor communication

$$\text{(S-STORE)}\quad \frac{}{\Sigma \parallel i := v.S \parallel B \rightarrow \Sigma\{v/i\} \parallel S \parallel B}$$

A sensor stores the sensed value v in its reserved location i

A questo punto andiamo a vedere proprio l'idea della semantica che abbiamo a riduzione e vediamo solo le regole un pochino più caratteristiche di questa estensione del calcolo, quindi vediamo in particolare la regola della memorizzazione dei valori del sensore, quindi vi potete immaginare che siamo all'interno di un nodo, abbiamo che se ho sigma in parallelo con un sensore che ha i che prende v, allora io posso andare ad arricchire la mia store con il nuovo valore rilevato nella locazione riservata i, quindi questo è un modo per inserire nuovi elementi all'interno dello store.

invece il comando all' attuatore arriva attraverso questa regola semantica che mi dice se il processo che dà il comando gamma all'attuatore j è tale che gamma appartiene all' insieme gamma che l' attuatore j può fare allora semplicemente l'attuatore lo può recepire e successivamente metterlo tra le azioni che deve fare e quindi è quello che vedete nella regola. Invece vediamo come si comporta la interazione tra nodi.

A glimpse of reduction rules

Commands to actuators

$$\text{(A-COM)}\quad \frac{\gamma \in \Gamma}{\langle j, \gamma \rangle . P \parallel (\!| j, \Gamma |\!) . A \parallel B \rightarrow P \parallel \gamma . A \parallel B}$$

A process orders the actuator j to perform the action γ

In maniera simmetrica vi faccio vedere anche come funziona il comando attuatore, abbiamo da una parte e siamo sempre all'interno, vi potete immaginare di un nodo, vedete che all'interno di un nodo la composizione parallela è resa da una doppia barra verticale, quindi succede che ho da una parte il processo che controlla l'attuazione e quindi che comanda all'attuatore j di eseguire il comando gamma e dall'altra abbiamo una sorta di ricezione del comando dal lato dell' attuatore che sostanzialmente va a prendere il comando e lo accetta e poi lo fa eseguire solo se gamma fa parte dei comandi che lui può attuare, quindi lo riceve e successivamente, come vedete, abbiamo a che può eseguire l'azione gamma.

A glimpse of reduction rules

Interaction among nodes (sending a message)

(Ev-out)

$$\frac{\bigwedge_{i=1}^{k} v_i = [\![E_i]\!]_\Sigma}{\Sigma \parallel \langle\!\langle E_1, \cdots, E_k \rangle\!\rangle \triangleright L.\, P \parallel B \rightarrow \Sigma \parallel \langle\!\langle v_1, \cdots, v_k \rangle\!\rangle \triangleright L.0 \parallel P \parallel B}$$

1. Evaluate the expressions
2. Send the (asynchronous) message

Il messaggio diventa multi output e anche abbiamo detto che è asincrono, ovvero il fare questa azione di output, che vuol dire voglio mandare la tupla E1...Ek all' insieme di nodi con etichetta in l, semplicemente non continua con p come uno si aspetterebbe, ma semplicemente una volta che parte questa azione, quello che succede è che la tupla viene sganciata mentre il processo si libera e da lì in poi viene fatto in parallelo, quindi in particolare rimane diciamo a fluttuare dentro il sistema la tupla v1...vk che è il risultato della valutazione di tutti i vari elementi E1...Ek e rimane a disposizione di tutti i nodi che in parallelo corrono con il nodo che contiene questo output e che hanno l'etichetta tra quelle riservate in l.

Vediamo l'altra faccia della comunicazione, vediamo che cosa succede. Ci dobbiamo aspettare che una comunicazione di questo tipo multi viene eseguita in un contesto dove in un processo etichettato l1 ad esempio viene sganciata questa tupla e quindi non ci interessa più vedere la sua continuazione perché starà dentro la componente b1 che vi potete immaginare strutturata e dall'altra parte ho un nodo etichettato l2 la cui etichetta fa parte di l e che naturalmente espone l' input corrispondente, quindi una tupla della stessa dimensione con poi un'eventuale pattern matching. Allora nel caso in cui io vado a valutare, quindi ho v1...vk, poi vado a valutare i primi j elementi dell'input, quindi su cui devo fare i pattern matching e vale una certa condizione che ora vi spiego quello che succede che a questo punto la comunicazione può avvenire e quindi non leggo più le singole variabili in generale, ma le lego all'interno di sigma2 che è la store del nodo 2 e allo stesso tempo vado ad aggiornare la lista di chi deve ricevere il messaggio togliendo l2 perché l'ho appena preso. Quindi naturalmente uno si aspetta che ripetutamente finché non esaurisco tutti i nomi e tutte l'etichette all'interno di l, io vado a fare multicom con tutti i processi coinvolti. L'unica cosa che non vi ho spiegato in questo caso è che questa condizione Comp(l1,l2) sta a significare che io posso aggiungere sostanzialmente un tabella in cui dico se è possibile per due nodi poter comunicare questo può avere a che fare non tanto con la sicurezza, ma proprio quella fisicità perché se sono nodi che riescono a comunicare solo a basso range se magari la comunicazione dovesse avvenire tra due nodi che sono troppo lontani non ci sarebbero proprio le condizioni di compatibilità per fare questa comunicazione.

(Multi-com)

$$\frac{\ell_2 \in L \wedge \ Comp(\ell_1, \ell_2) \wedge \bigwedge_{i=1}^{j} v_i = [\![E_i]\!]_{\Sigma_2}}{\begin{array}{c}\ell_1 : [\langle\!\langle v_1, \cdots, v_k \rangle\!\rangle \triangleright L.0 \parallel B_1] \mid \ell_2 : [\Sigma_2 \parallel (E_1, \cdots, E_j; x_{j+1}, \cdots, x_k).Q \parallel B_2] \\ \rightarrow \ell_1 : [\langle\!\langle v_1, \cdots, v_k \rangle\!\rangle \triangleright L \setminus \{\ell_2\}.0 \parallel B_1] \mid \ell_2 : [\Sigma_2\{v_{j+1}/x_{j+1}, \cdots, v_k/x_k\} \parallel \ Q \ \parallel B_2]\end{array}}$$

1. Check if the node ℓ_2 is among receivers of the message and is compatible with the node ℓ_1, evaluate the expressions
2. In case of pattern matching bind the variables and update the set of receivers

Analysis overview

La prima cosa di diversa dall'analisi che abbiamo visto di CFA è che sarà un'analisi astratta, ora già è un'analisi astratta di suo, questa ha un grado maggiore di astrazione perché quello che mi porrò come obiettivo sarà non di controllare i valori concreti che vengono manipolati all'interno della mia rete del mio sistema, ma mi interesserà calcolare il flusso dei dati e la provenienza, quindi non mi interessa sapere ad esempio quali sono i valori calcolati e rilevati dal sensore ma mi interessa sapere che viene manipolato un valore che proviene da questo sensore e che viene manipolato essendo raccolto lì, essendo spedito che ne so al nodo che lo controlla che poi lo aggrega e lo spedisce a un livello superiore, eccetera eccetera quindi mi interesserà calcolare questo tipo di informazioni, quindi saranno valori astratti e mi interesserà capire in che forma vengono manipolati, quindi mi interessa sapere da dove provengono e in che modo vengono manipolati nei vari livelli gerarchici dei passaggi nel suo proprio sistema. Quindi la mia estimate sarà fatta di nuovo da una serie di componenti che sono sostanzialmente k, che ormai è noto e che proprio come sempre ci da l'astrazione dei messaggi che un nodo può ricevere, quindi adesso parliamo in termini di nodo e non di processo, ma vi potete immaginare che non sia molto diverso, poi abbiamo la abstract store sigma che mi dirà per ogni variabile quale potenzialmente possono essere i legami e poi avrò una astrazione dei dati che vengono raccolti a livello di ciascun nodo.

We abstract from concrete values and consider

- Where they come from (provenance)
- How they are manipulated and put together (shape)

An analysis estimate is made of

- An abstract store $\hat{\Sigma}$
  - Abstraction of data a node may store
- A network environment κ
  - Abstraction of messages a node may receive
- A data collection $\ominus$
  - Abstraction of data a node may compute

Abstract values

Trees with finite depth d

$$
\begin{array}{lll}
\hat{v} ::= & i^{\ell} & \text{data from sensor } i \text{ from the node } \ell \\
& v^{\ell} & \text{constant in node } \ell \\
& f^{\ell}(\hat{v}_1, \cdots, \hat{v}_n) & \text{function on abstract data in node } \ell \\
& \top^{\ell} & \text{term of depth greater than } d
\end{array}
$$

I valori astratti, quindi, sono fatti nel modo seguente, sono fatti in forma astratta, cioè io posso avere un valore i^l che sta per il generico dato che il sensore raccoglie al nodo l, quindi se fosse un sensore di temperatura, potrebbe essere 30 20 25 100 -50 quello che volete, ma non mi interessa saperne il valore, mi creerebbe troppo rumore all'interno della mia analisi, perché dovrei considerare tutti i possibili valori concreti, invece quello che mi interessa è tracciare il percorso di questi dati, quindi io identifico con i etichettato l tutti i possibili valori che il sensore può generare in quel punto lì, idem v^l, la costante nel nodo l, f^l(v1...vn) la funzione che aggrega a sua volta i valori v1...vn sempre valori astratti (hanno questo cappello sopra) e poi abbiamo anche la necessità che mano a mano che aggrego v1...vn possono essere a sua volta funzioni e quindi vi potete immaginare una struttura ad albero che rappresenta questi valori astratti allora a un certo punto l'analisi che vi presento qui decidiamo di fare un taglio, quindi a un certo punto taglio e quel valore astratto troppo fondo diventa un valore speciale che è top. Questo rappresenta il tipo di valori che vado raccogliendo, quindi i miei valori sono articolati quando poi vedo che sono troppo articolati, a un certo punto prendo un valore generico top che mi taglia come se facessi proprio un taglio netto nella profondità dell'albero questo lo faccio per motivi di semplificazione ovviamente, quindi a un certo punto se il termine è troppo strutturato decido di approssimarlo e lo approssimo con un generico valore T^l.

Analysis specification

The analysis is specified by three set of inference rules

1. $(\hat{\Sigma}, \Theta) \models_\ell E : \vartheta$ for expressions
2. $(\hat{\Sigma}, \kappa, \Theta) \models_\ell B$ for node internals
3. $(\hat{\Sigma}, \kappa, \Theta) \models N$ for nodes

- Abstract store $\hat{\Sigma}$
- Abstract data collection Θ
- Abstraction of an expression result ϑ
- Network environment κ

quindi l'analisi a questo punto viene specificata attraverso tre insiemi differenti di regole di inferenza, uno che mi controlla le espressioni, uno che mi controlla cosa succede all'interno dei nodi e uno per i nodi. In particolare, rispetto alle componenti che abbiamo visto prima, abbiamo anche un'astrazione di tutti i risultati per la singola espressione pensato un po come un tipo l'insieme di tutti i valori che può assumere a run time ed è theta minuscolo.

Andiamo a vedere la forma dell'analisi e vediamo che, visto che è così ricca la sintassi dell'analisi quindi la sintassi ha varie categorie sintattiche è chiaro che mi serve un jugement sia per le espressioni che per i componenti del corpo del nodo e per l'analisi dei nodi e quindi naturalmente possono essere diversi. Ora se voi guardate il judgement della singola espressione ricorda molto il judgement delle espressioni di LySa, solo che qui abbiamo sigma e theta mentre li avevamo rho sostanzialmente. Quindi che cosa mi raccoglierà theta? Mi raccoglierà tutti i possibili modi in cui E apparirà a run time, quindi laddove ci sarà una variabile rimpiazzerà con tutti i possibili valori che sigma di x sostanzialmente fornirà, invece l'analisi dei singoli nodi è quella che mi andrà a vedere nodo per nodo qual'è l'etichetta e lo passerà dentro, di modo che il judgement degli elementi del corpo dei nodi si porterà dietro l'etichetta l e questo se volete ricorderà il sistema che abbiamo usato in pi calculus per controllare la proprietà No read up e No write down, quindi se vi ricordate avevamo anche lì una struttura in cui ogni processo era racchiuso da parentesi angolate ed etichettato l quindi l'analisi, per tener conto di l, doveva quando trovava un processo con etichetta l, andarsela a ricordare nel simbolo che vedete qua con l minuscolo, in modo da scrivere a quel livello le azioni che faceva. Qui è molto simile, quindi se n viene fuori che è la composizione parallela di due nodi, uno con etichetta l e uno con etichetta l', quello che succede è che il judgement distribuirà le etichette quindi verrà fatta l'analisi del primo nodo sotto l'ipotesi l e l'analisi del secondo nodo sotto l'ipotesi l'.

A glimpse of analysis rules

Cominciamo dalle espressioni, quindi supponiamo che si vada a controllare prima l' identificatore allora l'analisi mi dice che sempre nell'ipotesi dell' indovinare la estimates e vedere se funziona, l'analisi è azzeccata quando mi dice che i sostanzialmente ha valori che fanno parte di theta, dove theta è un'insieme che deve contenere i^l perché sto facendo l'analisi dentro il nodo di etichetta l. Ora vedete che siccome i^l è un valore che appare nel nodo l, io vado a mettere questo valore anche nella componente theta_maiuscolo di l, che sostanzialmente mi va a radunare tutti gli elementi che emergono a livello del nodo l, cioè tutti quelli che vengono analizzati al livello del nodo l. In maniera simile, quando ho una variabile, questa variabile sarà ben predetta quando avrà valori che fanno parte dell'insieme theta e di nuovo theta dovrà contenere almeno v^l di theta e theta_maiuscolo di l ugualmente, quindi io calcolo il valore astratto e l'espressione risultante deve includere il valore astratto corrispondente e naturalmente devo ricordarmi in quale nodo sono. Pensatela un po come una regola di tipo, quindi il tipo di v è theta e l'analisi deve essere fatta in maniera tale che il suo corrispondente valore astratto deve essere ricordato dentro theta, che theta è locale a v e che ugualmente tutto ciò che sta in theta deve stare anche dentro theta_maiuscolo di l. Ora naturalmente questi due casi sono quelli chiusi, quindi sicuramente theta conterrà solo i^l nell'ipotesi di estimates minima e theta nel secondo caso conterà solo i^l. Vi potete invece immaginare che se avessi x allora dovrei metterci dentro l'analisi astratta tutti i possibili legami che astrattamente io prevedo per x , questo è un po l'idea ora non vi faccio vedere singolarmente tutte le regole perché sennò è over killing.

$$\frac{i^{\ell} \in \vartheta \subseteq \Theta(\ell)}{(\hat{\Sigma}, \Theta) \models_{\ell} i : \vartheta} \qquad \frac{v^{\ell} \in \vartheta \subseteq \Theta(\ell)}{(\hat{\Sigma}, \Theta) \models_{\ell} v : \vartheta} \qquad \frac{\hat{\Sigma}_{\ell}(x) \in \vartheta \subseteq \Theta(\ell)}{(\hat{\Sigma}, \Theta) \models_{\ell} x : \vartheta}$$

- Compute abstract value
- The expression result includes the abstract value
- Record that the node ℓ computes that value

A glimpse of analysis rules

Vediamo invece l'analisi dell' assegnamento, quindi della regola di assegnamento quindi andiamo adesso sui processi, quindi supponiamo di analizzare x che prende E, allora che cosa faccio? Sempre ricordatevi che l'analisi è fatta in maniera guidata dalla sintassi, quindi naturalmente io per prevedere che cosa può succedere quando a run time farò questo assegnamento, devo essere sicura che sia stata fatta l'analisi del valore sul lato destro e che una volta che io prevedo cosa succede nell'assegnamento, questa analisi deve continuare a essere valida anche per la continuazione P e quindi questa è la parte in rosso e la parte in azzurro è molto simile a quella che abbiamo visto per LySa. Invece in questo caso devo gestire la variabile in modo nuovo, perché adesso le variabili sono gestite dalla componente della memoria, quindi per ogni possibile valore di theta dove theta è l'insieme dei valori che possono essere assunti a run time da E, devo essere sicura che la mia analisi abbia previsto che possono essere legati alla variabile x, quindi la mia analisi è corretta quando prevede che in x possano finire tutti i valori v che sono legati possibilmente ad E. Ora che cosa succede? vi potrei far vedere altre regole ma sostanzialmente funziona, come vi potete aspettare, ci sarà la regola dell' input e dell' output che mette insieme le cose, quindi andrò a mettere in sigma,x per tutte le variabili dell'input andrò a mettere tutti i valori che stanno in sigma x, per esempio di E1...Ek insomma quindi i valori theta che sono legati a ciascuna delle variabili.

$$\frac{(\hat{\Sigma},\Theta) \models_\ell E : \vartheta \ \wedge \ \forall \hat{v} \in \vartheta \Rightarrow \hat{v} \in \hat{\Sigma}_\ell(x) \ \wedge \ (\hat{\Sigma},\kappa,\Theta) \models_\ell P}{(\hat{\Sigma},\kappa,\Theta) \models_\ell x := E.P}$$

- Analyse the expression E
- Store includes the abstract values possibly bound to E
- Analyse the continuation

Ho aggiunto le regole della comunicazione, guardando il caso binario per avere meno roba giro e quindi quando vado ad analizzare un processo che sta facendo multi output a tutti gli elementi di l quello che succede è vado a controllare se le informazioni sono state registrate in maniera corretta dalla mia stima, quindi, in particolare andrò a vedere a che cosa corrispondono nella mia analisi le due espressioni E1 ed E2 e poi quello che faccio vado a prendere tutti i possibili contenuti di E1 e E2, li mescoli in tutti i modi possibili e vado a ricordarmi che questi possono diventare tuple che vengono scambiate tra i due processi, in particolare vado a mettere che per ogni l' che può essere il destinatario di questa tupla di output, l è in grado di mandare la coppia v1,v2 ad l' e quindi in questo modo riesco a mantenere l'informazione sugli scambi, questo ricorda da vicino il tipo di analisi che facevamo per LySa, per gli output.

Output (binary case)

$$\frac{\bigwedge_{i=1}^{2} (\hat{\Sigma}, \Theta) \models_\ell E_i : \vartheta_i \ \wedge \ (\hat{\Sigma}, \kappa, \Theta) \models_\ell P \ \wedge \ \forall \hat{v}_i : \bigwedge_{i=1}^{2} \hat{v}_i \in \vartheta_i \Rightarrow \ell' \in L\,(\ell, \langle\langle \hat{v}_1, \hat{v}_2 \rangle\rangle) \in \kappa(\ell')}{(\hat{\Sigma}, \kappa, \Theta) \models_\ell \langle\langle E_1, E_2 \rangle\rangle \triangleright L.\, P}$$

- Analyse the expressions E_i
- record all the possible output tuples $\langle\langle \hat{v}_1, \hat{v}_2 \rangle\rangle$
- Analyse the continuation

allo stesso modo e vediamo come funziona l'analisi dell' input quando vado a controllare se la stima ha predetto correttamente il risultato di questo input, quello che succede è che devo andare a controllare se tutto ciò che può essere legato a E1, viene matchato da tutte le possibili componenti che possono fluire da chi fa l' output a chi sta facendo l' input e laddove la tupla corrisponde quindi v1 appartiene al theta1, allora io posso andare a controllare se sigma cappuccio di x2 contiene il secondo valore della tupla.

Input (binary case)

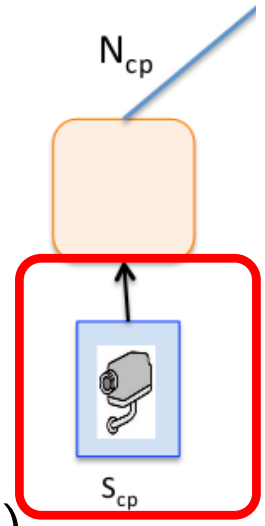
$$\frac{(\hat{\Sigma}, \Theta) \models_\ell E_1 : \vartheta_1 \ \wedge \ (\hat{\Sigma}, \kappa, \Theta) \models_\ell P \ \wedge \ \forall \ell' \in L : (\ell', \langle\langle \hat{v}_1, \hat{v}_2 \rangle\rangle \in \kappa(\ell) : \ \hat{v}_1 \in \vartheta_1 \Rightarrow \hat{v}_2 \in \hat{\Sigma}_\ell(x_2)}{(\hat{\Sigma}, \kappa, \Theta) \models_\ell (E_1; x_2).\, P}$$

- Analyse the expression E_1
- Store includes the abstract values possibly bound to E_2
- Analyse the continuation

Analysis: revisited

Question

1. How are car pictures manipulated?
2. Where do car pictures flow in the system?



Value manipulation (1)

- 1^{cp}: pictures taken by camera S_{cp} $(\hat{\Sigma}_{\ell_{cp}}(z) \ni 1^{\ell_{cp}})$
- $noiseRed^{cp}(1^{cp})$: pictures from S_{cp} elaborated inside N_{cp} $(\Theta(\ell_{cp}) \ni noiseRed^{\ell_{cp}}(1^{\ell_{cp}})$

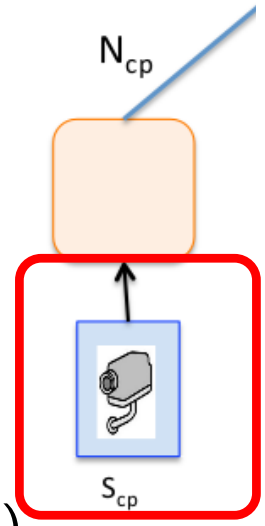
Flows (2)

- Car pictures are sent to N_a $(\kappa(\ell_a) \ni (\ell_{cp}, \langle\langle noiseRed^{\ell_{cp}}(1^{\ell_{cp}})\rangle\rangle)$ then to N_s and finally to lamp posts.
- Lamp post

Analysis: revisited

Question

1. How are car pictures manipulated?
2. Where do car pictures flow in the system?



Value manipulation (1)

- 1^{cp}: pictures taken by camera S_{cp} $(\hat{\Sigma}_{\ell_{cp}}(z) \ni 1^{\ell_{cp}})$
- $noiseRed^{cp}(1^{cp})$: pictures from S_{cp} elaborated inside N_{cp} $(\Theta(\ell_{cp}) \ni noiseRed^{\ell_{cp}}(1^{\ell_{cp}})$

Flows (2)

- Car pictures are sent to N_a $(\kappa(\ell_a) \ni (\ell_{cp}, \langle\langle noiseRed^{\ell_{cp}}(1^{\ell_{cp}})\rangle\rangle)$ then to N_s and finally to lamp posts.
- Lamp post (security violation if pictures are private)

Formal results

Quindi i risultati formali, come in ogni cfa, sono quelli che ci aspettiamo, quindi subject reduction l'analisi deve essere soundness, quindi l'analisi deve riflettere l'evoluzione dinamica ad ogni passo del mio sistema e l'esistenza di soluzioni, perché anche in questo caso abbiamo stime che si possono ordinare parzialmente e che costituiscono una Moore family con un elemento minimo e naturalmente a questi risultati possono seguire corollari attesi, del tipo che k correttamente approssima i messaggi che i nodi possono ricevere e theta approssima correttamente i dati che i nodi possono calcolare a livello di funzioni di aggregazione o altro.

Standard results

Subject reduction the analysis is sound w.r.t. the semantics

Existence of solution valid estimates are a Moore family

Corollaries

1. Θ correctly approximates the data nodes may compute
2. κ correctly approximates messages nodes may receive

Quali sono i risultati formali? Sono legati come sempre alla subreduction e all'esistenza delle soluzioni, quindi l'analisi è sound rispetto alla semantica, quindi di nuovo se io faccio l'analisi al punto di partenza, quello che succede che vado poi ad avere questa analisi sempre valida mano a mano che il sistema si evolve e anche in questo caso posso posso vedere che esiste sempre una soluzione perché le soluzioni si possono ordinare parzialmente e costituiscono ancora una Moore family come elemento minimo. Ora ovviamente, poi uno può andare ad aggiungere corollari a questi risultati formali, andando a vedere che la variabile theta approssima in maniera corretta i dati che i nodi possono calcolare, quindi per ogni nodo io avrò, come avete visto dalle poche regole che abbiamo discusso, ho tutti i dati che vengono calcolati a livello di quel nodo e k approssima correttamente tutti i messaggi che possono essere ricevuti dai nodi.

Contributions

The process algebra IoT-LySA extends LySA

- Network of nodes
- Sensors & actuators
- Group communication
- Local communication à la Linda

Tracking data analysis to predict

- Interaction among nodes
- How data flow in the network and are manipulated

Quindi questa algebra di processo estende LySa, lo fa attraverso una struttura gerarchica che LySa non aveva, per cui i processi si trovano ospitati all'interno di nodi che agiscono in parallelo, che quindi sono in zone diverse, probabilmente anche fisiche, i sensori e gli attuatori cominciano a far parte di questo quadro, abbiamo comunicazione di gruppo perché abbiamo il multicast, abbiamo una comunicazione locale alla Linda, cioè ogni singolo nodo ha la sua memoria, però i nodi tra di loro possono comunicare, così come facevano in LySa e quindi acquisire nuove informazioni che vengono non più lasciate semplicemente in variabili, ma messi in variabili che stanno dentro la store. Questa analisi viene utilizzata per fare il tracciamento dei dati, soprattutto interessa vedere come vengono manipolati e quindi quello che uno va a vedere è proprio la storia. Nel momento in cui voglio vedere che cosa succede (slide 21), io vado a seguire il tracciato dei valori degli identificatori e vado a vedere che questi passano dal sensore al nodo che lo controlla ma le decisioni che vengono fatte e che quindi vengono portate al livello ancora superiore, quindi a livello di questa struttura più complessa quindi diciamo che arrivano ai nodi supervisori quindi possiamo tracciare tutto il percorso che fanno questi dati. Questo è utile per molti scopi, quindi in particolare avevamo visto (alla slide 50) che la telecamera raccoglie dati, sta dentro il sensore, però vengono mandati al nodo CP che li elabora, eventualmente li può mandare anche più in alto e quindi tracciando i nodi tracciando le loro aggregazioni sono in grado di stabilire anche proprietà di sicurezza ad esempio o comunque posso stabilire come funzionano.

Security properties: examples

- Since car pictures are sensitive, we can check whether they are kept secret. By inspecting κ we discover that they are sent to all lamp posts, so possibly violating privacy
- We could verify if the predicted communications between nodes respect a given policy, e.g. that the interactions between the nodes Ns and Na are strictly unidirectional

Se vogliamo avere una proprietà di secrecy, ad esempio uno deve andare a vedere k e deve stabilire qual è la proprietà di secrecy a tempo statico e quale quella a tempo dinamico. Oppure possiamo avere proprietà di sicurezza che hanno a che fare con il flusso dei dati da una parte all'altra e quindi magari io potrei decidere di avere una certa policy e controllare se il mio sistema la rispetta ovvero potrei decidere che i flussi da N_s a N_a non si può andare da un l'uno all'altro e andare a controllare se il mio sistema fa esattamente quello che voglio. Altra cosa posso naturalmente estendere la mia analisi per darmi ulteriori livelli di dettaglio questo è quello che vediamo nella prossima lezione.

Further developments

Da leggere alla fine

- Tracking sensitive and untrustworthy data in IoT with a *taint analysis* of IoT-LySA
- Tracking data trajectories and provenance of data in IoT with IoT-LySA and with Klaim (language designed for specifying the behaviour of distributed and coordinated processes at a suitable level of abstraction)

Ulteriori sviluppi di questo tipo di analisi sono proprietà di sicurezza e in particolare una taint analisi su cui accennerò la prossima lezione e cui ci chiederemo visto che stiamo tracciando il percorso che fanno sostanzialmente i nostri dati, che cosa succede quando le decisioni di attuazione dipendono da dati che sono provenienti da fonti non trusted oppure che sono presi da sensori che possono essere per esempio tampered, allora uno può stabilire se alcune decisioni sono messe a rischio dal fatto che i valori su cui vengono basate possono non essere affidabili questo è un tipo di analisi che si può fare quindi proprio di tipo taint che avete già visto col professor ferrari e che quindi prendere quella che abbiamo visto come analisi di partenza, perché se pensate a come sono fatti i valori astratti (slide 43), potete immaginare che quindi potete seguire il flusso di questi dati che partono magari dai sensori e fluiscono mano a mano in questa gerarchia di nodi sempre più in alto, con aggregazioni diverse e poi sarà il risultato dell'aggregazione che sarà quello che porta a prendere una decisione, l'abbiamo vista anche qua che l'attuazione dell' accensione del lampione dipende da dati raccolti sopra, quindi io devo andare a controllare se passa un pedone, se le condizioni di luce lo richiedono e infine se c'è abbastanza batteria per poterlo accendere.