Corso di Elementi di Software Dependability
Proposta di elaborato
A.A. 2011-12

"In the railway signaling domain, an ***interlocking*** *is the safety-critical system that controls the movement of trains in a station and between adjacent stations. The interlocking monitors the status of the objects in the railway yard (e.g., points, switches, track circuits) and allows or denies the routing of trains in accordance with the railway safety and operational regulations that are generic for the region or country where the interlocking is located."*

Si richiede la modellazione di un algoritmo di interlocking distribuito.

L'algoritmo di interlocking si basa sul concetto di **itinerario** . Gli oggetti fisici sono modellati come **nodi** di una rete. L'itinerario è un concetto globale.
In prima approssimazione, ogni nodo conserva una tabella di itinerari a cui appartiene; un nodo è connesso ad almeno due nodi vicini.

Tipi di nodi (ente):
circuito di binario (track circuit): stato: (libero, *occupato* , riservato su route x)
incrocio (o tratto di binario interallacciato) : stato: libero, *occupato* , riservato su route x
scambio (point): stato: (normale, rovescio, comandato a normale, comandato a rovescio) in parallelo a (libero, *occupato* , riservato su route x)
segnale (signal): verde, *rosso,* comandato a verde, comandato a rosso

Topologia:

- un cdb è adiacente al più a due cdb/scambi, detti adiacente sinistro, adiacente destro (in caso di assenza di un adiacente, è un binario morto)
- A ha come adiacente destro B se e solo se B ha come adiacente sinistro A
- uno scambio è adiacente a tre cdb/scambi,
- uno scambio sinistro ha un adiacente destro, un adiacente normale sinistro, un adiacente rovescio sinistro
- uno scambio destro ha un adiacente sinistro, un adiacente normale destro, un adiacente rovescio destro
- un incrocio è adiacente a quattro cdb/scambi, detti adiacente normale sinistro, un adiacente normale destro, un adiacente

obliquo sinistro, un adiacente obliquo destro
- un segnale è connesso ad un unico cdb
- un cdb ha al più due segnali: segnale di uscita sinistro segnale di uscita destro (un segnale si comporta come segnale di protezione del cdb adiacente)

Queste regole impongono che il grafo diretto di adiacenza sia aciclico

- *Un itinerario è costituito da un cammino diretto formato da enti di binario adiacenti*
- *Un itinerario il cui primo nodo è A è un cammino possibilmente non completo dello spanning tree da A sul grafo di adiacenza.*
- *il primo nodo e l'ultimo nodo di ogni itinerario sono cdb*
- *il primo nodo definisce il punto di richiesta itinerario*
- *il primo nodo deve essere occupato alla richiesta*
- *la richiesta verifica la libertà degli enti successivi*
- *la richiesta viaggia solo in direzione destra o in direzione sinistra*

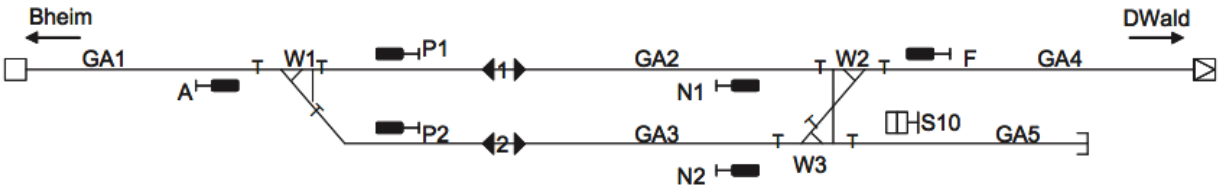
"The working of the 2-phase-commit protocol – based algorithm is basically the following:

- *A train is assumed to request a route at its start node. The request is propagated through adjacency by the nodes of the route, in a linear fashion.*
- *When the request reaches the end node, and if all the nodes are free, the acknowledge message flows back to the start node. At the end of this phase, all the nodes are reserved for the requesting train, and the points have been commanded to the proper position.*
- *A second round then is started by the start node to commit the reservation, and after the agree message has flowed back through all the nodes, a consensus message is given to the train that can move safely through the route.*
- *Either if a node is not free or if the movement of the point does not succeed (this is modelled by simple nondeterminism) the algorithm aborts the request, again by means of an abort message flowing back through the nodes of the route, and the train receives a no go message."*

Prima fase (a.a. 2009-10): solo richiesta e occupazione, assenza di segnali. Nello stato iniziale, gli enti di binario sono tutti liberi; l'utente può posizionare liberamente treni sui cdb, con uno specifico evento su ogni nodo cdb. Dopo la fase di configurazione, parte l'algoritmo vero e proprio, in cui e' possibile richiedere un itinerario mediante richiesta al nodo punto di inizio. Al termine del bloccamento di un itinerario (e

durante il bloccamento, vista la natura distribuita e quindi concorrente dell'interlocking), si possono anche richiedere altri itinerari. In ogni istante si può ritornare allo stato iniziale con un pulsante di reset
Seconda fase (2010-11): liberazione e distruzione itinerari:
distruzione : nel nodo in cui si e' prevista la richiesta iniziale di un itinerario, e' prevista anche la sua distruzione, che puo' essere effettuata solo prima del commit finale
liberazione : il movimento di un treno da un ente a quello adiacente di un itinerario bloccato libera gli enti precedenti, che sono quindi disponibili per una prenotazione di un nuovo itinerario.
Terza fase (2101-2012): introduzione dei segnali, focus sulla verifica:
I segnali si trovano di default nello stato rosso.
Nella fase di commit, ogni segnale che si trova su un itinerario viene comandato a verde. Dal momento che vi è la possibilità che un segnale si guasti, un segnale comandato a verde che non diventi verde a causa di un guasto non permette il commit.
Un sganle disposto a verde è comandato a rosso appena è transitato il treno che aveva prenotato il realtivo itinerario. Un segnale verde comandato a rosso che non si disponga a verde per effetto di un guasto non permette la liberazione dell'itinerario.

La topologia di stazione da considerare è quella in figura:



con la legenda:
GA1, GA2, …, GA5: circuiti di binario
P1, P2, F, S10 : segnali (incontrati da un treno che viaggia verso sinistra)
A, N1, N2: segnali (incontrati da un treno che viaggia verso destra)
W1, W2, W3: scambi
Gli itinerari da considerare sono:
1: GA1-A-W1-GA2
2: GA1-A-W1-GA2-N1-W2-GA4
3: GA1-A-W1-GA3
4: GA1-A-W1-GA3-N2-W3-W2-GA4
5: GA1-A-W1-GA3-N2-W3-GA5
6: GA4-F-W2-GA2
7: GA4-F-W2-GA2-P1-W1-GA1
8: GA4-F-W1-GA3
9: GA4-F-W1-GA3-P2-W1-GA1
10: GA5-S10-W3-GA3
11: GA2-P1-W1-GA1
12: GA2-N1-W2-GA4

13: GA3-P2-W1-GA1
14: GA3-N2- GA3-P2-W1-GA1
15: GA3-N2-W3-GA5

Ai fini di semplificare il progetto, si può considerare solo un sottoinsieme significativo di questi itinerari.

Attraverso gli strumenti di verifica disponibili, o attraverso adeguate simulazioni, si provi la proprietà di *stabilizzazione:*

"per ogni richiesta di itinerario compiuta da un treno posto al punto di inizio dell'itinerario, esiste una computazione che porta il treno al punto di fine dell'itinerario, e tutte le computazioni alternative (per effetto di guasti o altri motivi) producono un abort della richiesta."