



Automazione industriale dispense del corso

19. Supervisione e controllo: LD e SFC

Luigi Piroddi
piroddi@elet.polimi.it

Introduzione

Nella modellizzazione con reti di Petri di un processo discreto abbiamo distinto il modello delle sequenze produttive (o ricette) dal modello di allocazione delle risorse (e dall’eventuale controllo supervisivo):

- ▶ il *modello delle ricette* rappresenta le sequenze desiderate di operazioni
 - ▼ è dettagliato quanto basta per descrivere correttamente il meccanismo di allocazione delle risorse
 - ▼ il dettaglio delle singole operazioni è rappresentato in altri moduli di livello inferiore del sistema di controllo (*controllore*)
- ▶ il *modello di allocazione delle risorse* rappresenta i vincoli a cui le operazioni sono soggette a causa della limitatezza delle risorse
- ▶ il *controllo supervisivo* complementa i due modelli con vincoli aggiuntivi (assimilabili a risorse virtuali)

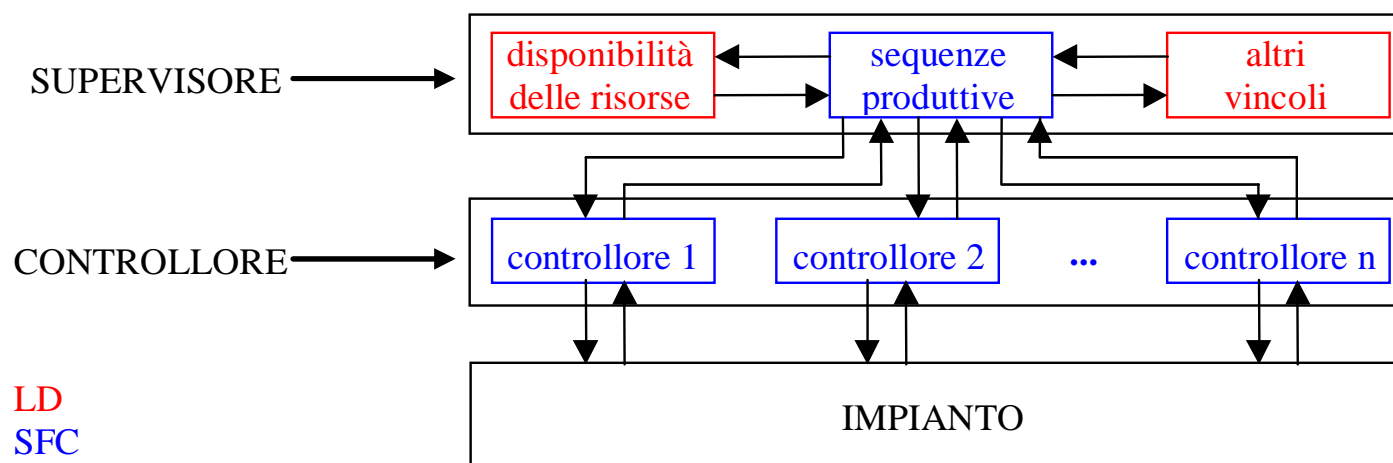
Il modello a rete di Petri risultante dalla composizione di questi elementi, che serve a gestire la attivazione delle operazioni nel rispetto dei vincoli, costituisce il livello superiore del sistema di controllo (*supervisore*).

SFC e LD

Un programma SFC descrive il ciclo operativo del sistema di controllo, prescrivendo come cambia lo stato del sistema a fronte di eventi. A differenza di LD, in SFC non si “vede” il ciclo operativo del PLC.

Di conseguenza, è innaturale esprimere dei vincoli in SFC, mentre è comodo rappresentare delle sequenze. Nel progettare un sistema di controllo useremo:

- ▶ SFC per le sequenze (sequenze prodotti, sequenze singole operazioni)
- ▶ LD per i vincoli (p.es. disponibilità di risorse, altri vincoli di supervisione)



Esempio

Si consideri un processo di preparazione di un dolce con una macchina semi-automatica. La sequenza produttiva principale prevede 3 operazioni:

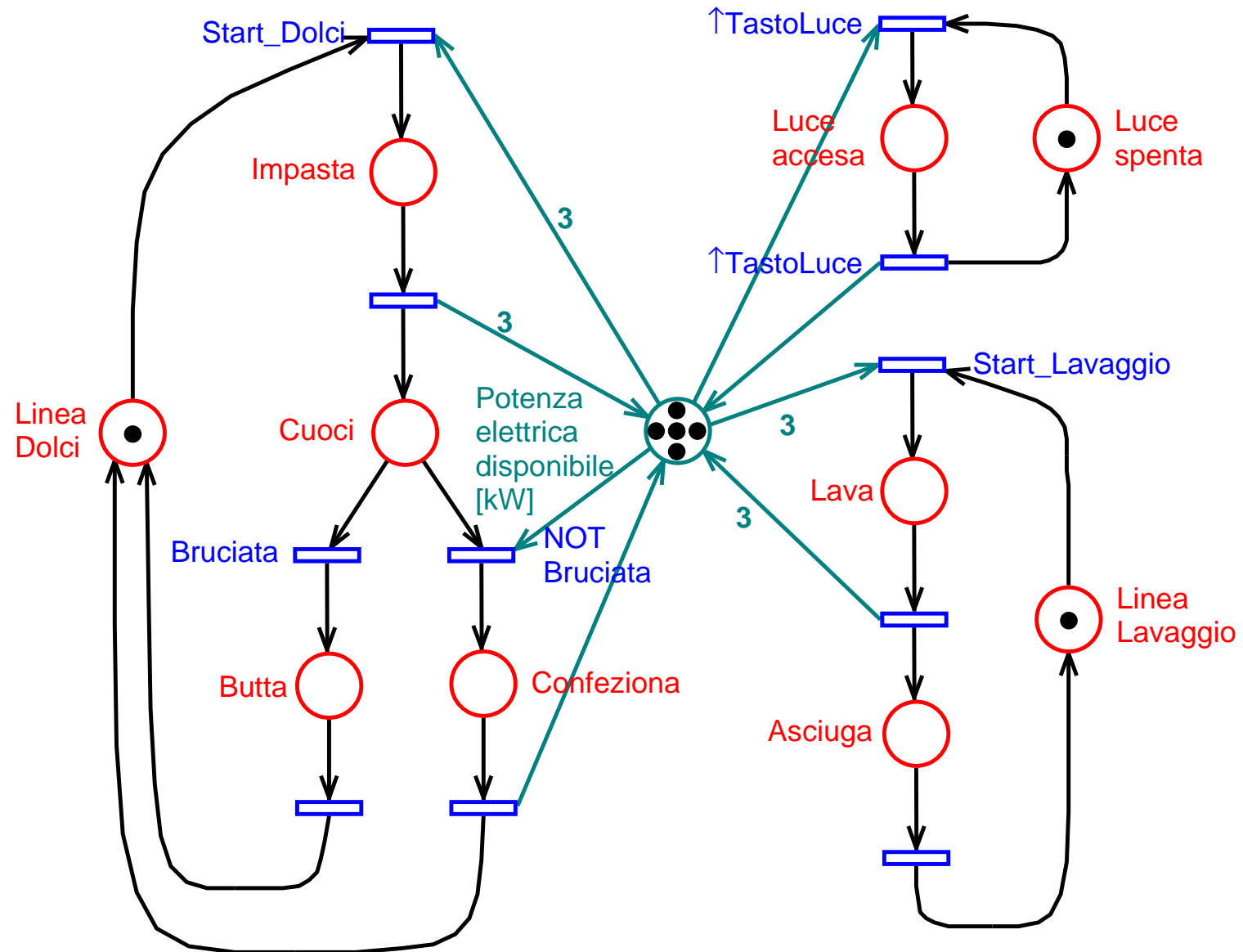
- ❶ impastamento del dolce
- ❷ cottura
- ❸ confezionamento (oppure eliminazione nel caso il dolce si bruci).

Inoltre, l'operatore può attivare altre due sequenze accessorie:

- ▶ un ciclo di lavaggio
- ▶ l'accensione la luce

Ciascuna delle 3 sequenze va eseguita per intero prima di poter essere riattivata, ma ci sono dei vincoli sulla potenza massima massima utilizzabile per le operazioni, pari a 5 kW (risorsa condivisa):

- ▶ le operazioni di impastamento e lavaggio richiedono 3 kW ciascuna
- ▶ confezionamento e luce richiedono 1 kW



Vincoli:

- ▶ i posti risorsa *Linea Dolci*, *Linea Lavaggio* e *Luce spenta*, determinano, con la loro marcatura e i loro collegamenti al resto della rete, delle pre-condizioni all'esecuzione di alcune operazioni, costituendo quindi una parte del supervisore
- ▶ c'è un vincolo sulla potenza massima utilizzabile per le operazioni, pari a 5 kW (risorsa condivisa); il posto che rappresenta la potenza disponibile determina degli ulteriori vincoli alle varie operazioni, completando il supervisore

Si noti che gli eventi “*Bruciata*” e “*NOT Bruciata*” sono non controllabili.

La rete non si blocca e quando ci sono conflitti effettivi il supervisore può risolverli. Tuttavia, la rete non è a scelta libera:

- ▶ non lo è per via dell'esistenza del vincolo sulla potenza
- ▶ ad esempio, se non si sta impastando il supervisore è libero di scegliere se consentire l'accensione della luce o il lavaggio, se si sta impastando, no

Il supervisore “vede” un impianto esteso:

- ▶ per realizzare il supervisore non serve (e non è opportuno) affinare tutte le attività
- ▶ è sufficiente che si vedano gli eventi che esso deve osservare e/o controllare

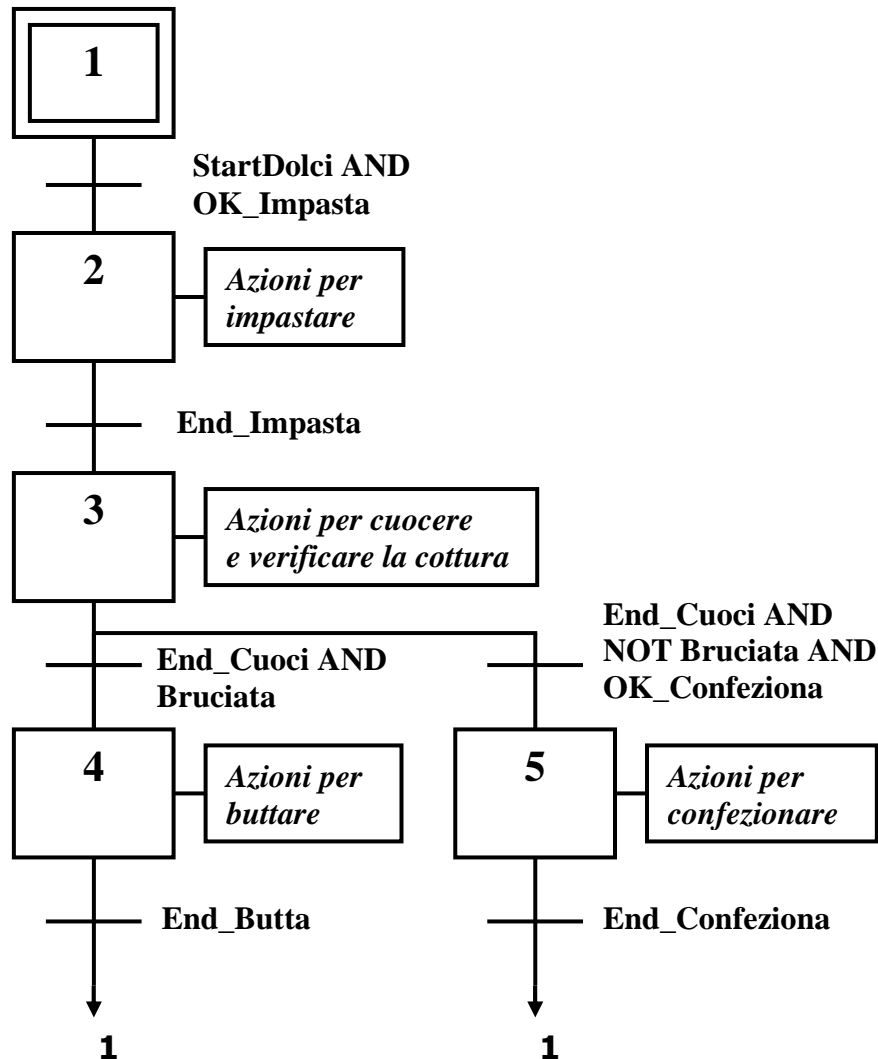
Il supervisore deve comandare le sequenze produttive, preoccupandosi di verificare se le condizioni per svolgere le operazioni sono verificate.

Tali condizioni hanno a che fare principalmente con il sequenziamento delle operazioni e con la disponibilità delle risorse necessarie al loro svolgimento.

A livello di implementazione, il supervisore dovrà risolvere i conflitti dando o negando i *consensi* all’inizio delle attività (e questo si farà in LD):

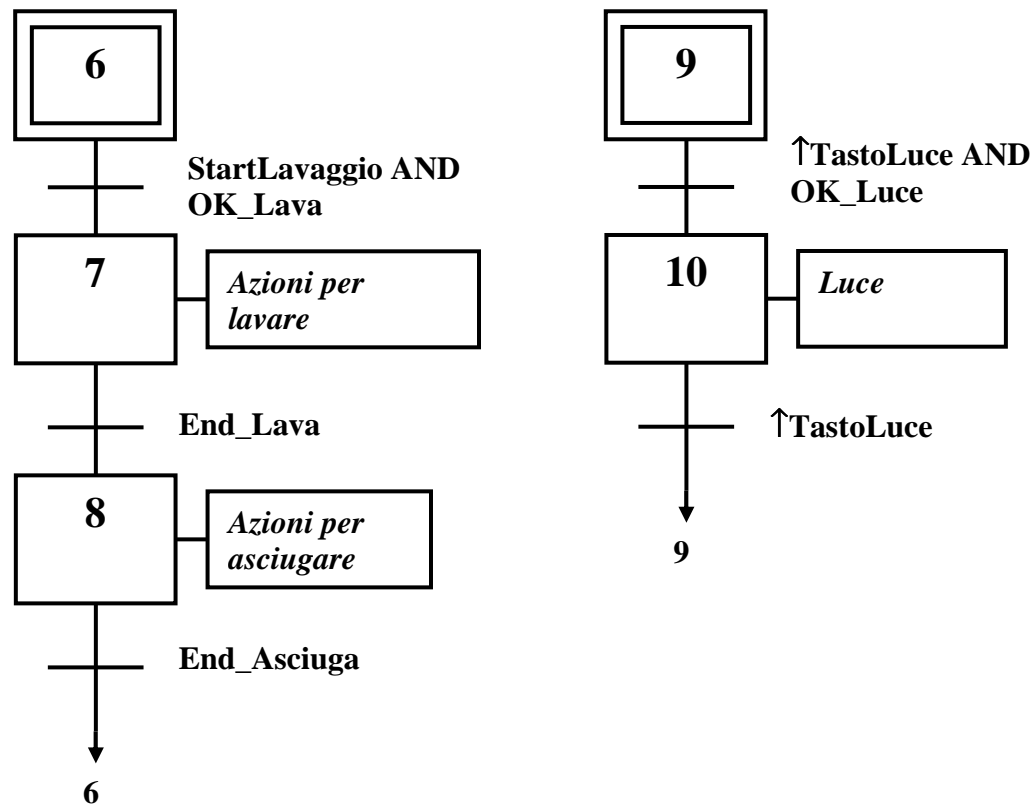
- ▶ il codice finale dovrà rendere o meno superabili alcune transizioni (abilitate) dello schema SFC che descrive le sequenze produttive
- ▶ nel codice si possono anche rendere superabili tutte le transizioni abilitate (ad esempio, quando è concesso dal vincolo, si può consentire contemporaneamente l’inizio del lavaggio e l’accensione della luce)
- ▶ questo differenzia il codice del PLC (sincrono) dal suo modello (asincrono), di cui è l’implementazione
- ▶ peraltro, lo scatto contemporaneo di tutte le transizioni abilitate di una rete di Petri non è in contraddizione con la scelta casuale purché esse non siano in conflitto

Controllo delle sequenze (SFC): linea dolci



Comandi dell'operatore	StartDolci
Consensi dal supervisore	OK_Impasta OK_Confeziona
Segnali di eventi dall'impianto	End_Impasta End_Cuoci End_Butta End_Confeziona
Segnali di condizione dall'impianto	Bruciata

Controllo delle sequenze (SFC): linea lavaggio e luce



Comandi dell'operatore	StartLavaggio ↑TastoLuce
Consensi dal supervisore	OK_Lava OK_Luce
Segnali di eventi dall'impianto	End_Lava End_Asciuga
Segnali di condizione dall'impianto	

Le azioni non sono descritte in dettaglio: possono essere macrofasi, oppure SFC locali sincronizzati con questo SFC.

I vincoli imposti dai posti risorsa Linea Dolci, Linea Lavaggio e Luce spenta sono automaticamente rappresentati nei 3 SFC dalle variabili di consenso.

Implementazione dei vincoli (LD)

Per eseguire il programma SFC di gestione delle ricette, ad ogni ciclo il PLC “esplora” lo schema SFC e fa quanto segue:

- ❶ esegue le azioni corrispondenti alle fasi attive
- ❷ determina le transizioni superabili
- ❸ aggiorna lo stato, ovvero l’insieme delle fasi attive

Pertanto, è logico che il codice LD che descrive i vincoli preceda il codice SFC nel programma del PLC (per poter definire il valore della variabili di consenso).

Esso svolge le seguenti operazioni ad ogni ciclo:

- ❶ aggiornare il suo stato (ovvero l’allocazione delle risorse)
in base agli eventi osservati (che temporalmente sono avvenuti nel ciclo prima)
- ❷ determinare quali sono i consensi da dare o negare in base allo stato suo e del resto del sistema (controllo e impianto), risolvendo gli eventuali conflitti in essere
- ❸ aggiornare nuovamente il suo stato in base ai consensi dati (e che produrranno effetto nella corrente esplorazione del codice SFC, stanti le regole di evoluzione di quest’ultimo)

Aggiornamento dello stato del supervisore in base agli eventi osservati

Per comodità, prima di scrivere il LD vero e proprio, lo esprimeremo come pseudo-codice (se si scrivesse il supervisore in ST, cosa possibile, questo sarebbe praticamente già il codice).

```
if End_Impasta,          kWdisp = kWdisp+3, endif
if End_Confeziona,      kWdisp = kWdisp+1, endif
if End_Lava,            kWdisp = kWdisp+3, endif
if X10 AND ↑TastoLuce, kWdisp = kWdisp+1, endif
```

Lo stato del supervisore è costituito dal numero di unità disponibili della risorsa, associata alla potenza disponibile.

La variabile kWdisp rappresenta tale stato e si assume inizializzata a 5.

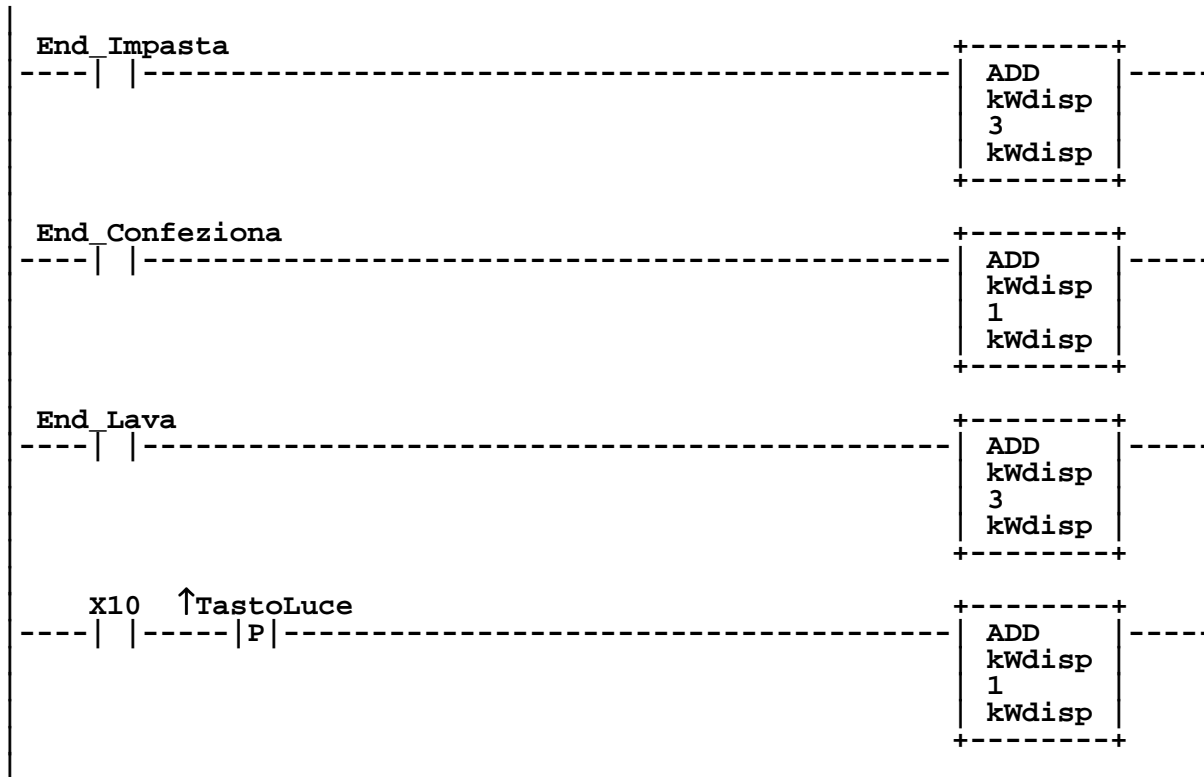
Determinazione dei consensi e ri-aggiornamento dello stato del supervisore

Si assume che i consensi dati producano effetto nel ciclo PLC in corso (si usa un meccanismo di priorità banale, implementato attraverso l'ordine delle istruzioni, come poi si fa in LD).

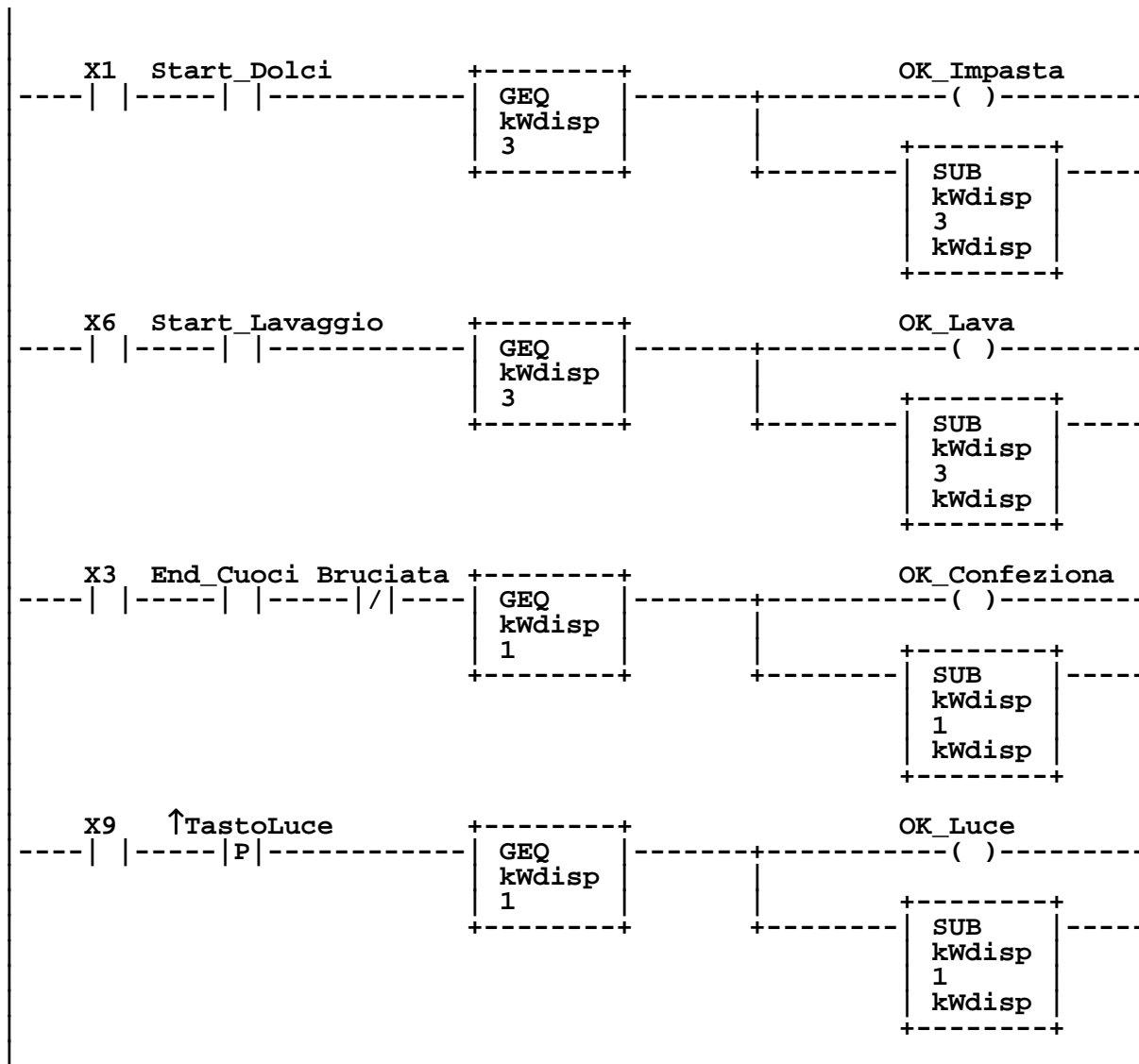
```
OK_Impasta = 0
OK_Confeziona = 0
OK_Lava = 0
OK_Luce = 0
if (X1 AND StartDolci) AND kWdisp>=3
    OK_Impasta = 1, kWdisp = kWdisp-3, endif
if (X6 AND StartLavaggio) AND kWdisp>=3
    OK_Lava = 1, kWdisp = kWdisp-3, endif
if (X3 AND End_Cuoci AND NOT Bruciata) AND kWdisp>=1
    OK_Confeziona = 1, kWdisp = kWdisp-1, endif
if (X9 AND ↑TastoLuce) AND kWdisp>=1
    OK_Luce = 1, kWdisp = kWdisp-1, endif
```

Traduzione in LD del supervisore per la gestione delle risorse

Aggiornamento dello stato del supervisore in base agli eventi osservati:



Determinazione dei consensi e ri-aggiornamento dello stato del supervisore:



La porzione di codice LD che gestisce l’allocazione delle risorse e determina i consensi va inserita prima della sezione di valutazione delle transizioni, dato che queste ultime dipendono dai consensi.

Poiché dipende solo dagli ingressi e dallo stato corrente del SFC può essere inserita subito all’inizio di ogni ciclo, dopo la lettura degli ingressi.

La struttura del codice prodotto non corrisponde soltanto al modello formale, ma anche alla struttura di programma adottata da diversi ambienti di sviluppo IEC 61131-3.

Ad esempio in ISaGRAF è naturale mettere il codice LD del supervisore nella sezione “begin” e il codice SFC del supervisore e del controllore nella sezione “sequential”.