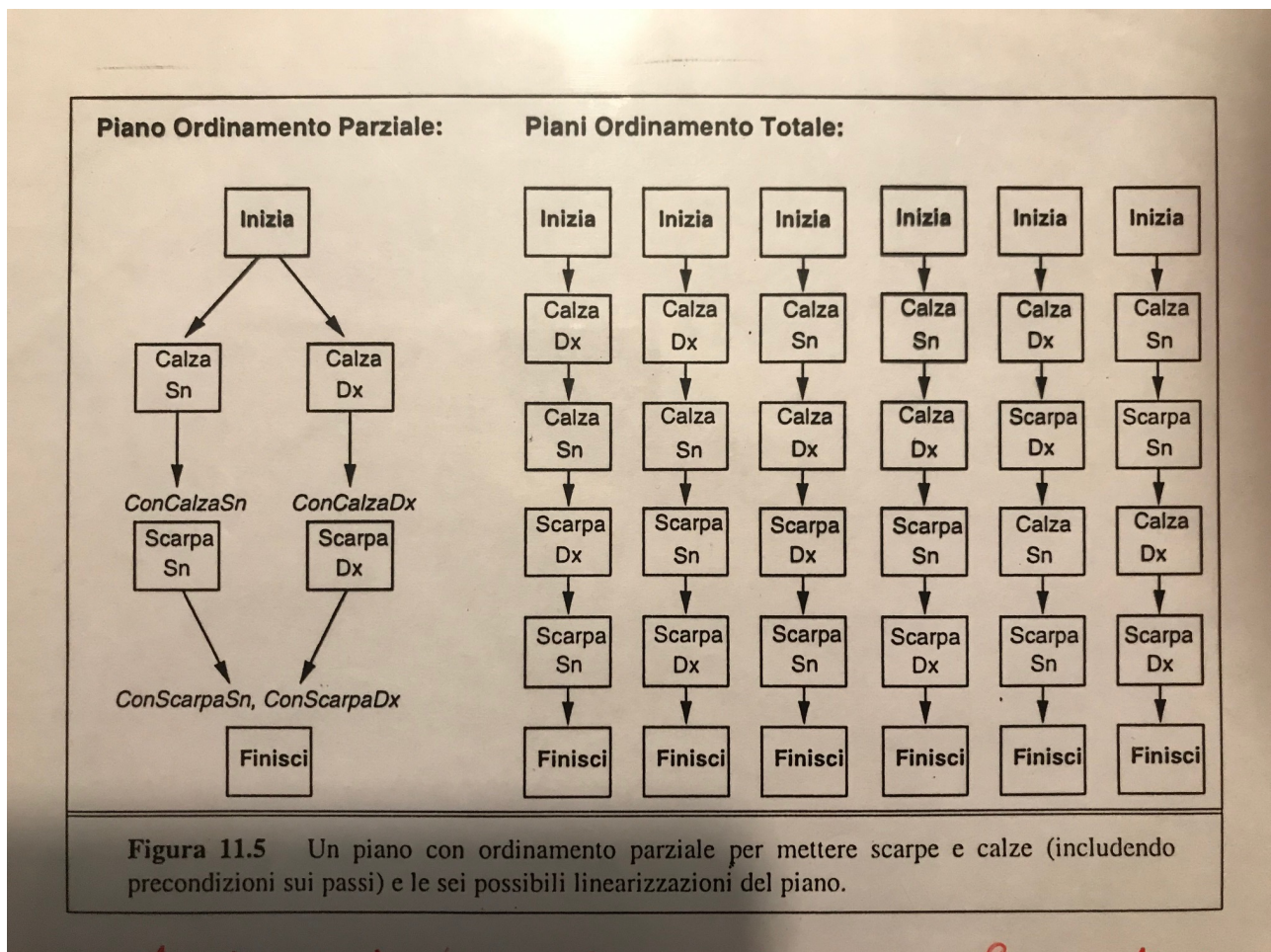


Pianificazione con Ordinamento Parziale (POP)

Perché ordinamento parziale?

- Le azioni in un piano non devono necessariamente essere sequenziali
 - Esempio “calzare calze e scarpe” (vedi lavagna)
- L’ordinamento parziale consente parallelismo nell’esecuzione
- Generare piani con ordinamento parziale è meno vincolante:
 - Ordinamento totale deciso successivamente in fase di esecuzione
 - Piano meno vincolato: maggiore efficienza in generazione?



Rappresentazione Piani in POP

- Quadrupla formata da:
 - Insieme di **operatori** (passi/azioni del piano)
 - Insieme **vincoli di ordinamento** tra i passi. Es.: $S1 < S2$
 - Insieme di **vincoli sui legami dei parametri** negli operatori (opzionale: solo se gli operatori *non* sono “pre-istanziati” o “groundizzati”):
ad es., ?Truck = Truck208
 - Insieme di **relazioni causali**, ad es.:
“S1 supporta la precondizione C di S2” $S1 \xrightarrow{C} S2$
Memorizzano a cosa serve un operatore inserito nel piano

Rappresentazione piani in POP

Azioni “speciali”:

- **INIZIA:**
 - no precondizioni
 - effetti = i fatti veri nello stato iniziale
- **FINISCI:**
 - Precondizioni = i goal del problema
 - No effetti
- INIZIA e FINISCI appartengono a tutti i piani
- Piano iniziale in POP contiene sono queste azioni (esempio piano iniziale problema calze-scarpe alla lavagna)

Piano Soluzione in POP

- **Soluzione** = piano completo e consistente
- **Completezza** = ogni precondizione C di un passo/azione S_j nel piano è “raggiunta” (soddisfatta) da un altro passo S_i nel piano:
 - (1) $S_i < S_j$ e C è un effetto di S_i
 - (2) Non esiste S_k t.c. C in EffettiNegativi(S_k) e $S_i < S_k < S_j$ è consistente (esiste una linearizzazione del piano in cui vale $S_i < S_k < S_j$)
- **Consistenza** = l’insieme dei vincoli di ordinamento (e degli eventuali legami dei parametri) è consistente. (Esempi alla lavagna)

“Minacce” in un piano POP (con operatori pre-istanziati)

- Un passo S_k nel piano **minaccia** una relazione causale $S_i \xrightarrow{C} S_j$ nel piano se:
 - $C \in \text{EffettiNegativi}(S_k)$
 - $S_i < S_k < S_j$ è consistente con i vincoli di ordinamento temporale nel piano
- Una minaccia può essere risolta
 - *promuovendo* S_k (aggiungo $S_j < S_k$)
 - *retrocedendo* S_k (aggiungo $S_k < S_i$)
- Esempi alla lavagna

Algoritmo POP (con operatori pre-istanziati)

- Esempio Mondo dei Blocchi: vedi lavagna
- Schema e pseudocodice: vedi lavagna

Estensioni algoritmo POP

- L' algoritmo visto è facilmente estendibile per gestire un linguaggio contenente:
 - Effetti con variabili quantificate universalmente
 - Negazione nelle precondizioni
 - Disgiunzione nelle precondizioni
 - Effetti condizionali
- Esempi di questi costrutti (alla lavagna)

Gestione Effetti Condizionali in POP

- Ad es., azione “rilascia(vaso)”:
“Se l' oggetto è fragile allora si rompe”
When fragile(vaso), rotto(vaso)
When not(incollato(vaso)), not(in-mano(vaso))
- Risoluzione di minacce:
 - Se mi serve usare l' effetto *rotto(vaso)* allora *fragile(vaso)* viene aggiunto alle precondizioni dell' azione
 - Se *rilascia(vaso)* minaccia una relazione causale con condizione *in-mano(vaso)*, posso aggiungere *incollato(vaso)* alle precondizioni dell' azione minacciante...
 - Pseudocodice alla lavagna...