Università di Parma
Dipartimento di Ingegneria e Architettura

Introduzione all'Intelligenza Artificiale

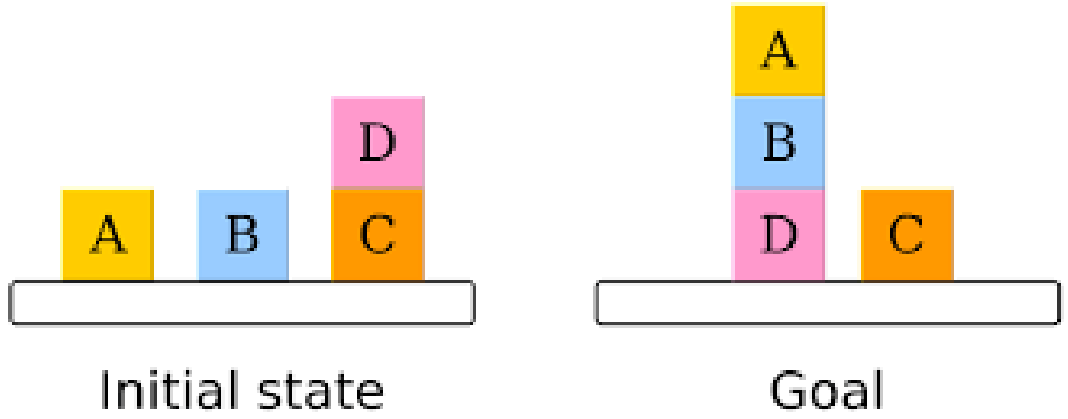
Big Data & Business Intelligence

A.A. 2022/2023

Corso di «Introduzione all'Intelligenza Artificiale»
Corso di «Big Data & Business Intelligence»

Reasoning and Planning

Monica Mordonini (monica.mordonini@unipr.it)



PIANIFCAZIONE

CENNI

Pianificazione

- Sequenza di azioni capace di raggiungere un obiettivo
- Abbiamo visto
  - Risolutore di problemi basato sulla ricerca
  - Pianificatore logico

Pianificazione

- Abbiamo definito l'IA come lo studio dell'azione razionale, il che significa che la pianificazione, cioè l'elaborazione di un piano d'azione per raggiungere i propri obiettivi, è una parte fondamentale dell'IA.
- Sequenza di azioni capace di raggiungere un obiettivo
- Abbiamo visto
  - Risolutore di problemi basato sulla ricerca
  - Pianificatore logico

Pianificazione:
Caratteristiche dei Problemi Reali

- Sono complessi:
  - Non è pensabile l'utilizzo del metodo della ricerca nello spazio degli stati o del pianificatore logico senza l'utilizzo di un altro attore: il pianificatore
  - Il pianificatore si occupa della ***scalabilità*** delle teniche di pianificazione

- Ma necessita di metodi per:
  - scomporre il problema.
  - applicare gli operatori in base alla parte dello stato del problema che li riguarda (frame problem).
  - trattare la conoscenza incompleta e l'evoluzione non del tutto controllabile e prevedibile dello stato.

Caratteristiche dei Problemi Reali

- Come possono essere affrontati questi problemi?

  1. Considerando tutti i possibili piani.
  2. Selezionando un piano con alta probabilità di successo.

- Se si esegue una azione di un piano e l'azione fallisce?

  √ Ripianificare dallo stato in cui l'azione ha fallito.
  √ Fare delle modifiche locali e mantenere il resto del piano.

Pianificazione

- ***Linguaggio dei problemi di pianificazione***
  - **Rappresentazione degli stati**
    - Si scompone il mondo in condizioni logiche e uno stato è dato come congiunzione di letterali positivi
    - (es, povero & sconosciuto &triste)
  - **Rappresentazione degli obiettivi**
    - Uno stato parzialmente specificato che deve essere soddisfatto da uno stato
    - (povero & sconosciuto obiettivo raggiunto nello stato povero & sconosciuto &triste)
  - **Rappresentazione delle azioni**
    - Specificata per mezzo delle precondizioni che devono valere prima della sua esecuzione e degli effetti che ne scaturisce

Funzioni Base dei Pianificatori

- Scegliere l'operatore migliore per produrre l'azione che avvicinera' l'agente all'obiettivo.
- Applicare un operatore e calcolare il nuovo stato.
- Individuare quando la soluzione è stata trovata.
- Individuare i vicoli ciechi.
- Individuare quando la soluzione è quasi corretta e applicare delle tecniche speciali per renderla corretta.

Applicare un Operatore e Calcolare il Nuovo Stato

- La rappresentazione dei problemi di pianificazione (stati azioni obiettivi) dovrebbe permettere agli algoritmi di trarre vantaggio dalla struttura logica del problema

- La chiave sta nel trovare un linguaggio sufficientemente espressivo da descrivere una grande varietà di problemi, ma abbastanza ristretto da essere utilizzabile da algoritmi efficienti

Applicare un Operatore e Calcolare il Nuovo Stato

- **STRIPS** (STandford Research Institute Problem Solver)
  - Linguaggio di rappresentazione dei pianificatori classici
- Rappresentazione degli stati
  - Si scompone il mondo in azioni logiche
  - ✓ Ipotesi di mondo chiuso (tutte le condizioni non menzionate in uno stato saranno considerate false)
- Rappresentazione degli obiettivi
  - Obiettivo: stato parzialmente specificato
  - ✓ Uno stato s soddisfa un obiettivo se contiene tutte le condizioni previste nell'obiettivo (possibilmente anche altri)
- Rappresentazione delle azioni

Applicare un Operatore e Calcolare il Nuovo Stato

- STRIPS (STandford Research Institute Problem Solver)
  - Linguaggio di rappresentazione dei pianificatori classici

- Rappresentazione delle azioni
  - Schema di azione: un'azione è specificata per mezzo delle precondizioni che devono valere prima della sua esecuzione e degli effetti che ne scaturiscono
  - ✓ Un azione è applicabile in ogni stato che soddisfa la precondizione
  - ✓ Sussunto di STRIPS: i letterali non menzionati nell'effetto non sono mai modificati

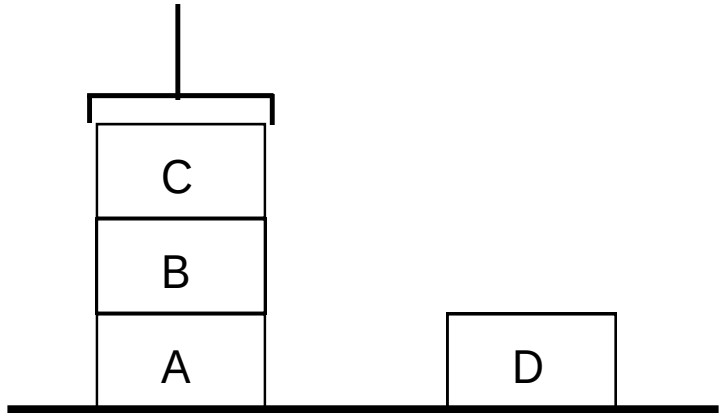
Applicare un Operatore e Calcolare il Nuovo Stato

- STRIPS (STandford Research Institute Problem Solver)
  - Linguaggio di rappresentazione dei pianificatori classici

- *Soluzione di un problema di pianificazione*
  - E' costituita nella sua forma più semplice da una sequenza di azioni che, una volta eseguite, porteranno in uno stato che soddisfa l'obiettivo

Esempio: Il Mondo dei Blocchi

Dominio: insieme di blocchi di forma cubica. Essi possono essere impilati ma solo un blocco può stare direttamente sopra un altro.



- Lo stato del mondo è descritto da un insieme di predicati.
- Il mondo può essere modificato solo dal braccio.

Il Mondo dei Blocchi in Strips

- I predicati sono del tipo:
  - on(B,A) ontable(A)
  - clear(D) armempty
- Le azioni sono del tipo:
  - unstack(C,B) stack(C,D)
  - pickup(D) putdown(D)

Applicare un Operatore e Calcolare il Nuovo Stato

- Operatori in STRIPS
  unstack(x,y)

  - Precondizione: on(x,y) $\wedge$ clear(x) $\wedge$ armempty
  - Effetto: on(x,y) $\wedge$ armempty
  - Azione: holding(x) $\wedge$ clear(y)

Individuare i Vicoli Ciechi e le Soluzioni Quasi Corrette

- Vicoli ciechi

  - Consistenza dello stato corrente.
  - Consistenza degli effetti dell'azione da applicare con la meta.

Individuare i Vicoli Ciechi e le Soluzioni Quasi Corrette

- Soluzioni quasi corrette

È utile nel caso di problemi quasi scomponibili:

- Si risolve il problema come se fosse scomponibile e si combinano le soluzioni.
- Si risolvono le varie parti senza definire un ordine di esecuzione per gli operatori e si combinano le soluzioni aggiungendo i vincoli di precedenza.
- Se lo stato ottenuto è più lontano dalla meta dello stato iniziale, allora si ripianifica dallo stato iniziale altrimenti si ripianifica dallo stato raggiunto.

STRIPS: Pianificazione Basata sullo Stack di Mete

- Ad ogni passo viene considerato la meta in cima allo stack.
- Viene cercata la sequenza di operatori che permettono di soddisfare la meta.
- Tale sequenza di operatori vengono applicati allo stato.
- Si passa alla meta successiva fino a vuotare lo stack.
- Si confronta la meta originaria con lo stato raggiunto.
- Le componenti della meta non raggiunte si ri-inseriscono nello stack.

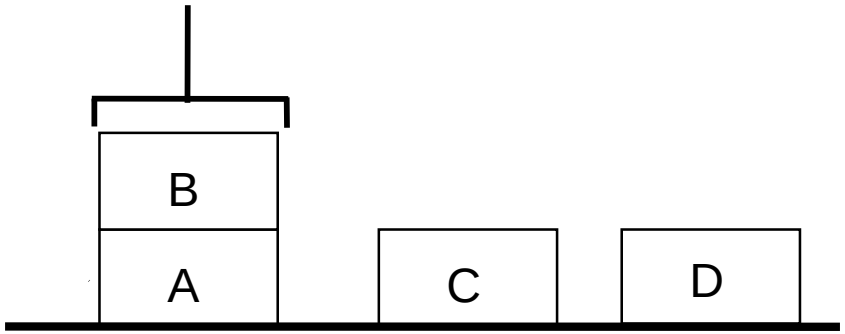
STRIPS: Pianificazione Basata sullo Stack di Mete

- Fa parte della pianificazione in avanti nello spazio degli stati, chiamata pianificazione di progressione.
- Si comincia dallo stato iniziale del problema e si considerano sequenze diverse di azioni finché non se ne trova una che raggiunge un stato obiettivo
- Formulazione dei problemi di pianificazione:
  - Lo stato iniziale della ricerca
  - Le azioni applicabili a quello stato (tutte quelle le cui precondizioni sono soddisfate dallo stato)
  - Il test obiettivo che controlla se lo stato corrente soddisfa l'obiettivo del problema di pianificazione
  - Il costo di ogni pass è sempre considerato uguale a 1

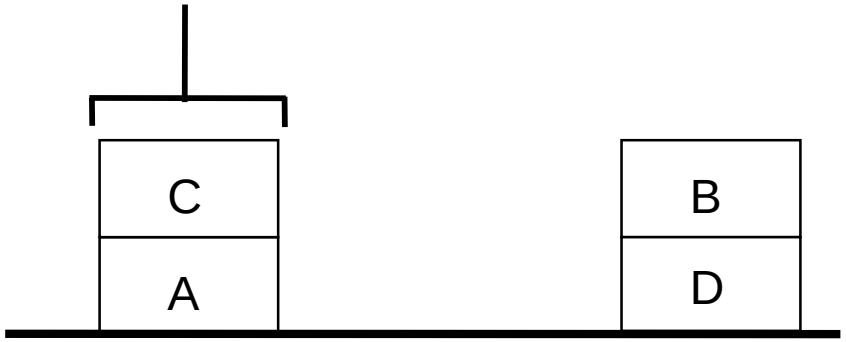
STRIPS: Pianificazione Basata sullo Stack di Mete

- Consideriamo il seguente problema:

Stato iniziale:



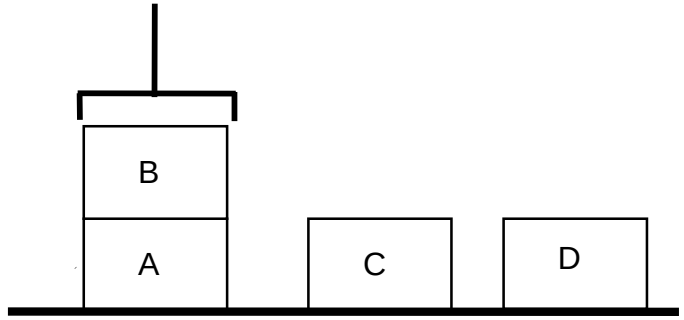
Stato finale:



STRIPS: Pianificazione Basata sullo Stack di Mete

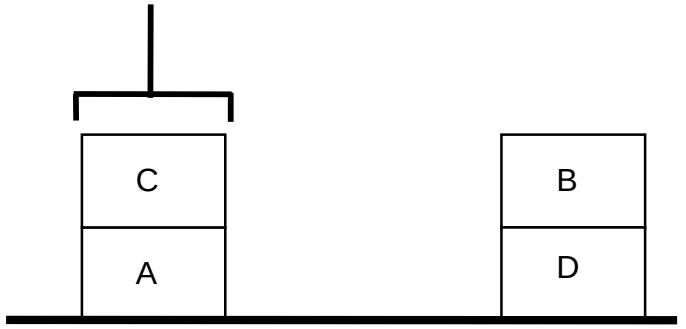
- Lo stack inizialmente contiene la meta:

  on(C,A) ∧ on(B,D) ∧ ontable(A) ∧ ontable(D)



- Scomponendo la meta posso ottenere due stack:

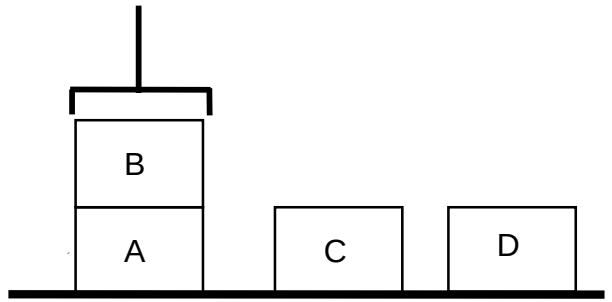
  (1) on(C,A)
  on(B,D)
  on(C,A) ∧ on(B,D) ∧ ontable(A) ∧ ontable(D)



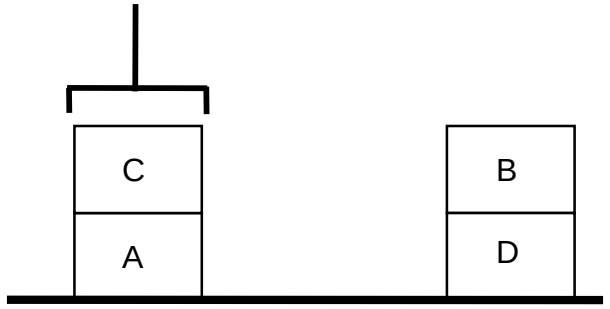
  (2) on(B,D)
  on(C,A)
  on(C,A) ∧ on(B,D ∧ ontable(A) ∧ ontable(D)

STRIPS:Pianificazione Basata sullo Stack di Mete

- Considero lo stack 1 ottengo:
  stack(C,A)
  on(B,D)
  on(C,A) on(B,D) $\wedge$ ontable(A) $\wedge$ ontable(D)

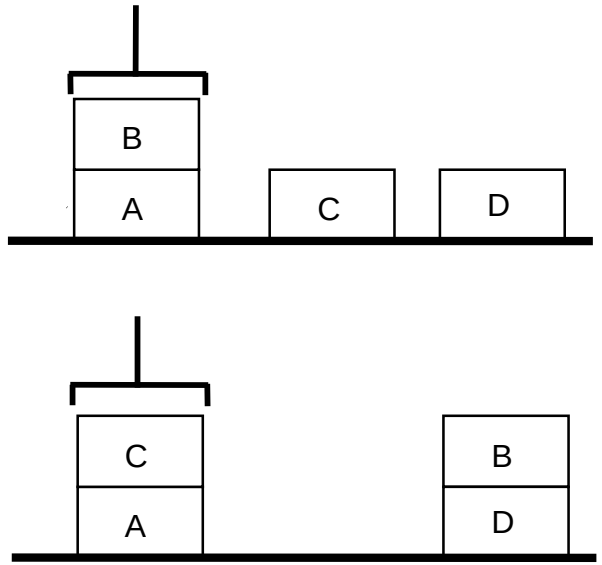


- Aggiungo le precondizioni di stack(C,A)
  clear(A)
  holding(C)
  clear(A) $\wedge$ holding(C)
  stack(C,A)
  on(B,D)
  on(C,A) $\wedge$ on(B,D) $\wedge$ ontable(A) $\wedge$ ontable(D)



STRIPS: Pianificazione Basata sullo Stack di Mete

- clear(A) non è vera, applico unstack(B,A):

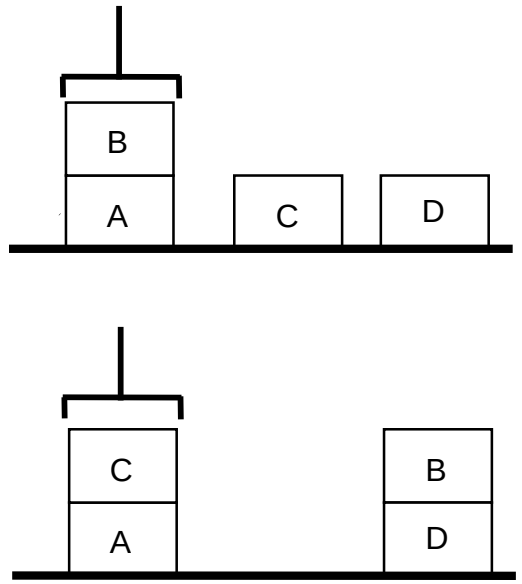


on(B,A)
clear(B)
armempty
on(B,A) $\wedge$ clear(B) $\wedge$ armempty
unstack(B,A)
holding(C)
clear(A) $\wedge$ holding(C)
stack(C,A)
on(B,D)
on(C,A) $\wedge$ on(B,D) $\wedge$ ontable(A) $\wedge$ ontable(D)

STRIPS:Pianificazione Basata sullo Stack di Mete

- Le precondizioni di unstack(B,A) sono vere, quindi le rimuovo dallo stack:

  holding(C)
  clear(A) $\wedge$ holding(C)
  stack(C,A)
  on(B,D)
  on(C,A) $\wedge$ on(B,D) $\wedge$ ontable(A) $\wedge$ ontable(D)



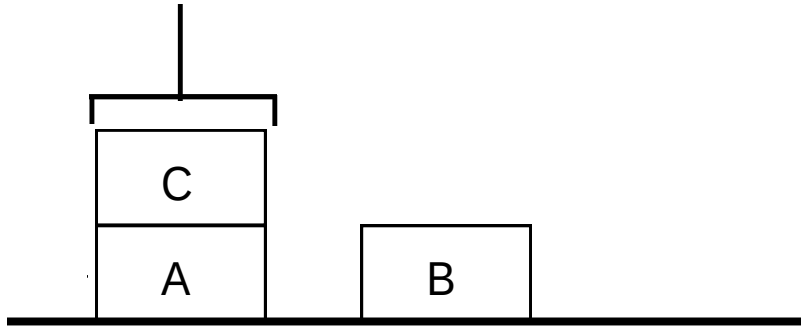
- Alla fine ho il piano:

  unstack(B,A)
  stack(B,D)
  pickup(C)
  stack(C,A)

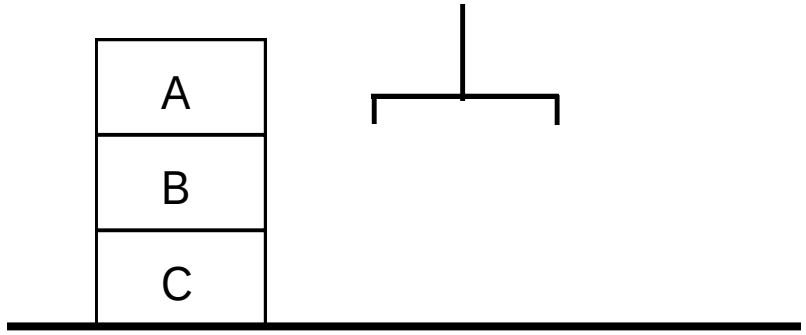
STRIPS: Pianificazione Basata sullo Stack di Mete

- Consideriamo il seguente problema:

Stato iniziale:

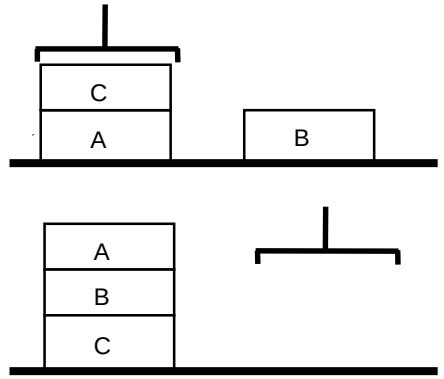


Stato finale:



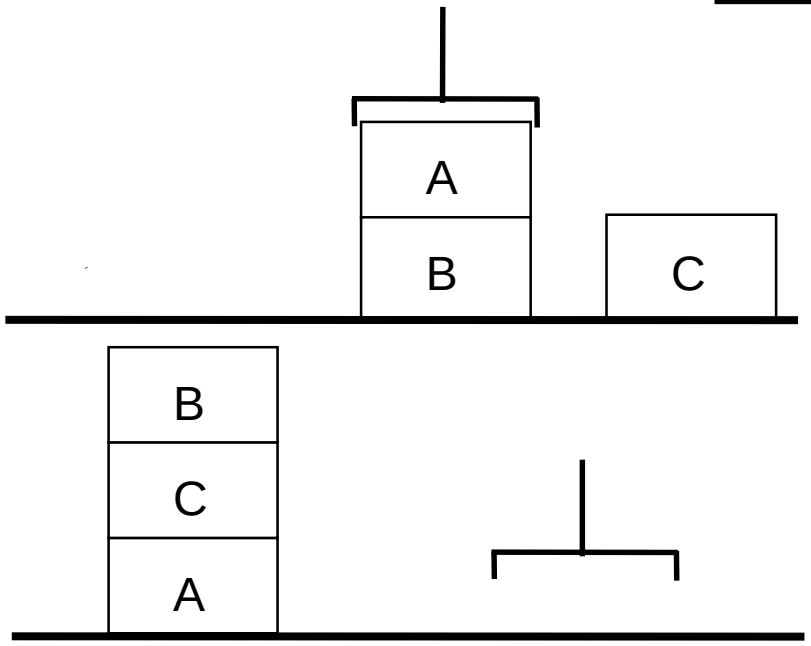
STRIPS: Pianificazione Basata sullo Stack di Mete

- Lo stack inizialmente contiene la meta:
  on(A,B) ∧ on(B,C)
- Scomponendo la meta posso ottenere due stack:



1 **on(A,B)**
  on(B,C)
  on(A,B) ∧ on(B,C)

2 **on(B,C)**
  on(A,B)
  on(A,B) ∧ on(B,C)



L'anomalia di Sussman

- I due piani generati non risolvono il problema:
- Il primo piano generato è:

  [Unstack(c), Stack(a, b), Unstack(a),
  Stack(b, c), Stack(a ,b)]
- Il secondo piano generabile è:

  [Stack(b, c), Unstack(b), Unstack(c),
  Stack(a, b), Unstack(a), Stack(b, c), Stack(A,B)]

- Il piano ideale non è ottenibile con una pianificazione "**lineare**":

  [Unstack(c), Stack(b, c), Stack(a, b)]

Pianificazione Non Lineare

- Questi metodi non risolvono le mete in sequenza.
- Passano dalla soluzione di una meta alla soluzione di un'altra meta se ciò diventa vantaggioso.
- Realizzano dei piani provvisori che lasciano ad esempio indeterminati i rapporti temporali tra le diverse azioni del piano.
- Si basano principalmente sul fatto di considerare le mete indipendenti le une dalle altre.

Pianificazione Non Lineare Metodo di Nilsson

- Basato su una procedura all'indietro a partire dalla meta.
- ***Pianificazione di regressione***: quali sono gli stati in cui l'applicazione di una data azione conduce ad un obiettivo?
- Si suppone che tutte le mete tranne una siano state raggiunte e quindi si trovano gli operatori che potrebbero soddisfarla.
- Ogni volta che applichiamo un operatore dobbiamo aggiungere le sue precondizioni alle mete originarie.

Pianificazione Non Lineare Metodo di Nilsson

- Questo metodo genera un albero di ricerca molto complesso:
  - tutti i possibili ordinamenti delle mete;
  - tutti gli operatori che possono essere utilizzati.

Pianificazione Non Lineare Metodo di Nilsson

- Molti cammini possono essere abbandonati perché l'esecuzione dell'operatore annulla una meta:
  - ✓ *Oltre a richiedere che le azioni rendano vero qualche letterale desiderato, occorre richiedere anche che esse non falsifichino altri letterali desiderati*

  - regressione(on(A,B), pickup(C)) = on(A,B)
  - regressione(on(A,B), stack(A,B)) = True
  - regressione(on(A,B), unstack(A,B)) = False

Pianificazione Non Lineare Metodo di Nilsson

- Molto buono se le mete interagiscono tra di loro.
- Molto dispendioso se le mete non interagiscono.
- Non fa distinzione tra mete importanti e mete secondarie.

Pianificazione Basata sull'Ordinamento Parziale

- Pianificazione di progressione e regressione considerano solo sequenze di azioni strettamente lineari collegati direttamente allo stato iniziale e all'obiettivo.
- Sono dette ricerche di pianificazione ***con ordinamento totale.***
- Ciò significa che non possono trarre vantaggio dalla scomposizione del problema:
  - Non si lavora indipendentemente ogni sotto-problema,ma le decisioni riguardano sequenze di azioni che coivolgono tutti.

Pianificazione Basata sull'Ordinamento Parziale

- Un approccio che tratta indipendentemente sotto-obiettivi diversi, li risolve per mezzo di più sottopiani diversi ed infine li integra in un unico piano sarebbe preferibile.

- Potrebbe essere più flessibile sull'ordine con cui costruisce il piano:
  - Può considerare prima le decisioni evidenti o importanti senza essere obbligato a seguire un ordine cronologico
  - Durante la ricerca il principio generale di rimandare una scelta sarà qello del minimo impegno

Pianificazione Basata sull'Ordinamento Parziale

- Un algoritmo di pianificazione che può aggiungere due azioni a un piano senza specificare quale delle due è eseguita per prima prende il nome di pianificatore con ordinamento parziale
- La soluzione con ordinamento parziale può corrispondere a più piani con ordinamento totale
- Ognuna di esse prende il nome di **linearizzazione** del piano parzialmente ordinato
- ***Ogni linearizzazione di una soluzione con ordinamento parziale è una soluzione ordinata la cui esecuzione a partire dallo stato iniziale raggiungerà uno stato obiettivo***

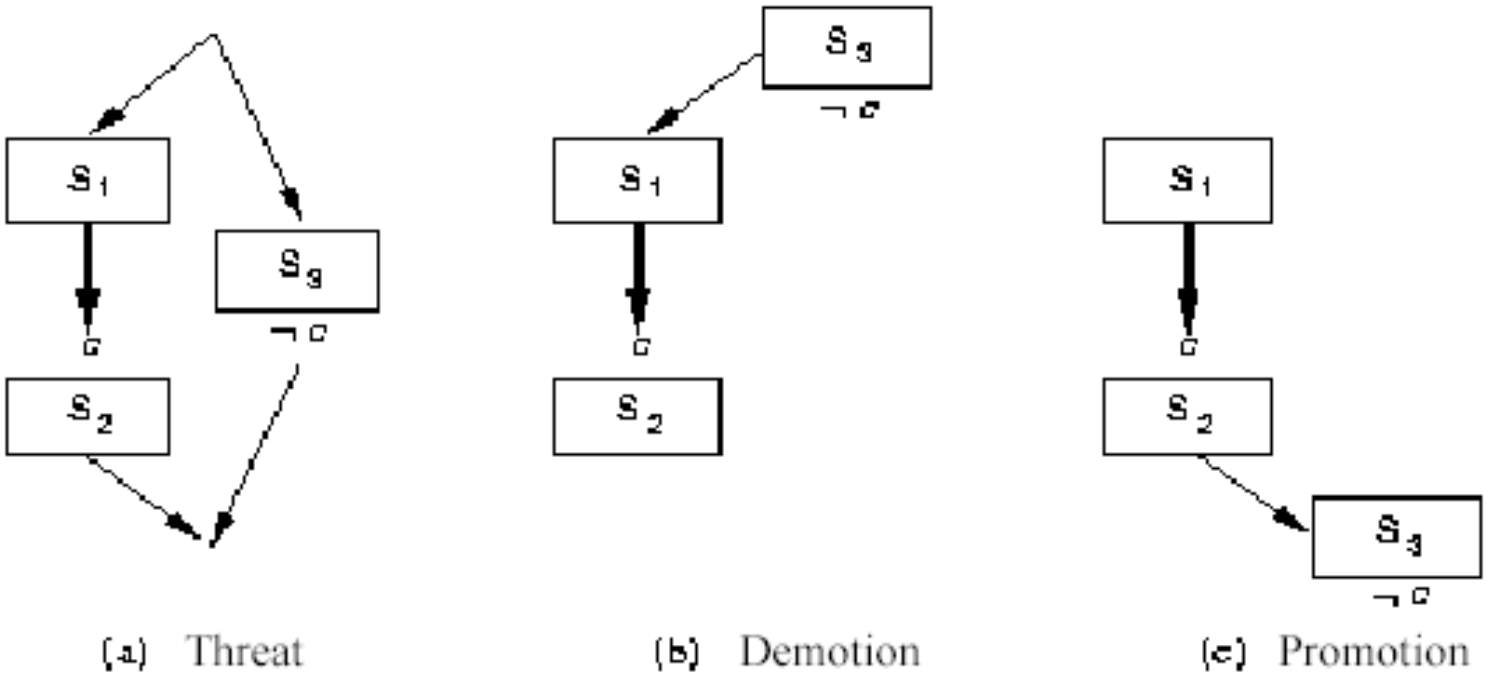
Pianificazione Basata sull'Ordinamento Parziale

- Ad ogni passo viene aggiunto un passo s_i che soddisfa una precondizione c_j necessaria per raggiungere il goal: l'azione del passo s_i è unita a c_j da un link causale.
- Se viene introdotto un passo s_k la cui azione distrugge una precondizione c_h protetta da un link causale dell'azione s_m (s_{jk} threatens c_h), allora il threat viene risolto aggiungendo un link di ordinamento (se possibile).

Pianificazione Basata sull'Ordinamento Parziale

- Rispettare i link causali
- Sia il passo S1 -> S2 che **protegge** una condizione c
  - non si deve aggiungere uno step S3 che viola *c*
- Se una azione parallela **minaccia** *c* (i.e., ha l'effetto di negarlo), risolverlo aggiungendo link di ordinamento:
  - S3 prima di S1 (**demotion**)
  - S3 dopo S2 (**promotion**)
- Se queste due strategie non risolvono il threat, si torna indietro modificando delle scelte precedenti.

Pianificazione Basata sull'Ordinamento Parziale



Pianificazione Basata sull'Ordinamento Parziale

Operators on partial plans:

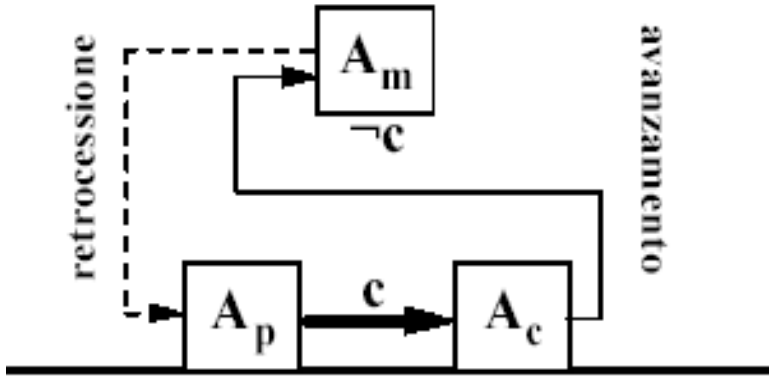
add a link from an existing action to an open condition
add a step to fulfill an open condition
order one step wrt another to remove possible conflicts

Gradually move from incomplete/vague plans to complete, correct plans

Backtrack if an open condition is unachievable or
if a conflict is unresolvable

Pianificazione Basata sull'Ordinamento Parziale

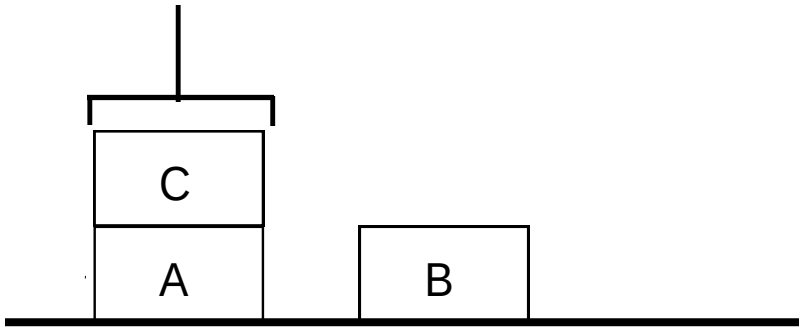
- Operazioni sui piani
  - aggiunta di un legame causale da un'azione esistente ad una condizione aperta
  - aggiunta di un'azione per soddisfare una condizione aperta
  - ordinamento di un'azione rispetto ad un'altra per annullare una minaccia:
    - retrocessione (Am < Ap)
    - avanzamento (Am > Ac)



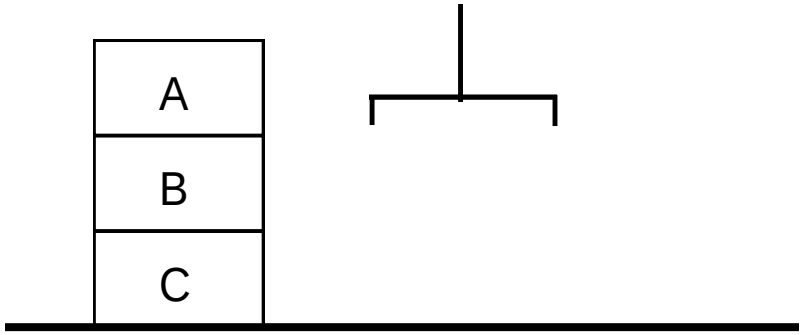
Pianificazione Basata sull'Ordinamento Parziale

- Consideriamo ancora il problema:

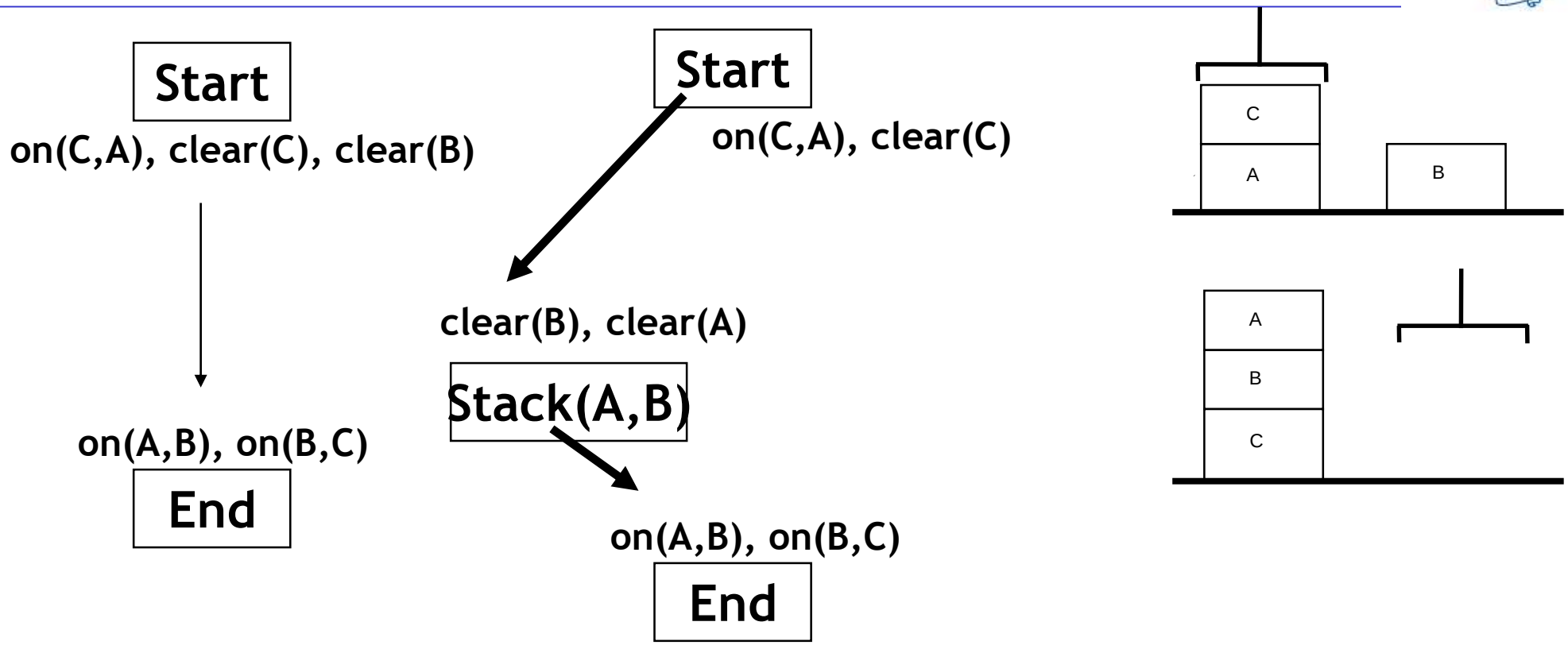
Stato iniziale:



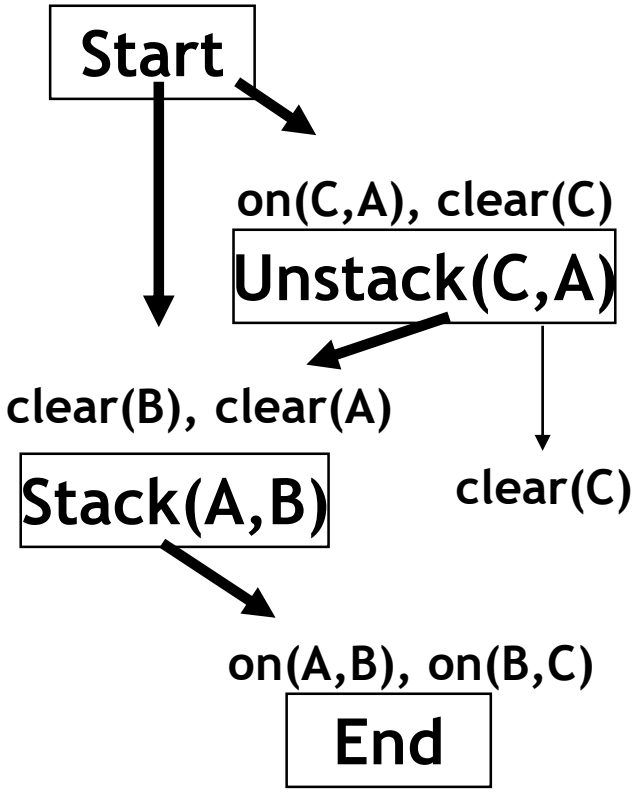
Stato finale:

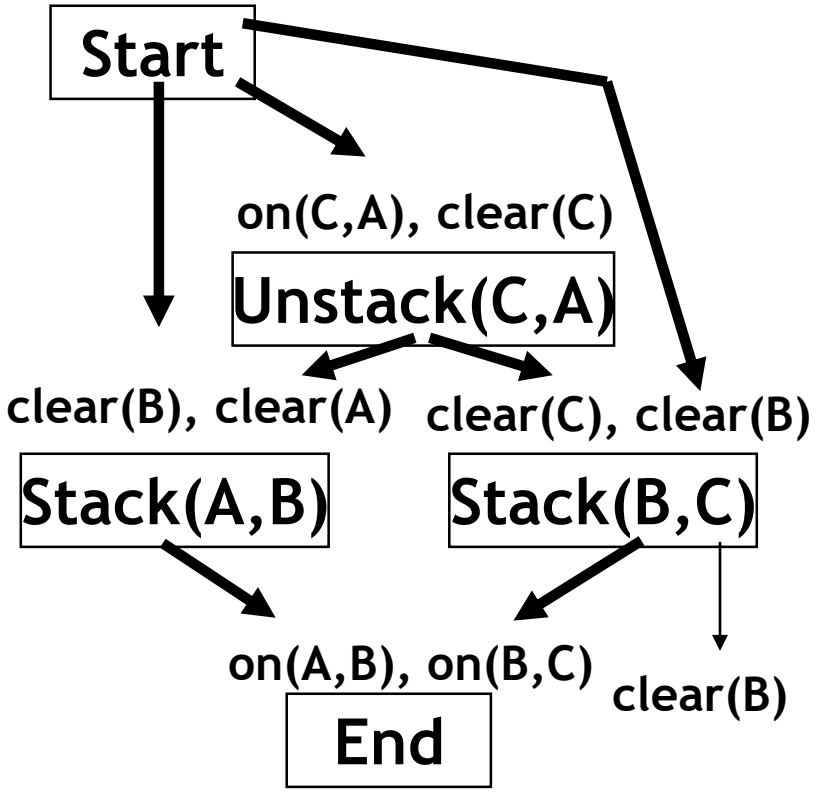


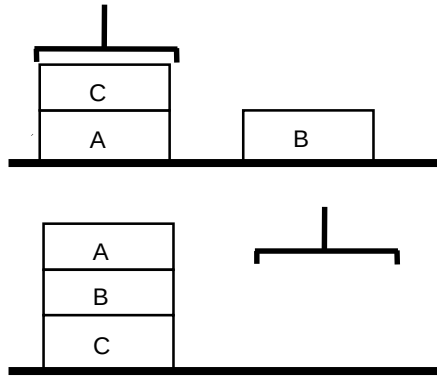
Pianificazione Basata sull'Ordinamento Parziale



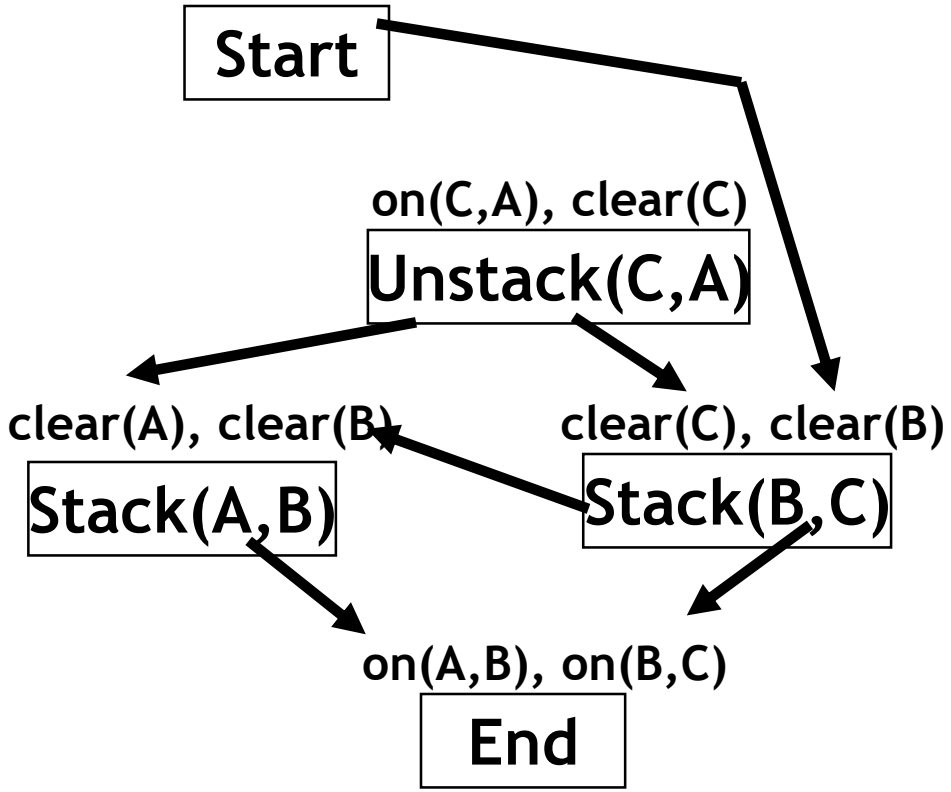
Pianificazione Basata sull'Ordinamento Parziale

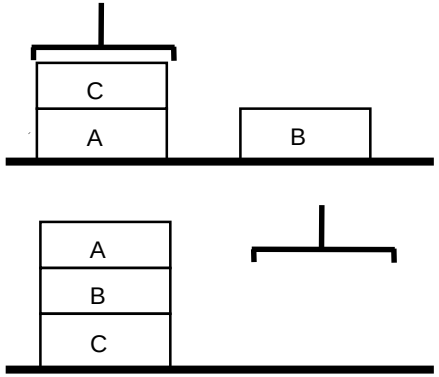






Pianificazione Basata sull'Ordinamento Parziale





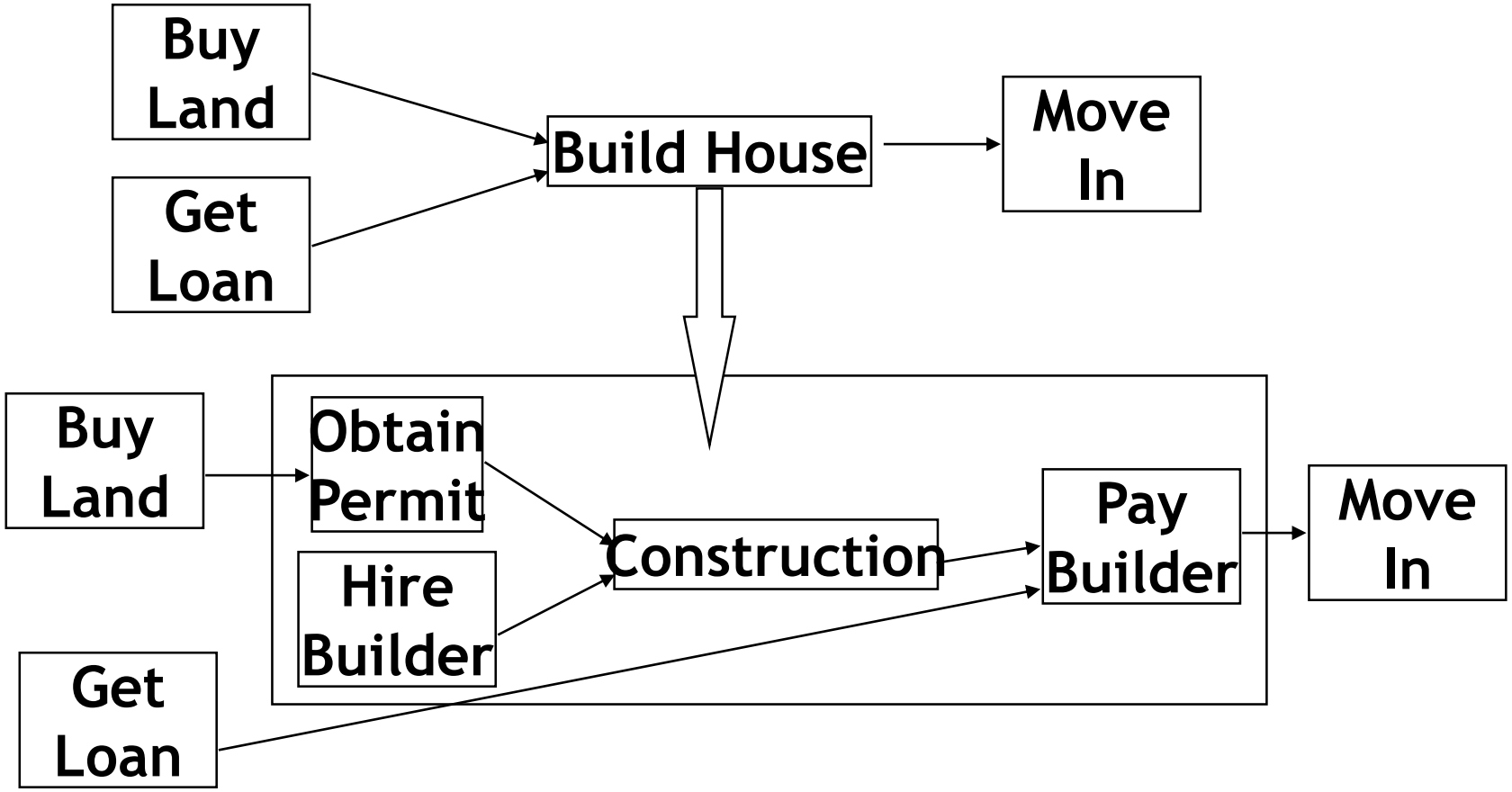
Pianificazione Gerarchica

- Cerca di affrontare il problema della ***diversa importanza delle mete.***
- Usa degli operatori astratti che sono decomposti in piani che implementano l'operatore.
- Abbiamo una pianificazione a diversi livelli di astrazione.
- Ogni livello di gerarchia si riduce ad un piccolo numero di attività al livello immediatamente inferiore, cosicché il costo di trovare il modo migliore di organizzare tale attività risulta parimenti piccolo

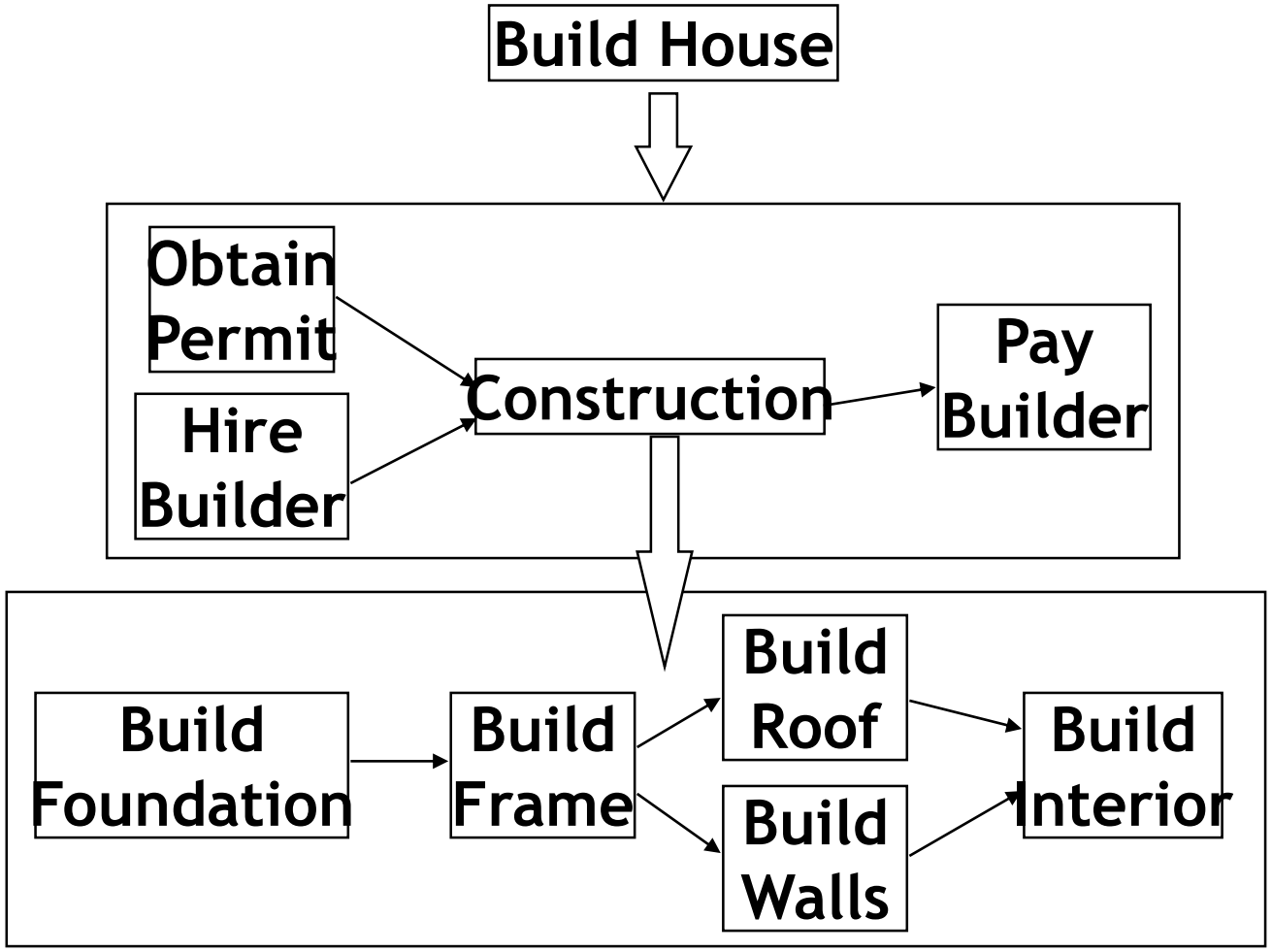
Pianificazione Gerarchica

- Un piano p implementa correttamente un operatore o se:
  - p è un piano consistente.
  - Ogni effetto di o deve essere raggiunto da un passo di p.
  - Ogni precondizione di un passo di p deve o essere raggiunto da un altro passo di p o essere una precondizione di o.

Pianificazione Gerarchica



Pianificazione Gerarchica



Pianificazione Gerarchica

- Si può dimostrare che con un pianificatore gerarchico se i piani vengono decomposti correttamente la soluzione si trova con costo computazionale inferiore che con la pianificazione non lineare.

Pianificazione nel Mondo Reale

- Le tecniche di pianificazione precedenti sono applicabili a domini in cui:
  - Ogni entità è nota.
  - Le proprietà di ogni entità sono statiche e deterministiche.
- Nel mondo reale le tecniche di pianificazione precedenti non sono sufficienti perché:
  - Si ha una conoscenza incompleta sul mondo.
  - Si ha una conoscenza errata del mondo.
- Per gestire queste situazioni si possono usare due diverse tecniche di pianificazione:
  - Pianificazione condizionale (conditional planning).
  - Controllo dell'esecuzione (execution monitoring).

Pianificazione Condizionale

- Se l'ambiente non è deterministico, l'agente non potrà predire completamente il risultato delle sue azioni
- Un agente di pianificazione condizionale gestisce il non determinismo includendo al momento di costruzione del piano dei passi condizionali in cui verificherà, durante l'esecuzione, lo stato dell'ambiente
- La pianificazione condizionale introduce dunque delle azioni sensoriali per decidere tra sequenze di azioni alternative.
- Il problema si riduce quindi alla costruzione di piani condizionali

Pianificazione Condizionale

- Esempio: controllo e cambio di una ruota bucata

- Obiettivo:
  - avere una ruota montata e gonfia
- Azioni:
  - montare una ruota,
  - rimuovere una ruota,
  - gonfiare una ruota
- Stato iniziale:
  - una ruota montata ma sgonfia
  - un ruota di ricambio intatta e gonfia ma nel bagagliaio

Pianificazione Condizionale

- Ad esempio, se abbiamo:
  Goal: $On(x) \wedge Inflated(x)$
  Operators: Remove(x), PutOn(x), Inflate(x)
  Init State: $Inflated(Spare) \wedge Intact(Spare)$
  $\wedge Off(Spare)$
  $\wedge On(Tire_1) \wedge Flat(Tire_1)$

- Un pianificatore precedente comporrà il piano:
  [Remove($Tire_1$), PutOn(Spare)]

- Invece un pianificatore condizionale:
  if (Intact($Tire_1$)) [Inflate($Tire_1$)],
  [Remove($Tire_1$), PutOnt(Spare)]

Controllo dell'Esecuzione

- Il controllo dell'esecuzione permette di ripianificare quando si incontrano situazioni impreviste nel modello del mondo su cui si è definito il piano.
- Il funzionamento di un sistema del genere può essere descritto dai seguenti passi:
  - Finché le precondizioni della prossima azione del piano sono soddisfatte nello stato corrente s, si esegue l'azione.
  - Dunque si cerca un punto p' del piano che sia facilmente raggiungibile dallo stato corrente s.
  - Si genera un piano da s a p'.

Pianificazione Condizionale Vs Controllo dell'Esecuzione

- Ogni azione di un piano ha la possibilità di un malfunzionamento ed si può avere una conoscenza sbagliata di ogni stato del mondo.
- Il numero di condizioni che bisogna introdurre cresce esponenzialmente con il numero dei passi del piano.
- Ad ogni passo del piano può essere necessario ripianificare (pianificazione continua).
- Una soluzione accettabile è integrare i due approcci introducendo condizioni:
  - Per eventi che hanno un'alta probabilità.
  - Per eventi che pur avendo una bassa probabilità, sono catastrofici.

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

- L'agente durante l'esecuzione può scegliere con quanta attenzione monitorare l'ambiente. Distinguiamo tre livellivel:
- **Action monitoring**: prima di eseguire un'azione, l'agente verifica che tutte le precondizioni siano ancora valide.
- **Plan monitoring**: prima di eseguire un'azione, l'agente verifica che il piano rimanente abbia ancora esito positivo.
- **Goal monitoring**: prima di eseguire un'azione, l'agente verifica se esiste un insieme migliore di obiettivi che potrebbe tentare di raggiungere.

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

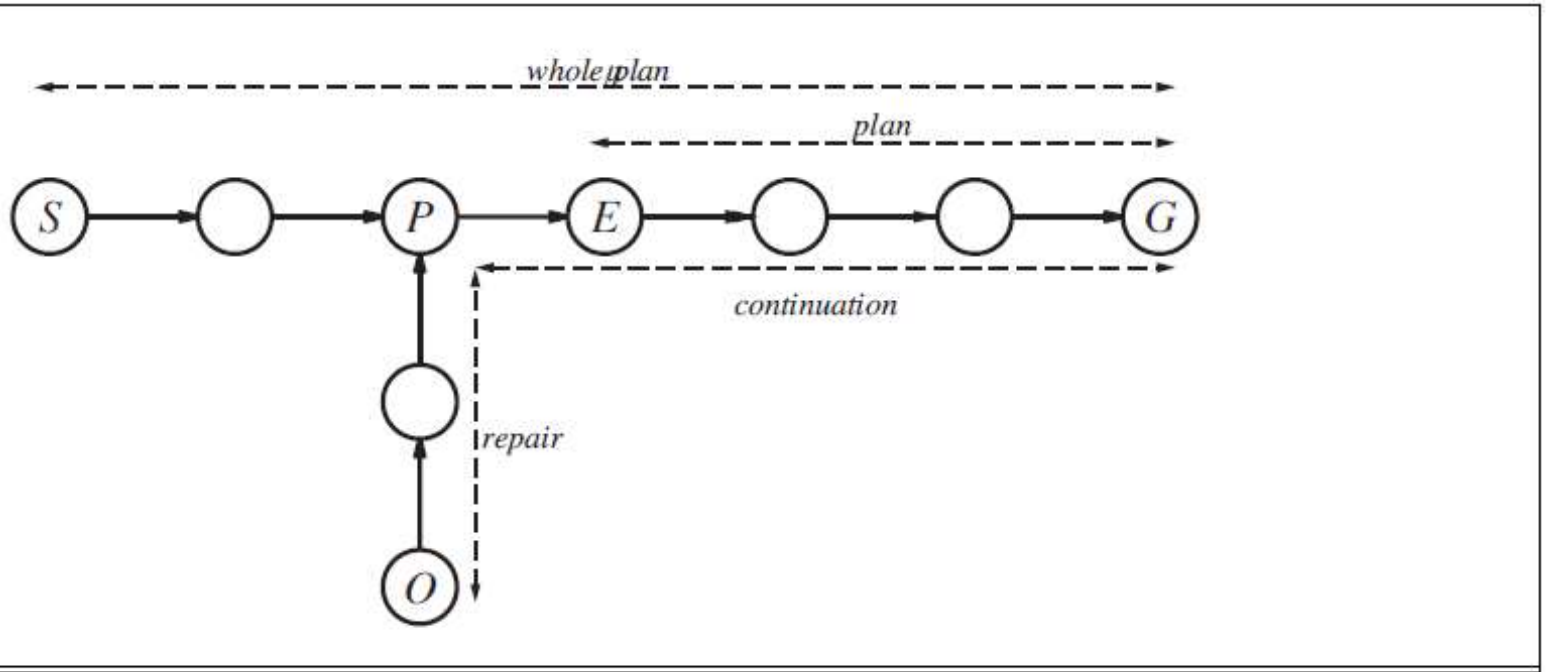


Figure 11.9 Before execution, the planner comes up with a plan, here called *whole plan*, to get from S to G. The agent executes steps of the plan until it expects to be in state E, but observes it is actually in O. The agent then replans for the minimal repair plus continuation to reach G.

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

- **Action monitoring**: è un metodo semplice di monitoraggio dell'esecuzione, ma a volte può portare a un comportamento meno che intelligente (non guarda se ci sono altre possibilità per ragiungere obiettivi
- **Plan monitoring**: può rilevare il fallimento ogni volta che lo stato attuale è tale che il resto del piano non funziona più. Quindi, non perderebbe tempo. Il monitoraggio del piano interrompe il prima possibile l'esecuzione di un piano condannato, piuttosto che continuare fino a quando il fallimento si verifica effettivamente

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

- **Plan monitoring**: "Funziona?"
- Se intendiamo "Possiamo garantire che l'agente raggiunga sempre l'obiettivo?" allora la risposta è no, perché l'agente potrebbe inavvertitamente arrivare a un vicolo cieco dal quale non c'è rimedio. Ad esempio, l'agente del vuoto potrebbe avere un modello difettoso di se stesso e non sapere che le sue batterie si stanno esaurendo.
- Se escludiamo vicoli ciechi - assumiamo che esista un piano per raggiungere l'obiettivo da qualsiasi stato dell'ambiente - e che un tale piano ha sempre qualche possibilità di successo in ogni tentativo, l'agente alla fine raggiungerà l'obiettivo
- .

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

- I problemi si verificano quando un'azione in realtà non è deterministica, ma dipende piuttosto da qualche precondizione che l'agente non conosce.
- Una soluzione è scegliere casualmente tra l'insieme di possibili piani di rimedio, piuttosto che provare lo stesso ogni volta.
- Un approccio migliore è quello di imparare un modello del mondo migliore. Ogni errore di previsione è un'opportunità di apprendimento; un agente dovrebbe essere in grado di modificare il suo modello del mondo in accordo con le sue percezioni.

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

- Il primo pianificatore online con monitoraggio dell'esecuzione è stato PLANEX (Fikes et al., 1972), che ha lavorato con il pianificatore STRIPS per controllare il robot Shakey.
- A metà degli anni '80, il pessimismo sui tempi di esecuzione lenti dei sistemi di pianificazione portò alla proposta di agenti riflessi chiamati sistemi di pianificazione reattivi(Brooks, 1986)

Pianificazione Condizionale Vs Controllo dell'Esecuzione: online planning

- I "piani universali" (Schoppers, 1987, 1989) sono stati sviluppati come metodi basati sulla ricerca in una lookup table per la pianificazione reattiva, ma si sono rivelati una riscoperta dell'idea di politiche che erano state a lungo utilizzate nei processi decisionali di Markov.
- Un piano universale (o una politica) contiene una mappatura da qualsiasi stato all'azione che dovrebbe essere intrapresa in quello stato. Koenig (2001) esamina le tecniche di pianificazione online, sotto il nome*Agent-Centered Search*.

TIME, SCHEDULES, AND RESOURCES

- La pianificazione classica tratta di cosa fare e in quale ordine, ma non ti tempoquanto tempo impiega un'azione e quando si verifica.
- Questo è l'argomento dello scheduling.
- l mondo reale impone anche molti vincoli di risorse; ad esempio, una compagnia aerea ha un numero limitato di dipendenti e il personale che si trova su un volo non può essere su un altr contemporaneamente.

TIME, SCHEDULES, AND RESOURCES

- Un approccio è “plan first, schedule later”: cioè, dividiamo il problema complessivo in una fase di planning in cui le azioni vengono selezionate, con alcuni vincoli di ordinamento, per raggiungere gli obiettivi del problema, e una fase successiva di scheduling, in cui vengono aggiunte informazioni temporali al piano per garantire che esso soddisfa i vincoli di risorse e tempo.
- Questo approccio è comune negli ambienti produttivi e logistici del mondo reale, dove la fase di pianificazione è spesso eseguita da esperti umani.

TIME, SCHEDULES, AND RESOURCES

- Un tipico problema di scheduling è quello di un lavoro in officina che consiste in un insieme di lavori, ognuno dei quali consiste in una raccolta di azioni con vincoli di ordinamento tra di loro.
- Ogni azione ha una durata e una serie di vincoli di risorse richiesti dall'azione. Ogni vincolo specifica un tipo di risorsa (ad esempio, bulloni, chiavi inglesi o piloti), il numero di oggetti di quella risorsa richiesti e se tale risorsa è consumabile (ad esempio, i bulloni non sono più disponibili per l'uso) o riutilizzabile (ad esempio, un pilota è occupato durante un volo ma è nuovamente disponibile al termine del volo).
- Le risorse possono essere prodotte anche da azioni con consumi negativi, tra cui la produzione, la crescita e l'attività di rifornimento

TIME, SCHEDULES, AND RESOURCES

- Una soluzione a un problema di schedulazione in una officina deve specificare gli orari di inizio di ciascuna azione e deve soddisfare tutti i vincoli di ordinamento temporale e di risorse.
- Come per i problemi di ricerca e pianificazione, le soluzioni possono essere valutate in base a una funzione di costo;
- questo può essere piuttosto complicato, con costi di risorse non lineari, costi di ritardo dipendenti dal tempo e così via.

TIME, SCHEDULES, AND RESOURCES

- Un semplice esempio: un problema che riguarda l'assemblaggio di due auto.
- Il problema consiste in due lavori, ciascuno del tipo
- [AddEngine,AddWheels, Inspect].
- Ci sono quattro tipi di risorse e per ognuna di esse si ha disponibile all'inizio: 1 paranco motore, 1 stazione della ruota, 2 ispettori e 500 dadi.
- Gli action schemas forniscono la durata e le esigenze di risorse di ciascuna azione.
- I dadi vengono consumati quando le ruote vengono aggiunte all'auto, mentre le altre risorse vengono "prese in prestito" all'inizio di un'azione e rilasciate alla fine dell'azione.

TIME, SCHEDULES, AND RESOURCES

$Jobs(\{AddEngine1 \prec AddWheels1 \prec Inspect1\},$
$\{AddEngine2 \prec AddWheels2 \prec Inspect2\})$

$Resources(EngineHoists(1), WheelStations(1), Inspectors(2), LugNuts(500))$

$Action(AddEngine1,$ DURATION:30,
USE:$EngineHoists(1))$
$Action(AddEngine2,$ DURATION:60,
USE:$EngineHoists(1))$
$Action(AddWheels1,$ DURATION:30,
CONSUME:$LugNuts(20)$, USE: $WheelStations(1))$
$Action(AddWheels2,$ DURATION:15,
CONSUME:$LugNuts(20)$, USE: $WheelStations(1))$
$Action(Inspect_i,$ DURATION:10,
USE:$Inspectors(1))$

Figure 11.1 A job-shop scheduling problem for assembling two cars, with resource constraints. The notation $A \prec B$ means that action A must precede action B.

TIME, SCHEDULES, AND RESOURCES

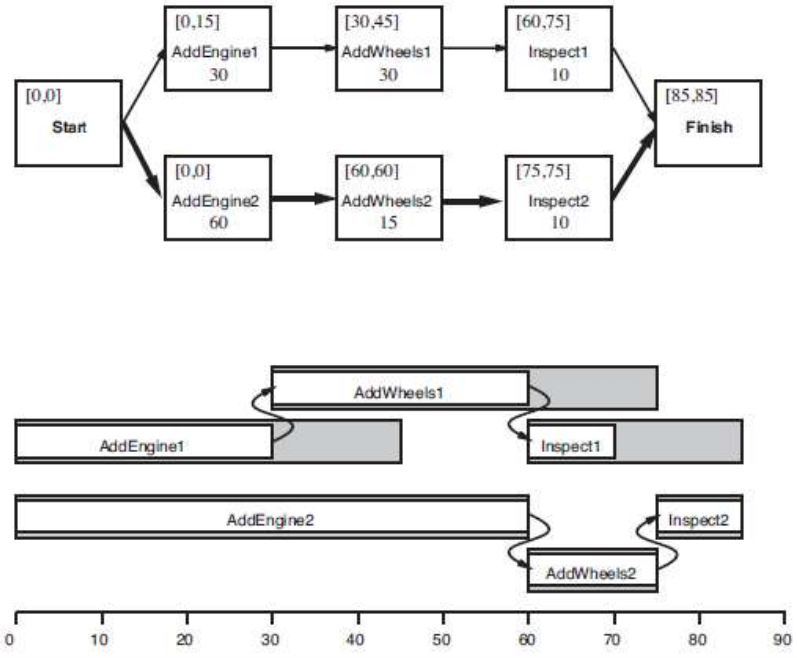


Figure 11.2 Top: a representation of the temporal constraints for the job-shop scheduling problem of Figure 11.1. The duration of each action is given at the bottom of each rectangle. In solving the problem, we compute the earliest and latest start times as the pair $[ES, LS]$, displayed in the upper left. The difference between these two numbers is the *slack* of an action; actions with zero slack are on the critical path, shown with bold arrows. Bottom: the same solution shown as a timeline. Grey rectangles represent time intervals during which an action may be executed, provided that the ordering constraints are respected. The unoccupied portion of a gray rectangle indicates the slack.