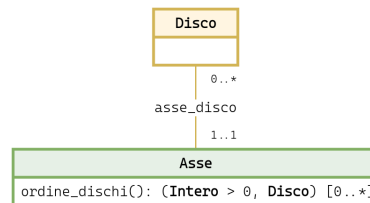


E.B.2.1 (PDDL: Torri di Hanoi, modelling)

1.1 Modellazione (FOL)

L'idea è quella di partire da una modellazione ad alto livello (FOL) e trasformarla per ricavarne una specifica in PDDL (*come si fa tipicamente nella progettazione*).



`ordine_dischi(): (Intero > 0, Disco) [0..*]`

descrizione – l'ordine di inserimento dei dischi su un asse è dato dalla relazione < su `Disco`

postcondizioni

\exists dischi_asse
 dischi_asse = { disco | `asse_disco(this, disco)` } \rightarrow
 result = sorted(dischi_asse, <)

– $\mathcal{P}_{\text{FOL}} = \{ \text{Asse}/1, \text{Disco}/1, \text{asse_disco}/2, </2 \}$

– $\phi_{\text{FOL}} =$

– invariante tipi

$\forall a \text{ Asse}(a) \rightarrow \neg \text{Disco}(a) \wedge$
 $\forall a, d \text{ asse_disco}(a, d) \rightarrow \text{Asse}(a) \wedge \text{Disco}(d) \wedge$
 $\forall d1, d2 (d1 < d2 \rightarrow \text{Disco}(d1) \wedge \text{Disco}(d2)) \wedge$

– 1..1

$\forall d \text{ Disco}(d) \rightarrow \exists a \text{ asse_disco}(a, d) \wedge$
 $(\neg \exists d, \alpha, \beta$
 $\text{asse_disco}(\alpha, d) \wedge \text{asse_disco}(\beta, d)) \wedge$

– relazione <

– antiriflessività

$\forall \delta \neg (\delta < \delta) \wedge$

– antisimmetria

$\forall \delta, \gamma (\delta < \gamma) \rightarrow \neg (\gamma < \delta) \wedge$

– transitività

$\forall \delta, \gamma, \eta (\delta < \gamma \wedge \gamma < \eta) \rightarrow \delta < \eta \wedge$

– totalità

$\forall \delta, \gamma (\text{Disco}(\delta) \wedge \text{Disco}(\gamma) \wedge \delta \neq \gamma) \rightarrow \delta < \gamma \vee \gamma < \delta$

1.2 Modellazione (PDDL)

Si modifica il predicato *asse_disco* in *asse_ord_disco* in modo da tenere conto dell'operazione *ordine_dischi()*, modellando sostanzialmente ogni asse come una **stack**, e si tiene traccia dell'elemento **top** tramite la relazione *minore*. La stack è **vuota** quando l'elemento *minore* è Tavolo.

- $\mathcal{P}_{\text{PDDL}} =$
 { *Asse*/1, *Disco*/1, </2 } \cup
 { *asse_ord_disco*/3, *minore*/2 }
- $\mathcal{F}_{\text{PDDL}} = \{$
 Tavolo/0, A/0, B/0, C/0,
 *D*₁/0, *D*₂/0, *D*₃/0, *D*₄/0
 }
- Stato iniziale:
 - invariante tipi
 Asse(A) \wedge *Asse*(B) \wedge *Asse*(C) \wedge
 Disco(*D*₁) \wedge *Disco*(*D*₂) \wedge *Disco*(*D*₃) \wedge *Disco*(*D*₄) \wedge
 - relazione <
 (*D*₁ < *D*₂) \wedge (*D*₁ < *D*₃) \wedge (*D*₁ < *D*₄) \wedge
 (*D*₂ < *D*₃) \wedge (*D*₂ < *D*₄) \wedge (*D*₃ < *D*₄) \wedge
 - posizione iniziale + *ordine_dischi()*
 asse_ord_disco(A, *D*₁, *D*₂) \wedge *asse_ord_disco*(A, *D*₂, *D*₃) \wedge
 asse_ord_disco(A, *D*₃, *D*₄) \wedge *asse_ord_disco*(A, *D*₄, Tavolo) \wedge
 minore(B, Tavolo) \wedge *minore*(C, Tavolo) \wedge *minore*(A, *D*₁)
- **Stato finale:**
 - posizione finale + *ordine_dischi()*
 asse_ord_disco(C, *D*₁, *D*₂) \wedge *asse_ord_disco*(C, *D*₂, *D*₃) \wedge
 asse_ord_disco(C, *D*₃, *D*₄) \wedge *asse_ord_disco*(C, *D*₄, Tavolo) \wedge
 minore(A, Tavolo) \wedge *minore*(B, Tavolo) \wedge *minore*(C, *D*₁)

1.2.1 Schemi di azione

```
SpostaSuDiscoDaDisco(  
  disco,  
  disco_sottostante,  
  disco_minore_asse_di_arrivo,  
  asse_di_partenza,  
  asse_di_arrivo  
)  
  
precondizioni  
- tipi  
Disco(disco) ∧ Disco(disco_sottostante) ∧  
Disco(disco_minore_asse_di_arrivo) ∧  
Asse(asse_di_partenza) ∧ Asse(asse_di_arrivo)  
disco < disco_minore_asse_di_arrivo ∧  
  
asse_ord_disco(asse_di_partenza, disco, disco_sottostante) ∧  
minore(asse_di_partenza, disco) ∧  
minore(asse_di_arrivo, disco_minore_asse_di_arrivo)  
  
effetto  
- disco = pop(asse_di_partenza)  
→ asse_ord_disco(asse_di_partenza, disco, disco_sottostante) ∧  
→ minore(asse_di_partenza, disco) ∧  
→ minore(asse_di_arrivo, disco_minore_asse_di_arrivo) ∧  
  
- push(asse_di_arrivo, disco)  
asse_ord_disco(asse_di_arrivo, disco, disco_minore_asse_di_arrivo) ∧  
minore(asse_di_partenza, disco_sottostante) ∧  
minore(asse_di_arrivo, disco)
```

```
SpostaSuTavoloDaDisco(  
  disco,  
  disco_sottostante,  
  asse_di_partenza,  
  asse_di_arrivo  
)  
  
precondizioni  
- tipi  
Disco(disco) ∧ Disco(disco_sottostante) ∧  
Asse(asse_di_partenza) ∧ Asse(asse_di_arrivo) ∧  
  
asse_ord_disco(asse_di_partenza, disco, disco_sottostante) ∧  
minore(asse_di_partenza, disco) ∧  
minore(asse_di_arrivo, Tavolo)  
  
effetto  
- disco = pop(asse_di_partenza)  
→ asse_ord_disco(asse_di_partenza, disco, disco_sottostante) ∧  
→ minore(asse_di_partenza, disco) ∧  
→ minore(asse_di_arrivo, Tavolo) ∧  
  
- push(asse_di_arrivo, disco)  
asse_ord_disco(asse_di_arrivo, disco, Tavolo) ∧  
minore(asse_di_partenza, disco_sottostante) ∧  
minore(asse_di_arrivo, disco)
```

```
SpostaSuDiscoDaTavolo(  
  disco,  
  disco_minore_asse_di_arrivo,  
  asse_di_partenza,  
  asse_di_arrivo  
)  
  
precondizioni  
- tipi  
Disco(disco) ∧ Disco(disco_minore_asse_di_arrivo) ∧  
Asse(asse_di_partenza) ∧ Asse(asse_di_arrivo) ∧  
disco < disco_minore_asse_di_arrivo ∧  
  
asse_ord_disco(asse_di_partenza, disco, Tavolo) ∧  
minore(asse_di_partenza, disco) ∧  
minore(asse_di_arrivo, disco_minore_asse_di_arrivo)  
  
effetto  
- disco = pop(asse_di_partenza)  
→ asse_ord_disco(asse_di_partenza, disco, Tavolo) ∧  
→ minore(asse_di_partenza, disco) ∧  
→ minore(asse_di_arrivo, disco_minore_asse_di_arrivo) ∧  
  
- push(asse_di_arrivo, disco)  
asse_ord_disco(asse_di_arrivo, disco, disco_minore_asse_di_arrivo) ∧  
minore(asse_di_partenza, Tavolo) ∧  
minore(asse_di_arrivo, disco)
```

```
SpostaSuTavoloDaTavolo(  
  disco,  
  asse_di_partenza,  
  asse_di_arrivo  
)  
  
precondizioni  
- tipi  
Disco(disco) ∧ Asse(asse_di_partenza) ∧ Asse(asse_di_arrivo) ∧  
  
asse_ord_disco(asse_di_partenza, disco, Tavolo) ∧  
minore(asse_di_partenza, disco) ∧  
minore(asse_di_arrivo, Tavolo)  
  
effetto  
- disco = pop(asse_di_partenza)  
→ asse_ord_disco(asse_di_partenza, disco, Tavolo) ∧  
→ minore(asse_di_partenza, disco) ∧  
→ minore(asse_di_arrivo, Tavolo) ∧  
  
- push(asse_di_arrivo, disco)  
asse_ord_disco(asse_di_arrivo, disco, Tavolo) ∧  
minore(asse_di_partenza, Tavolo) ∧  
minore(asse_di_arrivo, disco)
```