

# PLANNING DEDUTTIVO

---

Prendiamo l'esempio visto a lezione e usiamo la formulazione di Kovalski:

Stato iniziale

*holds(on(a,d),s0).*

*holds(on(b,e),s0).*

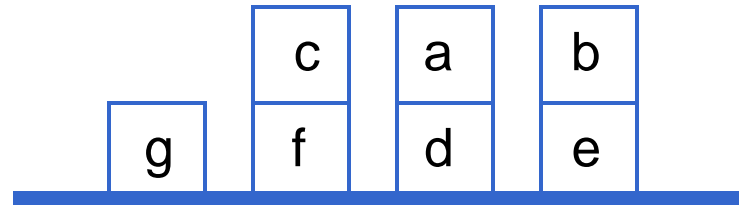
*holds(on(c,f),s0).*

*holds(clear(a),s0).*

*holds(clear(b),s0).*

*holds(clear(c),s0).*

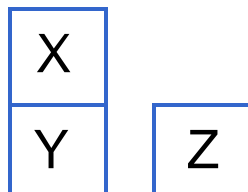
*holds(clear(g),s0).*



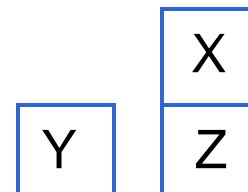
# PLANNING DEDUTTIVO

---

Consideriamo l'unica azione  $move(X,Y,Z)$



*on(X, Y)*  
*clear(X)*  
*clear(Z)*



*clear(Y)*  
*on(X, Z)*

# PLANNING DEDUTTIVO

---

%Effetti dell'azione move(X,Y,Z):

*holds(clear(Y),do(move(X,Y,Z),S)).*

*holds(on(X,Z),do(move(X,Y,Z),S)).*

%Clausola per esprimere le condizioni di frame:

*holds(V,do(move(X,Y,Z),S)):-*

*holds(V,S),*

*V\=clear(Z),*

*V\=on(X,Y).*

# PLANNING DEDUTTIVO

---

% Clausola che esprime le precondizioni dell'azione `move(X,Y,Z)`:

`pact(move(X,Y,Z),S):-`

`holds(clear(X),S), holds(clear(Z),S),`

`holds(on(X,Y),S), X\=Z.`

% Clausola per esprimere la raggiungibilità di uno stato:

`poss(s0).`

`poss(do(U,S)):-`

`poss(S),`

`pact(U,S).`

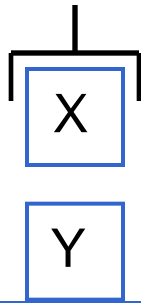
# PLANNING DEDUTTIVO

---

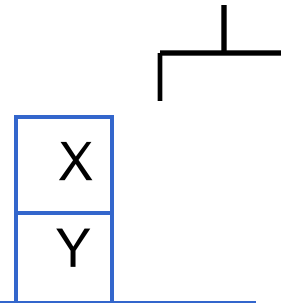
- NOTA: abbiamo una clausola che esprime condizioni di frame per ogni AZIONE
- Goal:  
$$:- \text{poss}(S), \text{holds}(\text{on}(a,b), S), \text{holds}(\text{on}(b,g), S).$$
- Attivare il trace per monitorare la risoluzione
- Verificare la costruzione di altri piani con altre query

# PLANNING DEDUTTIVO

- Secondo esercizio: usare il mondo a blocchi ma cambiare e modellare le azioni

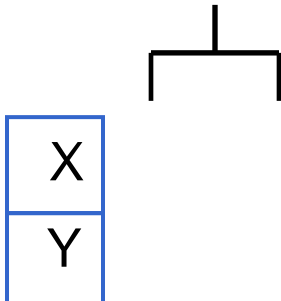


*clear(Y), holding(X)*

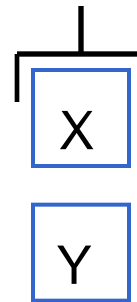


***STACK(X,Y)***

*on(X,Y), handempty, clear(X)*



*on(X,Y), handempty, clear(X)*

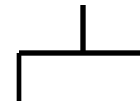
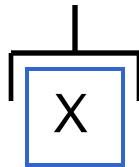


***UNSTACK(X,Y)***

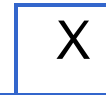
*clear(Y), holding(X)*

# PLANNING DEDUTTIVO

- Secondo esercizio: usare il mondo a blocchi ma cambiare e modellare le azioni

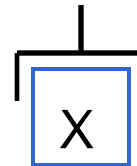
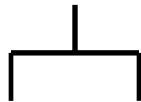


***PUTDOWN(X)***



*holding(X)*

*ontable(X), handempty, clear(X)*



***PICKUP(X)***



*ontable(X), handempty, clear(X)*

*holding(X)*

# PLANNING DEDUTTIVO

---

Si modellino ora le seguenti azioni

Caricamento di un oggetto

```
load(Oggetto, Carrello, Location)
```

```
PREC: at(Oggetto, Location), at(Carrello, Location)
```

```
ADD LIST: in(Oggetto, Carrello)
```

```
DELETE LIST: at(Oggetto, Location)
```

Trasporto

```
drive(Carrello, Location1, Location2)
```

```
PREC: at(Carrello, Location1), connected(Location1, Location2)
```

```
ADD LIST: at(Carrello, Location2)
```

```
DELETE LIST: at(Carrello, Location1)
```

Scaricamento di un oggetto

```
unload(Oggetto, Carrello, Location)
```

```
PREC: at(Carrello, Location), in(Oggetto, Carrello)
```

```
ADD LIST: at(Oggetto, Location)
```

```
DELETE LIST: in(Oggetto, Carrello)
```



# PLANNING DEDUTTIVO

---

Con il seguente stato iniziale e goal

Stato iniziale:

```
in(carico1,carrello1), at(carrello1,milano)  
connected(milano,bologna), connected(bologna,roma)
```

Stato goal: at(carico1,roma)