L'algoritmo POP dal libro

Costruttori (i dati)

- `plan(Azioni, OrdineParziale, LinkCausali)`
 - Azioni : lista di azioni,
 - OrdineParziale: lista di coppie $a < b$
 - LinkCausali: terne `cl(A,P,B)` A produce la precondizione P di B
- `Agenda = lista [G1,...,Gn]` di goals (cose da fare)
 - `goal G: goal(P,A)` P una precondizione di A

Le minacce

- Consistenza di Ord
 - non ci sono cicli (es: $a < b < c < a$ è inconsistente, per transitività porterebbe ad $a < a$; caso particolare $a < b < a$)
- A1 minaccia `cl(A,P,B)` in Ord se
 - $A < A1 < B$ è consistente con Ord e
 - P appartiene alla deletelist di A1 (not P è un effetto di A1)
- Consistenza di un piano:
`plan(Azioni, Ordine, LinksCausali)` consistente se Ordine consistente, le azioni nel piano sono Azioni, e nessun link causale in LinkCausali è minacciato, in Ord, da una azione in Azioni.

Le operazioni di base

- `add_constraint(in: A < B, in: Ord, out: Ord1)`
`Ord1` = estensione minimale di Ord consistente contenete $A < B$
- `incorporate_action(in: A, in: LinkCausali, in: Ord, out: Ord1)`
è possibile incorporare A perchè trovo un'estensione `Ord1` di Ord in cui nessuno dei LinkCausali è minacciato da A
- `incorporate_causal_link(in:cl(A,P,B), in: LAzioni, in: Ord, out: Ord1)`
è possibile incorporare `cl(A,P,B)` perchè trovo un'estensione `Ord1` di Ord in cui nessuna delle azioni in LAzioni minaccia `cl(A,P,B)`
- `achieves(A,P) : P` è reso vero da A
- `add_preconds(A,Agenda,NAgenda) : NAgenda` contiene Agenda e tutti i nuovi goals `goal(P,A)`, con P precondizione di A

L'algoritmo

```
/* pop(Piano1,Agenda,Piano2) : Piano1 consistente, Piano2 piano finale,
ottenuto con agenda vuota *****/
```

```
pop(Plan,[],Plan).
pop(Plan1,Agenda,PlanF) :- select(G,Agenda,Agenda1),
solve_goal(G,Plan1,NPlan,Agenda1,NAgenda),
pop(NPlan,NAgenda,PlanF).
```

```
/** il cuore è solve_goal *****/
```

```
/** il cuore è solve_goal: due casi:
caso 1: trovo una azione già nel piano che risolve il goal *****/
```

```
solve_goal(goal(P,A1), plan(Ain,Oin,Lin),
plan(Ain,Ofin,[cl(A0,P,A1)]LS), Ag, Ag) :-
member(A0,Ain),
achieves(A0,P),
add_constraint(A0<A1,Oin,O1),
incorporate_causal_link(cl(A0,P,A1),Ain,O1,Ofin).
```

```
/** il cuore è solve_goal: due casi:  
caso 2: trovo una azione nuova *****/
```

```
solve_goal(goal(P,A1), plan(Ain,Oin,Lin),  
           plan(Afin,Ofin,[cl(A0,P,A1)]LS), Ag, NAg) :-  
    action(A0),  
    achieves(A0,P),  
    add_constraint(A0<A1,Oin,O1),  
    add_constraint(start < A0,O1,O2),  
    incorporate_action(A0, Lin,O2,O3),  
    incorporate_causal_link(cl(A0,P,A1),Ain,O3,Ofin),  
    add_preconds(A0,Ag,NAg).
```

Una esecuzione

- Riprendiamo alla lavagna la esecuzione con il mondo dei blocchi