Partial-Order Planning

- traditional planner enforce a total ordering on the actions
- leads to a high number of alternative plans, even if the sequence of actions is irrelevant
- partial-order planning:
  - leave the order of actions underspecified
  - only commit to an ordering when forced to do
- also called non-linear planner

Partial Order

- explicit linear precedence relation between the possible actions of a plan

$$A_0 < A_1$$

  - linear precedence is transitive and asymmetric
- additionally assumed pseudo actions *start* and *finish*
- any total ordering consistent with the partial ordering is a solution
- two step procedure:
  - obtain a partial plan
  - extract a solution

Causal Links

- every precondition P of A_1 will have an action A_0 associated that achieves that precondition

$$cl(A_0, P, A_1)$$

- if there is a causal link between A_0 and A_1, $A_0 < A_1$ must be part of the plan
- A_0 is said to support P
- any action A_2 which deletes P must be either before A_0 or after A_1

Partial Plan

- a partial plan is a 3-tupel $(\mathcal{A}, \mathcal{O}, \mathcal{L})$ with
  - $\mathcal{A}$ a set of actions
  - $\mathcal{O}$ a linear precedence relation over $(\mathcal{A})$
  - $\mathcal{L}$ a set of causal links
- plan $P_1 = (\mathcal{A}_1, \mathcal{O}_1, \mathcal{L}_1)$ is an extension of plan $P_2 = (\mathcal{A}_2, \mathcal{O}_2, \mathcal{L}_2)$ if $\mathcal{A}_2 \subseteq \mathcal{A}_1$ and $\mathcal{O}_2 \subseteq \mathcal{O}_1$ and $\mathcal{L}_2 \subseteq \mathcal{L}_1$
- action A threatens a causal link $cl(A_0, P, A_1)$ if A_0 deletes P
- a plan is safe whenever action A threatens $cl(A_0, P, A_1)$, the partial order $\mathcal{A}$ entails either $A < A_0$ or $A_1 < A$

Algorithm

- agenda: list of subgoals $goal(P, A_1)$
- initially: preconditions for finish, i.e. the final goal to be achieved
- choose a subgoal P which is a precondition for action A_1
- choose an action A_0 which supports P
  - if A_0 is already in the agenda:
    - add an ordering constraint $A_0 < A_1$
    - add a causal link between $cl(A_0, P, A_1)$
    - for any action A_2 in the plan that threatens P add a precedence constraint $A_2 < A_0$ or $A_1 < A_2$
  - if A_0 is a new action
    - add its preconditions to the agenda
- continue until the agenda is empty

Algorithm

- nondeterministic procedure with two choice points
  - Which action A_0 to selected to achieve P?
  - Whether to place action A_2 which deletes P before A_0 or after A_1?
- if actions may occur twice in a plan, they need to be indexed to be able to order instances of actions

Example

- goal: $carrying(rob, parcel) \wedge sitting\_at(rob, lab2)$
- call:

$$
\begin{aligned}
&?-pop(\\
&\qquad plan([start, finish], [start < finish], [\,]),\\
&\qquad\qquad [goal(carrying(rob, parcel), finish),\\
&\qquad\qquad\ goal(sitting\_at(rob, lab2), finish)],\\
&\qquad [\,]).
\end{aligned}
$$

Example (cont.)

- select the first subgoal: $goal(carrying(rob, parcel), finish)$
- resulting plan:

$$
\begin{aligned}
plan(\ & [pickup(rob, parcel, P), start, finish], \\
& [start < finish, \\
& \ start < pickup(rob, parcel, P), \\
& \ pickup(rob, parcel, P) < finish], \\
& [cl(pickup(rob, parcel, P), carrying(rob, parcel), finish)])
\end{aligned}
$$

- resulting agenda:

$$
\begin{aligned}
& [goal(sitting\_at(rob, lab2), finish), \\
& \ goal(sitting\_at(parcel, P), pickup(rob, parcel, P)), \\
& \ goal(at(rob, P), pickup(rob, parcel, P))]
\end{aligned}
$$

Example (cont.)

- select the next subgoal: $goal(sitting\_at(rob, lab2), finish)$
- resulting plan:

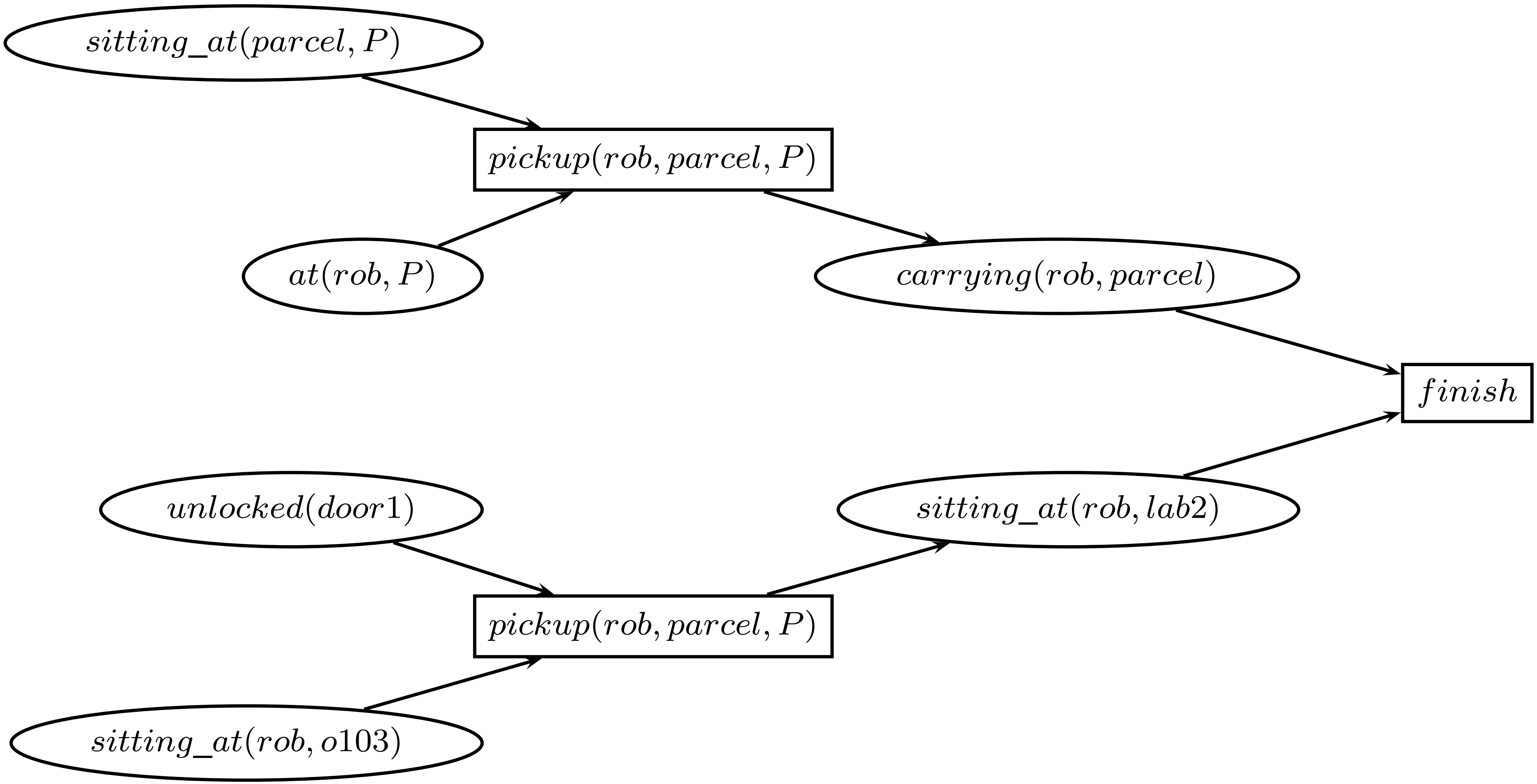
$$
\begin{aligned}
plan(\ & [move(rob, o103, lab2), pickup(rob, parcel, P), start, finish], \\
& [start < finish, \\
& \ start < pickup(rob, parcel, P), \\
& \ pickup(rob, parcel, P) < finish, \\
& \ start < move(rob, o103, lab2), \\
& \ move(rob, o103, lab2) < finish], \\
& [cl(move(rob, o103, lab2), sitting\_at(rob, lab2), finish), \\
& \ cl(pickup(rob, parcel, P), carrying(rob, parcel), finish)])
\end{aligned}
$$

Example (cont.)

- resulting agenda:

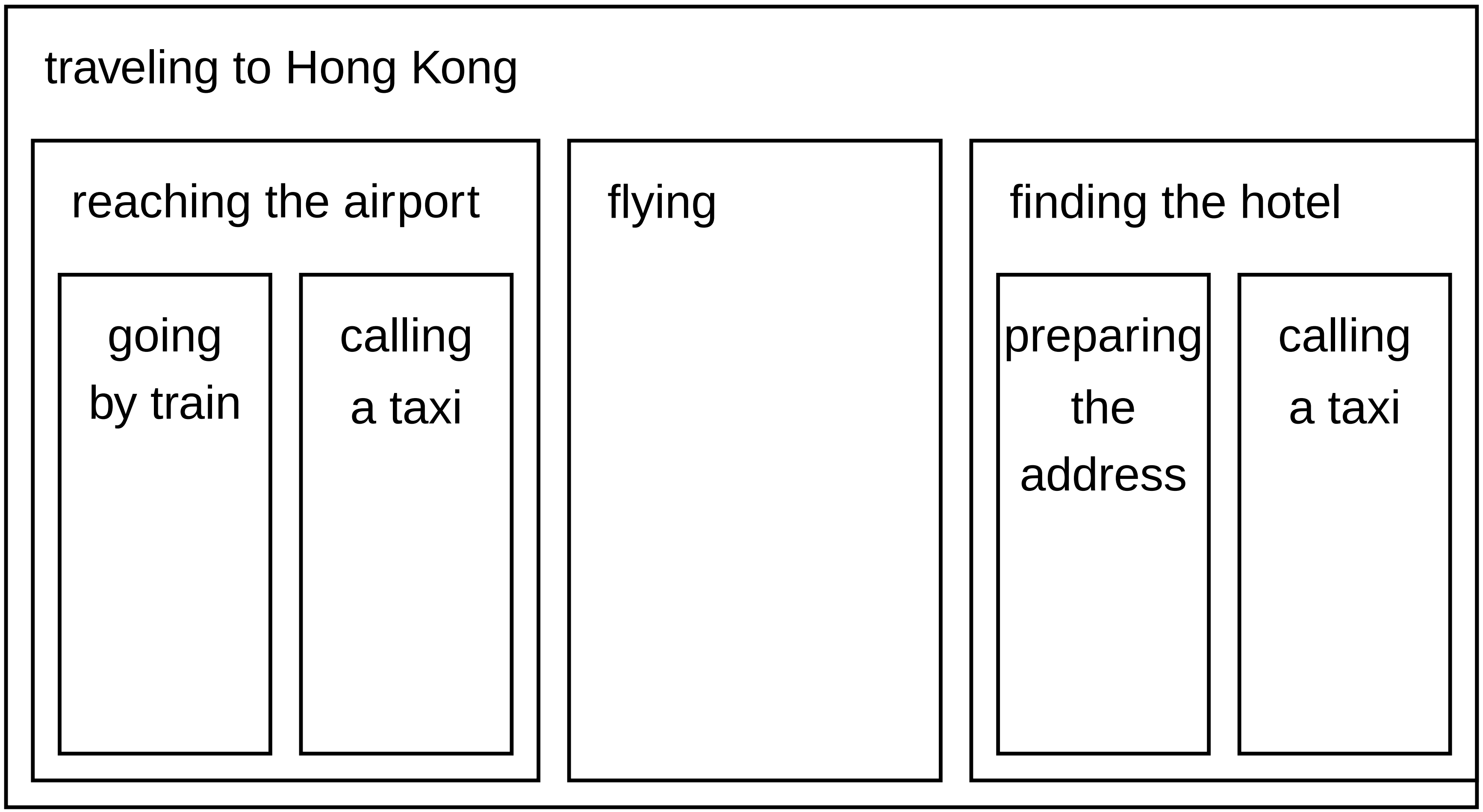
$$[goal(sitting\_at(parcel, P), pickup(rob, parcel, P)),$$
$$goal(at(rob, P), pickup(rob, parcel, P)),$$
$$goal(unlocked(door1), move(rob, o103, lab2)),$$
$$goal(sitting\_at(rob, o103), move(rob, o103, lab2))]$$

Example


sitting_at(parcel, P)
pickup(rob, parcel, P)
at(rob, P)
carrying(rob, parcel)
finish
unlocked(door1)
sitting_at(rob, lab2)
pickup(rob, parcel, P)
sitting_at(rob, o103)

Efficiency issues

- hierarchical planning
  - describe the problem space on different levels of granularity



Efficiency issues

- preprocessing
  - collect information about the planning problem
- reordering goals
  - computing of a goal agenda
  - fundamental relation: B needs to be achieved before B
  - domain dependent: $washing(X) \prec drying(X)$
  - domain independent: B cannot be achieved after A has been achieved
- incremental plan extension
  - start with a subset of goals
  - extend it by including an increasing number of goals

Reactive planning

- the plan has to be found within certain temporal bounds
- incremental planning
  - planning and action needs to be interleaved
- plan refinement
  - start with a tentative plan and try to improve it by transformation