Outline

- Deterministic sequential planning
- Best planners of IPC 2011
- Plan-space planning
- My Basic Planner
Restricted State-Transition System $\Sigma = (S, A, \gamma)$

- $S$ - Set of states
- $A$ - Set of actions
- $(E$ - Set of events$)$
- $\gamma$ - Transition function $\gamma : S \times A \rightarrow P(S)$
Planning Problem \( P = (\Sigma, s_0, g) \):

- \( \Sigma \) - System modeling \textit{states} and \textit{transitions}
- \( s_0 \) - The \textit{initial state}
- \( g \) - The \textit{goal states}
States and Goals

- Represented as **sets of facts**
- **Closed World Assumption (CWA)**
  - Fact not listed in a state are assumed to be false
- **Goal state** - any state with all the goal facts
Operators and Actions

- **Operator** \( o = (\text{name}(o), \text{precond}(o), \text{effects}(o)) \)
- An **action** is any ground instance of an operator
- **Move**\( (r, l, m) // \) Example of an operator
  - **Precond:**
    adjacent\( (l, m) \), \text{at}(r, l), \text{not occupied}(m)
  - **Effects:**
    \text{at}(r, m), \text{occupied}(m), \text{not occupied}(l), \text{not at}(r, l)
Solution of Planning Problem

- Sequence of actions \( \langle a_1, a_2, ..., a_k \rangle \)
- Sequence of states \( \langle s_0, s_1, ..., s_k \rangle \)
- Such that:
  - \( s_i = \gamma(s_{i-1}, a_i) \)
  - \( s_k \) satisfies \( g \)
Extensions of the Classical Representation

- Typed variables
- Non-negative costs
- Conditional effects
- Optional - derived predicates
Sequential Satisfying Domains

- Sokoban
- ScanAnalyzer
- Peg solitaire
- TSP
- Elevators
- Transport
Best planners of IPC 2011

<table>
<thead>
<tr>
<th>pegsol</th>
<th>scanl2</th>
<th>parcpr</th>
<th>openst</th>
<th>nomys</th>
<th>tidybot</th>
<th>woodw</th>
<th>sokoban</th>
<th>trnsptr</th>
<th>vistall</th>
<th>elevat</th>
<th>parking</th>
<th>barman</th>
<th>frtite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lama-2011</td>
<td>fdss-1</td>
<td>fdss-2</td>
<td>fd-autotune-1</td>
<td>roamer</td>
<td>fd-autotune-2</td>
<td>forkuniform</td>
<td>probe</td>
<td>arvand</td>
<td>lama-2008</td>
<td>lamar</td>
<td>randward</td>
<td>brt</td>
<td>cbp2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>daeyahsp</td>
<td>yahsp2</td>
<td>yahsp2</td>
<td>yahsp2-mt</td>
<td>cbp</td>
<td>lprpgp</td>
<td>madagascar-p</td>
<td>popf2</td>
<td>madagascar</td>
<td>cpt4</td>
<td>satplanlm-c</td>
<td>sharaabi</td>
<td>acoplan</td>
<td>acoplan2</td>
</tr>
</tbody>
</table>
Some used techniques

- Forward search, planning graph
- **Landmarks** - variable assignments that must occur at some point in every solution plan
- ACOPlan - Ant colony optimalization
- Arvand - Monte Carlo random walks (MRW)
- BRT - (Biased Rapidly exploring Tree)
- Divide-and-Evolve – Evolutionary computation
- Fast downward (autotune), lama – various algorithms and heuristics
- Madagascar – SAT
- POPF2 - Forward-Chaining Partial Order Planner
Plan-Space Planning

- Partially specified plans
- Refinement operations
- Least commitment principle
Partial Plan $\Pi = (A, <, B, L)$

- A - Set of partially instantiated operators $\{a_1, \ldots, a_k\}$
- $<$ - Partial order on A $(a_i < a_j)$
- B - Set of constraints $x=y, x\neq y$ or $x\in D_x$
- L - Set of causal relations $(p: a_i \rightarrow a_j)$
Partial Plan $\Pi = (A, <, B, L)$
Plan-Space Planning

- Start with an **empty plan**
- **Repair** all **flaws** in partial plan step by step
  - Add actions to satisfy **open goals**
  - Remove threats
    - Bind variables
    - Add ordering between actions
    - Add causal relations
Solution for problem $P = (E, s_0, g)$

- Partial plan $\Pi = (A, <, B, L)$
  - Partial ordering $<$ and constraints $B$ are globally consistent
  - Any linearly ordered sequence of fully instantiated actions from $A$ satisfying $<$ and $B$ goes from $s_0$ to a state satisfying $g$
Solution for problem $P = (E, s_0, g)$

- Partial plan $\Pi = (A, <, B, L)$
  - Partial ordering $<$ and constraints $B$ are globally consistent
  - There are no flaws
    - No open goals
    - No threats
PSP procedure

PSP(\(\pi\))

\[
\begin{align*}
\text{flaws} & \leftarrow \text{OpenGoals}(\pi) \cup \text{Threats}(\pi) \\
\text{if} \text{ flaws} = \emptyset & \text{ then return } \pi \\
\text{select any flaw } \varphi & \in \text{ flaws} \\
\text{resolvers} & \leftarrow \text{Resolve}(\varphi, \pi) \\
\text{if} \text{ resolvers} = \emptyset & \text{ then return failure} \\
\text{non-deterministically choose} & \text{ a resolver } p \in \text{ resolvers} \\
\pi' & \leftarrow \text{Refine}(p, \pi) \\
\text{return} & \text{ PSP}(\pi')
\end{align*}
\]
Algorithm PoP

PoP(π, agenda) // where π = (A, <, B, L)

if flaws = Ø then return π
setect any pair (aj, p) in and remove it from agenda
relevant ← Providers(p, π)
if relevant = Ø then return failure
nondeterministically choose an action ai ∈ relevant
L ← L U { (p: ai → aj) }
update B with the binding constraints of this causal link
if ai is a new action in A then
    update A with ai
    update < with (ai < aj), (a0 < ai < ai∞)
    update agenda with all preconditions of ai
for each threat on (p: ai → aj) or due to ai do
    resolvers ← set of resolvers for this threat
    if resolvers = 0 then return failure
    non-deterministically choose a resolver in resolvers
    add that resolver to < or to B
return PoP(π, agenda)
Algorithm PoP - extensions

- Conditional operators
- Flaw-Selection Heuristics
- Resolver-Selection Heuristics
My Basic Planner

- **PDDL parser**
- **Preprocessor**
  - Analyze operators/actions
    - map possible predecessors/successors for each action
    - Replace some operators with meta-operators
  - Analyze domain/instance of the problem
    - Derive some restrictions
- **Plan-Space planning**
- (CSP for some sub-problems)
Possible future extensions

- Durative actions
- Qualitative (temporal) relations
- Preferences
- Learning (domain specific properties)
References

- Automated Planning: Theory and Practice; M. Ghallab, D. Nau, P. Traverso; Morgan Kaufmann, 2004
- http://sokoban-jd.blogspot.cz/
- http://en.wikipedia.org/wiki/Peg_solitaire
- The Scanalyzer Domain: Greenhouse Logistics as a Planning Problem; M. Helmert, H. Lasinger; ICAPS 2010
- http://www.fast-downward.org/
- http://ipc.icaps-conference.org/
- http://www.plg.inf.uc3m.es/ipc2011-deterministic/