Partial-Order Causal-Link Planning with Preferences

Fabian Ginter

Master thesis at the Institute of Artificial Intelligence

March 14th, 2012

Content

- Motivation
- Task of the master thesis
- Definition of a planning problem with preferences
- Solving planning problems with preferences
  - Approaches to resolve preferences
  - Selection methods for preferences
- Evaluation
- Summary

Motivation

Solving problem "travel from Ulm to Munich":

- Example for P1: 1 Task, Costs 100€, Time 3h, 0 Transfers

  Ulm → Car → Munich

- Example for P2: 2 Tasks, Costs 130€, Time 2.5h, 1 Transfer

  Ulm → Taxi → Station → Train → Munich

How to measure the quality of those plans?

- Typical quality measurements of a plan are costs like money, time or number of tasks.

Motivation

We can clearly measure which plan is cheaper, faster or has more tasks, but what about other properties?

- P1: Car is at Munich in the final state

  Ulm → Car → Munich

- P2: Car is at home in the final state

  Ulm → Taxi → Station → Train → Munich

→ Preferences can be used to describe properties of plans to support each user individually

→ Preferences are state features, called Soft-Goals, that are optional

Motivation

Features provided by preferences:

- Quality of plans can be measured by taking Soft-Goals into account
- Soft-Goals are optional whereas "normal" goals must be fulfilled in order to find a solution (which might be too restrictive, since humans tend to define more goals than possibly satisfiable)

Task of the master thesis

- Extend PANDA_2
  - Integrate and handle Soft-Goals, which correspond to "at-end" preferences in PDDL3.0
  - Develop different strategies to handle Soft-Goals
  - Implement planning graph
- Evaluation of the implementation

Planning problem with preferences

Definition of a planning problem: $\pi = \langle V, A, s_{init}, s_{goal}, Prefs \rangle$

- V is a finite set of state variables
- A is a finite set of actions where:
  - $a = \langle pre, add, del \rangle \in A$ where $pre, add, del \subseteq V$
  - a is applicable in a state $s \in 2^V$ iff $pre \subseteq s$
  - If a is applied in s, it leads to $(s \setminus del) \cup add$
- $s_{init} \in 2^V$ and $s_{goal} \subseteq V$

Preferences: Can be arbitrary formulæ over literals which can be transformed into "Disjunctive Normal Form" (DNF)

- $Prefs$ is a finite set of preferences
- $\{disjunct_1, \ldots, disjunct_n\} \in Prefs$ represents a formula in DNF, where $disjunct_j \subseteq V, j = 1, \ldots, n$

Planning problem with preferences

Action sequence $\overline{a}$ is a solution iff $\overline{a}(s_{init})$ fulfills s_{goal}

Possible quality measurement called "Benefit":

- $\mathtt{Ben}(\overline{a}) = \sum_{\substack{p \in Prefs \\ \overline{a}(s_{init}) \text{ fulfills } p}} \mathtt{benefit}(p)$

Used quality measurement called "NetBenefit":

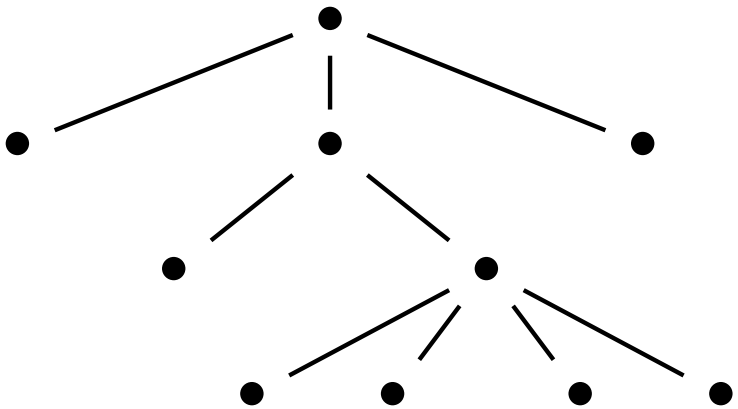
- $\mathtt{NetBen}(\overline{a}) = \sum_{\substack{p \in Prefs \\ \overline{a}(s_{init}) \text{ fulfills } p}} \mathtt{benefit}(p) - \sum_{a \in \overline{a}} \mathtt{costs}(a)$

Solving planning problems with preferences

- Based on a fringe which contains a set of partial plans
- Planselection selects a partial plan out of the fringe
- Partial plan has flaws, which need to be addressed in order to get a plan
- Flawselection decides which flaw of the partial plan is addressed

  $\rightarrow$ Results in new partial plans that are branched into the fringe

Illustration of the search space:



Approaches to resolve preferences

- Developed and implemented two different strategies:
  - "Split strategy"
  - "No-split strategy"
- Basis for both strategies are preferences in DNF

$\rightarrow$ Presentation of strategies on next slides

Approaches to resolve preferences — "Split strategy"

→ Basic idea: Split DNF into its disjuncts

First selection of the preference p (example):

$$p = \psi_{1_1} \vee (\psi_{2_1} \wedge \psi_{2_2})$$

$$p_1 = \psi_{1_1} \qquad\qquad p_2 = \psi_{2_1} \wedge \psi_{2_2}$$

Each subsequent selection of the preference:

$$p_1 = \psi_{1_1} \qquad\qquad p_2 = \psi_{2_1} \wedge \psi_{2_2}$$

Supporters: 2

Supporters: 1, 3

Approaches to resolve preferences — "No-split strategy"

→ Basic idea: Stay flexible and work on different disjuncts

$$p = \psi_{1_1} \vee (\psi_{2_1} \wedge \psi_{2_2})$$

Supporters

2 1 3

- Literals are protected according to the used flaw selection strategy which will be explained on the next slides

Selection methods for preferences

How do we decide which literal of which preference we protect next?

$\rightarrow$ Basic idea: LCFR (Least Cost Flaw Repair)

$$LCFR_{DNF}\text{-}\circ(P) = \min_{i}\left[\min_{\varphi_j}\left(\circ_{\psi_{j_k}} |\texttt{supporter}(P, f_i(\psi_{j_k}))|\right)\right]$$

$$\text{with } i = 1, \ldots, |Prefs(P)| \text{ and } \circ \in \left\{\prod, \sum, \min\right\}$$

$p_i \in Prefs, p_i = \varphi_1 \vee \ldots \vee \varphi_n, \varphi_j = \psi_{j_1} \wedge \ldots \wedge \psi_{j_k}, j = 1, \ldots, n$

$\rightarrow$ 3 Strategies:

- $LCFR_{DNF}\text{-}\prod$
- $LCFR_{DNF}\text{-}\sum$
- $LCFR_{DNF}\text{-}\min$

Selection methods for preferences — $LCFR_{DNF}$-$\prod$

$$LCFR_{DNF}\text{-}\prod(P) = \min_i \left[\min_{\varphi_j} \left(\prod_{\psi_{j_k}} |\texttt{supporter}(P, f_i(\psi_{j_k}))| \right) \right]$$

Example: Plan P has a preference p

$$p = \psi_{1_1} \vee (\psi_{2_1} \wedge \psi_{2_2})$$

Supporters

Calculated values: 2 $\quad 1 * 3 = 3$

$\rightarrow LCFR_{DNF}\text{-}\prod = 2 \rightarrow \varphi_1$ is selected

$\rightarrow$ Literal with fewest supporters of φ_1 is ψ_{1_1}

Selection methods for preferences — $LCFR_{DNF}$-$\sum$

$$LCFR_{DNF}\text{-}\sum(P) = \min_i \left[\min_{\varphi_j} \left(\sum_{\psi_{j_k}} |\texttt{supporter}(P, f_i(\psi_{j_k}))| \right) \right]$$

Example: Plan P has a preference p

$$p = \psi_{1_1} \vee (\psi_{2_1} \wedge \psi_{2_2})$$

Supporters

Calculated values: 2 $\quad 1 + 3 = 4$

$\rightarrow LCFR_{DNF}\text{-}\sum = 2 \rightarrow \varphi_1$ is selected

$\rightarrow$ Literal with fewest supporters of φ_1 is ψ_{1_1}

Selection methods for preferences — $LCFR_{DNF}$-min

$$LCFR_{DNF}\text{-min}(P) = \min_{i} \left[\min_{\varphi_j} \left(\min_{\psi_{j_k}} |\texttt{supporter}(P, f_i(\psi_{j_k}))| \right) \right]$$

Example: Plan P has a preference p

$$p = \psi_{1_1} \vee (\psi_{2_1} \wedge \psi_{2_2})$$

Supporters

Calculated values: 2 1 3

$\rightarrow LCFR_{DNF}\text{-min} = 1 \rightarrow \psi_{2_1}$ is selected.

Evaluation

Evaluation compares:

- Solve "normal" goals first vs. solve Soft-Goals first
- "Split strategy" vs. "No-split strategy"
- $LCFR_{DNF}$-min vs. $LCFR_{DNF}$-$\prod$ vs. $LCFR_{DNF}$-$\sum$
- "A* NetBenefit" vs. "Uniform NetBenefit"

  $\rightarrow$ 24 resulting configurations

Used planning domains and problems:

- 200 randomly generated planning problems

  $\rightarrow 24 * 200 = 4800$ test cases
- 6 user assistance planning problems modeled by Claudia Bolch

  $\rightarrow 24 * 6 = 144$ test cases

Evaluation — Which kind of goals should be solved first?

Solve Soft-Goals first

Random-Problems:

- 2400 tests done
- 6 solutions found

  → Only 0.25% solved

User-Assistance-Problems:

- 72 tests done
- 49 solutions found

Solve "normal" goals first

Random-Problems:

- 2400 tests done
- 2400 solutions found

  → 100% solved

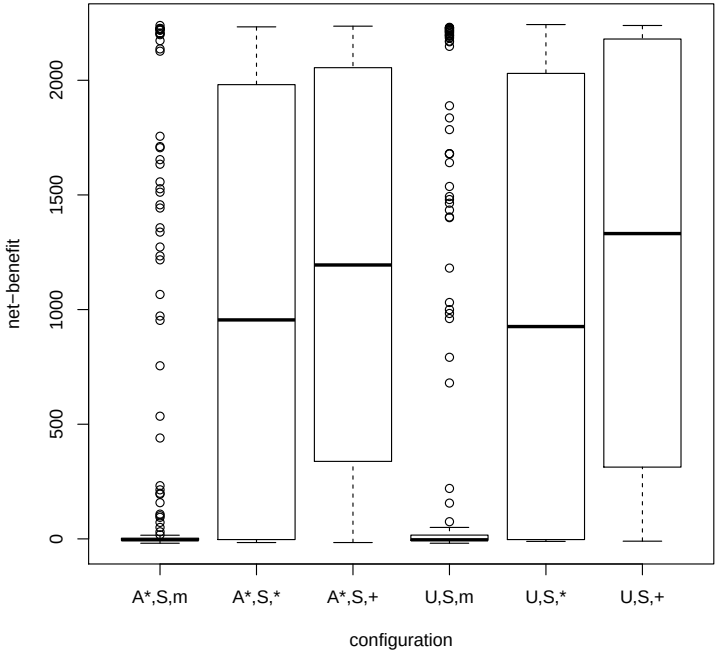
User-Assistance-Problems:

- 72 tests done
- 63 solutions found

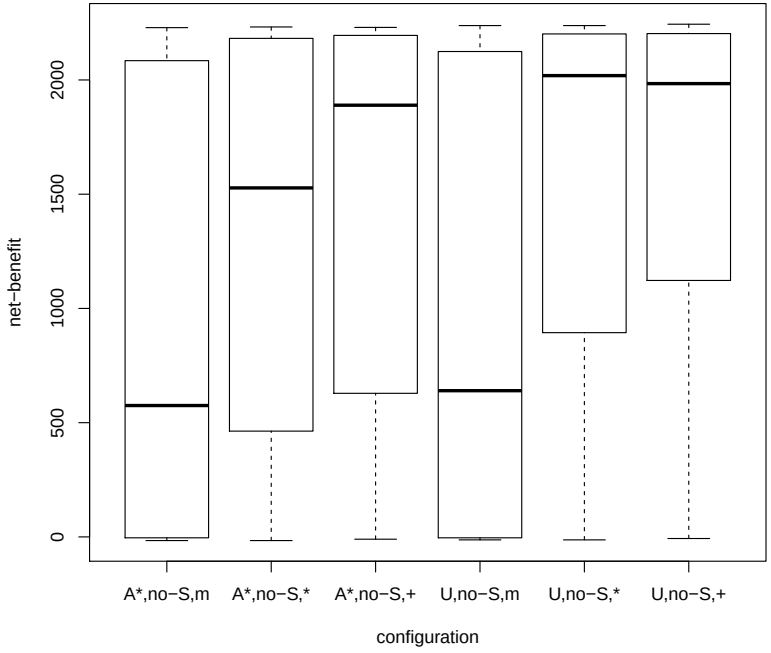
→ It's more important to find a solution than solving preferences
→ Therefor "normal" goals must be solved first

Evaluation — Approaches to resolve preferences

"Split strategy"



"No-split strategy"

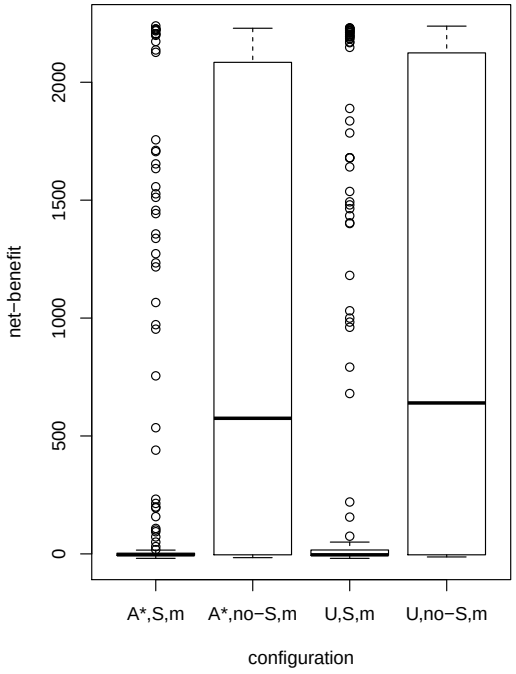


→ "No-split strategy" dominates "Split strategy"

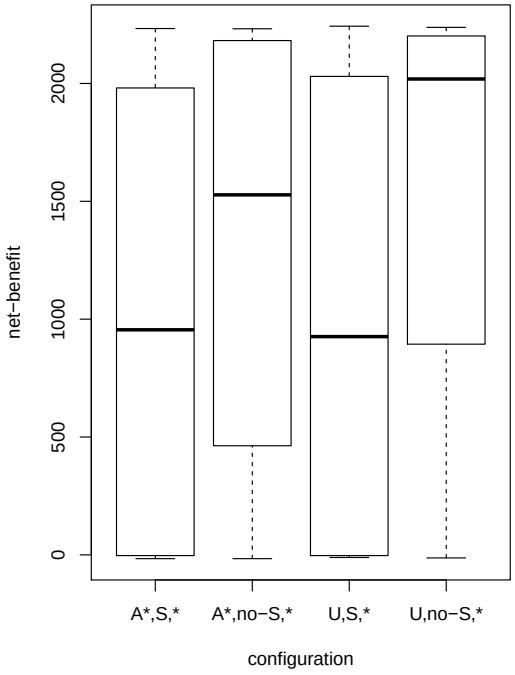
→ But: "Split strategy" has room for improvement

Evaluation — Selection methods for preferences

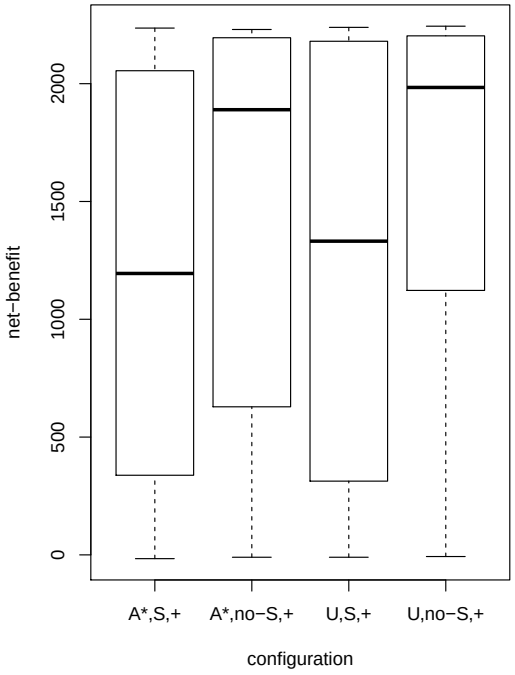
$LCFR_{DNF}$-min



$LCFR_{DNF}$-$\prod$



$LCFR_{DNF}$-$\sum$



→ As expected: $LCFR_{DNF}$-min is dominated by other strategies

→ Reason: $LCFR_{DNF}$-min is not aware of DNF

Summary

- Planning algorithm was extended to enable the use of preferences
  - Developed and implemented different methods to resolve preferences
  - Developed and implemented different methods for soft-flaw selection
  - Evaluated combinations of different strategies
- Implemented the planning graph (used for heuristic estimation)

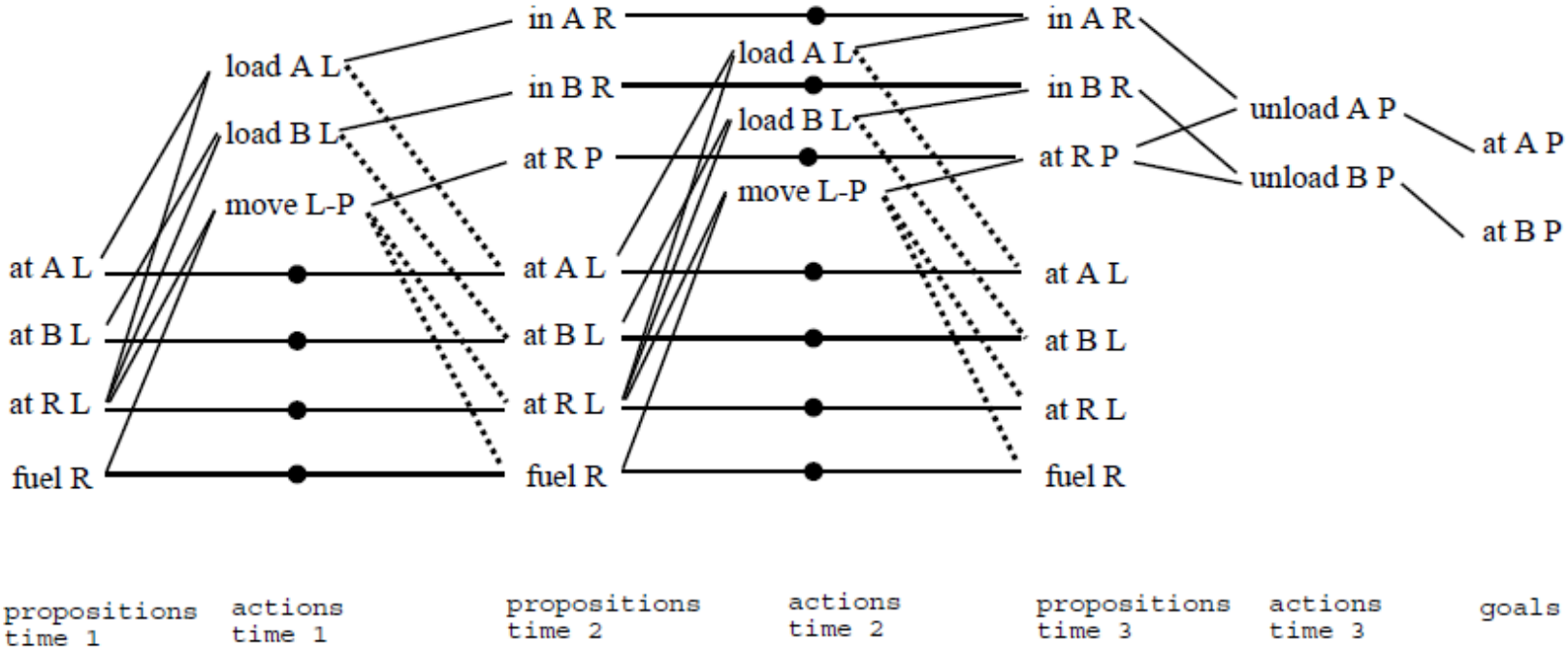
Appendix — Planning algorithm

Algorithm 1: Preference-based POCL planning algorithm.

Input : The fringe $\texttt{Fringe} = \{P_{init}\}$.
Output : A plan or $fail$.

```
best-nB := −∞
best-P := fail
while Fringe ≠ ∅ and "no timeout" do
    P = planSel(Fringe)
    Fringe = Fringe \ {P}
    if flaws_h(P) = ∅ and netBen(P) > best-nB then
        best-nB := netBen(P)
        best-P := P
    f = flawSel(flaws_h(P) ∪ flaws_s(P))
    Fringe = Fringe ∪ resolveFlaw(P, f)
return best-P
```

Appendix — Example: Planning Graph



Appendix — Semantics of preferences in PDDL3.0

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{at end}\ \phi) \quad \text{iff} \quad S_n \models \phi$$

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models \phi \quad \text{iff} \quad S_n \models \phi$$

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{always}\ \phi) \quad \text{iff} \quad \forall i : 0 \leq i \leq n \cdot S_i \models \phi$$

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{sometime}\ \phi) \quad \text{iff} \quad \exists i : 0 \leq i \leq n \cdot S_j \models \phi$$

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{within}\ t\ \phi) \quad \text{iff} \quad \exists i : 0 \leq i \leq n \cdot S_i \models \phi \text{ and } t_i \leq t$$

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{at-most-once}\ \phi) \quad \text{iff} \quad \forall i : 0 \leq i \leq n \cdot \text{ if } S_i \models \phi \text{ then } \exists j : j \geq i \cdot \forall k : i \leq k \leq j \cdot S_k \models \phi \text{ and } \forall k : k > j \cdot S_k \models \neg\phi$$

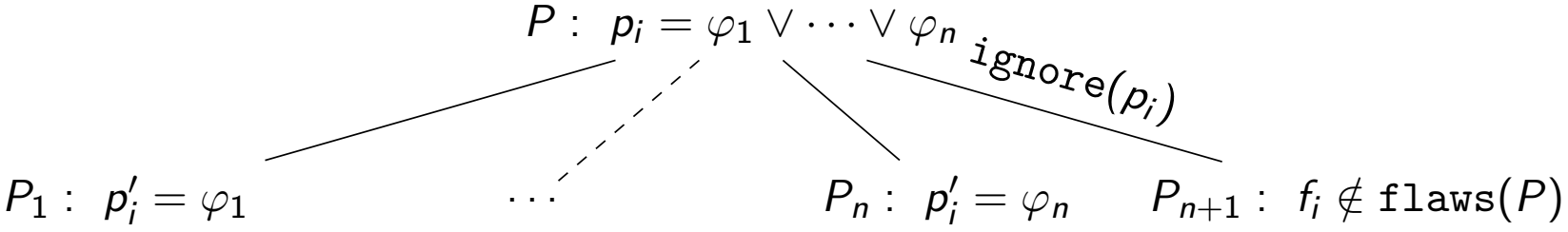
$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{sometime-after}\ \phi\ \psi) \quad \text{iff} \quad \forall i \cdot \text{if } S_i \models \phi \text{ then } \exists j : i \leq j \leq n \cdot S_j \models \psi$$

$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{sometime-before}\ \phi\ \psi) \quad \text{iff} \quad \forall i \cdot \text{if } S_i \models \phi \text{ then } \exists j : 0 \leq j < i \cdot S_j \models \psi$$

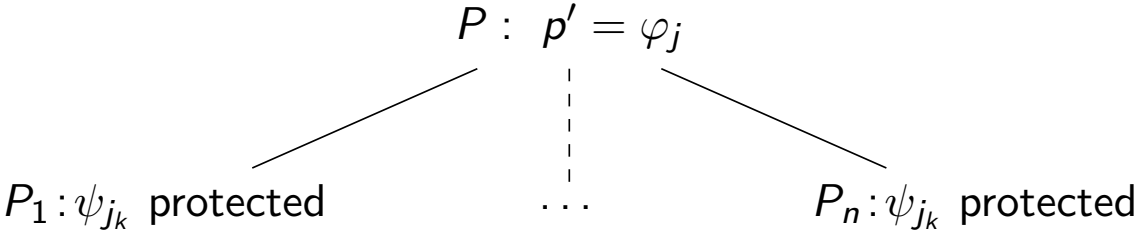
$$\langle (S_0,0),(S_1,t_1),...,(S_n,t_n)\rangle \models (\texttt{always-within}\ t\ \phi\ \psi) \quad \text{iff} \quad \forall i \cdot \text{if } S_i \models \phi \text{ then } \exists j : i \leq j \leq n \cdot S_j \models \psi \text{ and } t_j - t_i \leq t$$

Appendix — More details: "Split strategy"

First selection of the preference p_i in plan P:



Each following selection of the simplified preference p':



Appendix — Room for improvement of "Split strategy"

- Fringe contains partial plans
- "Split strategy" splits preference and branches the resulting partial plans into the fringe

$$P: \; p_i = \varphi_1 \vee \cdots \vee \varphi_n$$

$$P_1: \; p_i' = \varphi_1 \qquad \cdots \qquad P_n: \; p_i' = \varphi_n$$

- Currently $\textsc{Panda}_2$ is not able to differentiate between the partial plans $P_1, \ldots, P_n$

  $\rightarrow$ Random selection of a partial plan among $P_1, \ldots, P_n$

  $\rightarrow$ But: Preference was selected because of a specific disjunct φ_j thus the partial plan P_j with $1 \leq j \leq n$ should be preferred

Appendix — More details: "No-split strategy"

Branching of generated partial plans if preference p in partial plan P is selected:

$P: \; p = \varphi_1 \vee \cdots \vee \varphi_j$

$\texttt{first}(\varphi_j)$

$\texttt{first}(p_i)$

$P_1 : \psi_{j_k}$ protected $\cdots$ $P_n : \psi_{j_k}$ protected $P_{n+1} : f_i \notin \texttt{flaws}(P_{n+1})$ $P_{n+2} : p_i' = p_i \setminus \varphi_j$