

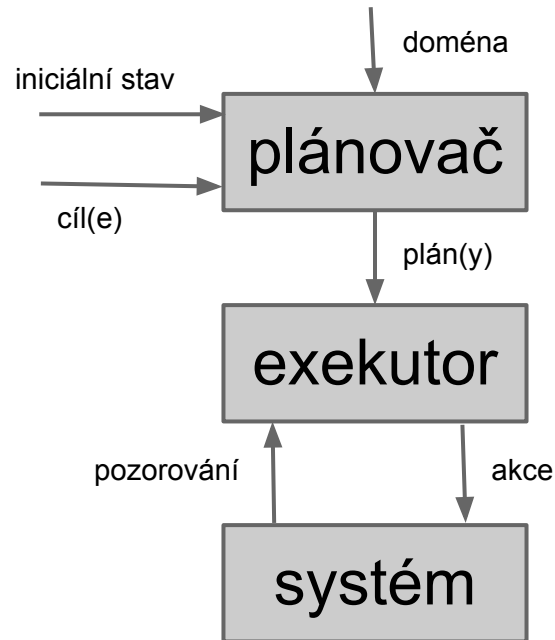
Plánování s nejistotou

Antonín Komenda
33PAH - 19.3.2012

Doménově nezávislé plánování

- plánovací problém
 - plánovací doména
 - instance plánovacího problému
- plánovač
 - vstup: plánovací problém
 - výstup: plán || *false*
- plán
 - řešení konkrétního plánovacího problému

Konceptuální model systému s plánováním



Typy prostředí

Z pohledu:

- pozorovatelnosti
 - plně pozorovatelný stav
 - částečně pozorovatelný stav
 - chyby pozorování se známou prpostí
 - chyby pozorování s neznámou prpostí (uniformní)
- výsledků akcí
 - deterministické akce
 - nedeterministické akce se známou prpostí možností
 - nedet. akce s neznámou prpostí možností

Plánování dle typu prostředí

	deterministické akce	nedet. se známou P	nedet. s neznámou P
plně pozorovatelný	klasické plánování	MDP, RR (RP)	RR (RP)
část. pozorovatelný	POMDP, TCP	POMDP, TCP+RR	TCP+RR

(PO)MDP = (Partially Observable) Markov Decision Processes (Markovské rozhodovací procesy)

RR = Re-planning & Plan Repair

(Replánování a opravy plánů)

RP = Reactive Planning (Reaktivní plánování)

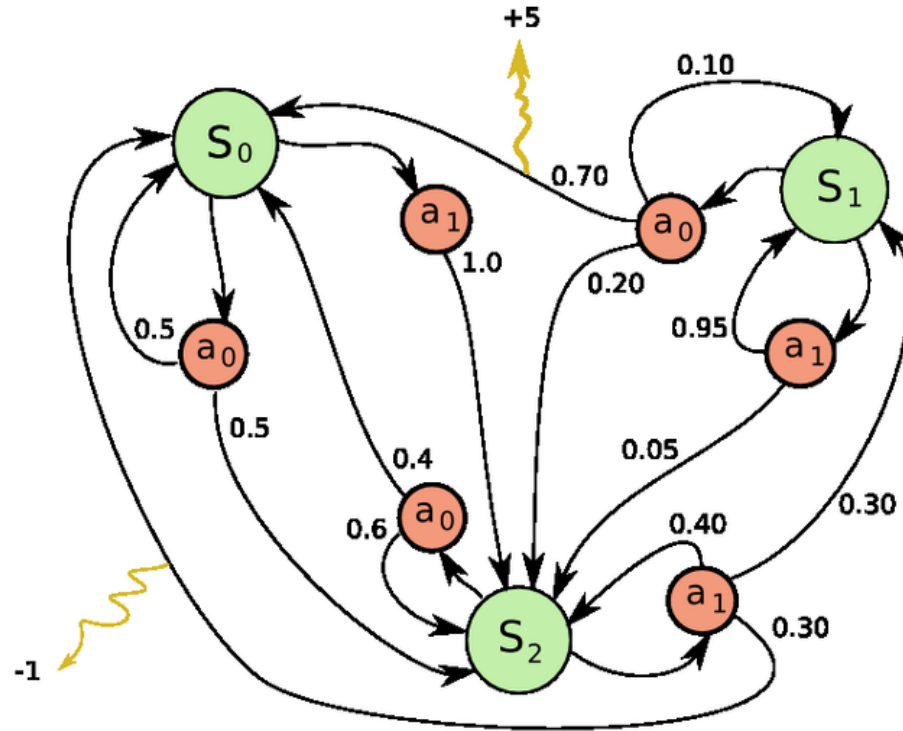
TCP = Transformation to Classical Planning

(Transformace probl. do klasického plánování)

(PO)MDP

- plánovací problém
 - plánovací doména
 - instance plánovacího problému
- plánovač
 - vstup: plánovací problém
 - výstup: plán || *false*
- policy (plán)
 - řešení konkrétního plánovacího problému
- exekuce policy

(PO)MDP

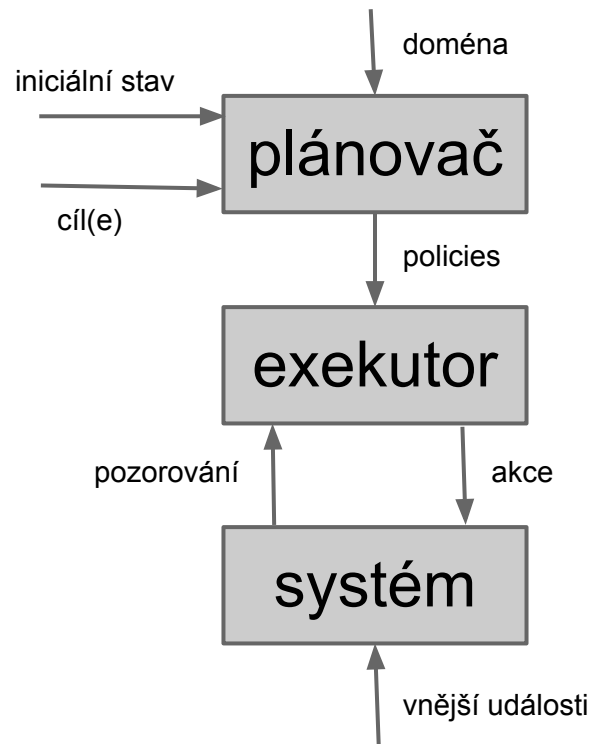


Exekuce Policy

- policy:
 - $p1 = \{(S0,a1),(S1,a0),(S2,a1)\}$
 - $p2 = \{(S0,a0),(S1,a0),(S2,a1)\}$

- loop
 - sense the current state s
 - $a \leftarrow p(s)$
 - execute a

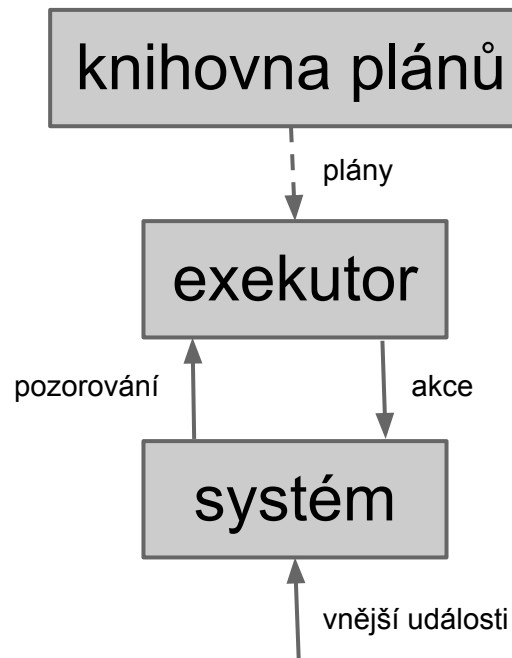
Konceptuální model systému s (PO)MDP



Reaktivní plánování

- předpřipravená knihovna plánů
 - známé *preconditions* a *effects* pro každý plán
- plánovač a exekutor
 - pozoruje aktuální stav
 - použije vhodný plán z knihovny
- postupná stavba plánu od makro-akcí ke konkrétním akcím
 - backtracking na úrovni rozkladu akcí

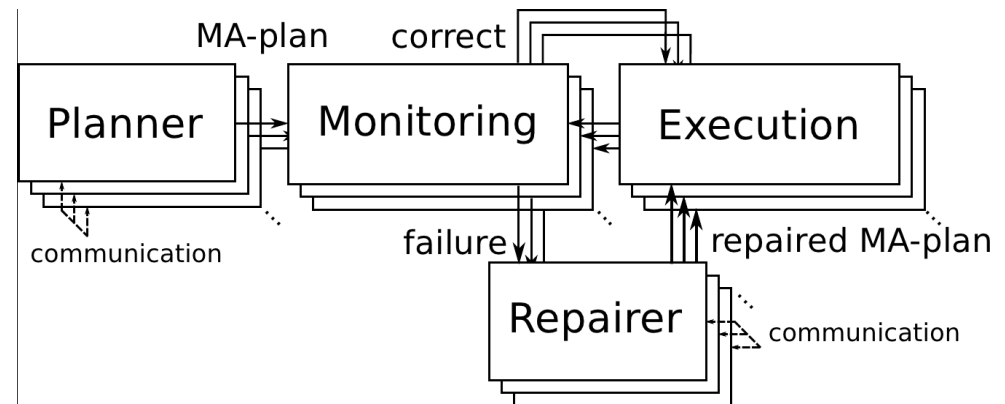
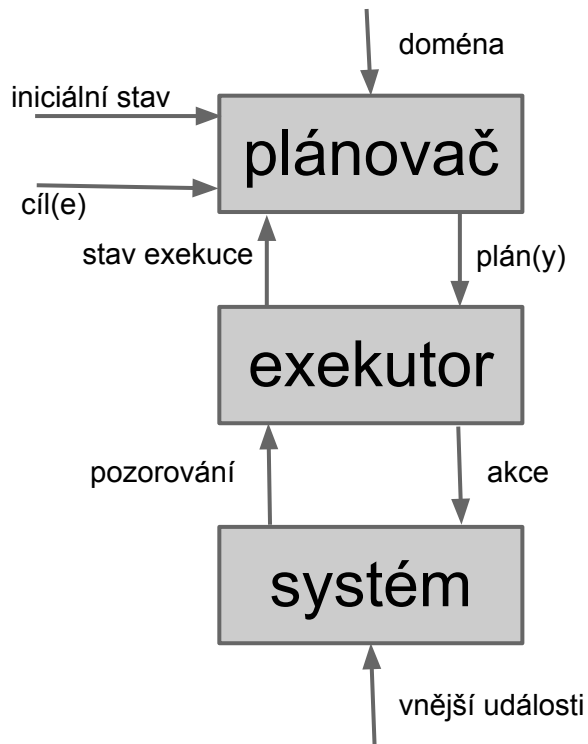
Konceptuální model systému s reaktivním plánováním



Replánování a opravy plánů

- plánovací problém
 - plánovací doména
 - instance plánovacího problému
 - plánovač
 - vstup: plánovací problém
 - výstup: plán || *false*
 - plán
 - řešení konkrétního plánovacího problému
 - exekuce plánu
 - monitoring exekuce plánu
- případné replánování nebo oprava plánu

Konceptuální model systému s dynamickým plánováním



Replánování

- spuštění plánovače znovu z aktuálního stavu
- jednoduchá technika
- často dostačující rychlostí
- optimální v délce výsledného plánu
 - za předpokladu optimálního plánovače

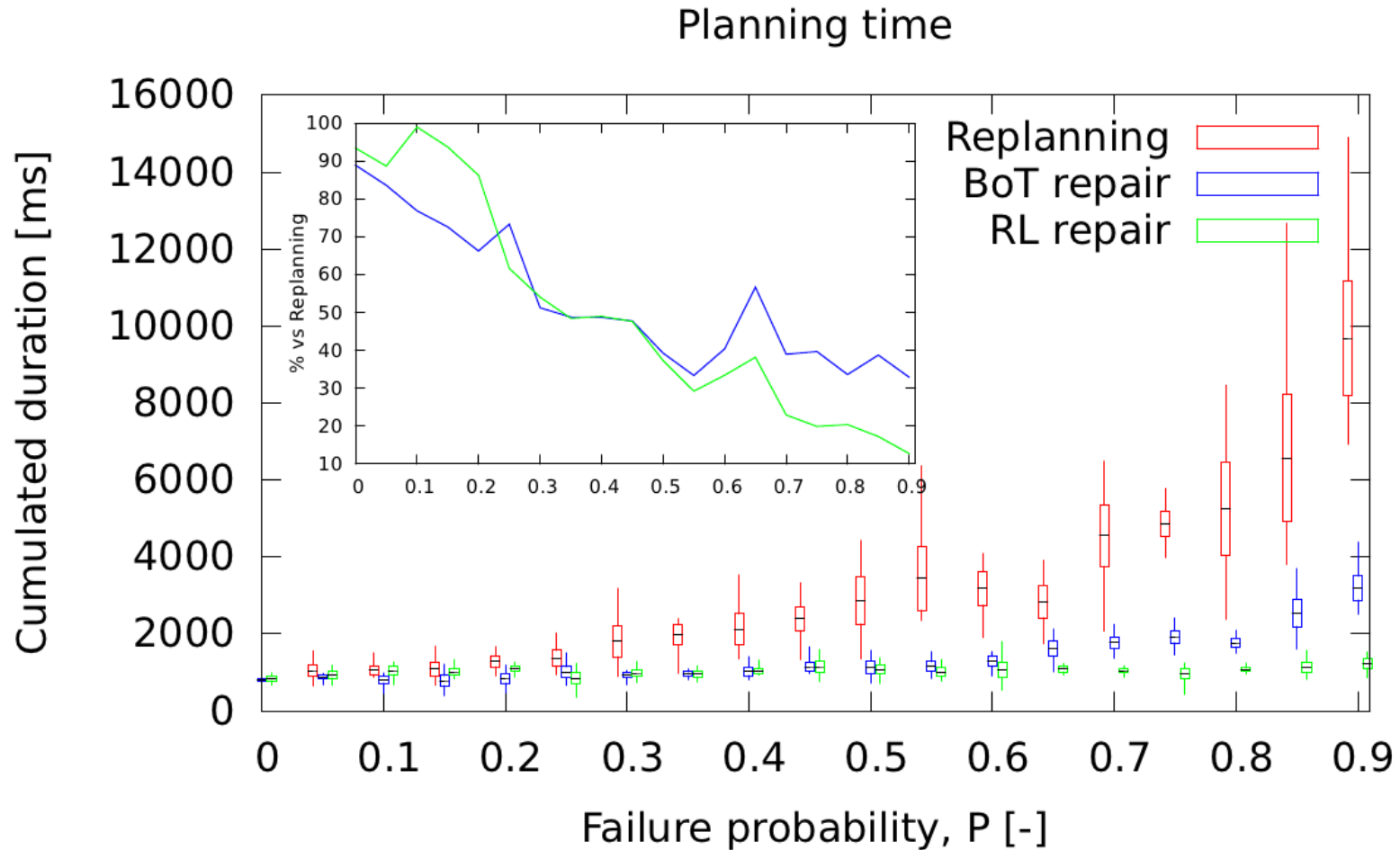
Back-on-track technika opravy plánu

- použití plánovače pro návrat k originálnímu plánu
- vyšší požadavky na možnosti plánovače
- může mít vyšší rychlost oproti replánování

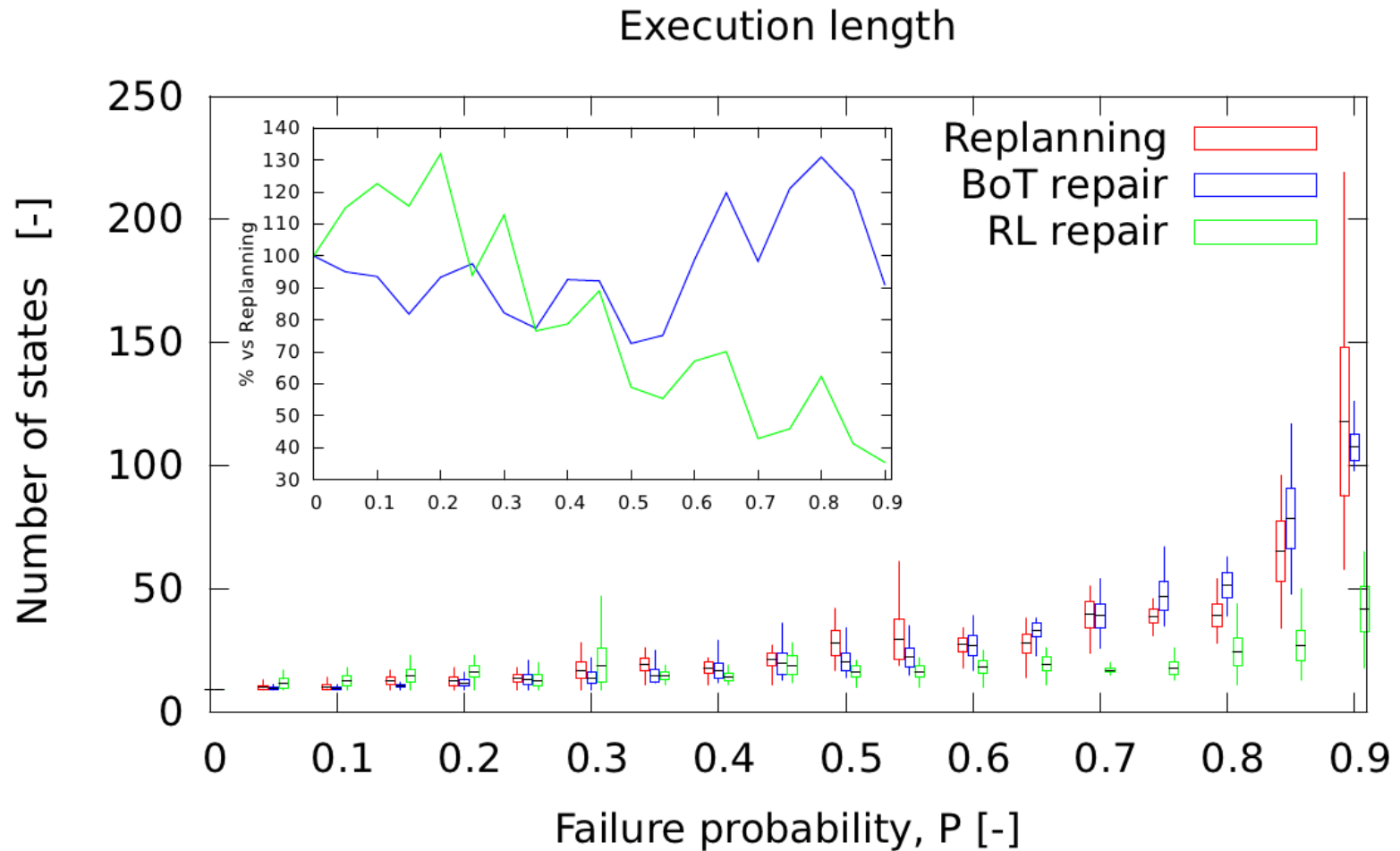
Lazy-repair technika opravy plánu

- ignoruj chyby v exekuci dokud je to možné, potom naplánuj znovu
- opakování: proces opakuj dokud nejsou cíle splněné
- může mít vyšší rychlost jak oproti replánování, tak oproti back-on-track

Opravy plánů vs. replanning

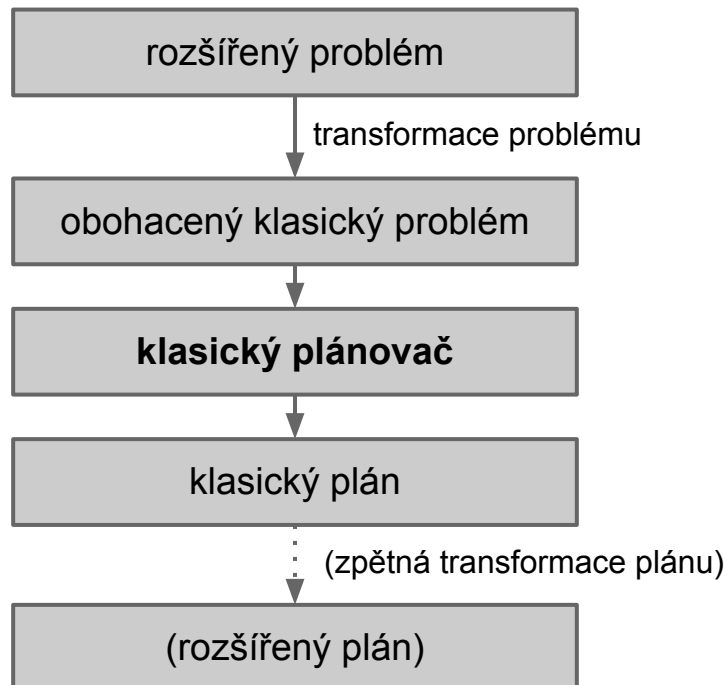


Opravy plánů vs. replanning



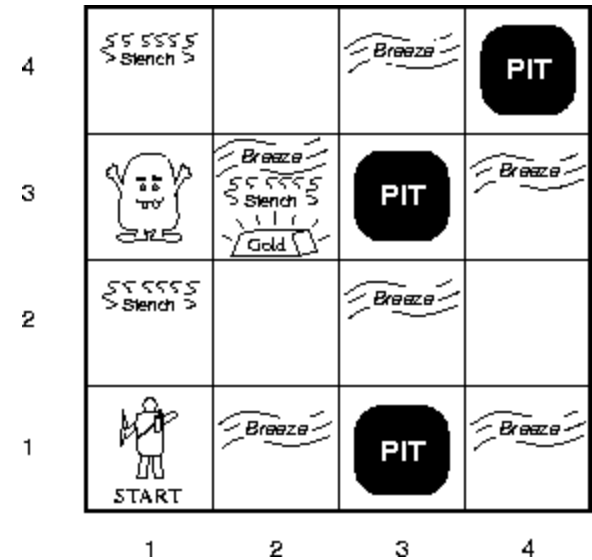
Transformace plánovacích problémů

- obecně používaný princip
- transformace rozšířených vlastností problému do speciálních akcí a proměnných stavu



Tranformace pro částečnou pozorovatelnost

- stav rozšíříme o:
 - `know-at(loc, what)`
- akce rozšíříme o:
 - `sense-at(loc)`



Wumpus problem

Opravy plánů v hierarchickém plánování

- princip *derivační analogie* (*derivational analogy*)
- přepoužijme částečné výsledky předešlých plánovacích procesů
- rozbalený hierarchický strom možných akcí
- nové rozvíjení stromu na místě nejbližší chybě v plánu