Abstraction refinement and plan revision for control synthesis under high level specifications

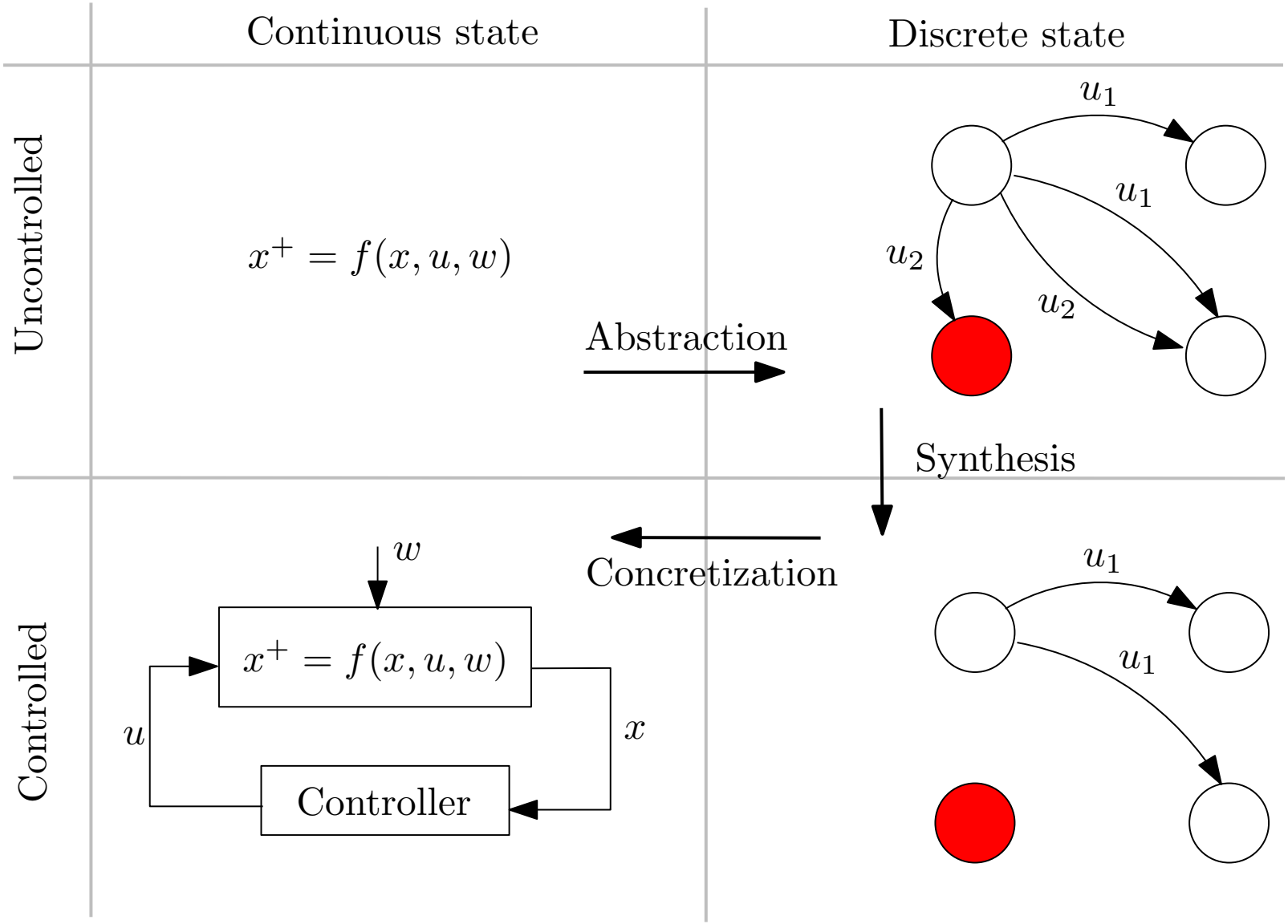
Pierre-Jean Meyer Dimos V. Dimarogonas

KTH, Royal Institute of Technology

July 12th 2017

Outline

Abstraction-based synthesis



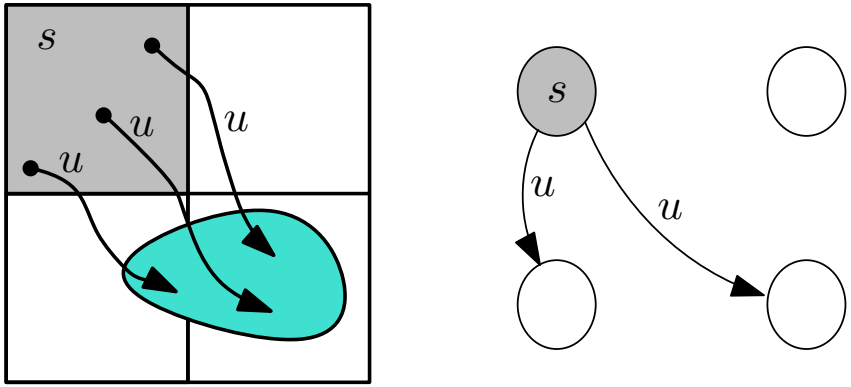
Abstraction procedure

$x^+ = f(x, u, w)$

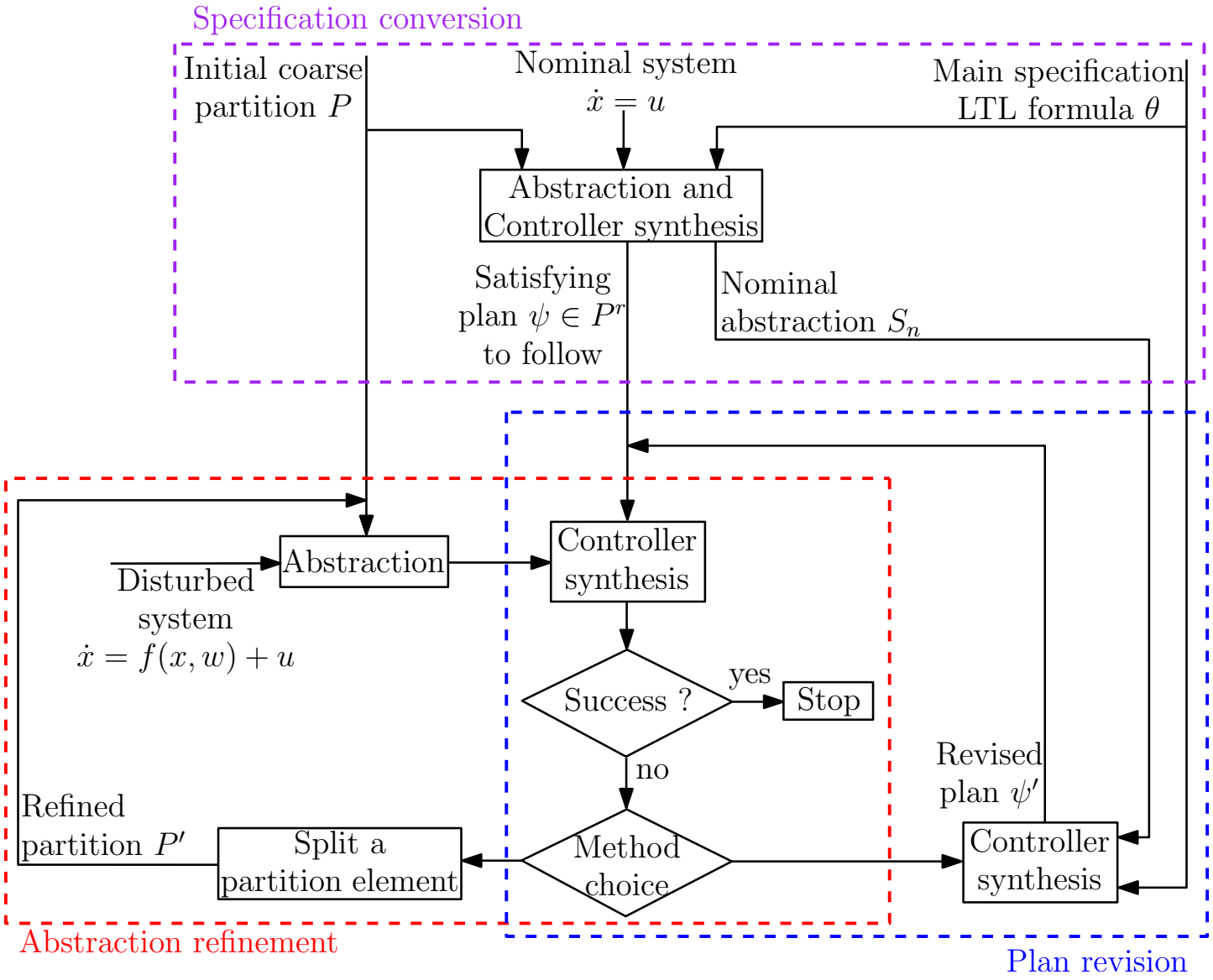
- Define the **reachable set** for any disturbance $w \in W$:

$$RS(X, U) = \{f(x, u, w) \mid x \in X,\ u \in U,\ w \in W\}$$

- Partition of the state space
- For each partition cell s and control u: compute the **reachable set** $RS(s, \{u\})$
- Obtain a **non-deterministic** transition system: each pair (s, u) may have several successors



Global framework

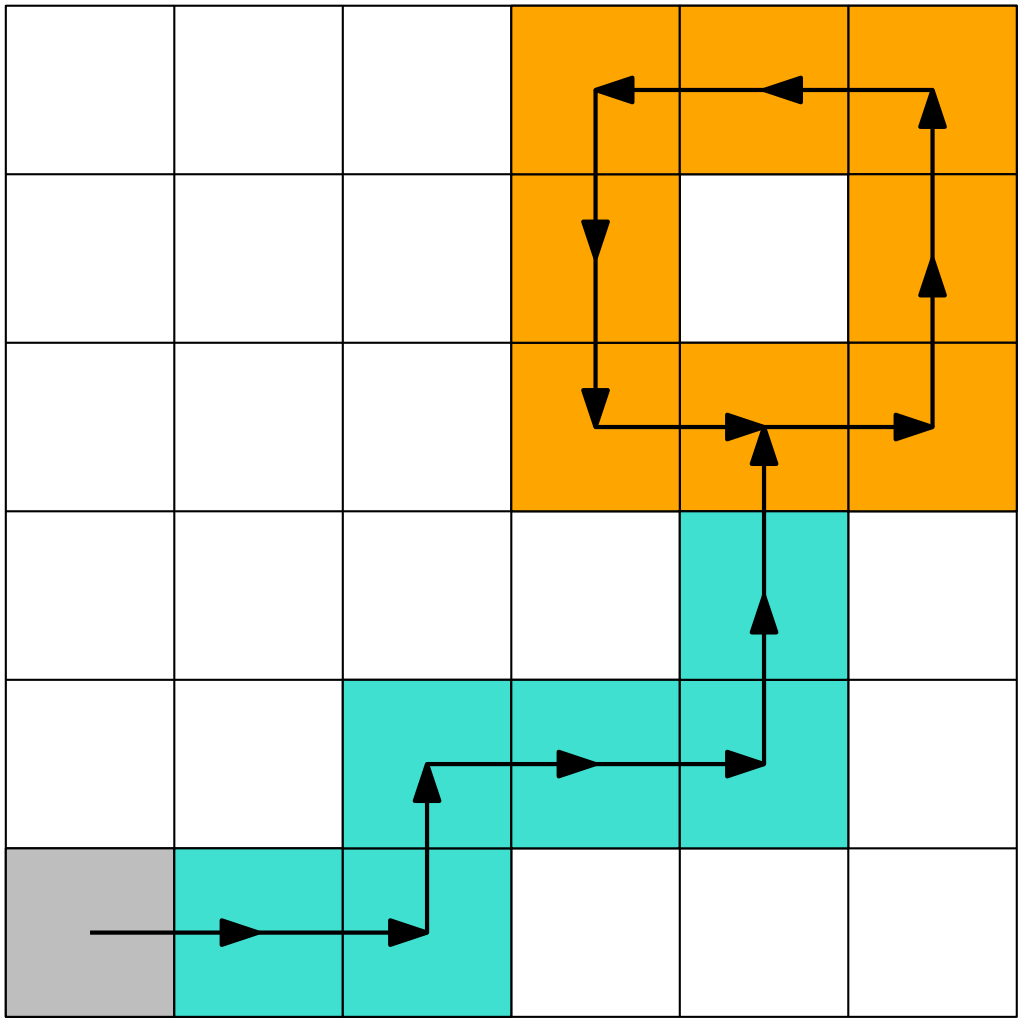


Specifications

General Linear Temporal Logic (**LTL**) formula:

Satisfying plans are **lasso-shaped** sequences in the state partition

- **prefix**: finite path from the initial cell
- **suffix**: finite path looping on itself

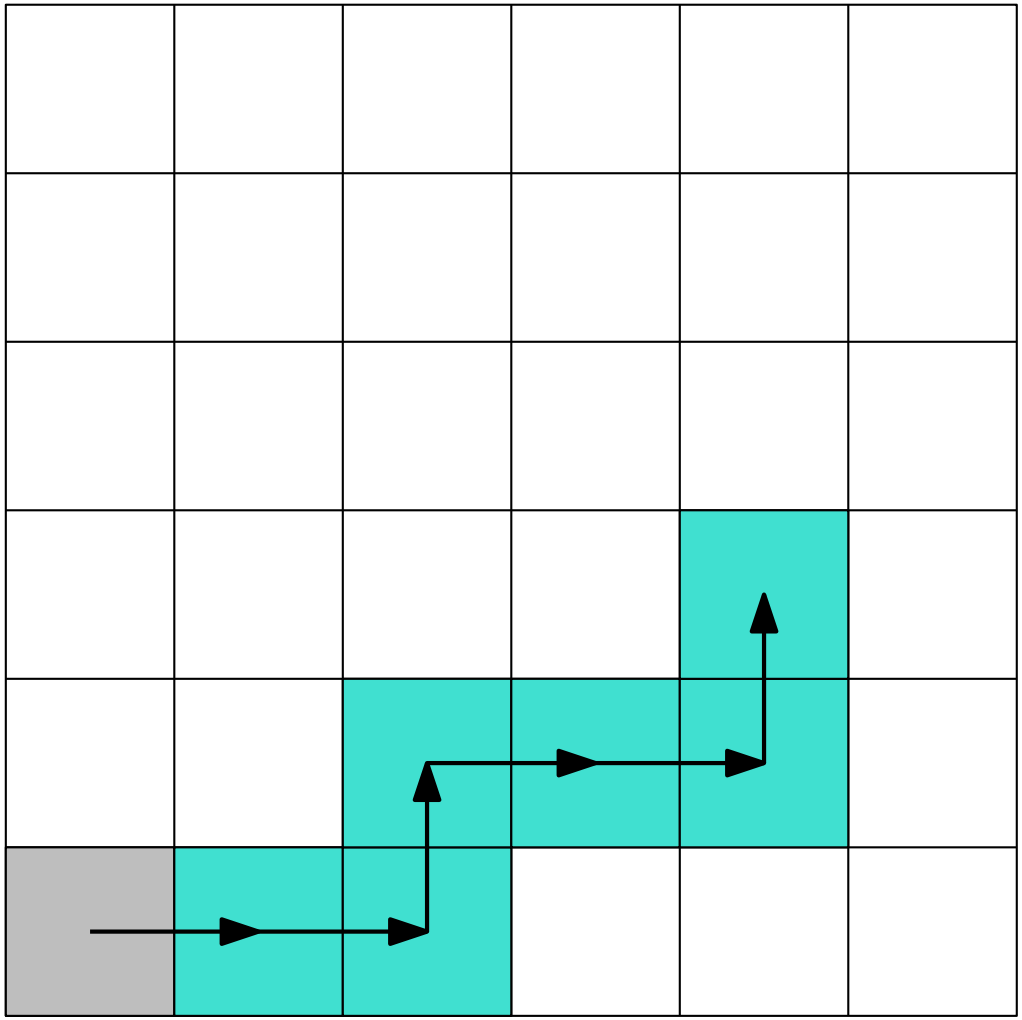


Specifications

Syntactically co-safe LTL formula:

Satisfiable in finite time

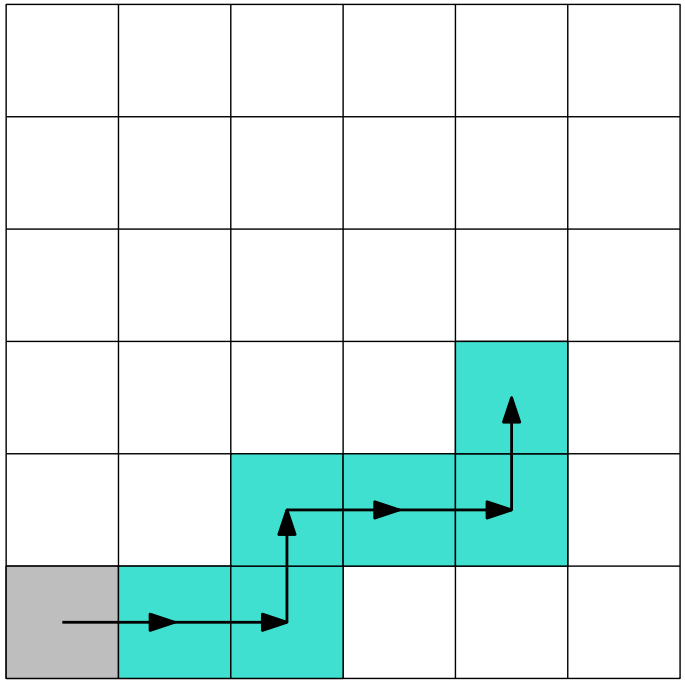
- **prefix**: finite path from the initial cell



Valid sets

Definition (Valid set)

Elements of the refined partition X_a that can be controlled to follow the desired sequence of cells

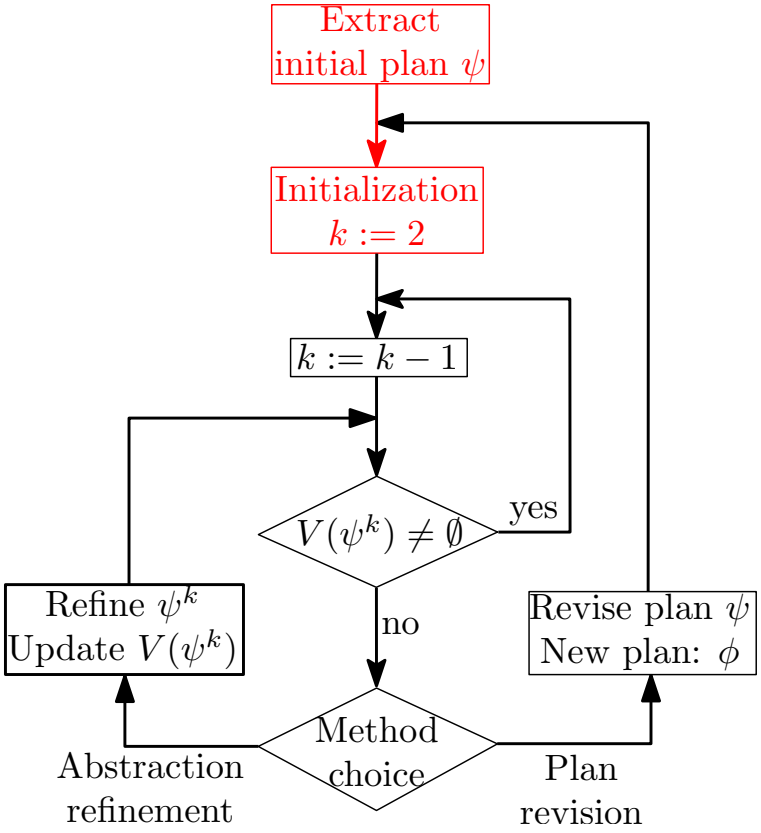


Finite plan: $\psi = \psi^0 \psi^1 \ldots \psi^r$

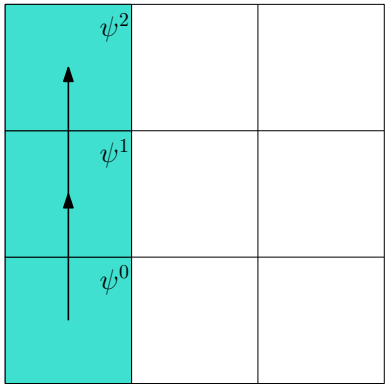
- Final cell: $V(\psi^r) = \{\psi^r\}$
- for all $k \in \{0, \ldots, r-1\}$:

$$V(\psi^k) = \{s \in X_a | s \subseteq \psi^k, \exists u \in U_a, RS(s, \{u\}) \subseteq V(\psi^{k+1})\}$$

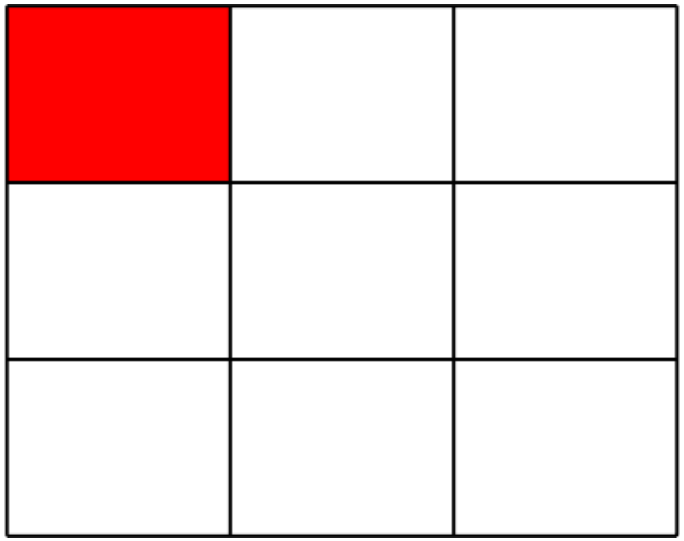
Algorithm



Specification:
reach top-left
from bottom left

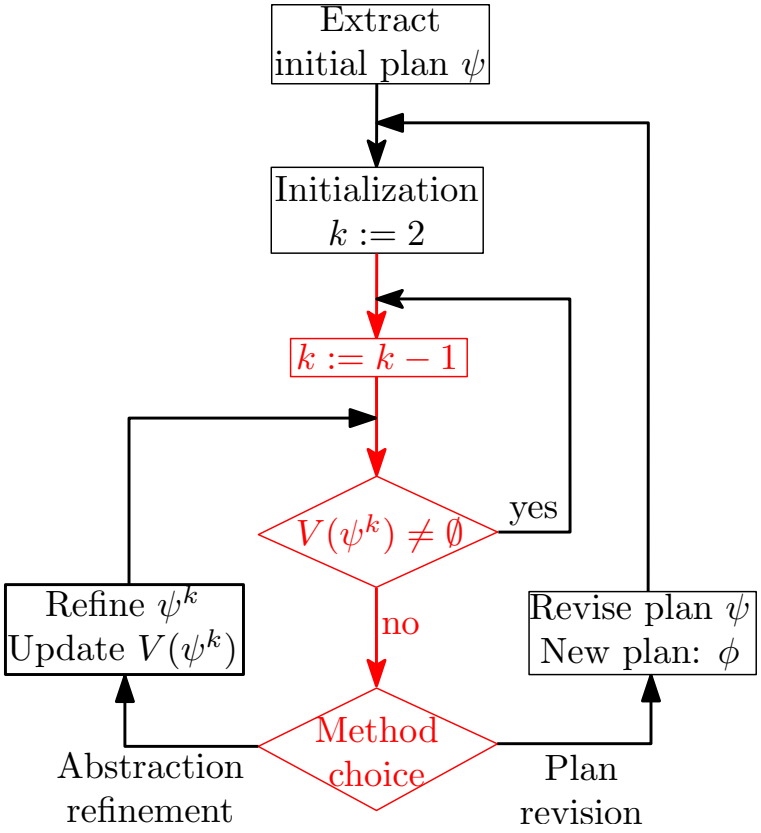


Initial plan:
$\psi^0 \psi^1 \psi^2$



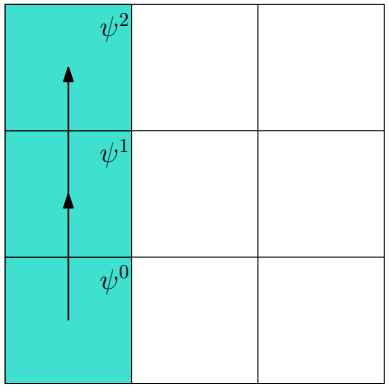
Last step of the plan: ψ^2
Initial valid set: $V(\psi^2) = \{\psi^2\}$

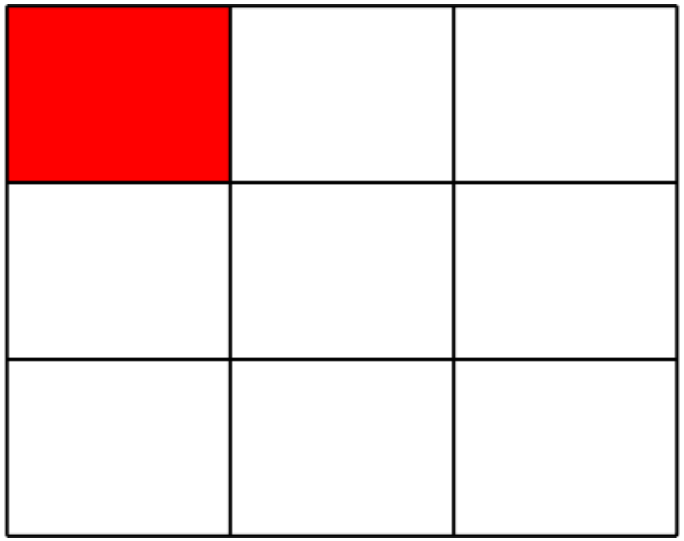
Algorithm



Specification:
reach top-left
from bottom left

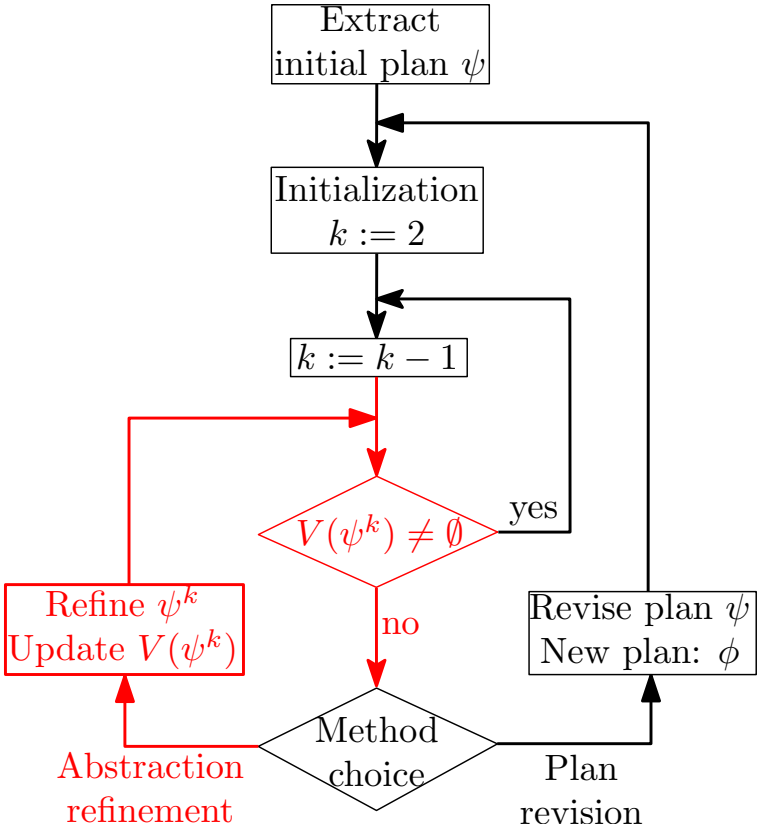
Initial plan:
$\psi^0 \psi^1 \psi^2$





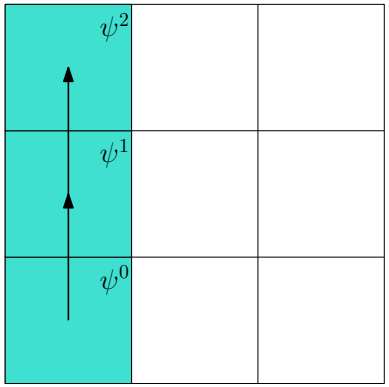
Transition $\psi^1 \rightarrow \psi^2$
Empty valid set: $V(\psi^1) = \emptyset$

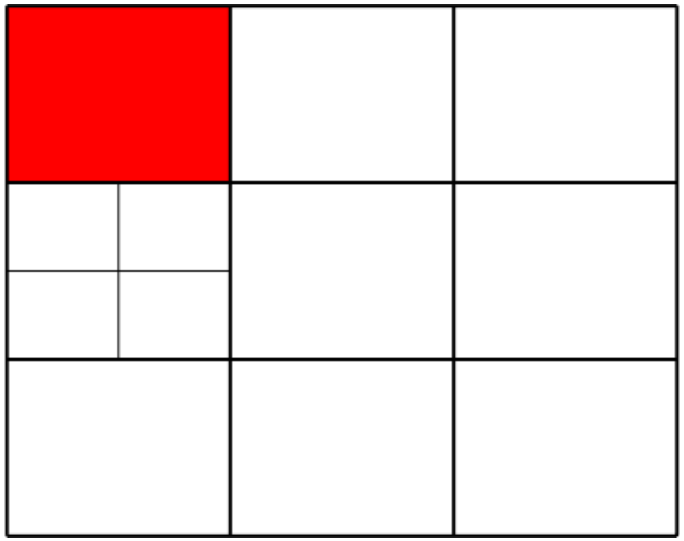
Algorithm



Specification:
reach top-left
from bottom left

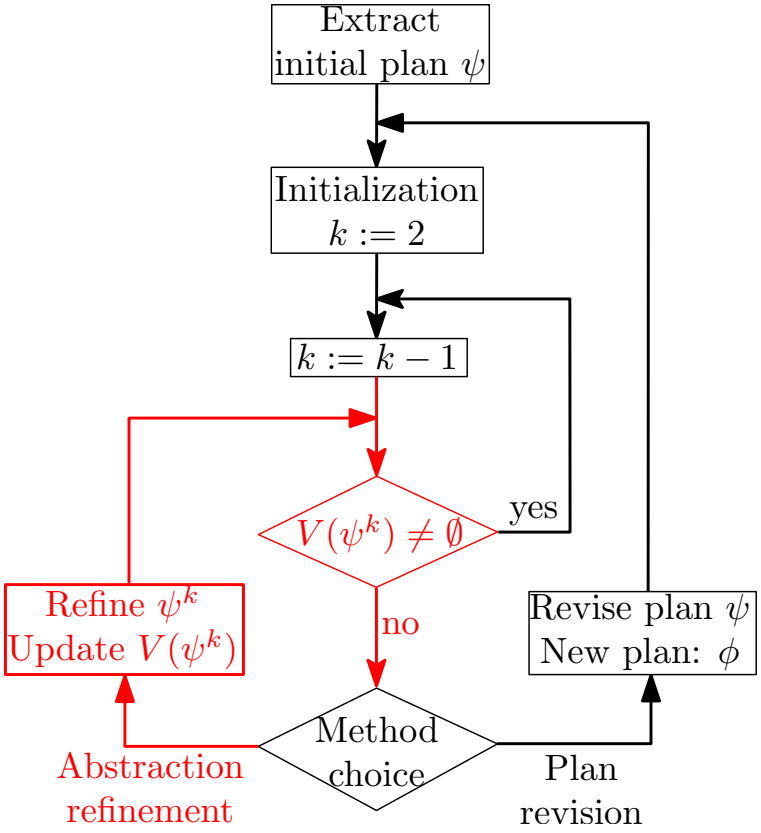
Initial plan:
$\psi^0 \psi^1 \psi^2$





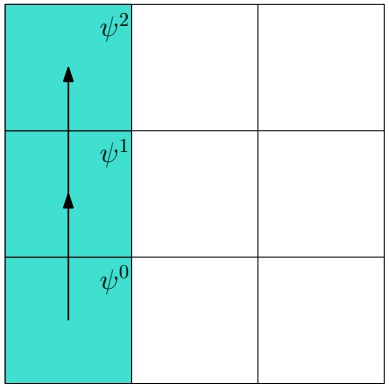
Transition $\psi^1 \rightarrow \psi^2$
Refine ψ^1
Valid set still empty: $V(\psi^1) = \emptyset$

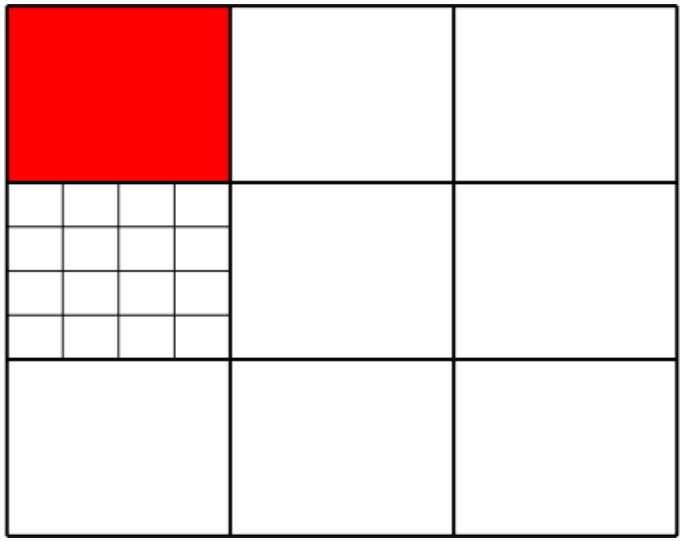
Algorithm



Specification:
reach top-left
from bottom left

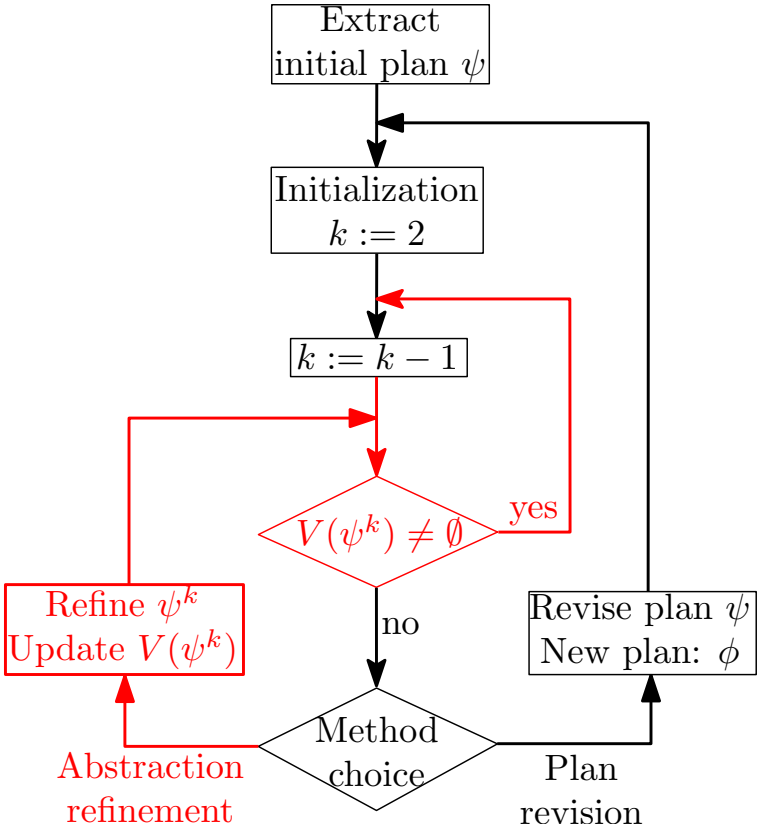
Initial plan:
$\psi^0 \psi^1 \psi^2$





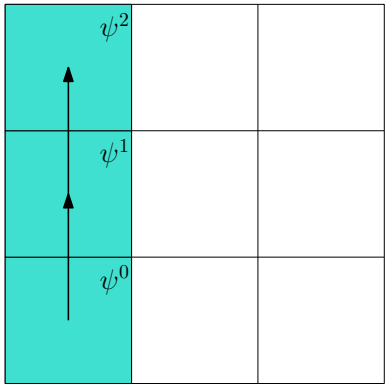
Transition $\psi^1 \rightarrow \psi^2$
Refine ψ^1
Valid set still empty: $V(\psi^1) = \emptyset$

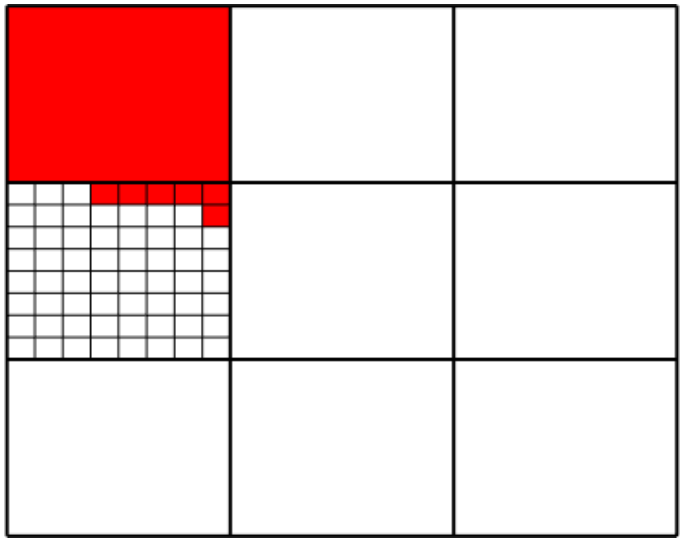
Algorithm



Specification:
reach top-left
from bottom left

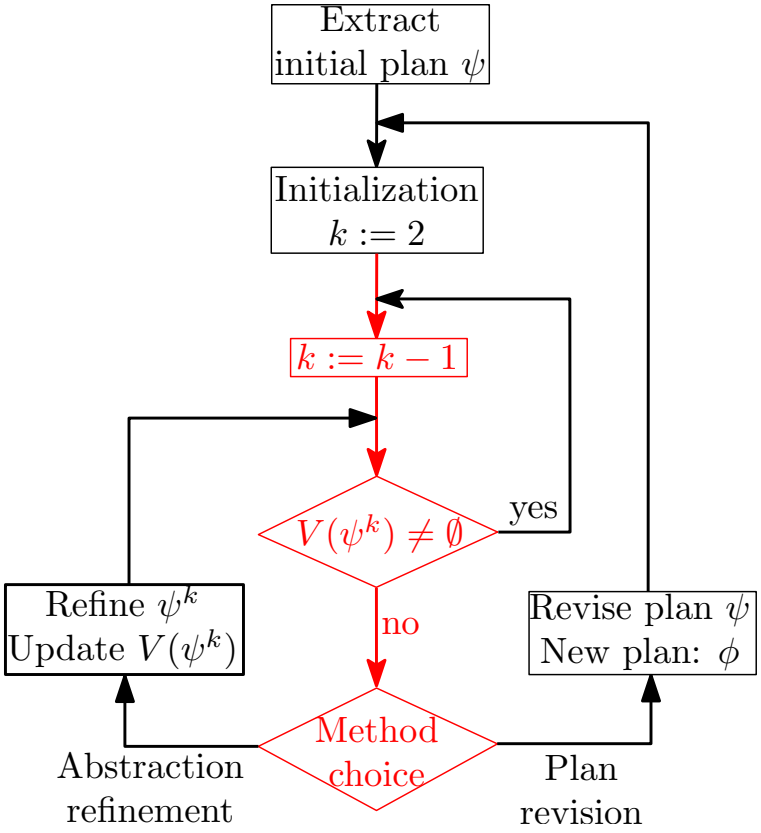
Initial plan:
$\psi^0 \psi^1 \psi^2$





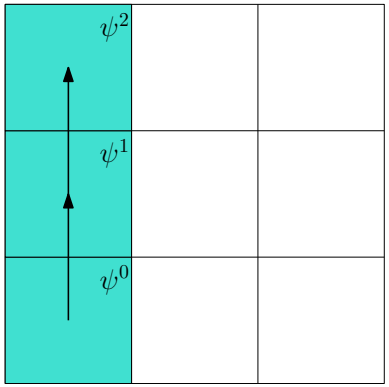
Transition $\psi^1 \rightarrow \psi^2$
Refine ψ^1
New valid set for ψ^1: $V(\psi^1) \neq \emptyset$

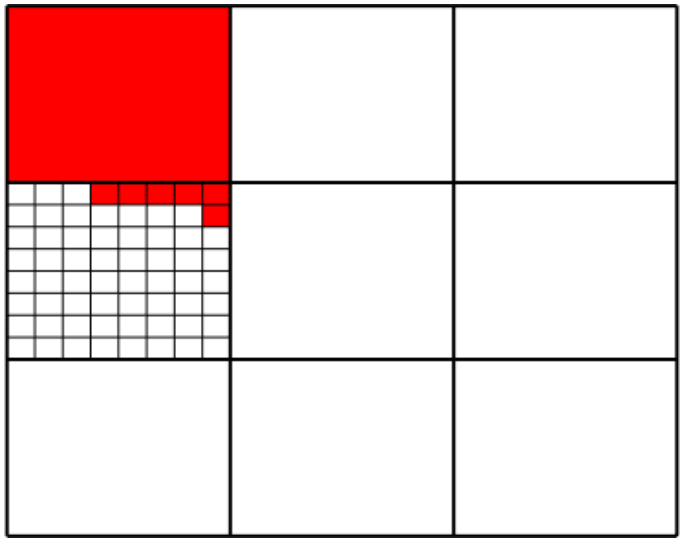
Algorithm



Specification:
reach top-left
from bottom left

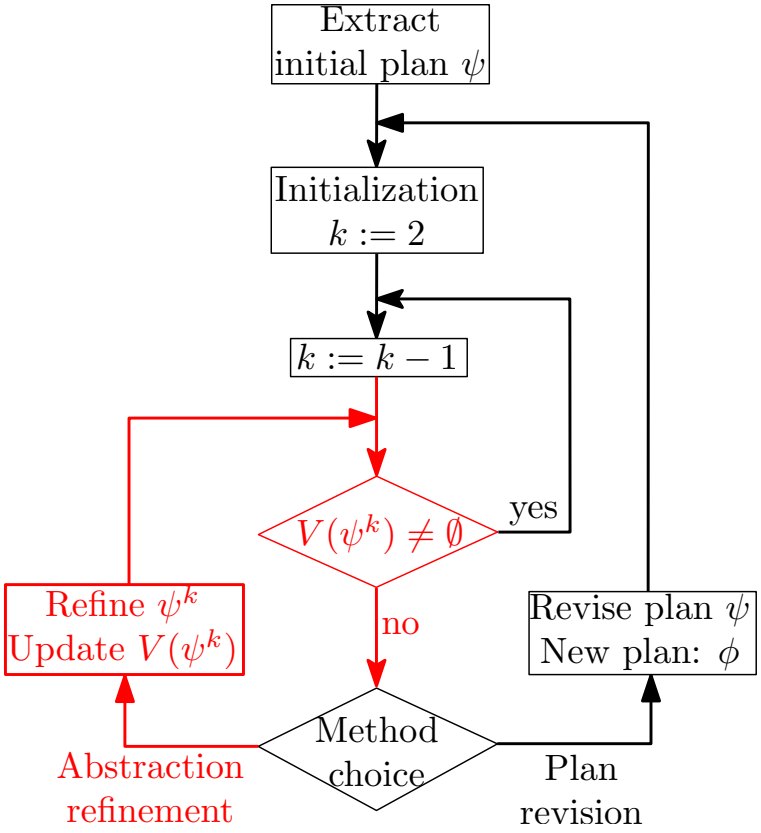
Initial plan:
$\psi^0 \psi^1 \psi^2$





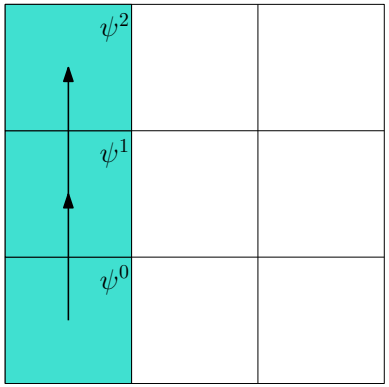
Transition $\psi^0 \rightarrow \psi^1$
Empty valid set: $V(\psi^0) = \emptyset$

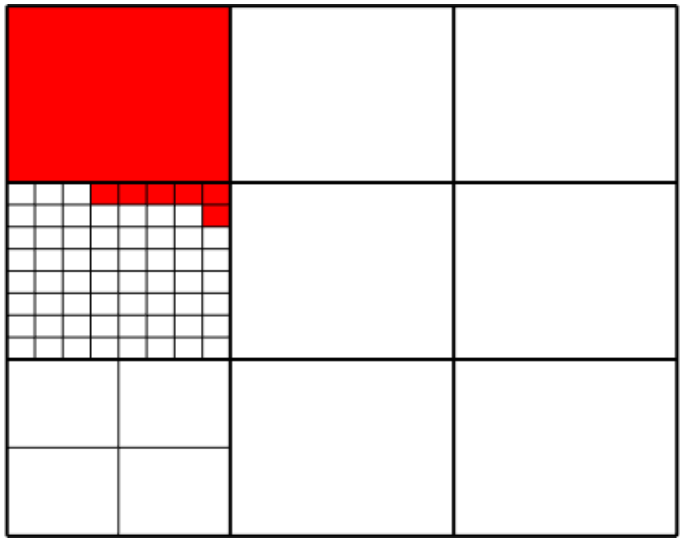
Algorithm



Specification:
reach top-left
from bottom left

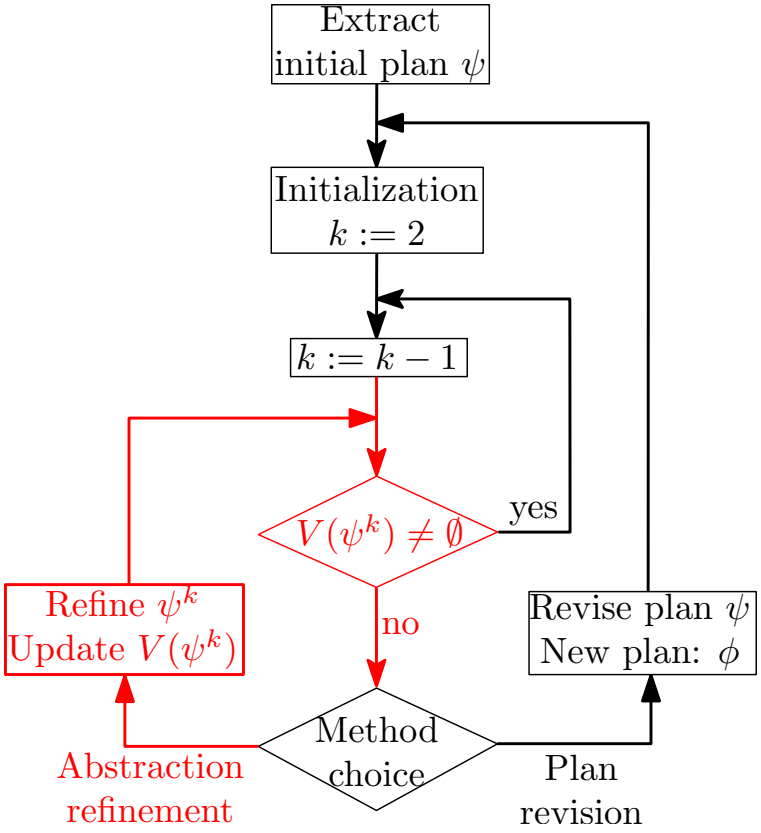
Initial plan:
$\psi^0 \psi^1 \psi^2$





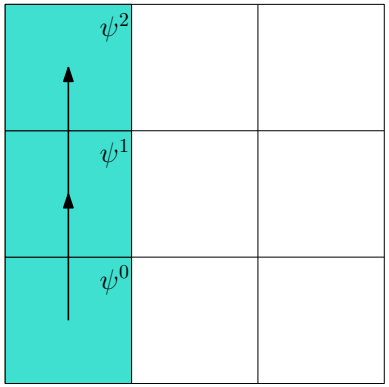
Transition $\psi^0 \to \psi^1$
Refine ψ^0
Valid set still empty: $V(\psi^0) = \emptyset$

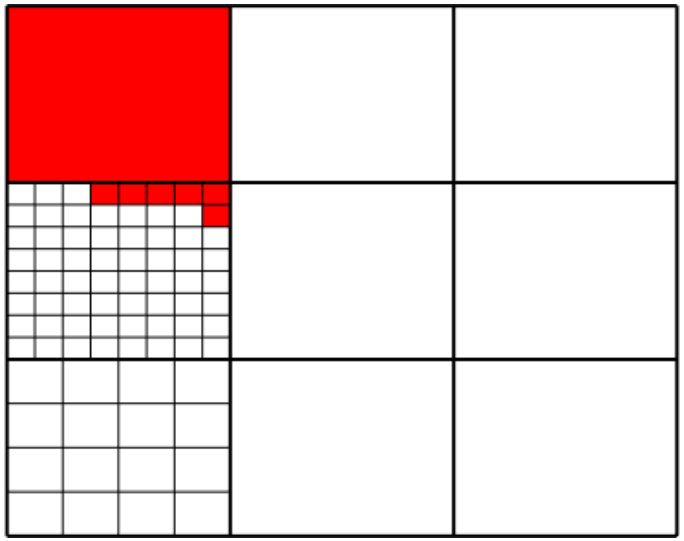
Algorithm



Specification:
reach top-left
from bottom left

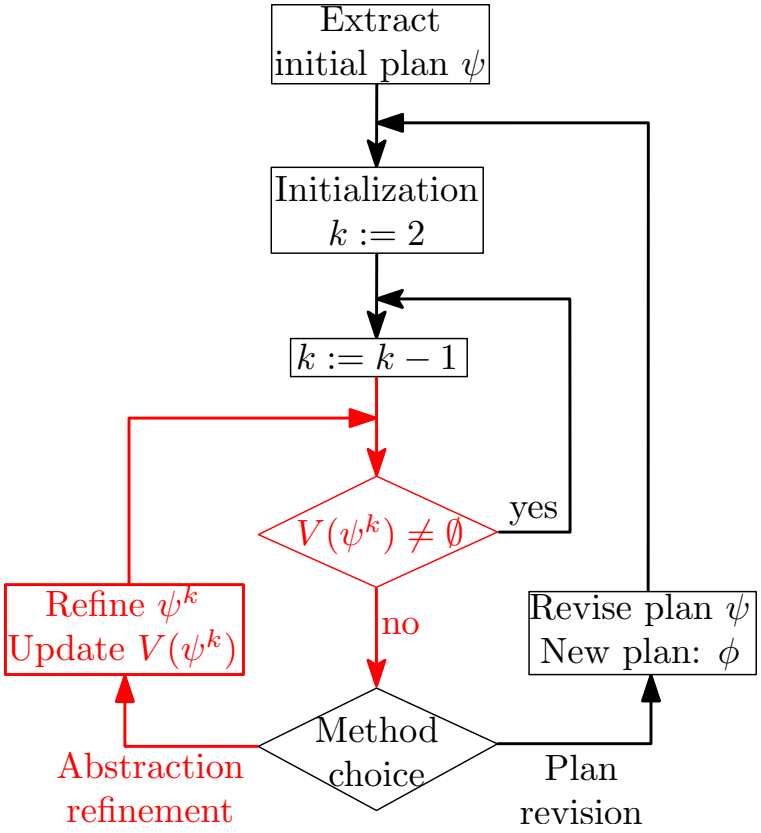
Initial plan:
$\psi^0 \psi^1 \psi^2$





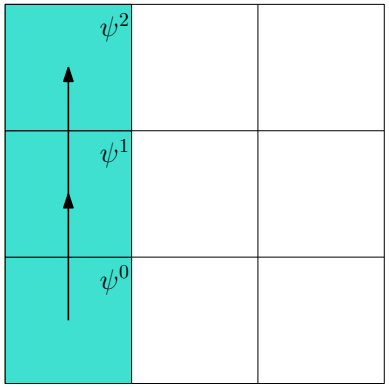
Transition $\psi^0 \to \psi^1$
Refine ψ^0
Valid set still empty: $V(\psi^0) = \emptyset$

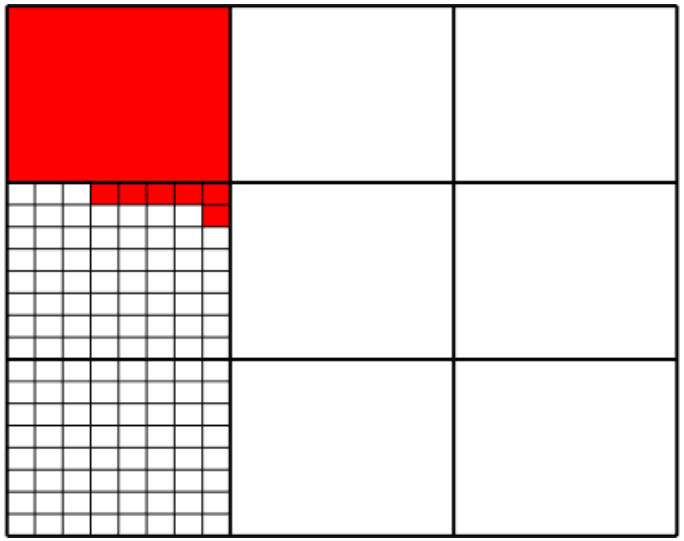
Algorithm



Specification:
reach top-left
from bottom left

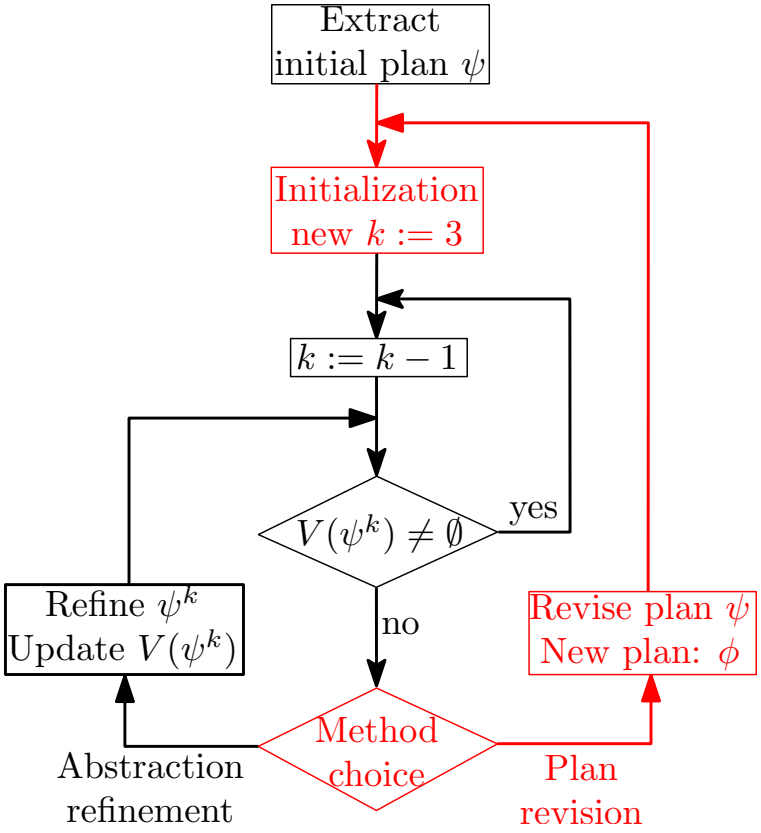
Initial plan:
$\psi^0 \psi^1 \psi^2$





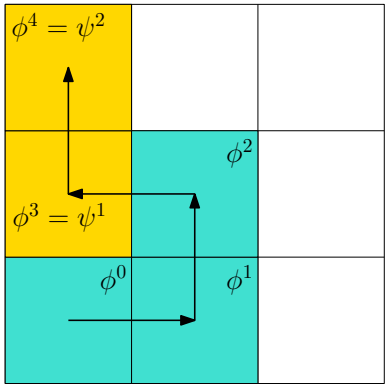
Transition $\psi^0 \to \psi^1$
Refine ψ^0
Valid set still empty: $V(\psi^0) = \emptyset$

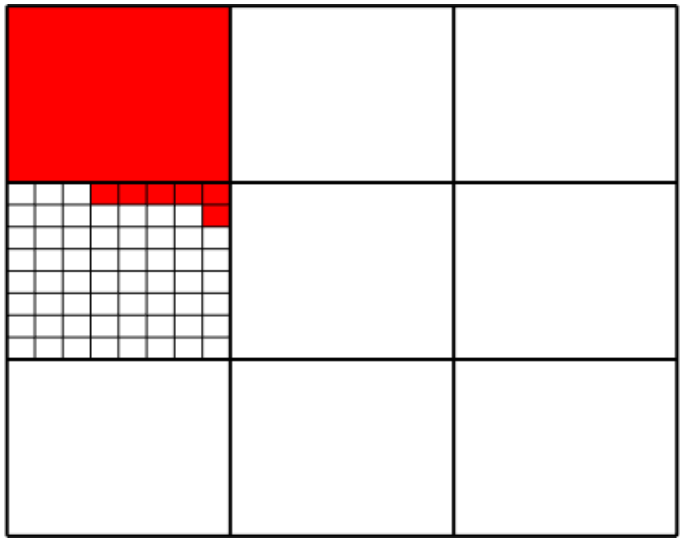
Algorithm



Specification: reach top-left from bottom left

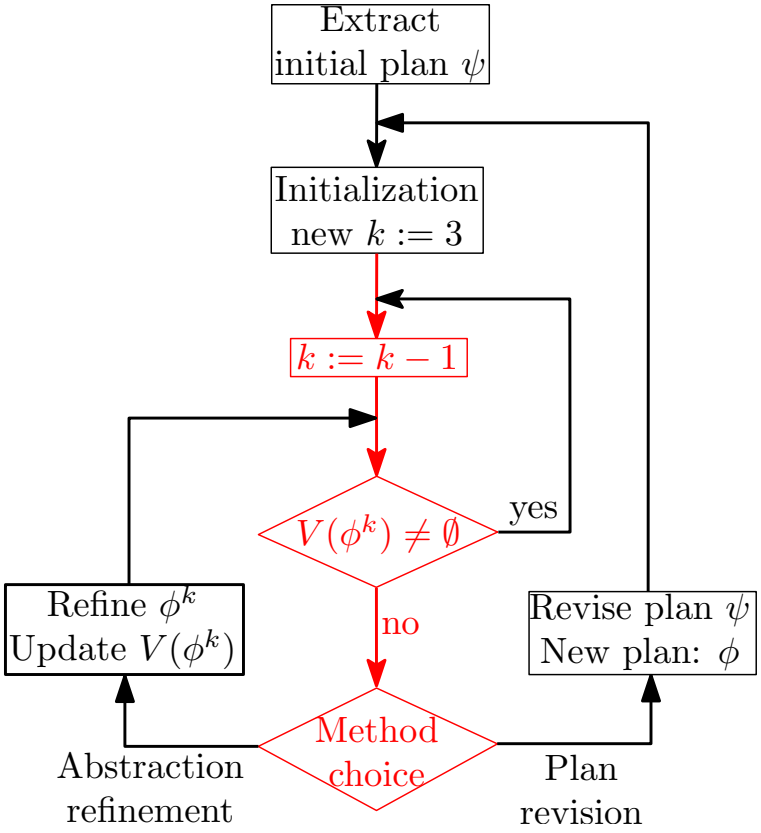
Revised plan: $\phi^0\phi^1\phi^2\phi^3\phi^4$





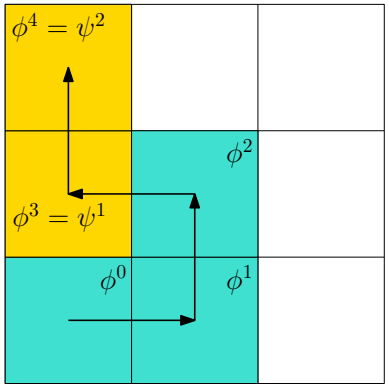
Plan revision: keep partial progress: $\phi^3\phi^4 = \psi^1\psi^2$

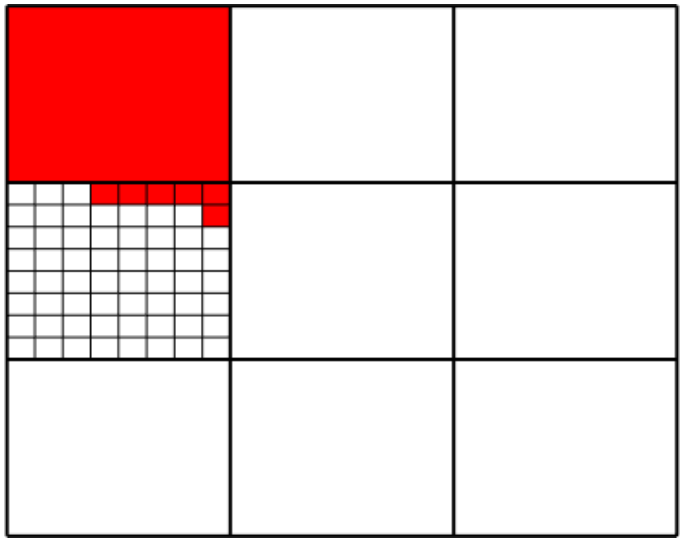
Algorithm



Specification:
reach top-left
from bottom left

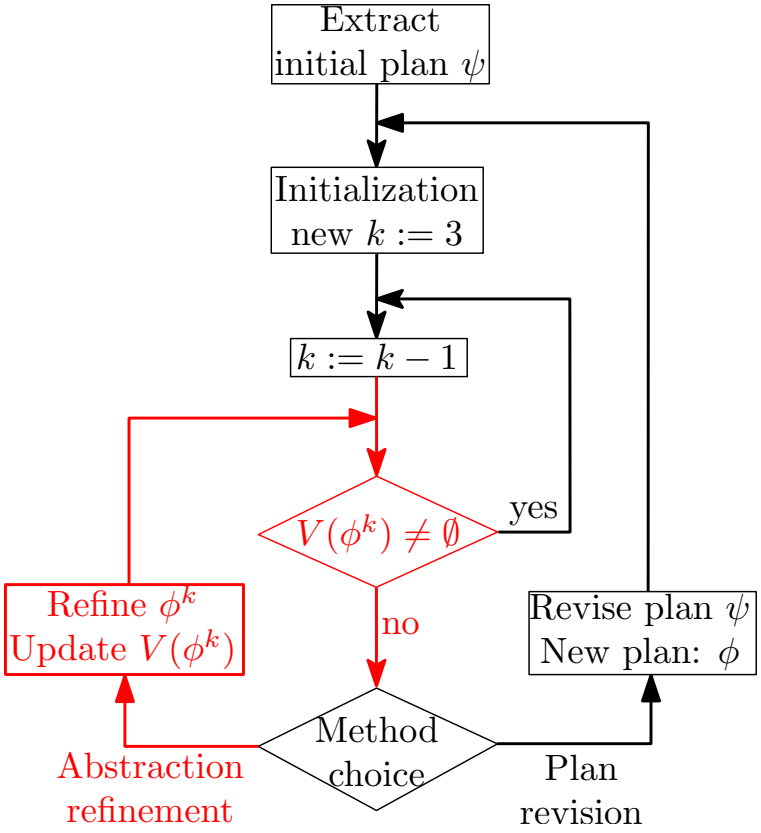
Revised plan:
$\phi^0\phi^1\phi^2\phi^3\phi^4$





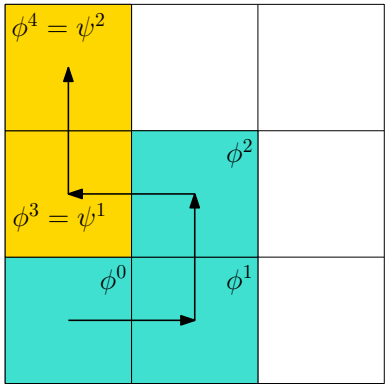
Transition $\phi^2 \to \phi^3$
Empty valid set: $V(\phi^2) = \emptyset$

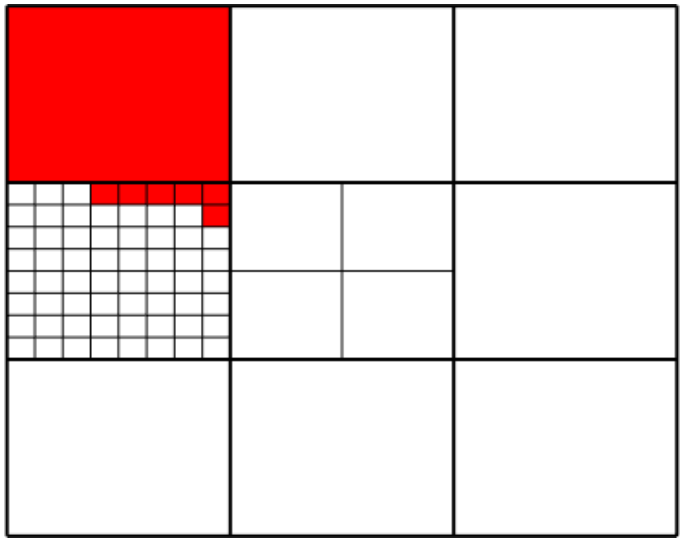
Algorithm



Specification:
reach top-left
from bottom left

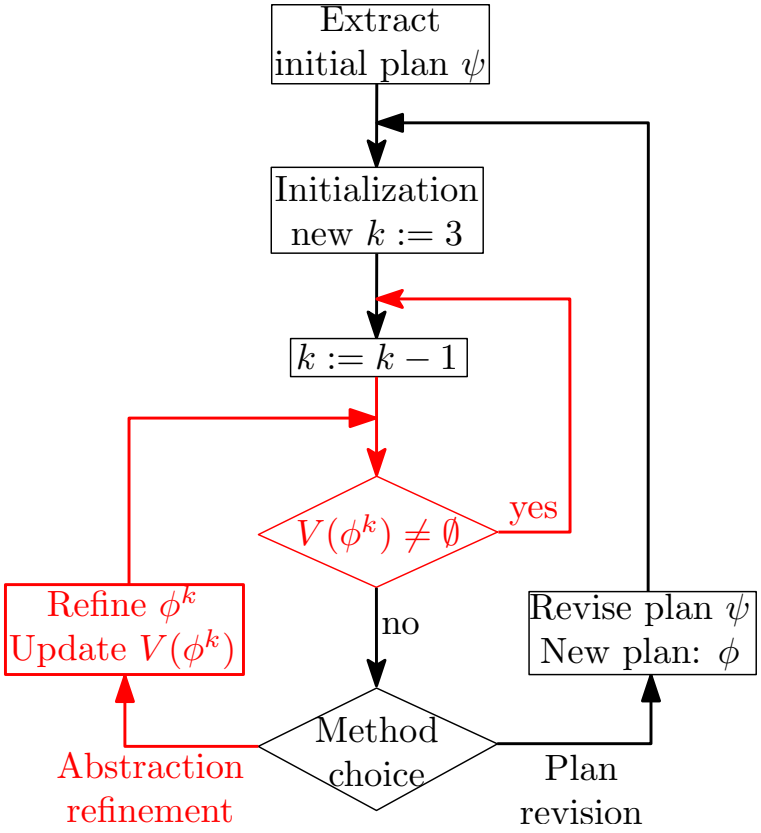
Revised plan:
$\phi^0\phi^1\phi^2\phi^3\phi^4$





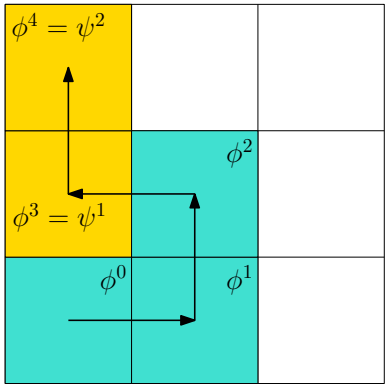
Transition $\phi^2 \to \phi^3$
Refine ϕ^2
Valid set still empty: $V(\phi^2) = \emptyset$

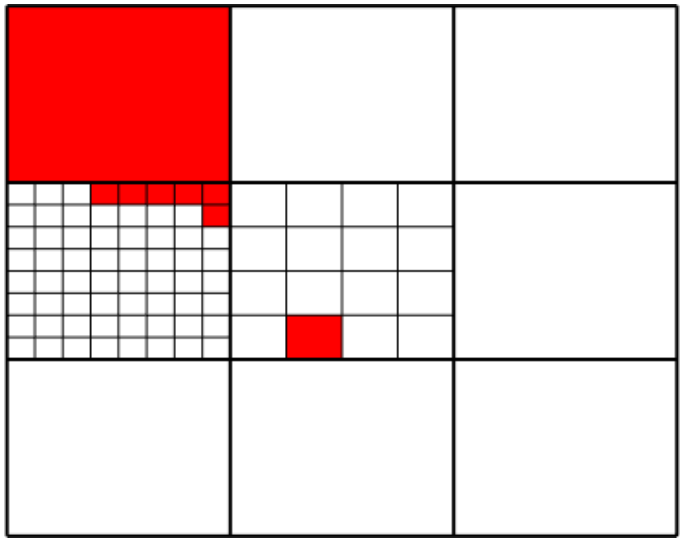
Algorithm



Specification:
reach top-left
from bottom left

Revised plan:
$\phi^0 \phi^1 \phi^2 \phi^3 \phi^4$



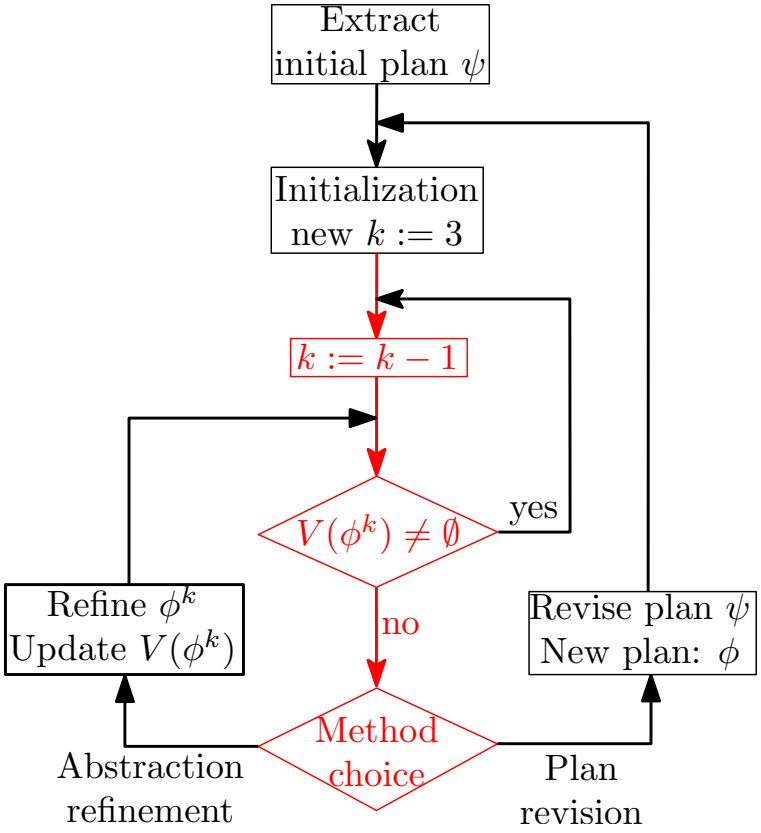


Transition $\phi^2 \to \phi^3$

Refine ϕ^2

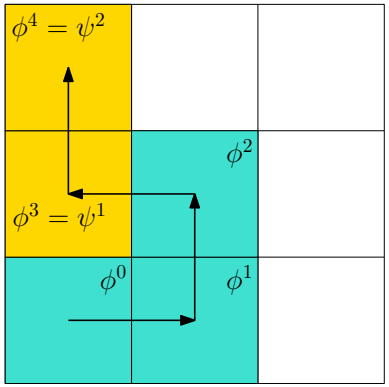
New valid set for ϕ^2: $V(\phi^2) \neq \emptyset$

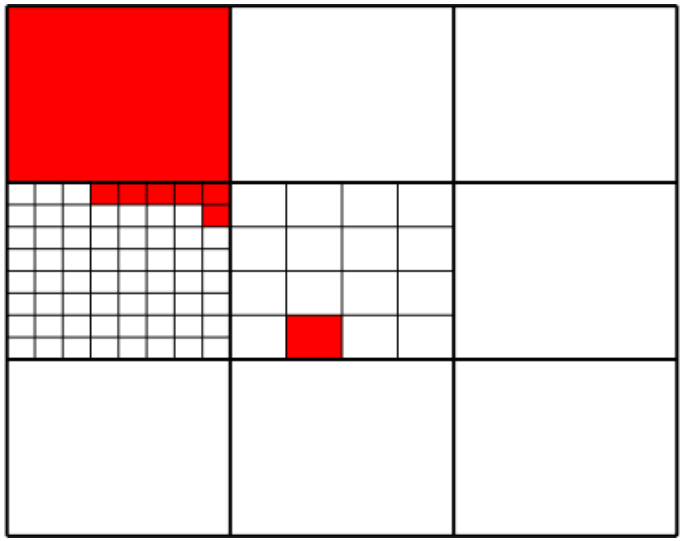
Algorithm



Specification:
reach top-left
from bottom left

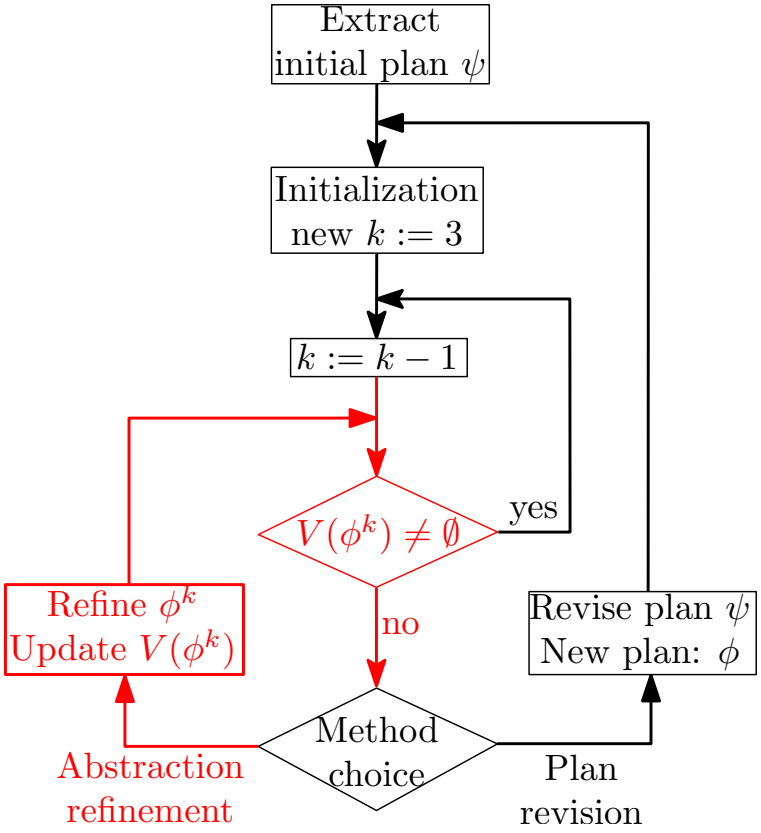
Revised plan:
$\phi^0\phi^1\phi^2\phi^3\phi^4$





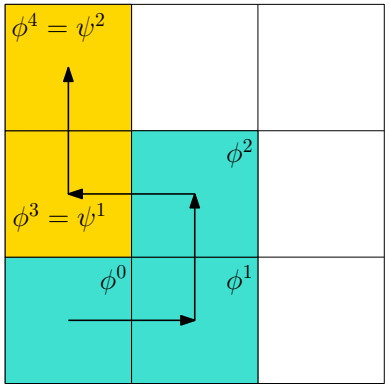
Transition $\phi^1 \to \phi^2$
Empty valid set: $V(\phi^1) = \emptyset$

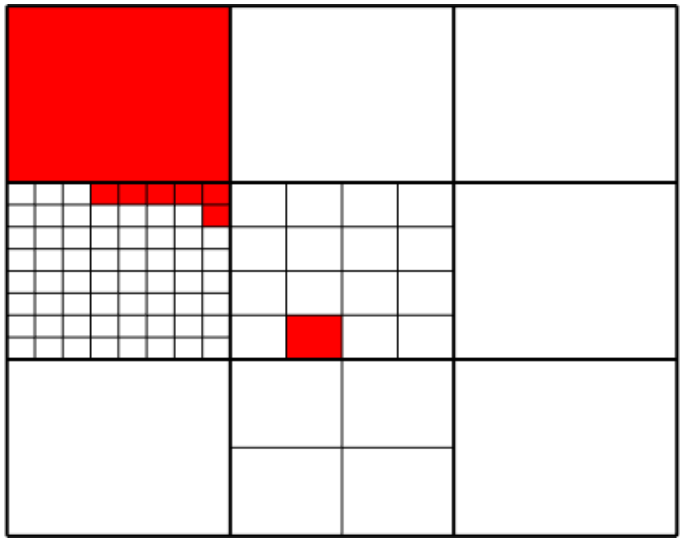
Algorithm



Specification:
reach top-left
from bottom left

Revised plan:
$\phi^0 \phi^1 \phi^2 \phi^3 \phi^4$



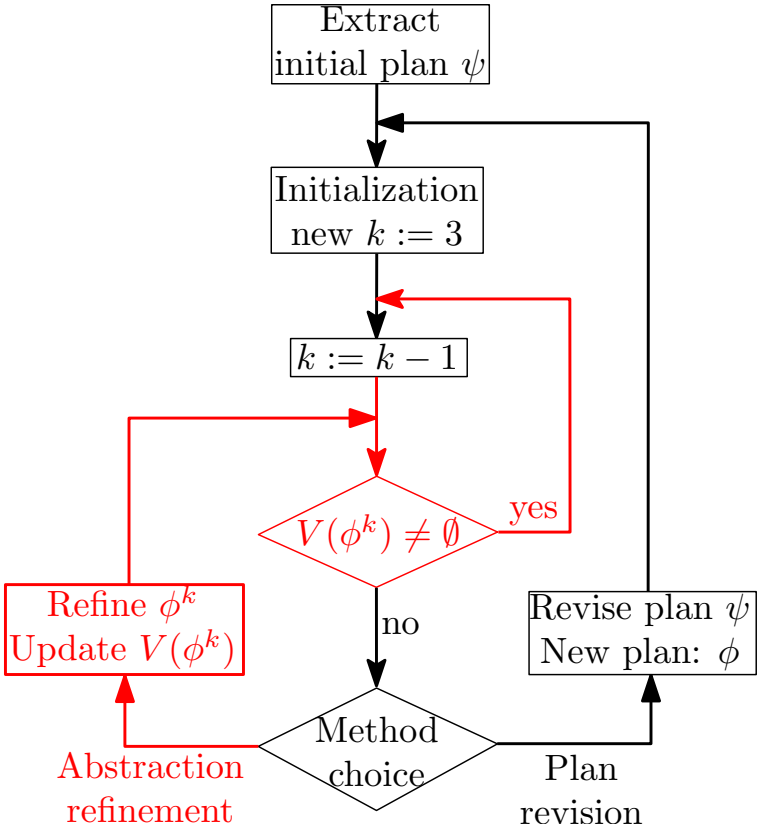


Transition $\phi^1 \to \phi^2$

Refine ϕ^1

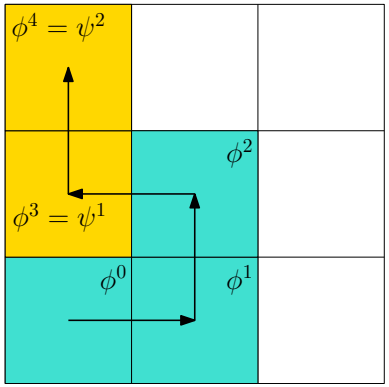
Valid set still empty: $V(\phi^1) = \emptyset$

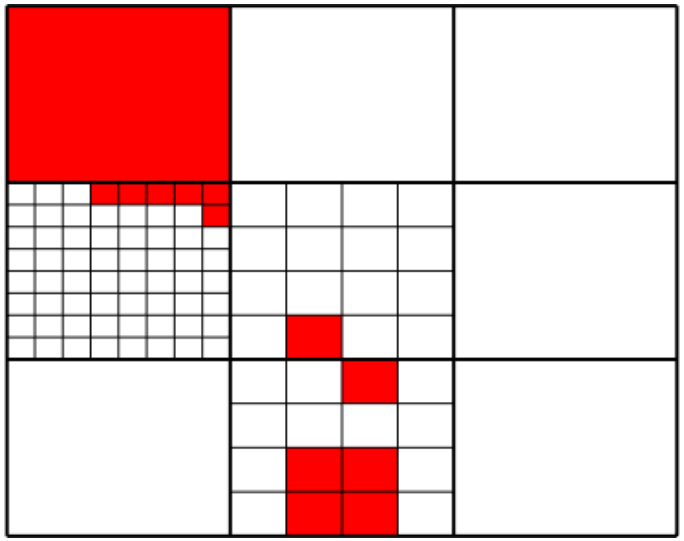
Algorithm



Specification:
reach top-left
from bottom left

Revised plan:
$\phi^0 \phi^1 \phi^2 \phi^3 \phi^4$



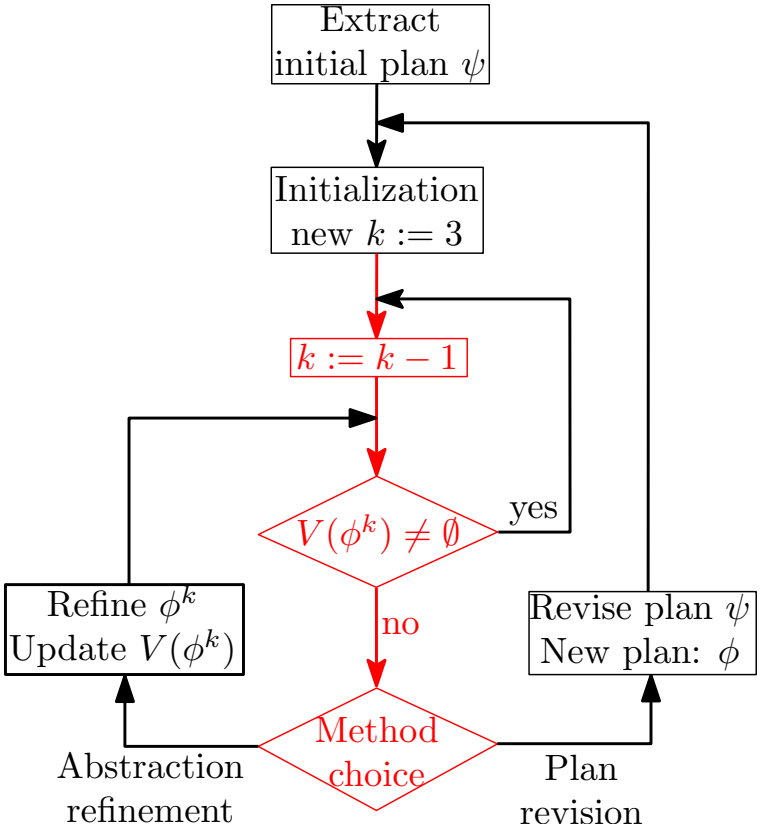


Transition $\phi^1 \rightarrow \phi^2$

Refine ϕ^1

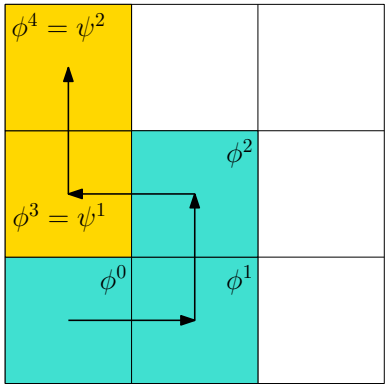
New valid set for ϕ^1: $V(\phi^1) \neq \emptyset$

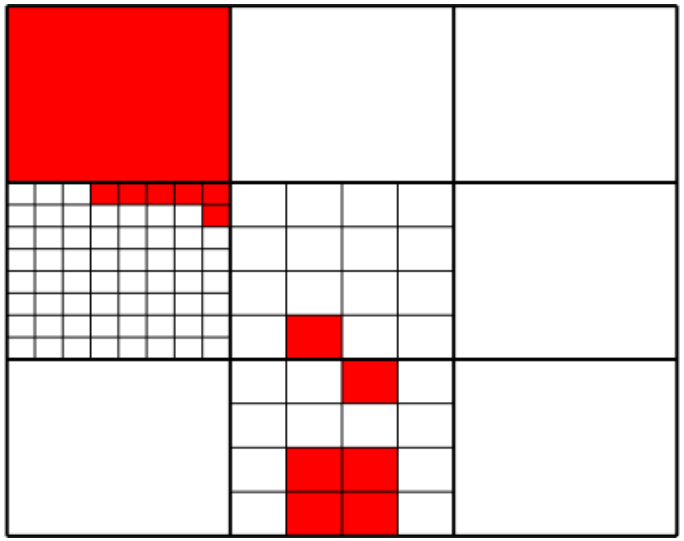
Algorithm



Specification:
reach top-left
from bottom left

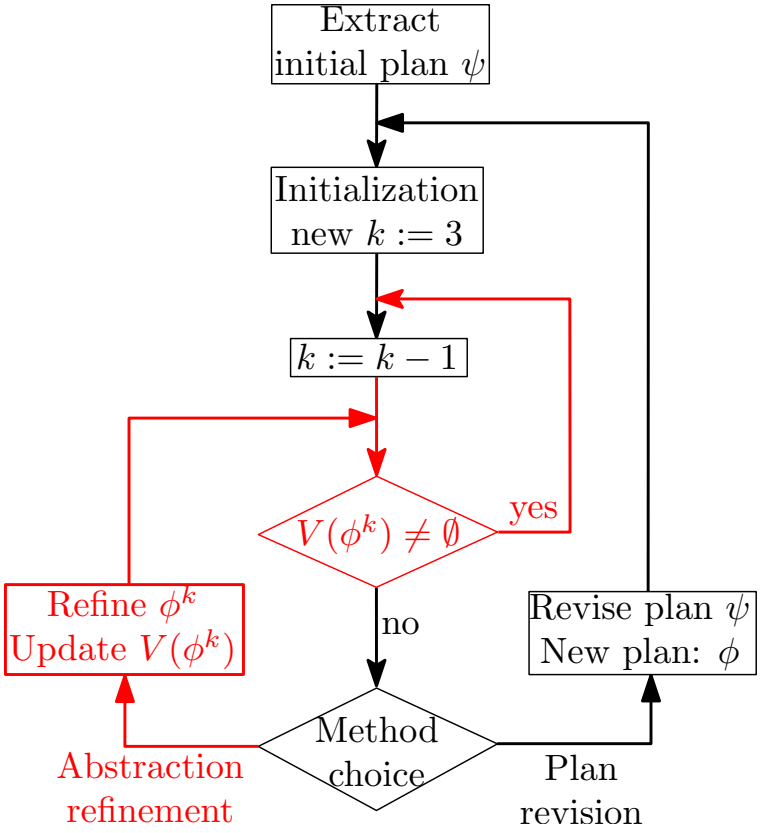
Revised plan:
$\phi^0 \phi^1 \phi^2 \phi^3 \phi^4$





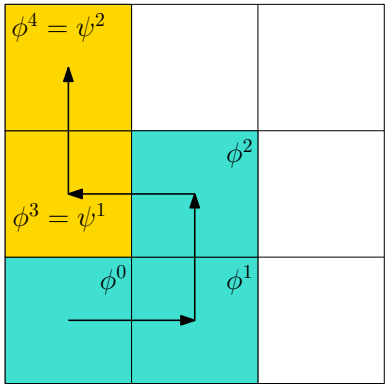
Transition $\phi^0 \to \phi^1$
Empty valid set: $V(\phi^0) = \emptyset$

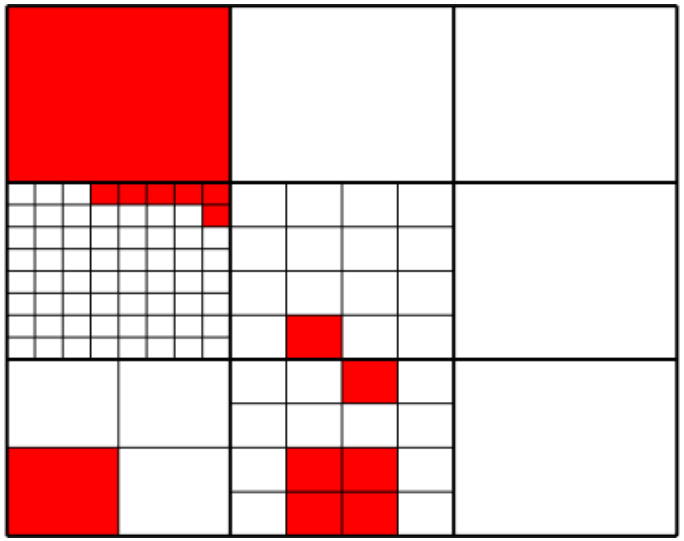
Algorithm



Specification:
reach top-left
from bottom left

Revised plan:
$\phi^0 \phi^1 \phi^2 \phi^3 \phi^4$





Transition $\phi^0 \to \phi^1$

Refine ϕ^0

New valid set for ϕ^0: $V(\phi^0) \neq \emptyset$

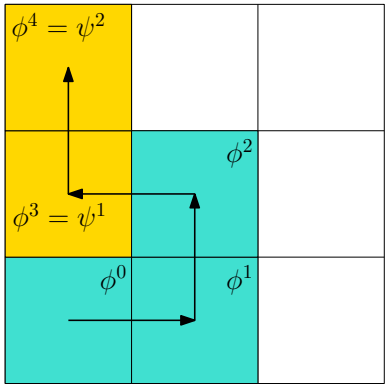
Algorithm

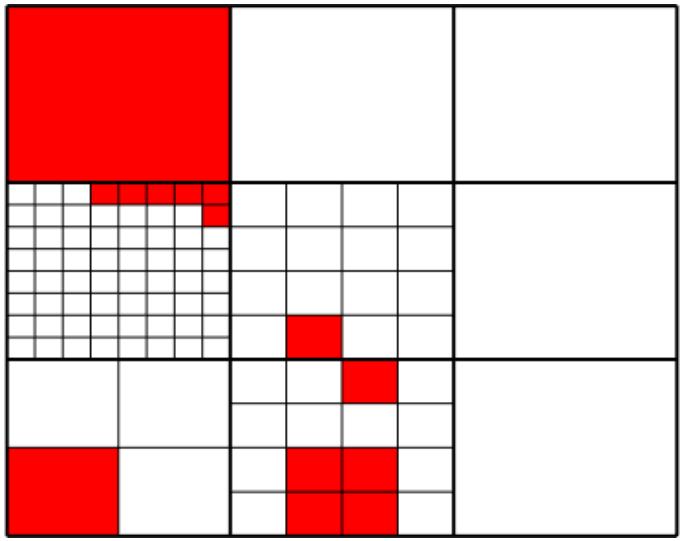
Outputs

- **Final plan**
- **Refined partition**
- **Valid set** for each cell in the plan
- **Controller** associated to each element of the valid sets

Specification: reach top-left from bottom left

Revised plan: $\phi^0\phi^1\phi^2\phi^3\phi^4$





Non-empty valid set for the initial cell: $V(\phi^0) \neq \emptyset$

Terminates the algorithm

Cost functions

Cost for abstraction refinement: refinement of cell ψ^j

- number of new subsymbols after refinement of ψ^j
  $2^n *$(number of invalid subsymbols of ψ^j)
- number of invalid subsymbols in $\psi^k \ldots \psi^{j-1}$ to be updated
- predicted number of subsymbols in future refinements

$$J_{AR}(\psi^j) = 2^n * |P_a(\psi^j) \setminus V(\psi^j)| + \sum_{l=k}^{j-1} |P_a(\psi^l) \setminus V(\psi^l)| + (k+1) * (2^n)^2$$

Cost for plan revision up to cell ψ^j

- predicted number of subsymbols in future refinements (for the new length of the revised plan)

$$J_{PR}(\psi^j) = (|\texttt{Revise}(\psi, j)| - |\psi| + j + 1) * (2^n)^2$$

Can add a weight to give more priority to one method.

Main result

- $S_\tau = (X_\tau, U_\tau, \underset{\tau}{\longrightarrow})$: sampled version of the continuous dynamics
- $S_a = (X_a, U_a, \underset{a}{\longrightarrow})$: abstraction of S_τ with the refined partition X_a

Definition

A map $H : X_\tau \to X_a$ is a **feedback refinement relation** from S_τ to S_a if: $\forall x \in X_\tau, s = H(x), \forall u \in U_a \subseteq U_\tau$, $\forall x' \in Post_\tau(x, u), H(x') \in Post_a(s, u)$.

Theorem

If the **algorithm terminates in finite time***, the obtained controller C_a of S_a can be refined into a controller $C : X_\tau \to U_\tau$ defined by $C(x) = C_a(H(x))$ such that S_τ satisfies the main LTL specification.*

Conclusion and perspectives

Contributions

- framework combining abstraction refinement and plan revision
- for non-deterministic abstractions with LTL specification

Degrees of freedom

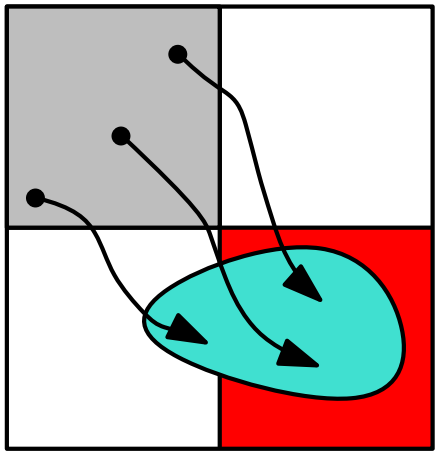
- cost functions giving priority to refinement or revision and picking which cell is refined or revised
- splitting strategy when refining

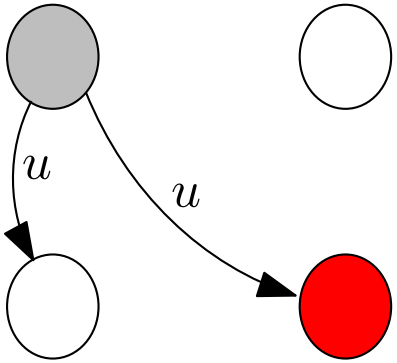
Perspectives

- guarantees of termination of the algorithm
- scalability: application within a compositional framework
- online approach

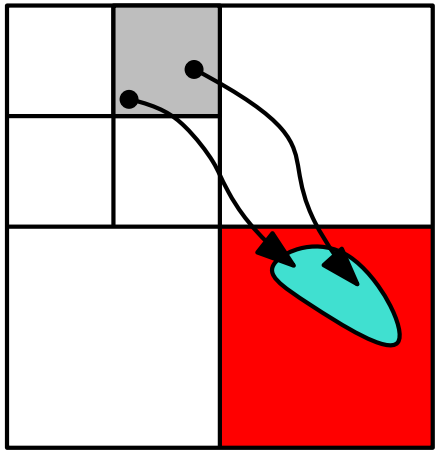
Abstraction refinement

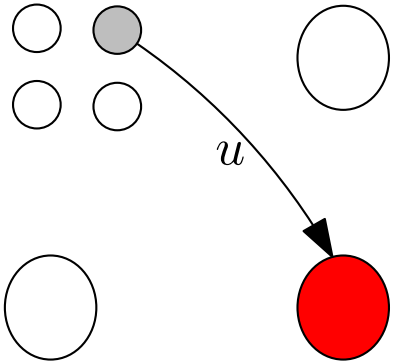
No guarantee of reaching the red cell





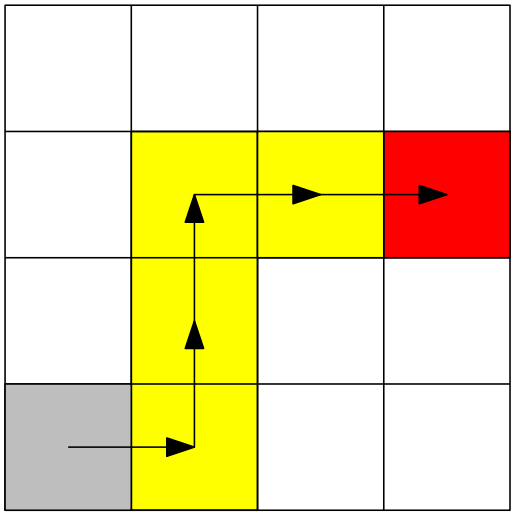
Refine the abstraction by **splitting the partition**



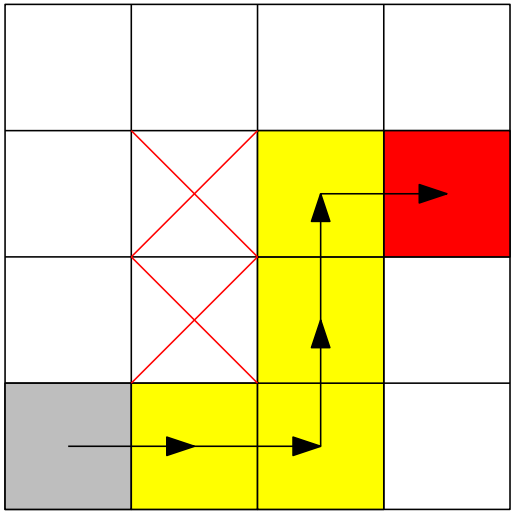


Plan revision

- Main specification: **LTL formula** (reach the red cell from the gray one)
- Find a **satisfying plan** (sequence of cells to be followed)



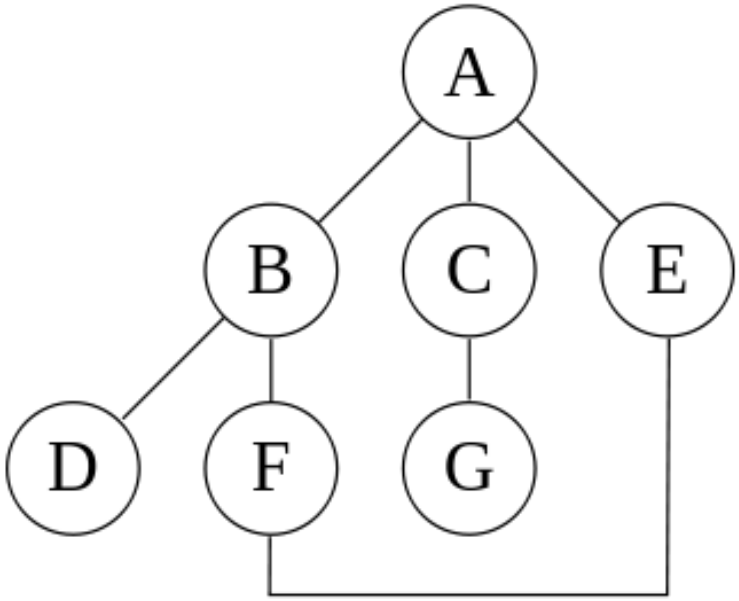
- Failure of the control synthesis to follow this plan ?
- **Revise the plan** by looking for an alternative satisfying sequence
- Repeat the synthesis attempt



Iterative deepening depth-first search

Search algorithm on the product automaton

- set $DepthLimit = 0$
- call a *Limited-Depth Depth First Search* with *DepthLimit*
- if no result found, repeat with $DepthLimit = DepthLimit + 1$



- $DepthLimit = 0$: A
- $DepthLimit = 1$: A, B, C, E
- $DepthLimit = 2$:
  A, B, D, F, C, G, E, F
- $DepthLimit = 3$:
  A, B, D, F, E, C, G, E, F, B

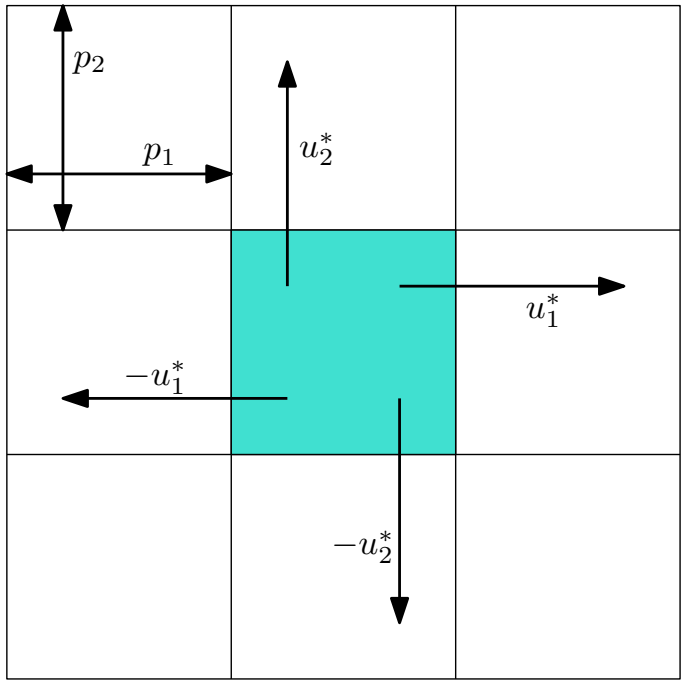
Sampling period

Real system: $\dot{x} = f(x, w) + u$
$u \in [-u^*, u^*]$

Nominal system: $\dot{x} = u$

Minimal sampling time such that we can steer the state of the nominal system between any two neighbor cells of the uniform partition:

$$\tau = \max_{i \in \{1,\ldots,n\}} \frac{p_i}{u_i^*}$$



Guarantees to obtain a **deterministic** abstraction of the nominal system