```
FSSP(S_0, G)
1  S ← S_0
2  π ← empty plan
3  loop
4      if G ⊆ S
5          return π
6      actions ← set of applicable actions in S
7      if actions is empty
8          return failure
9      choose a ∈ actions          ▷ choose an action
10     S ← γ(S, a)                 ▷ progress to new state
11     π ← π ∘ a                   ▷ update plan
```

Has plan generation step @ 11
MoveGen NOT specified
Search Strategy NOT specified
Node -- State of the World
MoveGeneration @ 6
Non-Det selection @ 9
-- handled by in full impltn:
selecting each choice 1/1 while
backtracking on failure

```
BSSP(S_0, G)
1  G' ← G
2  π ← empty plan
3  loop
4      if G' ⊆ S_0
5          if G ⊆ γ(S_0, π) then return π
6          else return failure
7      actions ← set of relevant actions for G'
8      if actions is empty
9          return failure
10     choose a ∈ actions          ▷ choose an action
11     G' ← γ^-1(G', a)            ▷ regress to new goal
12     π ← a ∘ π                   ▷ update plan
```

Has plan generation step @ 12
MoveGen NOT specified
Search Strategy NOT specified
Node -- Goal/Subgoal
MoveGeneration @ 7
Non-Det selection @ 10
-- handled by in full impltn:
selecting each choice 1/1 while
backtracking on failure

```
GSP(givenState, givenGoal, actions)
1  π ← empty plan
2  stack ← empty stack
3  S ← givenState
4  PushSet(givenGoal, stack)
5  while stack is not empty
6      X ← pop stack
7      if X is an action a
8          π ← π ∘ a
9          S ← Progress(S, a)
10     else if X is a compound goal G, and G is not true in S
11         PushSet(G, stack)
12     else if X is a goal g, and g is not true in S
13         a ← choose a relevant action that achieves g
14         if a is null then return failure
15         push a to stack
16         PushSet(pre(a), stack)
17 return π

PushSet(G, stack)
1  push G to stack
2  for each goal g in G
3      push g to stack
4  return

Progress(S, a)
1  return (S ∪ add-effects(a)) \ del-effects(a)
```

Has plan generation step @ 8
MoveGen NOT specified
Search Strategy IS(?) specified
Node -- (?)
MoveGeneration @ 13
Non-Det selection @ 13
-- handled by in full impltn:
selecting each choice 1/1 while
backtracking on failure

```
PSP(π)
1  flaws ← OpenGoals(π) ∪ Threats(π)
2  if flaws is empty
3      return π                    ▷ a plan with no flaws
4  choose φ ∈ flaws                ▷ choose a flaw
5  resolvers ← Resolve(φ, π)
6  if resolvers is empty
7      return failure              ▷ a flaw with no resolvers
8  choose ρ ∈ resolvers            ▷ choose a resolver
9  π' ← Refine(ρ, π)
10 return PSP(π')
```

Has plan generation step @ yes but where
MoveGen NOT specified
Search Strategy NOT specified
Node -- Partial Plan
MoveGeneration @ 1,4,5
Non-Det selection @ 4,8
-- handled by in full impltn:
selecting each choice 1/1 while
backtracking on failure