STRIPS (Stanford Research Institute of Problem Solver)

- ➢ STRIPS is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 while working at SRI (Stanford Research Institute)
- ➢ It's a first major planning system
- ➢ This was initially developed to guide a Robot's (named "Shakey") actions to move toward goal from present location and was developed in LISP programming language
- ➢ STRIPS attempt to find sequence of actions which transform initial state(model) to goal state. Strips represent a state as a collection of first order predicate calculus formulas
- ➢ Starting state would be list of all facts that were true in Robot's present situation. Goal state would also represent list of facts that would be true once Robot moves to that end(goal) state.

> STRIPS instance composed of
> 1. P - Set of conditions
> 2. O - Set of Operators
> 3. I - Initial state
> 4. G - Goal state

- ➢ A sample STRIPS problem

> Initial state: At(A), Level(low), BoxAt(C), BananasAt(B)
> Goal state: Have(bananas)
>
> Actions:
>   // move from X to Y
>   _Move(X, Y)_
>   Preconditions: At(X), Level(low)
>   Postconditions: not At(X), At(Y)
>
>   // climb up on the box
>   _ClimbUp(Location)_
>   Preconditions: At(Location), BoxAt(Location), Level(low)
>   Postconditions: Level(high), not Level(low)
>
>   // climb down from the box
>   _ClimbDown(Location)_
>   Preconditions: At(Location), BoxAt(Location), Level(high)
>   Postconditions: Level(low), not Level(high)
>
>   // move monkey and box from X to Y
>   _MoveBox(X, Y)_
>   Preconditions: At(X), BoxAt(X), Level(low)
>   Postconditions: BoxAt(Y), not BoxAt(X), At(Y), not At(X)
>
>   // take the bananas
>   _TakeBananas(Location)_
>   Preconditions: At(Location), BananasAt(Location),
Level(high)
>   Postconditions: Have(bananas)

Reference: Artificial intelligence a modern approach 3rd edition by stuart russell peter norvig,
https://en.wikipedia.org/wiki/STRIPS
https://www.youtube.com/watch?v=mQ7M-zhiu7U

ADL and PDDL

ADL (Action Description Language) - An extension of STRIPS by relaxing some restrictions.

PDDL (Planning Domain Description Language) -

> ➤ It was an attempt to standardize artificial intelligence planning languages.
> ➤ It was first developed by Drew McDermott and his colleagues in 1998 to make 1998/2000 International Planning Competition.
> ➤ PDDL separates the model of planning into two major parts
>> 1. Domain Description - Present every problem related to domain
>>> - Domain name definition and definition of requirements
>>> - Definition of predicates
>>> - Definition of possible actions with pre-conditions and effects
>>
>> 2. Problem Description - Present every problem related to planning
>>> - Problem name definition
>>> - Definition of domain name
>>> - Definition of all possible objects
>>> - Initial conditions
>>> - Goal states
>>
>> These two complement each other by separating problems into plan related and domain related.

A sample PDDL problem

Init(On(A, Table) ∧ On(B, Table) ∧ On(C, A) ∧ Block (A) ∧ Block (B) ∧ Block (C) ∧ Clear (B) ∧ Clear (C))
Goal(On(A, B) ∧ On(B,C))

Action(Move(b, x, y),
    PRECOND: On(b, x) ∧ Clear (b) ∧ Clear (y) ∧ Block (b) ∧ Block (y) ∧ (b!=x) ∧ (b!=y) ∧ (x!=y),
    EFFECT: On(b, y) ∧ Clear (x) ∧ ¬On(b, x) ∧ ¬Clear (y))
Action(MoveToTable(b, x),
    PRECOND: On(b, x) ∧ Clear (b) ∧ Block (b) ∧ (b!=x),
    EFFECT: On(b, Table) ∧ Clear (x) ∧ ¬On(b, x))

Reference: Artificial intelligence a modern approach 3rd edition by Stuart Russell and Peter Norvig ,

https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language#De_facto_official_versions_of_PDDL