# **RESEARCH REVIEW by *Sean Alexander Frenn***

# *Stanford Research Institute Problem Solver (STRIPS) and Language*

# Goals and Techniques

STRIPS (**St**anford **R**esearch **I**nstitute **P**roblem **S**olver) is a First-Order Logic (FOL) language with an associated linear solver to produce an automated planner. I was developed in 1971 by Richard Fikes and Nis Nilsson at SRI International. STRIPS has been designed to be a general-purpose problem solver for robot tasks. The initial version was implemented in LISP programming language. For searching through the space of world models, it uses a GPS-like means-end analysis strategy (Ernst and Newell 1969).

The language became the basis for most of the AI planning languages for expressing automated planning problems, i.e. action languages, such as PDDL and ADL.

It is an action-centric representation (as opposed to a feature-centric representation) which, for each action, specifies the effect of the action.

The **STRIPS representation** for an action consists of

* the **precondition**, which is a set of assignments of values to features that must be true for the action to occur, and
* the **effect**, which is a set of resulting assignments of values to those primitive features that change as the result of the action.

## Planning Domain Definition Language (PDDL)

PDDL is a standard encoding language for “classical” planning tasks. Which was a result of an attempt by Drew McDermott to standardize AI planning languages.

It supports among others, basic STRIPS-style actions, conditional effects, object creation and destruction, domain axioms over stratified theories and specification of safety constraints. This language is composed of:

• Objects: Things in the world that interest us.

• Predicates: Properties of objects that we are interested in; can be true or false.

• Initial state: The state of the world that we start in.

• Goal specification: Things that we want to be true.

• Actions/Operators: Ways of changing the state of the world.

The PDDL was derived from several other languages such as the Action Description Language (ADL) and the original STRIPS (Russell and Norvig 2009).

Most modern planners are based on Strips plus the first threes extensions. Unfortunately, PDDL does mainly provide a syntactic framework for these features but does give no or only an informal description of their semantics.

## Graphplan

Graphplan is an algorithm developed by Avrim Blum and Merrick Furst in 1995 for automated planning (Blum and Furst 1997). The algorithm takes a planning problem expressed in STRIPS as an input and is guaranteed to return a shortest possible partial-order plan, if there exists one.

This algorithm reduces the branching factor by searching in a special data structure.

Unlike the standard planning methods, the Graphplan algorithm does not immediately begin searching, instead it explicitly constructs and analyses a compact, directed, and leveled structure called a planning graph to guide its search.

It is composed of two phases: a Creation of a Planning Graph and Solution Extraction phase.

One of the key limitations of the Graphplan algorithm is that it is only applicable to STRIPS-like domains, e.g. actions cannot create new objects, the effects can only be determined statically.

# REFERENCES

* **Wikipedia.**
* **Fikes, R. E. and N. J. Nilsson (1971). "STRIPS: A new approach to the application of theorem proving to problem solving." Artificial intelligence 2(3-4): 189-208.**
* **McDermott, D., et al. (1998). "PDDL-the planning domain definition language."**
* **Russell, S. J. and P. Norvig (2009). Artificial intelligence: a modern approach (3rd edition), Prentice Hall.**