# Research review

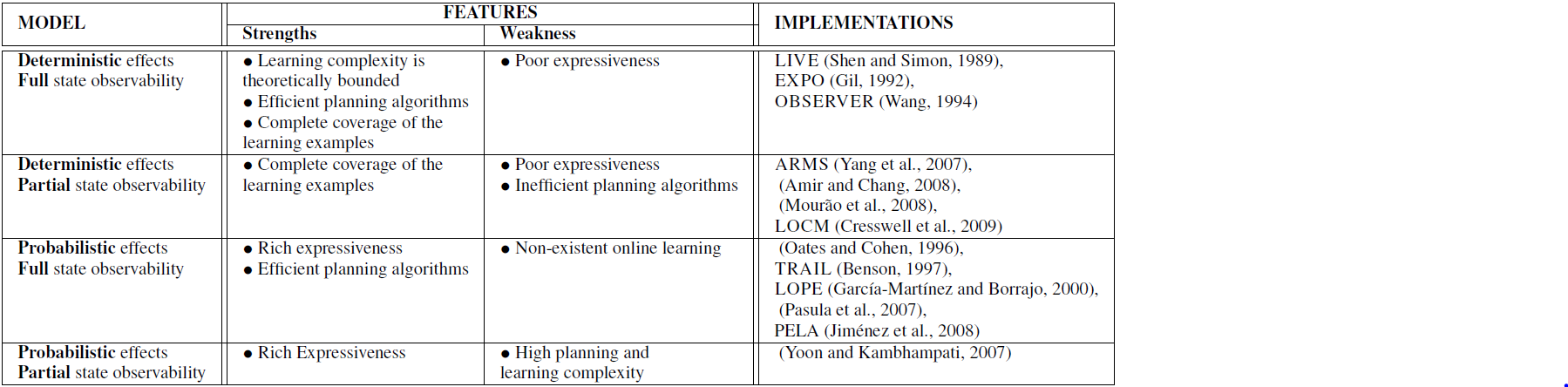
In a survey published in 2009, Jimenez et.al. reviewed several applications of machine learning to automated planning problems (Jimenez, 2009). Figure 1 (Jimenez, 2009) provides an overview of different approaches, where *Model* refers to the planning and not the machine learning model. 

Figure 1: Implementations of systems for planning action modelling

The third model comprises problems where the state is fully observable but the actions can have non-deterministic effects. Problems in this domain can be represented by the *Probabilistic Planning Domain Definition Language (PPDDL)* (Younes, Littman, Weissman, & Asmuth, 2005) which is an extension of the classical *Probabilistic Planning Domain Definition Language (PDDL)* (Fox & Long, 2003). *PPDDL* defines the syntax of probabilistic action defects as

Where is the probability that outcome occurs as result of the specific action. The three systems *TRAIL*, *LOPE* and *PELA* use different approaches, to learn these probabilities.

*TRAIL* (Benson, 1997)learns relational models of probabilistic actions by exploring a given planning problem, starting with an initial action model. If the current action model is insufficient to generate a plan, *TRAIL* asks an external expert for an execution plan and learns from executing this plan. If it is able to generate a plan based on the current model, it learns a refined model from executing this plan.

The *LOPE* system (Garcıa-Martınez & Borrajo) uses an approach similar to *reinforcement learning.* It starts without any model and selects a random action in the initial state of a given environment. It then loops over the following iteration to learn an action model:

1. Execute an action
2. Perceive the *resulting* state and calculate its utility
3. Update the model based on the perception and utitily
4. Select the next action based on 2 and 3

The *PELA* system (Jimenez, 2009) is based on a model definition used by the *Stanford* Institute *Problem Solver (STRIPS).* It automatically updates a *STRIPS* model by using probabilities learned during plan executions and learns a first order decision tree.

# Literature

Benson, S. S. (1997). Learning Action Models for Reactive Autonomous Agents. Stanford.

Borrajo, D., Fernandez, F., & Jimenez, S. (2008). The PELA architecture: integrating planning and learning to improve execution. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08)*.

Fox, M., & Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning. *Journal of Artificial Intelligence Research*, S. 61–124.

Garcıa-Martınez, R., & Borrajo, D. (kein Datum). An integrated approach of learning, planning, and execution. *Journal of Intelligent and Robotics Systemss, 29*, S. 47–78.

Jimenez, D. (2009). A Review of Machine Learning for Automated Planning. *The Knowledge Engineering Review, Vol. 00:0, 1–24*.

Younes, H., Littman, M., Weissman, D., & Asmuth, J. (2005). The first probabilistic track of the international planning competition. *Journal of Artificial Intelligence Research, 24*, S. 851–887.