STRIPS Representation

- State-based view of time.
- The actions are external to the logic.
- Given a state and an action, the STRIPS representation is used to determine
  - whether the action can be carried out in the state
  - what is true in the resulting state
STRIPS Representation: Idea

- Predicates are **primitive** or **derived**.
- Use normal rules for derived predicates.
- The STRIPS representation is used to determine the truth values of primitive predicates based on the previous state and the action.
- Based on the idea that most predicates are unaffected by a single action.
- **STRIPS assumption:** Primitive relations not mentioned in the description of the action stay unchanged.
The STRIPS representation for an action consists of:

- **preconditions**: A list of atoms that need to be true for the action to occur.
- **delete list**: A list of those primitive relations no longer true after the action.
- **add list**: A list of the primitive relations made true by the action.
STRIPS Representation of “pickup”

The action \texttt{pickup(Ag, Obj)} can be defined by:

preconditions \[\text{autonomous}(Ag), Ag \neq Obj, at(Ag, Pos), sitting\_at(Obj, Pos)]

delete list \[sitting\_at(Obj, Pos)]

add list \[\text{carrying}(Ag, Obj)]
STRIPS Representation of “move”

The action \textit{move}(Ag, \textit{Pos}_1, \textit{Pos}_2) can be defined by:

**preconditions** \[\textit{autonomous}(Ag), \textit{adjacent}(\textit{Pos}_1, \textit{Pos}_2, S), \textit{sitting\_at}(Ag, \textit{Pos}_1)\]

**delete list** \[\textit{sitting\_at}(Ag, \textit{Pos}_1)\]

**add list** \[\textit{sitting\_at}(Ag, \textit{Pos}_2)\]
Example Transitions

```
sitting_at(rob, o109).
sitting_at(parcel, storage).
sitting_at(k1, mail).

move(rob, o109, storage) →
```

```
sitting_at(rob, storage).
sitting_at(parcel, storage).
sitting_at(k1, mail).

pickup(rob, parcel) →
```

```
sitting_at(rob, storage).
carrying(rob, parcel).
sitting_at(k1, mail).
```