CHEF

- A case-based planner for Chinese cooking.
- Creates recipes by retrieving and modifying existing ones.
- Notices failed recipes and repairs them.
- Learns rules for predicting plan failure.
- Uses failure rules to predict potential problems with new goals and uses as additional retreival index to find plans that avoid the failure.

Example

- Want to cook stir-fry dish with beef and with broccoli.
- Retrieve most similar plan which is for beef with green beans (both vegetables).
- “Object critics” replace green beans with broccoli, insert step for chopping broccoli and adjusts cooking time.
- Simulate execution of plan expecting achievement of several associated goals.
- Notice failure in that the broccoli is soggy not crisp as expected.
- Simulator also returns a causal explanation of the why the failure occured (hack).

  The fat from the beef left in the pan is absorbed by the broccoli and makes it soggy.

Example (cont)

- Plan is repaired by applying general plan critics (20):
  Determine failure type:
  Side-effect:disabled-condition:concurrent

  Determine appropriate fix:

  Split-and-Reform (break problematic step in two)
  Alter-Plan:Side-Effect (replace step so side-effect avoided)
  Adjunct:Plan (Add a white-knight step to fix plan)

  For the example, pick Split-and-Reform fix: Stir fry the broccoli seperate and then add it back later.

- Learn new failure prediction rule by compiling failure explanation (EBL).

  Meat sweats when it is stir-fried.
  Stir-frying in too much liquid makes vegetables soggy.

  Failure indicated by stir-frying meat with crisp vegetable.

- Store and index new plan by generalized goal features and by failure conditions (stir-frying crisp vegetable with meat).

Example (final)

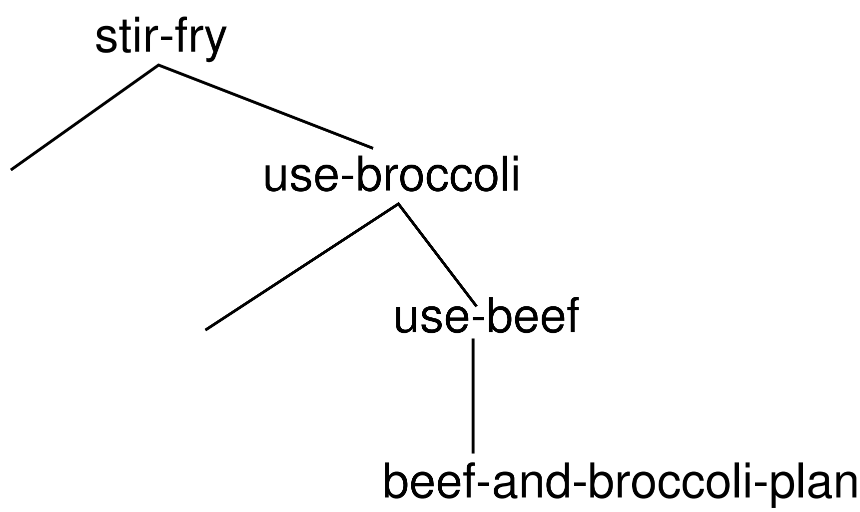
- Use new compiled rule to predict failures on new cases:

  New goal: stir-fry chicken with snow peas.
  Failure indicated: stir-frying meat with crisp vegetable will make vegetable soggy.

- Use failure condition as an additional index to retrieve plan. In this case, choose the reparied beef and broccoli plan. Failure condition causes to prefer to chicken and green beans plan.

Learning Plans

- Plans are stored into memory using a descrimination net in which features are ordered by importance.

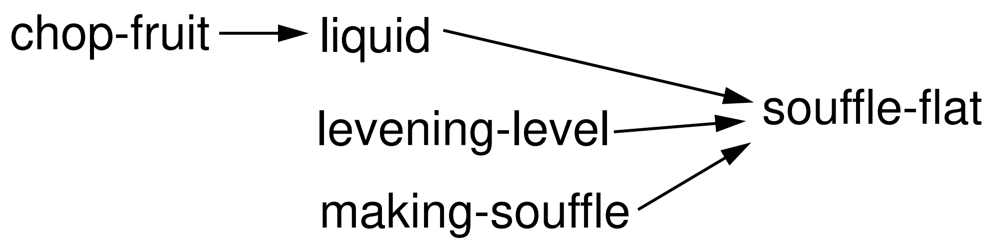


- Plans indexed under multiple levels of abstraction (e.g. beef and meat).

- Unclear how multiple indices are integrated or how to index on relational (non-featural) aspects.

Learning Failure Rules

- Explanation of failure used to compile failure rules.
  While making souffle with strawberries:



- Multiple levels of explanation used to produce failure rules.

  fruit $\wedge$ souffle $\rightarrow$ fall
  liquid $\wedge$ souffle $\rightarrow$ fall

  Rule 1 triggered when making orange souffle.
  Rule 2 triggered when making kirsch souffle.

Learning Critics

- Repairs can also be saved to allow repairing plans with a known failure but for which there is no similar stored plan that fixes it.

- Some repairs too "complex" to save and retrieve. Only simple ones based on remove-feature or reorder-steps.

  Duck with dumplings plan lead to greasy dish.
  Repair by removing fat from duck first.
  Save this repair indexed under duck.

CHEF Comments

- Case-based planning is an important approach to dealing with the computational complexity of planning.
- Indexing, retrieval, and modification are the difficult problems.
- Chef does not provide clear, robust algorithms for performing these tasks, mainly a set of specific hacks and critics that work for the specific domain.
- Kambhampati and Hendler have provided clearer algorithms for retrieval and reuse on nonlinear plans.
- Must be concerned that the total cost of failure prediction, retrieval, and adaptation do not exceed the benefit as the case and knowledge-base grows (the utility problem).
- Veloso has experimented with a “derivational analogy” planner and demonstrated that overall performance improvement is possible with a large case library.

Learning in Planning

- Integrating learning into planning is an important way to deal with the complexity of the task and can direct and give purpose to a learning method.

- An alternative to case based is generalizing specific plans as far as possible while still maintaining their validity.

  These so called “macro-operators” are generalized plans that can later be instantiated to solve similar problems.

- Another approach is to learn control rules that decide at each point in the search space, what choice to take.

  Control rules can be learned in two ways:

  – Inductively from many examples.
  – Analytically from one example.