Backtracking and Induction in ACL2

John Erickson

Introduction

- Often, ACL2 must choose between several promising alternatives
- Eg, there may be multiple induction schemes suggested by a given theorem
- Usually, ACL2 will choose one and proceed
- By implementing backtracking, we can find more proofs automatically

Backtracking Applications

- Cross Fertilization and Generalization
- Unmeasured Induction Variable Matching

Generalization

- Often a theorem must be generalized before it can be proved
  - Eg: (REV1 X NIL) = (RV X) usually generalized to (REV1 X A) = (APPEND (RV X) A)
- ACL2 already does generalize some theorems
- Generalization with cross-fertilization can be a powerful combination
  - Eg: ACL2 discovers (RV (APPEND X (LIST A))) = (CONS A (RV X))) during proof of (RV (RV X)) = X
- Often generalizes to non-theorems or non-inductive theorems, or cross-fertilizes when it is not helpful

Reverse Example

```
(defun rv (x)
  (if (endp x)
      nil
    (append (rv (cdr x)) (list (car x)))))

(defun rv1 (x a)
  (if (endp x)
      a
    (rv1 (cdr x) (cons (car x) a))))

(defthm rv1-rv
  (equal (rv1 x nil) (rv x)))
```

Reverse Almost Wins

Subgoal *1/2'4'

```
(IMPLIES (EQUAL (RV1 X2 NIL) (RV X2))
         (EQUAL (RV1 X2 (LIST X1))
                (APPEND (RV X2) (LIST X1)))).
```

We now use the hypothesis by substituting (RV1 X2 NIL) for (RV X2) and throwing away the hypothesis. This produces

Subgoal *1/2'5'

```
(EQUAL (RV1 X2 (LIST X1))
       (APPEND (RV1 X2 NIL) (LIST X1))).
```

- Name the formula above *1.1.

Reverse Wins

Subgoal *1/2'4'

```
(IMPLIES (EQUAL (RV1 X2 NIL) (RV X2))
         (EQUAL (RV1 X2 (LIST X1))
                (APPEND (RV X2) (LIST X1)))).
```

We now generalize by substituting X3 for (LIST X1) and throwing away the hypothesis. This produces

Subgoal *1/2'5'

```
(EQUAL (RV1 X2 X3)
       (APPEND (RV X2) X3)).
```

- Name the formula above *1.1.

Unmeasured Variable Instantiation

- Based on paper by Kapur and Subramaniam
- Represent unmeasured induction variables with constrained functions in induction hypothesis
- Match induction conclusion and induction hypothesis to find instantiations for unmeasured induction variables
- Use mismatches to discover lemmas

Rotate Example

```
(defun rot (n x)
    (if (zp n)
          x
       (rot (1- n) (ap (cdr x) (list (car x))))))
```

- Goal: (rot (len x) (ap x y)) = (ap y x)

Rotate Example (2)

- Induct:
  - IH: (rot (len (cdr x)) (ap (cdr x) (F x y))) = (ap (F x y) (cdr x))
  - IC’: (rot (len (cdr x)) (ap (ap (cdr x) y) (list (car x)))) = (ap y x)

Rotate Example (2)

- Induct:
  - IH: (rot (len (cdr x)) (ap (cdr x) (F x y))) = (ap (F x y) (cdr x))
  - IC’: (rot (len (cdr x)) (ap (ap (cdr x) y) (list (car x)))) = (ap y x)
- Decompose and Match:
  - IH: (ap (cdr x) (F x y))
  - IC: (ap (ap (cdr x) y) (list (car x)))

Rotate Example (2)

- Induct:
  - IH: (rot (len (cdr x)) (ap (cdr x) (F x y))) = (ap (F x y) (cdr x))
  - IC’: (rot (len (cdr x)) (ap (ap (cdr x) y) (list (car x)))) = (ap y x)
- Decompose and Match:
  - IH: (ap (cdr x) (F x y))
  - IC: (ap (ap (cdr x) y) (list (car x)))
- Speculate (endp (cdr x)) and Simplify:
  - IH: (F x y)
  - IC: (ap y (list (car x)))
- Found definition for (F x y)

Rotate Example (3)

- Definition
  - (F x y) = (ap y (list (car x)))
- Attempted Match:
  - IH: (ap (cdr x) (F x y))
  - IC: (ap (ap (cdr x) y) (list (car x)))
- Test Match:
  - IH: (ap (cdr x) (ap y (list (car x))))
  - IC: (ap (ap (cdr x) y) (list (car x)))
  - Proved as lemma, can be generalized first to obtain assoc-append

Implementation

- Prover implemented in ‘bprove’ book
- Use ACL2’s simplification, generalization routines
- Custom routines for generating induction subgoals with constrained functions
- Successful proofs submitted to ACL2 using make-event

Future Work

- Implement more alternatives
- Explore parallelization
- Find better ways to present output

Conclusions

- Computing power keeps growing
- Search is one way to use these cycles
- Unmeasured induction variable instantiation and generalization are two places this idea can be applied