Use ACL2 to Prove the Correctness of Clebsch-Gordan Coefficient

Xiaohui Chen

04/13/2015

Timeline

- 04/13/2015-04/19/2015

  Finish coding and finish verify the theorem
- 04/20/2015-04/26/2015

  Finish project report (first draft) and presentation slides
- 04/26/2015-05/03/2015

  Finish presentation and modify project report
- Finish final draft of project report by 05/06/2015

My model

- ((A . (j . m_j)) .

  ((B . ((l . m_l) . (s . m_s)))

  (C. ((l . m_l) . (s . m_s)))

  ....))

- We have to verify that after a series of operations on the above pair, A=B+C+D+...

Operations

- Input:
- ((A . (j . m_j)) .

  ((B . ((l . m_l) . (s . m_s)))

  (C. ((l . m_l) . (s . m_s)))

  ….))
- It satisfies the following conditions:
- (1) A=B+C+D+...
- (2) j>=|m_j|

- (3) For all l, m_l, s, m_s, l>= |m_l|, s>=|m_s|
- (4) l and m_l are either integers or half integers at the same time
- (5) s and m_l have the same condition as (4)
- (6) j=l+s, m_j=m_l+m_s
- (7) j and m_j have the same condition as (4)
- (8) m_j= -j, -j+1, …, j-1, j
- (9) m_l= -l, -l+1, …, l-1, l
- (10) m_s= -s, -s+1, …, s-1, s

- Step 1: normalize
- ((A . (j . m_j)) .

  ((B . ((l . m_l) . (s . m_s)))

  (C. ((l . m_l) . (s . m_s)))

  ….)) →

  ((1 . (j . m_j)) .

  ((B/A . ((l . m_l) . (s . m_s)))

  (C/A . ((l . m_l) . (s . m_s)))

  ….))

- Step 2: apply lowering operators
- e.g J-lowering (1 . (j . m_j))= (A' . (j . m_j-1))
- Special case: j=-m_j

  J-lowering (1 . (j . m_j)) = 0
- Condition (11): j-state and coupled states are either both 0 or j-pair and coupled-list
- After the lowering operators, we have coefficients A'=B'+C'+D'+...

Progress

- Finished working on j-lowering operator
- Working on l-lowering and s-lowering operators
- After each operation, we have to verify that the model satisfies the 11 conditions
- A lot of splitting

Example of Large Splitting

- Function l-lowering returns a list of coupled states
- Same for s-lowering
- We have to append the two lists, merge the states with same quantum numbers, and clean up 0s in the list
- If both of the list are lists of 0s, then return 0

```
(defthm append-valid
        (implies (and (true-coupled-state x)
                      (true-coupled-state y))
                 (true-coupled-state (append-and-merge-states x y)))
 :hints (("Goal" :in-theory (disable same-denominator-add
                                      remove-strict-inequalities
                                      remove-weak-inequalities
                                      merge-same-property-4))))
```

- This theorem takes more than 84,000,000 steps, 983 seconds

Solutions

- Tried to use control-C and inspect the check-pioints

```
(implies (and a
              b
              c
              (not d)
              (not e))
         (integerp x))
```

- Another check-point gives:

```
(implies (and a
              b
              (not c)
              d
              e)
         (integerp x))
```

- Prove a lemma:

  (implies (and a

  b)

  (integerp x))

  - This saves a lot of time
  - But be careful when choosing lemmas, sometimes it could cause the theorem proof fails