Università degli Studi di Firenze
Facoltà di S.M.F.N.

Corso di Laurea in

Matematica

Formalizzazione in HOL del Metodo Gosper/WZ e Applicazione alla Dimostrazione di Identità Binomiali

Giovanni Gherdovich
26 Aprile 2007

Relatore:

Dott. Marco Maggesi

Anno Accademico 2005/2006

1 Introduzione: obiettivo di questo lavoro

Esistono diversi tipi di theorem provers; in questo lavoro ci si riferisce a theorem provers interattivi (in particolare si è utilizzato il sistema HOL Light). In questo tipo di sistema l'utilizzatore non solo imposta la tesi del problema, ma fornisce anche alcuni passi (se non tutti) del ragionamento che conduce alla dimostrazione.

Scrivere una dimostrazione matematica con un theorem prover significa, in linea di principio, dimostrare ciascuna delle proposizioni soggiacenti alla tesi, risalendo fino agli assiomi della logica usata. Naturalmente ogni theorem prover viene distribuito con una libreria più o meno vasta di teoremi (una sorta di cultura matematica inclusa) da cui l'utente può partire per dimostrare i propri *goals*. Ogni theorem prover possiede inoltre alcune procedure di calcolo (deduzione logica, aritmetica, ecc.) che consentono all'utente di non occuparsi di tutti i dettagli delle dimostrazioni.

L'effettiva possibilità di dimostrare interessanti fatti matematici con un theorem prover dipende molto da quanta "cultura" possiede il theorem prover, nel senso appena descritto, e dalla potenza delle procedure automatiche implementate.

L'ambizioso obiettivo che ci si è posti, e che questo lavoro non ha esaurito, è quello di implementare una procedura automatica certificata in HOL Light che, usando il metodo Gosper/WZ descritto in seguito, riesca a dimostrare almeno una parte delle identità su cui il metodo ha successo.

Se una simile procedura entrasse a far parte degli strumenti a disposizione degli utenti di HOL Light, consentirebbe di sfruttare la potenza del metodo Gosper/WZ all'interno di ogni dimostrazione in cui si presenti, come passaggio intermedio, la necessità di provare un'identità combinatorica.

L'approccio che si è seguito è quello dell'"utilizzo scettico" del Computer Algebra System Maxima, lasciando trovare a quest'ultimo la funzione certificante per ciascuna identità e quindi verificando la bontà di questo certificato in HOL Light.

Allo stato attuale si è dimostrato in HOL il teorema principale su cui si basa il metodo Gosper/WZ e, con l'approccio appena descritto, si sono dimostrate alcune identità.

La verifica dei risultati ottenuti con il CAS Maxima prevede, almeno in parte, un'analisi caso per caso che ancora non si è automatizzata.

La necessità di verificare le risposte fornite dal software Maxima ci ha spinto a elaborare in HOL Light una strategia di calcolo simbolico mirata, insieme a delle riscritture condizionali per semplificare eventuali coefficienti binomiali presenti nell'espressione; l'equazione non viene verificata automaticamente, ma il lavoro svolto finora mostra che sono possibili sostanziali miglioramenti al riguardo.

2 Soluzioni e certificati

Il metodo Gosper/WZ è una strategia in grado di dimostrare molte indentità combinatoriche del tipo $\sum_k f(n,k) = r(n)$.

Come enfatizzato in [1], [6] e [4], se il metodo ha successo nel dimostrare un'identità allora produce una funzione $G(n,k)$ che, a posteriori, può essere considerata come il "certificato di validità" dell'identità stessa. Come sarà chiaro dall'esposizione dettagliata del metodo Gosper/WZ, chi voglia controllare questo risultato dovrà solo verificare che

$$\frac{f(n+1,k)}{r(n+1)} - \frac{f(n,k)}{r(n)} = G(n,k+1) - G(n,k) \tag{1}$$

Anche se adesso può sembrare strano che questa espressione abbia a che fare con il problema di partenza, si vuole solo far notare quanto possa essere più semplice verificare algebricamente un'espressione di questo tipo piuttosto che, ad esempio, $\sum_k (-1)^k \binom{n}{k}\binom{2k}{k} 4^{n-k} = \binom{2n}{n}$.

Esistono molti altri problemi la cui soluzione fornisce un qualche tipo di certificato che consente di verificare agevolmente la soluzione stessa:

- fattorizzare polinomi (o numeri)
- trovare MCD di polinomi (o numeri)
- risolvere equazioni
- trovare primitive

Questi esempi sono di grande interesse per lo sviluppo di calcolo simbolico certificato, e consentono di far interagire in modo proficuo Computer Algebra Systems (CASs) e Theorem Provers.

3 Computer Algebra Systems e Theorem Provers

La classe di problemi descritta nel paragrafo 2 offre un interessante punto di incontro tra CASs e Theorem Provers, consentendo a entrambi i sistemi di esprimere le loro migliori caratteristiche.

Se da un lato un CAS è in grado di eseguire efficientemente algoritmi di vario tipo, non produce alcuna dimostrazione sulla validità dei risultati ottenuti: fornisce soltanto la risposta. Esistono esempi in cui alcuni CASs ottengono risultati clamorosamente sbagliati. Inoltre un CAS usa tipicamente una semantica assai poco espressiva per rappresentare enunciati matematici; si pensi all'equazione

$$(x^2-1)/(x-1) = x+1$$

che può essere interpretata come uguaglianza nel campo delle funzioni razionali $\mathbb{R}(x)$ oppure come uguaglianza dei valori di due funzioni, nel qual caso sarebbe falsa per $x = 1$. In un CAS questa equazione verrebbe rappresentata senza alcuna indicazione aggiuntiva, e la sua interpretazione sarebbe ambigua.

Al contrario i theorem provers sono dotati dei costrutti necessari a catturare il significato di ogni affermazione matematica, e ogni risultato ottenuto è necessariamente accompagnato dalla sua dimostrazione, anche se, almeno allo stato attuale del loro sviluppo, è difficile con essi eguagliare le capacità di calcolo di un CAS.

Nel trattare i problemi di cui al paragrafo 2, compresa la verifica di identità binomiali col metodo Gosper/WZ, si può ottenere il certificato della soluzione con un CAS, quindi eseguire la "versione breve" della dimostrazione (la verifica del certificato) in un theorem prover.

4 Il teorema WZ

Si riporta il teorema WZ (dalle iniziali degli autori, Wilf e Zeilberger) così come è enunciato in [1] e [6].

Teorema 1. Siano $F, G : \mathbb{N} \times \mathbb{Z} \to \mathbb{R}$ tali che

$$\begin{array}{lll} F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k) & & [\text{GosperEq}] \\ \lim_{k \to \pm\infty} G(n,k) = 0 \qquad \forall n \in \mathbb{N} & & [\text{Bordo}] \\ \sum_{k \in \mathbb{Z}} F(n,k) \quad \text{converge} \qquad \forall n \in \mathbb{N} & & [\text{Convergenza}] \end{array}$$

Allora $\sum_{k \in \mathbb{Z}} F(n,k)$ è costante in n.

Dimostrazione 1. Dalla prima delle ipotesi si ottiene

$$\begin{aligned} \sum_{k=-M}^{M} \{F(n+1,k) - F(n,k)\} &= \sum_{k=-M}^{M} \{G(n,k+1) - G(n,k)\} \\ \lim_{M \to +\infty} \{ \sum_{k=-M}^{M} F(n+1,k) - \sum_{k=-M}^{M} F(n,k)\} &= \lim_{M \to +\infty} \sum_{k=-M}^{M} \{G(n,k+1) - G(n,k)\} \end{aligned}$$

visto che le serie a primo membro sono convergenti e che la serie al secondo membro è telescopica con gli estremi che tendono a zero, vale che $\sum_k F(n+1,k) - \sum_k F(n,k) = 0$. Chiamando $\mathcal{F}(n) = \sum_{k=-\infty}^{\infty} F(n,k)$, si è appena dimostrato che per ogni $n \in \mathbb{N}$ vale $\mathcal{F}(n+1) - \mathcal{F}(n) = 0$. Applicando il principio di induzione si ha la tesi.

4.1 Applicazione

Per dimostrare che $\sum_{k \in \mathbb{Z}} f(n,k) = r(n)$, si pone $F(n,k) \doteqdot f(n,k)/r(n)$. Per applicare il teorema è necessario che $\sum_{k \in \mathbb{Z}} F(n,k)$ converga per ogni n naturale. Quindi sia $h(k) \doteqdot F(n+1,k) - F(n,k)$ e supponiamo di trovare un successione che rende telescopica $h(k)$, ottenendo così la $G(n,k)$ del teorema. Se $G(n,k)$ verifica le condizioni al contorno, il teorema afferma che $\sum_{k \in \mathbb{Z}} F(n,k)$ non dipende da n. Basta quindi verificare che $\sum_{k \in \mathbb{Z}} F(0,k) = 1$ e l'identità è dimostrata.

Si pone il problema degli n tali che $r(n) = 0$: in questi punti non si può dividere per $r(n)$. Nell'enunciato del teorema che si è dimostrato in HOL si è posto in ipotesi che il supporto di $r(n)$ sia una semiretta oppure un intervallo in $\mathbb{Z}$; in questo modo si applica il metodo Gosper/WZ solo dove $r(n)$ non è zero, e altrove si verifica direttamente che $\sum_k f(n,k) = 0$.

5 L'algoritmo di Gosper

Nel teorema di Wilf e Zeilberg si richiede l'esistenza di una funzione $G(n,k)$ che renda telescopica $f_k = F(n+1,k) - F(n,k)$. Tutto ciò sarebbe di ben poco interesse se non si conoscesse la ricetta per trovare $G(n,k)$; questa ricetta esiste, ed è nota come *algoritmo di Gosper*. La procedura di decisione scoperta da Ralph William Gosper Jr. è stata suggestivamente definita *il teorema fondamentale del calcolo discreto*: data una successione a_n ipergeometrica, ossia tale che $\frac{a_{n+1}}{a_n}$ è una funzione razionale, è capace di dire se esiste oppure no la successione S_n che rende telescopica a_n, nel senso che $a_n = S_n - S_{n-1}$. In caso affermativo, la procedura di Gosper porta a determinare S_n. Ciò comporta che

$$\sum_{n=1}^{m} a_n = S_m - S_0$$

il ché, effettivamente, non può non richiamare alla mente il teorema fondamentale del calcolo integrale.

La tecnica utilizzata da Gosper per dimostrare il suo teorema si basa su un particolare cambio di variabile che riduce l'equazione $a_n = S_n - S_{n-1}$ ad un sistema di equazioni lineari, per il quale l'esistenza di soluzioni è equivalente all'esistenza di S_n.

6 Un esempio di applicazione

Si consideri l'identità

$$\sum_{k\in\mathbb{Z}} \binom{n}{k}^2 = \binom{2n}{n}$$

Tralasciando per un momento i dettagli e ponendo $F(n,k) = \binom{n}{k}^2 / \binom{2n}{n}$, l'algoritmo di Gosper trova che $G(n,k) = -\frac{(3n-2k+3)}{2(2n+1)}\binom{n}{k-1}^2 / \binom{2n}{n}$ rende telescopica in k la successione $F(n+1,k) - F(n,k)$.

Questa identità presenta caratteristiche comuni a molte altre formule su cui il teorema WZ viene applicato con successo, e precisamente:

- le condizioni [Convergenza] e [Bordo] sono banalmente verificate perché le funzioni coinvolte hanno supporto finito.
- La definizione più conveniente per i coefficienti binomiali comporta che entrambi i parametri siano numeri interi. In questo contesto, se $n \geq 0$ l'identità si dimostra grazie alla tecnica di Gosper, se invece $n < 0$ è banalmente verificata perché entrambi i membri sono nulli.

7 Alcune identità

Ecco alcune tra le identità a cui il metodo Gosper/WZ si applica con successo:

$\sum_k \binom{n}{k} = 2^n$ $\quad$ $\sum_k (-1)^k \binom{n}{k}\binom{2k}{k} 4^{n-k} = \binom{2n}{n}$ $\quad$ $\sum_k \frac{\binom{n}{k}}{k!(a-k)!} = \frac{(n+a)!}{n!}$

$\sum_k (-1)^{n-k} \binom{2n}{k}^2 = \binom{2n}{n}$ $\quad$ $\sum_k (-1)^k \frac{\binom{n}{k}}{\binom{k+a}{k}} = \frac{a}{n+a}$ $\quad$ $\sum_k \binom{a}{k}\binom{n}{k} = \binom{n+a}{a}$

$\sum_k \binom{n}{k}^2 = \binom{2n}{n}$

8 Il supporto finito

Nella maggior parte delle identità a cui si applica il metodo WZ, sia la funzione sommanda che la funzione di Gosper $G(n,k)$ hanno supporto finito.

Per questo motivo si è preferito sostituire le condizioni [Convergenza] e [Bordo] con la richiesta che $F(n,k)$ e $G(n,k)$ abbiano supporto finito. In questo modo sia nella dimostrazione del teorema che nella sua applicazione si ha a che fare con somme di funzioni nulle ovunque eccetto che per un numero finito di termini, e si può contare su una ricca libreria di teoremi già dimostrati nel sistema HOL.

È stato comunque necessario dimostrare qualche lemma sulle funzioni a supporto finito; in quelli che riportati di seguito si afferma che se il supporto delle funzioni f e g è finito, così è anche per la funzione $(f-g)$, e che nella somma sugli interi dei valori di una funzione f gli indici al di fuori del suo supporto non danno alcun contributo.

Avvertenza: Si riportano, adesso e anche più avanti, alcuni enunciati espressi nel linguaggio di HOL. Sapendo che questa sintassi non è di facile comprensione, si cercherà sempre di esprimere il significato degli enunciati nel linguaggio comune cosicché il lettore non esperto di HOL possa saltare il testo nei riquadri.

```
⊢ ∀f:int->real g:int->real.
    support (+) (λx. f x - g x) (:int) ⊆
    ((support (+) f (:int)) ∪ (support (+) g (:int)))

⊢ ∀(f:int->real) (g:int->real).
    FINITE (support (+) f (:int)) ∧ FINITE (support (+) g (:int))
      ⇒ FINITE (support (+) (λ. f x - g x) (:int))

⊢ ∀f:int->real s:int->bool.
    sum (support (+) f (:int)) f = sum ((support (+) f (:int)) ∪ s) f

⊢ ∀f:int->real s.
    support (+) f (:int) ⊆ s ⇒ sum s f = sum (support (+) f (:int)) f

⊢ ∀f:int->real a:int b:int.
    support (+) f (:int) ⊆ {i:int | a ≤ i ∧ i ≤ b}
      ⇒ sum {i:int | a ≤ i ∧ i ≤ b} f = sum (:int) f
```

9 I binomiali a parametri interi

In questo lavoro si considerano coefficienti binomiali con argomenti nei numeri interi:

$$\binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!} & \text{se } n \geq 0 \ \wedge \ k \geq 0 \ \wedge n \ \geq k \\ 0 & \text{altrimenti} \end{cases}$$

Questa definizione non era disponibile in HOL Light, e la si è ricavata da quella dei coefficienti binomiali a parametri naturali. È stato quindi necessario dimostrare alcuni lemmi inerenti alle proprietà di questa nuova definizione, del tutto analoghe a quelle dei binomiali a parametri nei numeri naturali.

10 Il teorema WZ in HOL Light

L'enunciato del teorema WZ che si è scelto di dimostrare coinvolge successioni con indici nei numeri interi; il primo lemma da procurarsi è dunque una versione adattata agli interi del principio di induzione.

```
⊢ ∀(P:int->bool). P 0 ∧ (∀(n:int). P n ⇔ P (n + 1)) ⇒ (∀n. P n)
```

Si è scelto di sostituire la condizione [Bordo] dell'enunciato originale con l'ipotesi che il supporto delle funzioni è finito; è pertanto necessario sviluppare una mini-teoria degli insiemi finiti nei numeri interi.

Partendo dalla definizione induttiva di insieme finito presente nelle librerie di HOL Light

```
⊢ FINITE ∅ ∧ ∀(x:A) s. FINITE s ⇒ FINITE ({x} ∪ s)
```

si è dimostrato che un segmento di numeri interi è finito, quindi un criterio per riconoscere gli insiemi finiti

```
⊢ ∀n:int m. FINITE {z | n ≤ z ∧ z ≤ m}

⊢ ∀s:int->bool. FINITE s ⇔ ∃a. ∀x. (0 ≤ a ∧ (x ∈ s ⇒ (-a ≤ x ∧ x ≤ a)))
```

Se la successione $f(n)$ è resa telescopica da $g(n)$ e quest'ultima ha supporto finito, allora sommando i valori di f su tutti gli interi si ottiene zero:

```
⊢ ∀(f:int->real) g.
    f = (λi. g(i + 1) - g(i)) ∧ FINITE (support (+) g (:int)) ⇒ sum (:int) f = 0
```

Per dimostrare questo enunciato occorrono due lemmi: nel primo si afferma che gli insiemi in cui f e g sono definitivamente nulle coincidono

```
⊢ ∀f:int->real g:int->real a:int b:int.
    (∀n. f n = g (n + 1) - g n) ∧
    support (+) f (:int) ⊆ {(z:int) | n ≤ z ∧ z ≤ m} ∧
```

```
    FINITE (support (+) g (:int)) ⇒
    (∀x:int. ¬(x ∈ {(z:int) | n ≤ z ∧ z ≤ m}) ⇒ g x = 0)
```

mentre l'altro serve a semplificare la somma di una successione telescopica.

```
⊢ ∀(a:int->real) n:int m:int.
    m ≤ n + 1 ⇒ sum {i:int | m ≤ i ∧ i ≤ n} (λi. a(i + 1) - a(i)) =
                 a(n + 1) - a(m)
```

Si consideri una successione $f : \mathbb{Z} \to \mathbb{R}$ per cui vale $f(n+1) - f(n) = 0$ per ogni n intero; la successione è costante. Nel corso della dimostrazione del teorema WZ si vorrà dimostrare che una certa successione è costante, ma l'ipotesi sulla derivata discreta sarà valida soltanto in un intervallo di $\mathbb{Z}$; si tratta dell'insieme di numeri su cui il secondo membro dell'identità binomiale è non nullo. Serve quindi un criterio per successioni costanti un po' più debole:

```
⊢ ∀(P:int->bool) (f:int->real) (n0:int).
    ((∀n m. n ≤ m ∧ n ∈ P ∧ m ∈ P ⇒ {x | n ≤ x ∧ x ≤ m} ⊆ P) ∧
    P n0 ∧ (∀n. P n ⇒ f (n + 1) - f n = 0))
      ⇒ (∀n. P n ⇒ f n = f n0)
```

Per dimostrare il teorema precedente l'ipotesi sull'insieme P non è direttamente spendibile: bisogna rendere esplicito che P può essere soltanto un intervallo limitato di numeri, una semiretta, tutto $\mathbb{Z}$ oppure l'insieme vuoto:

```
⊢ ∀(P:int->bool).
    (∀n m. (n ≤ m ∧ P n ∧ P m ⇒ {x | n ≤ x ∧ x ≤ m} ⊂ P))
      ⇒
        ((∃a. ∃b. P = {(x:int) | a ≤ x ∧ x ≤ b} ∧ a ≤ b) ∨
        (∃c. P = {(x:int) | c ≤ x}) ∨
        (∃d. P = {(x:int) | x ≤ d}) ∨
        (∀(k:int). P k) ∨ (∀(k:int). ¬(P k)))
```

Dal precedente enunciato si vede che si dovranno esibire gli estremi di un intervallo di numeri o di una semiretta; questi saranno di volta in volta il massimo o il minimo dell'insieme P. Nella libreria di Hol Light manca la teoria del massimo e minimo per insiemi di numeri interi, quindi è stato necessario scrivere le due definizioni e dimostrare qualche lemma al riguardo. Visto che la teoria dei numeri reali è ben trattata in HOL Light, è sembrato conveniente definire il massimo per numeri interi partendo dal *sup* di insiemi di numeri reali, definito nella libreria di HOL Light (ϵ indica l'operatore di scelta di Hilbert):

```
⊢ sup s = ϵa. (∀x. x ∈ s ⇒ x ≤ a) ∧ ∀b. (∀x. x ∈ s ⇒ x ≤ b) ⇒ a ≤ b
```

ecco quindi la definizione di massimo che serve, dedotta dalla precedente immergendo in $\mathbb{R}$ l'insieme di numeri interi:

```
⊢ int_sup S = int_of_real (sup (IMAGE real_of_int S))
```

Si possono ora dimostrare alcune proprietà del massimo e del minimo:

```
⊢ ∀(P:int->bool). ¬(P = ∅) ∧ FINITE P ⇒ int_sup P ∈ P

⊢ ∀(P:int->bool). ¬(P = ∅) ∧ FINITE P ⇒ (∀x:int. x ∈ P ⇒ x ≤ int_sup P)

⊢ ∀(P:int->bool). ¬(P = ∅) ∧ (∃b. ∀z. z ∈ P ⇒ z ≤ b) ⇒ int_sup P ∈ P

⊢ ∀(P:int->bool).
     ¬(P = ∅) ∧ (∃h. ∀z. z ∈ P ⇒ z ≤ h) ⇒ (∀x. x ∈ P ⇒ x ≤ int_sup P)

⊢ ∀(P:int->bool). ¬(P = ∅) ∧ FINITE P ⇒ int_inf P ≤ int_sup P
```

Analoghe proprietà sono state dimostrate per il minimo.

Adesso, avendo dimostrato che la somma su $\mathbb{Z}$ di una successione telescopica a supporto finito vale zero e disponendo di un adeguato criterio per riconoscere le successioni costanti, si può dimostrare il teorema WZ nella forma operativa, come discusso nel paragrafo 8.

```
∀F:int->int->real r:int->real.
     ((∀(n:int) m. (n ≤ m ∧
     n ∈ (support (+) (λi. r i) (:int)) ∧
     m ∈ (support (+) (λi. r i) (:int))) ⇒
     {x | n ≤ x ∧ x ≤ m} ⊆ (support (+) (λi. r i) (:int)))
∧
     (∃G:int->int->real. ∀n.
     (FINITE (support (+) (λk. f n k / r n) (:int))) ∧
     (FINITE (support (+) (λk. G n k) (:int))) ∧
     (¬(r n = 0) ⇒
     ∀k. f (n + 1) k / r (n + 1) - f n k / r n = G n (k + 1) - (G n k)))
∧
     (∀n. r n = 0 ⇒ (sum (:int) (λk. f n k) = r n))
∧
     (∃m:int. (sum (:int) (λk. f m k) =  r m) ∧ ¬(r m = 0))

  ⇒ (sum (:int) (λk. f n k) = r n)
```

11 Statistiche sul codice

Per dimostrare il teorema WZ in HOL è stato necessario provare circa 40 lemmi preliminari, per un totale di circa 1500 linee di codice; la dimostrazione del teorema WZ in sé consta di circa 100 linee. Questi lemmi riguardano le funzioni a supporto finito, gli insiemi di numeri interi, le successioni ad indici negli interi e le somme di successioni telescopiche.

La dimostrazione di alcuni di questi risultati è sorpendentemente più lunga della corrispondente dimostrazione fatta con carta e penna. In parte ciò è dovuto alla necessità di

affrontare completamente tutti i casi che l'enunciato di un teorema presenta: in un sistema logico formale come HOL non è possibile dire "quest'altro caso si affronta in modo analogo" come spesso si fa in matematica tradizionale.

Un'altra ragione che porta le dimostrazioni formali ad essere così lunghe è di natura psicologica: con carta e penna la ricerca della dimostrazione più breve e semplice è motivata dal fatto che una dimostrazione lunga e complicata è difficile da spiegare e da verificare, e quindi è più soggetta a errori. Questa spinta verso la semplicità cade quando si lavora con un *assistente di dimostrazione* come HOL Light, perché è pressoché impossibile che una dimostrazione accettata da un theorem prover sia sbagliata, per quanto contorta e complicata possa essere.

12 Semplificare i coefficienti binomiali

Nel dimostrare le identità col metodo WZ ci si trova a dover verificare la condizione di telescopicità [GosperEq]

$$F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k)$$

La difficoltà principale è semplificare i coefficienti binomiali presenti in questa espressione; eliminati questi, esistono varie procedure in HOL capaci di verificare identità aritmetiche semplici.

Per trattare i binomiali si sono scritte alcune tattiche per HOL che li riportano alla "forma base", ossia li riscrivono con i seguenti teoremi:

- Riscrittura `TOP STEP`

$$\binom{n+1}{k} = \begin{cases} 0 & \text{se } k = n+1 \text{ e } n+1 < 0 \\ 1 & \text{se } k = n+1 \text{ e } n+1 \geq 0 \\ \frac{n+1}{n+1-k}\binom{n}{k} & \text{altrimenti} \end{cases}$$

- Riscrittura `TOP BACKSTEP`

$$\binom{n-1}{k} = \begin{cases} 0 & \text{se } n = 0 \\ \frac{n-k}{n}\binom{n}{k} & \text{altrimenti} \end{cases}$$

- Riscrittura `BOTTOM STEP`

$$\binom{n}{k+1} = \begin{cases} 0 & \text{se } k+1 = 0 \text{ e } n < 0 \\ 1 & \text{se } k+1 = 0 \text{ e } n \geq 0 \\ \frac{n-k}{k+1}\binom{n}{k} & \text{altrimenti} \end{cases}$$

- Riscrittura `BOTTOM BACKSTEP`

$$\binom{n}{k-1} = \begin{cases} 0 & \text{se } k = n+1 \text{ e } n < 0 \\ 1 & \text{se } k = n+1 \text{ e } n \geq 0 \\ \frac{k}{n+1-k}\binom{n}{k} & \text{altrimenti} \end{cases}$$

Queste riscritture convertono i binomiali eliminando le somme e le sottrazioni al loro interno, facendo comparire delle funzioni razionali. Per agevolare le dimostrazioni è stato necessario sviluppare delle apposite tattiche in grado di effettuare le *riscritture condizionali* precedenti.

A titolo di esempio, ecco come la riscrittura `TOP STEP` scompone l'espressione $\binom{(a+b)+1}{k}$ = `qualcosa`:

```
val it : goalstack = 3 subgoals (3 total)

0 [‘k = (a + b) + 1‘]
1 [‘0 ≤ (a + b) + 1]
‘1 = qualcosa‘

0 [‘k = (a + b) + &1‘]
1 [‘¬(0 ≤ (a + b) + 1)‘]
‘0 = qualcosa‘

0 [‘¬(k = (a + b) + &1)‘]
1 [‘¬((a + b) + 1 - k = 0)‘]
‘(a+b+1)/(a+b+1-k) C(a+b,k) = qualcosa‘
```

13 Case study: $\sum_k \binom{n}{k}^2 = \binom{2n}{n}$

Questo è il goal che si vuole dimostrare:

```
‘sum (:int) (λk. (λn k. C(n,k)^2) n k) = (λj. C(2j,j)) n‘
```

Il primo passo è applicare l'enunciato del teorema WZ. Il sistema chiede dunque di verificare le ipotesi del teorema, istanziate sull'identità che si sta dimostrando.

La prima ipotesi da verificare riguarda il supporto della funzione a secondo membro: deve essere un intervallo di numeri. Per questo si utilizza il seguente lemma, dimostrato a parte:

```
⊢ support (+) (λi. C(2i,i)) (:int) = {x | 0 ≤ x}
```

Dopo aver riscritto con questo lemma, le procedure automatiche di HOL Light per la decisione di aritmetica semplice riescono a portare a termine la dimostrazione del subgoal.

Il teorema WZ prevede che adesso si esibisca la funzione di certificante $G(n,k)$, e grazie all'utilizzo del Computer Algebra System Maxima, trovare questa la funzione non rappresenta un problema.

I due successivi subgoals sono molto simili tra loro: si tratta di dimostrare che $F(n,k)$ e $G(n,k)$ hanno supporto finito nella variabile k. Queste condizioni sono verificate per motivi che variano caso per caso; generalmente, però, ci si riconduce sempre al fatto che i coefficienti binomiali hanno supporto finito. Dimostrando a parte questo lemma e usando le proprietà delle funzioni a supporto finito, queste condizioni si verificano in poche righe.

La prossima ipotesi da verificare è la condizione di telescopicità:

```
‘¬(C(2*n,n) = 0) ⇒ ∀k. C(n+1,k)^2 / C(2*n+1+1,n+1) − C(n,k)^2 / C(2*n,n) = −(3n−2(k+1)+3)C(n,k)^2 / (2(2n+1)C(2*n,n)) − −(3n−2k+3)C(n,k−1)^2 / (2(2n+1)C(2*n,n))‘
```

Per questo proposito si è sviluppato un modo standard di procedere, cercando di eseguire operazioni che siano applicabili a tutte le dimostrazioni di questo tipo, in modo che le semplificazioni effettuate possano venire, in futuro, automatizzate ed eseguite dal sistema HOL Light senza l'intervento dell'utente:

1. Il primo passo da fare è convertire l'ipotesi $\binom{2n}{n} \neq 0$ in condizioni esplicite su n, ossia $n \geq 0$.

2. Adesso si applicano le riscritture di cui al paragrafo 12. Come spiegato al paragrafo 12, queste semplificazioni producono nuovi subgoals, ognuno con le opportune ipotesi sui denominatori introdotti; spesso le ipotesi aggiunte da riscritture successive (ne viene eseguita una per ogni coefficiente binomiale presente nelle espressioni processate dalla precedente) sono in contraddizione tra loro. In questi casi il subgoal può essere eliminato semplicemente usando le procedure automatiche di deduzione, ed è proprio quello che si fa, con l'accortezza di effettuare questa "pulizia" ad ogni semplificazione, e non dopo che tutte le riscritture sono eseguite: in questo modo si riduce la crescita dei subgoals aumentando l'efficienza di questa strategia di calcolo.

3. Quando tutte le riscritture necessarie sono state eseguite ci si trova di fronte a varie espressioni, in ognuna delle quali i binomiali sono presenti solo in "forma base". A questo punto l'obiettivo è convertire l'equazione da una combinazione di funzioni razionali a una uguaglianza di polinomi, dopo aver discusso i denominatori. Si presentano due casi tipici: quello in cui sono presenti tutti i coefficienti binomiali di partenza, che possono essere quindi semplificati, e quello in cui le riscritture condizionali eseguite in precedenza hanno sostituito qualche binomiale con 0 oppure 1. In quest'ultima situazione si dovranno cercare tra le ipotesi le condizioni su n e k per cui questo è successo, e usarle per calcolare esplicitamente i binomiali rimasti. Ecco due esempi che illustrano queste situazioni, provenienti dalla dimostrazione che stiamo illustrando:

```
0 [‘¬(¬(0 ≤ 2 * n) ∨ ¬(0 ≤ n) ∨ 2 * n < n)‘]
1 [‘¬(n + 1 = (2 * n + 1) + 1)‘]
2 [‘¬((2 * n + 1) + 1 - (n + 1) = 0)‘]
3 [‘¬(k = n + 1)‘]
4 [‘¬(n + 1 - k = 0)‘]
5 [‘¬(n + 1 = 2 * n + 1)‘]
6 [‘¬((2 * n) + 1 - (n + 1) = 0)‘]
7 [‘¬(n + 1 = 0)‘]

‘(((n+1)/(n+1−k) C(n,k))^2) / ((2n+1)+1)/((2n+1)+1−(n+1)) (2n+1)/(2n+1−(n+1)) (2n−n)/(n+1) C(2*n,n)  −  C(n,k)^2 / C(2*n,n)  =
```

$\frac{-(3n-2(k+1)+3)\binom{n}{k}^2}{2(2n+1)\binom{2n}{n}} - \frac{-(3n-2k+3)(\frac{k}{n+1-k}\ \binom{n}{k})^2}{2(2n+1)\binom{2n}{n}}$ ‘

In questo caso sono presenti tutti i binomiali, ed è la situazione più semplice da trattare. Innanzi tutto occorre ridurre le funzioni razionali dell'espressione alla forma numeratore/denominatore. Successivamente l'applicazione del seguente teorema con la regola di inferenza Modus Ponens converte questo goal in una uguaglianza di polinomi con la condizione che i denominatori siano non nulli:

```
⊢ ∀a b c d e f g h.
        ¬(b = 0) ∧ ¬(d = 0) ∧ ¬(f = 0) ∧ ¬(h = 0) ∧
        (a * d - b * c) * f * h = (e * h - g * f) * b * d
        ⇒  a / b - c / d = e / f - g / h
```

Il goal che ne risulta viene dimostrato autonomamente dal sistema.

4. Il secondo tipo di goal da affrontare è quello che presenta i casi limite delle riscritture condizionali, per i quali i valori di n e k sono espliciti e le identità possono essere verificate per calcolo diretto. Per procedere si effettua la conversione vista nel caso precedente, ma stavolta il sistema riesce a condurre in autonomia solo la discussione dei denominatori; per la verifica dell'espressione occorre l'intervento dell'utente, che deve segnalare quali sono i parametri noti e che valore hanno. In futuro si pensa di automatizzare questo procedimento.

```
0 [‘¬(¬(0 ≤ 2 * n) ∨ ¬(0 ≤ n) ∨ 2 * n < n)‘]
1 [‘¬(n + 1 = (2 * n + 1) + 1)‘]
2 [‘¬((2 * n + 1) + 1 - (n + 1) = 0)‘]
3 [‘k = n + 1‘]
4 [‘0 ≤ n + 1‘]
5 [‘n + 1 = 2 * n + 1‘]
6 [‘0 ≤ 2 * n + 1‘]
7 [‘0 ≤ n‘]
```

‘$\frac{1^2}{\frac{(2n+1)+1}{(2n+1)+1-(n+1)}\ \cdot\ 1} - \frac{\binom{n}{k}^2}{\binom{2n}{n}} = \frac{-(3n-2(k+1)+3)\binom{n}{k}^2}{2(2n+1)\binom{2n}{n}} - \frac{-(3n-2k+3)\ \cdot\ 1^2}{2(2n+1)\binom{2n}{n}}$‘

Rimane da trattare il caso in cui il secondo membro è nullo, per il quale la verifica dell'identità si fa per calcolo diretto.

L'ultimo subgoal chiede di esibire un n nel supporto della funzione a secondo membro per il quale l'identà è valida (si tratta dell'innesco per l'induzione): il valore in questione è tipicamente 0 e la condizione, nel caso in esame, è facilmente verificabile.

14 Difficoltà

Segue un elenco dei principali ostacoli da superare per ottenere una procedura di verifica automatica del certificato prodotto dal metodo Gosper/WZ:

Conversioni tra numeri. In HOL un numero intero non può essere usato al posto di un numero reale senza una esplicita conversione. Lo stesso vale tra numeri naturali e reali, e tra naturali e interi. Le procedure automatiche per l'aritmetica in HOL non implementano queste conversioni; per esempio non riescono a dimostrare identità come `real_of_num (n+1) = real_of_num n + real_of_num 1` (`real_of_num` è la conversione da numero naturale a reale). Questo provoca qualche complicazione, perché si è costretti ad applicare continuamente teoremi come

```
⊢ ∀x y. x + y = int_of_real (real_of_int x + real_of_int y)
⊢ ∀m n. real_of_num m = real_of_num n ⇔ m = n
⊢ ∀m n. real_of_num m + real_of_num n = real_of_num (m + n)
```

È stato anche necessario scriverne uno inedito, che spesso è risultato indispensabile:

```
⊢ ∀n k. int_of_num n = k ⇔ real_of_num n = real_of_int k
```

Riscritture condizionali. Per semplificare i coefficienti binomiali si sono sviluppate delle riscritture condizionali, ma queste non sono efficienti: producono un numero di subgoals esponenziale rispetto al numero di volte che vengono applicate. In pratica, se nell'identità di partenza compaiono più di tre coefficienti binomiali il tempo di esecuzione della procedura che elimina i subgoals inconsistenti generati da queste riscritture è eccessivamente lungo.

Analisi caso per caso. L'equazione 1 non è l'unica condizione da verificare affinché il certificato dell'identità sia valido; le altre ipotesi da soddisfare necessitano di un'analisi caso per caso, che ancora non si è riusciti a rendere automatica. Inoltre non si può conoscere a priori la struttura dei binomiali che compaiono nella 1, quindi è difficile prevedere quali riscritture applicare alla 1, e quante volte applicarle: per $\binom{n+1}{k}$ basta una sola applicazione di `TOP_STEP`, ma per $\binom{2(n+1)}{k}$ ne servono due, e così via.

15 Conclusioni

In questo lavoro si è formalizzato in HOL il teorema principale della "teoria WZ", e lo si è usato per dimostrare in HOL alcune identità binomiali. La natura del problema ha permesso di utilizzare un software di calcolo simbolico esterno, Maxima, per ottenere una sorta di "certificato di validità" delle identità in esame, cioè una particolare funzione su cui verificare alcune proprietà corrisponde a dimostrare l'identità di partenza. Questa verifica è stata fatta, negli esempi studiati, col theorem prover HOL Light. L'analisi dei casi studiati ha mostrato che è possibile incrementare l'automazione della verifica del certificato, sfruttando al meglio la potenza di calcolo delle procedure di deduzione automatica integrate in HOL Light e intervenendo laddove queste non riescono ad usare convenientemente le ipotesi (tipicamente, nella discussione dei denominatori).

Riferimenti bibliografici

[1] Wilf, Herbert S.; Zeilberger, Doron: *Rational functions certify combinatorial identities.* J. Amer. Math. Soc. 3 (1990), no. 1, 147–158.

[2] Zeilberger, Doron: *A holonomic systems approach to special functions identities.* J. Comput. Appl. Math. 32 (1990), no. 3, 321–368.

[3] Wilf, Herbert S.; Zeilberger, Doron: *An algorithmic proof theory for hypergeometric (ordinary and q) multisum/integral identities.* Invent. Math. 108 (1992), no. 3, 575–633.

[4] Wilf, Herbert S.: *generatingfunctionology.* Third edition. A K Peters, Ltd., Wellesley, MA, 2006. x+245 pp. ISBN: 978-1-56881-279-3; 1-56881-279-5

[5] Zeilberger, Doron: *Closed form (pun intended!). A tribute to Emil Grosswald: number theory and related analysis*, 579–607, Contemp. Math., 143, Amer. Math. Soc., Providence, RI, 1993.

[6] Petkovšek, Marko; Wilf, Herbert S.; Zeilberger, Doron: *$A = B$. With a foreword by Donald E. Knuth. With a separately available computer disk.* A K Peters, Ltd., Wellesley, MA, 1996. xii+212 pp. ISBN: 1-56881-063-6

[7] Gosper, R. William, Jr.: *Decision procedure for indefinite hypergeometric summation.* Proc. Nat. Acad. Sci. U.S.A. 75 (1978), no. 1, 40–42.

[8] Caruso, Fabrizio: *A Macsyma implementation of Zeilberger's fast algorithm.* Sém. Lothar. Combin. 43 (1999), Art. S43c, 8 pp. (electronic).

[9] Harrison, John; Théry, L.: *A skeptic's approach to combining HOL and Maple.* J. Automat. Reason. 21 (1998), no. 3, 279–294.

[10] *Maxima Manual*,
`http://maxima.sourceforge.net/docs/manual/en/maxima.html`

[11] *The Maxima mailing list*,
`http://maxima.sourceforge.net/maximalist.html`

[12] Harrison, John: *The HOL Light 2.20 Reference Manual*,
`http://www.cl.cam.ac.uk/~jrh13/hol-light/reference_220.html`

[13] Harrison, John: *The HOL Light 2.20 Tutorial*,
`http://www.cl.cam.ac.uk/~jrh13/hol-light/tutorial_220.pdf`

[14] Harrison, John: *The HOL Light 1.00 User Manual*,
`http://www.cl.cam.ac.uk/~jrh13/hol-light/manual-1.1.pdf`

[15] *The hol-info mailing list*,
`https://lists.sourceforge.net/lists/listinfo/hol-info`