

# spreadtab

v0.2a

## Hướng dẫn sử dụng

Christian TELLECHEA  
[unbonpetit@gmail.com](mailto:unbonpetit@gmail.com)

Dịch bởi Lê Hữu Điền Khuê  
[huudienkhue.le@gmail.com](mailto:huudienkhue.le@gmail.com)

Ngày 25 tháng 1 năm 2010

---

### *Tóm tắt*

Gói này cho phép thực hiện những chức năng của một bảng tính trong mọi môi trường bảng của L<sup>A</sup>T<sub>E</sub>X.

Chức năng chính của nó là "công thức hóa" một ô trong bảng bằng cách tham chiếu đến các ô khác, tính giá trị của các ô chứa công thức và hiển thị kết quả của chúng.

---

# Mục lục

<b>1</b>	<b>Mở đầu</b>	<b>2</b>
1.1	Giới thiệu . . . . .	2
1.2	Động lực chính . . . . .	2
<b>2</b>	<b>Các chức năng thông thường</b>	<b>3</b>
2.1	Vị trí tuyệt đối . . . . .	3
2.2	Vị trí tương đối . . . . .	4
2.3	Ô chứa văn bản . . . . .	4
2.4	Ô hỗn hợp . . . . .	4
2.5	Sao chép công thức . . . . .	5
<b>3</b>	<b>Định dạng bảng</b>	<b>7</b>
3.1	Trở lại với hàng và đường kẻ ngang . . . . .	7
3.2	Ẩn một hàng hay một cột . . . . .	8
3.3	Lưu giá trị của một ô . . . . .	8
3.4	Sử dụng <code>\multicolumn</code> . . . . .	9
3.5	Gói <code>fp</code> . . . . .	9
3.6	Dấu thập phân . . . . .	10
<b>4</b>	<b>Macro hàm</b>	<b>10</b>
4.1	Macro hàm toán học . . . . .	11
4.1.1	Hàm tổng . . . . .	11
4.1.2	Hàm giai thừa . . . . .	11
4.1.3	Macro hàm <code>sumprod</code> . . . . .	11
4.1.4	Số ngẫu nhiên . . . . .	12
4.2	Các macro hàm kiểm tra . . . . .	12
<b>5</b>	<b>Một số lưu ý đặc biệt</b>	<b>13</b>
5.1	Định nghĩa lại lệnh vẽ đường kẻ ngang . . . . .	13
5.2	Sử dụng cùng lúc <code>\multicolumn</code> và <code>\SThidecol</code> . . . . .	14
5.3	Các thông báo lỗi . . . . .	15
5.4	Sửa lỗi . . . . .	15
<b>6</b>	<b>Một số ví dụ</b>	<b>17</b>
6.1	Lại là tam giác Pascal . . . . .	17
6.2	Chuỗi hội tụ . . . . .	17
6.3	Hội tụ đến tỉ số vàng . . . . .	18
6.4	Bảng hóa đơn . . . . .	19
6.5	Ô vuông ma thuật . . . . .	19
6.6	Kim tự tháp tổng . . . . .	20

# 1 Mở đầu

## 1.1 Giới thiệu

Mục đích của **spreadtab** là cho phép viết các công thức toán học vào một hay nhiều ô của một bảng mà trong các công thức này có chứa giá trị của những ô khác, hoàn toàn giống như khi tiến hành trên một bảng tính. Gói này sẽ tính các công thức theo một thứ tự thích hợp và hiển thị bảng chứa các giá trị được tính.

Gói này cần đến  $\varepsilon$ -TeX, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> cũng như gói **fp** để thực hiện việc tính toán. Gói **xstring** cũng được đòi hỏi (bản v1.5c [2009/06/05] hoặc mới hơn).

Ngay từ đầu tác giả đã mong muốn rằng **spreadtab** sẽ tương thích với *mọi* môi trường bảng mà ở đó các cột được phân chia bởi **&** và việc bắt đầu một hàng mới được thực hiện nhờ **\**. Điều này bắt buộc tác giả phải lập trình **spreadtab** hoàn toàn độc lập với môi trường bảng. Do vậy, **spreadtab** sẽ thực hiện việc "đọc" bảng và tính các công thức *trước khi* môi trường bảng làm nhiệm vụ của nó.

Như vậy, **spreadtab** sẽ thực hiện ba bước:

- đầu tiên, **spreadtab** đọc nội dung của bảng và chia nó thành hàng rồi thành ô (**spreadtab** sẽ tự nhận thấy các ô có công thức);
- tiếp theo nó sẽ thực hiện việc tính các công thức của các ô (các ô dùng để tính một ô nào khác sẽ được tính trước). Thứ tự của các ô được tính sẽ do **spreadtab** xác định. Các phép tính sẽ được thực hiện bởi gói **fp**;
- cuối cùng, nó sẽ xây dựng lại bảng bằng cách thay các công thức bằng giá trị đã tính của chúng và sau đó nhường việc hiển thị bảng cho môi trường bảng (tùy chọn bởi người sử dụng).

Cú pháp:

```
1 \begin{spreadtab}{<Tên của môi trường bảng><tham số của môi trường>}
2   bảng chứa công thức và số
3 \end{spreadtab}
```

và sau khi **spreadtab** thực hiện công việc của mình, bảng sẽ được hiển thị giống như khi ta thực hiện cú pháp:

```
1 \begin{<Tên của môi trường bảng><tham số của môi trường>
2   bảng chứa số
3 \end{<Tên của môi trường bảng>}
```

Việc thực hiện ba bước trên sẽ mất thời gian, hơn nữa **fp** chậm trong việc tính toán, do đó việc biên dịch bảng với **spreadtab** sẽ *chậm hơn rất nhiều* so với việc biên dịch một bảng bình thường.

Cần lưu ý rằng **spreadtab** *không thể thay thế một bảng tính*. Một điều hơi khó chịu đó là nó không trực quan, nhất là khi làm việc với những bảng lớn và phức tạp, và cú pháp không mấy đơn giản của **spreadtab** cũng sẽ là một trở ngại nữa. Ưu điểm của **spreadtab** là có thể thực hiện việc tính toán *ngay trong mã nguồn* nhận được khi chuyển đổi sang mã L<sup>A</sup>T<sub>E</sub>X một bảng của một chương trình bảng tính<sup>1</sup>. Chúng ta cũng nên tránh các khuyết điểm của các chương trình chuyển đổi nói trên như: việc định dạng bảng cần phải chỉnh sửa lại nếu muốn nhận được bảng hoàn toàn giống như trong chương trình bảng tính, không tương thích với mọi môi trường bảng, chỉ nhận được các giá trị hay kết quả (các công thức đều bị mất) sau khi chuyển đổi, cần phải tiến hành chuyển đổi lại nếu thay đổi một giá trị hay một công thức trong bảng.

## 1.2 Động lực chính

Một vài tháng trước khi viết **spreadtab**, Derek O'CONNOR đã nhận xét với tác giả rằng vào thời điểm đó không có một gói nào của L<sup>A</sup>T<sub>E</sub>X cho phép tính toán các công thức trong các bảng, tương tự như một

<sup>1</sup>Có hai chương trình chính cho phép thực hiện điều này : **cac12latex** đối với calc của Open Office, và **excel12latex** đối với excel của Microsoft Office.

bảng tính. Nhận thấy rằng đây là một thách thức thú vị, tác giả bắt tay vào viết `spreadtab`, cũng chỉ như làm một bài tập lập trình vậy.

Thuật toán chính trong `spreadtab` xác định thứ tự mà các ô được tính, nó không phải là "chuyên môn" của lập trình  $\text{\LaTeX}$ . Những thuật toán khác mang hơi hướng của  $\text{\LaTeX}$  nhiều hơn như phân tích các ô và các công thức để tìm ra những tham chiếu đến những ô khác. Đây đúng là một bài tập thú vị, bởi vì trên phương diện lập trình,  $\text{\TeX}$  không hề cung cấp cho chúng ta một thứ gì đó tương tự như con trỏ để xây dựng các danh sách liên kết của các ô cần tính.

Con đường để đi đến phiên bản đầu tiên này của `spreadtab` đúng là dài. Tác giả xin cảm ơn Christophe CASSEAU đã quan tâm ngay từ đầu và cho nhiều góp ý bổ ích, cũng như gần đây là Derek O'CONNOR đã giúp đỡ và trao đổi với tác giả. Đồng thời cũng xin cảm ơn Lê Hữu Điền Khuê đã dịch tài liệu này sang tiếng Việt.

## 2 Các chức năng thông thường

### 2.1 Vị trí tuyệt đối

Trong một bảng, các ô được định vị bởi vị trí tuyệt đối của chúng như sau:

- mỗi cột là một chữ cái từ **a** đến **z**, **a** là cột đầu tiên bên trái: như vậy số cột tối đa là 26 (đối với hầu hết các trường hợp thì như thế là quá đủ); `spreadtab` sẽ không phân biệt các chữ cái hoa và chữ cái thường;
- tiếp ngay sau chữ cái, một số nguyên dương sẽ biểu diễn số thứ tự của hàng, hàng số 1 là hàng đầu tiên trên cùng.

Một số ví dụ về cách viết vị trí tuyệt đối: "**b4**", "**C1**" hay "**d13**". Một cách trực quan, chúng ta có thể minh họa bằng bảng sau đây

	A	B	C	D	E
1					
2					
3					
4					
5					

Trong ví dụ sau, ta sẽ tính tổng của mỗi hàng và của mỗi cột và sau đó là tổng của tất cả các ô:

1	<code>\begin{spreadtab}{\tabular}{rr r}}</code>			
2	22	& 54	& a1+b1 \\	22 54   76
3	43	& 65	& a2+b2 \\	43 65   108
4	49	& 37	& a3+b3 \\	49 37   86
5	<code>\hline</code>			
6	a1+a2+a3	& b1+b2+b3	& a4+b4	114 156   270
7	<code>\end{spreadtab}</code>			

Còn ví dụ sau sẽ tính một số hàng của tam giác Pascal:

1	<code>\begin{spreadtab}{\tabular}{ccccc}}</code>					1
2	1	&	&	&	\\	1 1
3	a1	& a1	&	&	\\	1 2 1
4	a2	& a2+b2	& b2	&	\\	1 3 3 1
5	a3	& a3+b3	& b3+c3	& c3	\\	1 4 6 4 1
6	a2	& a4+b4	& b4+c4	& c4+d4	& d4	
7	<code>\end{spreadtab}</code>					

<sup>2</sup>Cách đánh số này tương tự như trong các bảng tính : chữ cái là thứ tự của cột và số là thứ tự của hàng.

## 2.2 Vị trí tương đối

Để tham chiếu đến một ô, sẽ rất thuận tiện nếu ta xác định vị trí của nó so với ô chứa công thức. "Vị trí tương đối" sẽ giúp ta làm điều này. Vị trí tương đối được cho bởi cú pháp  $[x,y]$ , nó sẽ tham chiếu đến ô thứ  $x$  theo chiều ngang và thứ  $y$  theo chiều dọc đối với ô chứa vị trí tương đối, tức chứa công thức (tạm gọi là ô hiện tại). Chẳng hạn  $[-2,3]$  sẽ tham chiếu đến ô ở vị trí thứ 2 về *phía trái* và vị trí thứ 3 *phía dưới* của ô hiện tại (nghĩa là nếu xuất phát từ ô hiện tại, để đi đến được ô  $[-2,3]$ , ta phải di chuyển sang trái 2 ô, sau đó di chuyển xuống phía dưới 3 ô).

Ví dụ dưới đây xây dựng một bảng giống như ví dụ trước nhưng sử dụng vị trí tương đối và môi trường "matrix" của gói **amsmath**:

```

1 $
2 \begin{spreadtab}{{matrix}}{}
3 1\\
4 [0,-1] & [-1,-1]\\
5 [0,-1] & [-1,-1]+[0,-1] & [-1,-1]\\
6 [0,-1] & [-1,-1]+[0,-1] & [-1,-1]+[0,-1] & [-1,-1]\\
7 [0,-1] & [-1,-1]+[0,-1] & [-1,-1]+[0,-1] & [-1,-1]+[0,-1] & [-1,-1]\\
8 \end{spreadtab}
9 $

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

Chúng ta thấy rằng đối với việc xây dựng bảng này, việc sử dụng vị trí tương đối là thích hợp hơn bởi vì chúng ta chỉ sử dụng đúng hai vị trí tương đối:  $[0,-1]$  tham chiếu đến ô ngay phía trên và  $[-1,-1]$  tham chiếu đến ô gần nhất ở phía trái-trên (hay phía Tây-Bắc trong ngôn ngữ bản đồ) so với ô chứa công thức.

Chúng ta cũng có thể sử dụng cả vị trí tuyệt đối lẫn tương đối trong cùng một công thức.

## 2.3 Ô chứa văn bản

Nếu muốn chèn văn bản vào một ô, chúng ta cần phải cho **spreadtab** biết rằng ô này sẽ không được tính. Chỉ cần thêm vào kí tự "@" ở một vị trí nào đó trong nội dung của ô, ô này sẽ được **spreadtab** bỏ qua và sẽ trở thành một ô "trơ": nó sẽ không thể được tham chiếu đến từ bất kì ô nào trong bảng.

Sau đây là một ví dụ:

```

1 \begin{spreadtab}{{tabular}{|r|ccc|}}
2 \hline
3 @ giá trị của $x$ & -5 & & -1 & & 4 \\
4 @ $f(x)=2x$ & & 2*[0,-1] & & 2*[0,-1] & & 2*[0,-1] \\
5 \end{spreadtab}

```

giá trị của $x$	-5	-1	4
$f(x) = 2x$	-10	-2	8

Ta có thể thay "@" bằng một kí tự (hoặc một chuỗi các kí tự) khác, chỉ cần định nghĩa lại lệnh **\STtextcell** theo cú pháp

**\renewcommand\STtextcell{kí tự hoặc chuỗi kí tự}**

Chẳng hạn nếu ta định nghĩa **\renewcommand\STtextcell{toto}** thì khi đó, mọi ô chứa "toto" đều được xem như những ô chứa văn bản.

Ngoài ra, ô trống cũng được **spreadtab** xem như là ô chứa văn bản.

## 2.4 Ô hỗn hợp

Trên thực tế, mỗi ô được tạo nên bởi *hai* vùng : *vùng số* chứa công thức và *vùng chữ* chứa văn bản (vùng này sẽ được fp bỏ qua khi tính toán):

- trong một ô, nếu không có gì đặc biệt, toàn bộ ô sẽ được xem giống như một vùng số, và vùng chữ là rỗng (xem ví dụ về tam giác Pascal ở phần trên của tài liệu);
- nếu ô có chứa "@" thì toàn bộ ô sẽ được xem như một vùng chữ, vùng số là rỗng;
- nếu ô có chứa ":" thì đối số nằm giữa hai dấu ngoặc nhọn theo sau là vùng số, và những thứ còn lại là vùng chữ. Ô này sẽ có cấu trúc như sau:

<vùng chữ>:={vùng số}<tiếp theo của vùng chữ>

Chúng ta có thể thay ":" bởi một hay một chuỗi các kí tự khác, chẳng hạn "toto", bằng cách định nghĩa lại macro `\STnumericfieldmarker` như sau:

```
\renewcommand\STnumericfieldmarker{toto}
```

Một khi vùng số đã được tính, nó sẽ được thay thế bởi giá trị của nó trong bảng.

Cần lưu ý rằng ":{vùng số}" có thể nằm giữa hai dấu ngoặc nhọn. Ví dụ, trong một ô ta có thể viết `\textbf{:{a1+1}}`. Nếu vùng số của ô a1 là 5 thì ô chứa công thức cuối cùng sẽ chứa giá trị là `\textbf{6}`.

Ví dụ đơn giản sau đây sẽ giúp các bạn hiểu rõ hơn:

```
1 \begin{spreadtab}{{\tabular}{|c|c||c|}}\hline
2 giá trị 1 : :={50} & giá trị 2 : :={29} & trung bình : \textbf{:{(a1+b1)/2}}\\ \hline
3 \end{spreadtab}
```

giá trị 1 : 50	giá trị 2 : 29	trung bình : <b>39.5</b>
----------------	----------------	--------------------------

Cũng cần lưu ý rằng ":{}" định nghĩa một công thức rỗng, nghĩa là ô được cho chỉ chứa văn bản như khi ta sử dụng "@". Tuy nhiên, ":{}" và "@" không hoàn toàn tương đương với nhau, bởi vì ô chứa ":{}" có thể "đọc" được các công thức sao chép (xem mục tiếp theo), còn với "@" thì không thể.

## 2.5 Sao chép công thức

Để tránh phải sao chép nhiều lần các công thức giống nhau, `spreadtab` cung cấp cho ta lệnh `\STcopy`.

Lệnh này có cú pháp như sau:

`\STcopy{>x,vy}{công thức}`

trong đó  $x$  và  $y$  là các số nguyên dương xác định "mảng ô"<sup>3</sup> sẽ nhận được công thức sao chép: mảng này được xác định bởi ô chứa cú pháp trên và ô nằm vị trí thứ  $x$  về phía phải và vị trí thứ  $y$  về phía dưới của nó<sup>4</sup>. Lệnh `\STcopy` không được đặt trong các ô có chứa vùng số (nghĩa là có ":").

Việc sao chép sẽ được bắt đầu từ ô chứa lệnh và sẽ thay đổi dần những chữ cái biểu diễn số thứ tự cột và những số biểu diễn số thứ tự hàng trong công thức, tùy theo vị trí của ô sẽ chứa công thức sao chép trong mảng. Ví dụ, nếu một ô chứa công thức `a1+b2+c3` và công thức này được sao chép trong ô thứ 2 về phía phải và thứ 5 về phía dưới của ô đó, khi đó công thức này sẽ trở thành `c6+d7+e8`. Công thức được sao chép cũng có thể chứa vị trí tương đối và các vị trí này sẽ không thay đổi trong quá trình sao chép.

Bằng cách đặt một dấu "!" trước "tọa độ" (số hay chữ cái) của một vị trí, tọa độ này sẽ không thay đổi trong suốt quá trình sao chép. Ta hãy lấy ví dụ ở trên với công thức `a!1+!b2+!c!3`. Nếu công thức này được sao chép trong ô thứ 2 về phía phải và thứ 5 về phía dưới của ô đó, khi đó công thức này sẽ trở thành `c1+d7+c3`. Chức năng này cũng có hiệu lực với vị trí tương đối. Chẳng hạn, nếu một ô chứa công thức `[-1,!-1]+[-1,1]+[!1,!2]` và công thức này được sao chép trong ô thứ 2 về phía phải và thứ 5 về phía dưới của ô đó, khi đó nó trở thành: `[-1,-6]+[-3,1]+[-1,-3]`.

Chúng ta cũng có thể thay dấu chấm than bởi một kí tự khác nhờ lệnh

<sup>3</sup>Xem định nghĩa về mảng ô mục 4.1.1 trang 11.

<sup>4</sup>Cách xác định này tương tự như khái niệm "Vị trí tương đối", xem mục 2.2 trang 4. Như vậy, việc sao chép chỉ có thể được thực hiện đối với các ô ở phía phải hay phía dưới của ô chứa lệnh sao chép.

`\renewcommand\STtransposecar{<kí tự>}`.

Trong cú pháp `\STcopy{>x,vy}{công thức}`, nếu tham số  $x$  vắng mặt thì việc sao chép theo chiều ngang sẽ được thực hiện cho đến vị trí cuối cùng bên phải của bảng, còn nếu tham số  $y$  vắng mặt thì việc sao chép theo chiều dọc sẽ được thực hiện cho đến vị trí đáy bảng.

Sau đây là một vài ví dụ:

---

<code>{&gt;3,v1}</code>	sao chép cho đến ô thứ 3 bên phải và thứ 1 phía dưới
<code>{&gt;3}</code>	sao chép cho đến ô thứ 3 bên phải
<code>{v1}</code>	sao chép đến ô ngay phía dưới
<code>{&gt;}</code>	sao chép công thức cho tất cả các ô ở về phía phải và trên cùng một hàng của ô này
<code>{v}</code>	sao chép công thức cho tất cả các ô ở về phía dưới và trên cùng một cột của ô này
<code>{v,&gt;}</code>	sao chép công thức đến tất cả các ô ở về phía phải và phía dưới của ô này

---

Chúng ta có thể dễ dàng tạo ra bảng phép nhân từ 1 đến 10:

```

1 \begin{spreadtab}{\tabular}{|c|*{10}{c|}}
2 \hline
3 @${\times}$ & 1 & & & & & & & & & & \\
4 1 & \STcopy{>,v}{!a2*b!1} & & & & & & & & & & \\
5 \STcopy{v}{a2+1} & & & & & & & & & & & \\
6 & & & & & & & & & & & \\
7 & & & & & & & & & & & \\
8 & & & & & & & & & & & \\
9 & & & & & & & & & & & \\
10 & & & & & & & & & & & \\
11 & & & & & & & & & & & \\
12 & & & & & & & & & & & \\
13 & & & & & & & & & & & \\
14 \end{spreadtab}

```

×	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Khi sao chép một công thức, nếu một ô nào đó trong mảng được sao chép đã chứa một giá trị nào đó (nghĩa là chứa vùng số khác rỗng) thì việc sao chép sẽ không được thực hiện và ô này sẽ giữ nguyên giá trị của nó.

Nếu có nhiều lệnh `\STcopy` cùng thực hiện việc sao chép cho một ô nào đó thì công thức mà ô này nhận được sẽ là công thức tương ứng với lệnh `\STcopy` cuối cùng mà `spreadtab` đọc được (`spreadtab` đọc nội dung của bảng từ trái sang phải và từ trên xuống dưới). Trong ví dụ minh họa dưới đây, mảng sao chép tạo ra bởi `\STcopy` trong ô màu hồng sẽ không bao gồm mảng được tạo ra bởi `\STcopy` trong ô màu xanh, bởi vì `spreadtab` sẽ đọc ô màu hồng trước ô màu xanh:

	A	B	C	D	E	F
1	1	\STcopy {v,>}{!a1+1}				
2	2					
3	3		\STcopy {>2,v1}{!a3*10}			
4	4					
5	5					

Ngay dưới đây ta sẽ áp dụng ví dụ minh họa ở trên. Trong bảng này, ô b5 đã có sẵn một giá trị và ô c5 là một ô hỗn hợp có vùng số khác rỗng, chúng sẽ được giữ nguyên.

```

1 \begin{spreadtab}{{\tabular}{|*6{c|}}}\hline
2 1 & \STcopy{v,>}{!a1+1} & & & & \\\hline
3 2 & & & & & \\\hline
4 3 & & \STcopy{>2,v1}{!a3*10} & & & \\\hline
5 4 & & & & & \\\hline
6 5 & -1 & & a:={0}b & & \\\hline
7 \end{spreadtab}

```

1	2	2	2	2	2
2	3	3	3	3	3
3	4	30	30	30	4
4	5	40	40	40	5
5	-1	a0b	6	6	6

Như đã đề cập trong mục trước, ta có thể sao chép một công thức cho một ô chứa vùng số rỗng (nghĩa là chứa ":{ }"). Trong trường hợp này, công thức sẽ được sao chép đến vị trí của ":{ }". Ngược lại, ta không thể làm điều tương tự đối với các ô chứa "@" (nghĩa là chỉ chứa văn bản).

Ví dụ:

```

1 \begin{spreadtab}{{\tabular}{|*6{c|}}}\hline
2 1 & 2 & 3 & 4 & 5 & 6 \\\hline
3 X\STcopy{>}{a1+1}Y & @XY & X:={ }Y & \textbf{:={ }} & & \\\hline
4 \end{spreadtab}

```

1	2	3	4	5	6
X2Y	XY	X4Y	5	6	7

## 3 Định dạng bảng

### 3.1 Trở lại với hàng và đường kẻ ngang

Để xác định điểm kết thúc của một hàng, `spreadtab` bắt buộc phải nhận ra sự xuống hàng và các đường kẻ ngang. Gói này cho phép sử dụng thêm một tham số của `\\` như sau: `\\[<dimension>]`.

Chúng ta có thể sử dụng một số tùy ý các đường kẻ ngang sau:

- `\hline`;
- `\cline{x-y}` trong đó x và y là số thứ tự của cột xuất phát và cột đến của đường kẻ;
- `\hhline{<type>}` trong đó <type> là kiểu đường kẻ tùy chọn (xem tài liệu của gói `hhline`);
- một lệnh bất kì của gói `booktabs`: `\toprule`, `\midrule`, `\bottomrule`, `\cmidrule`, `\addlinespace`, `\morecmidrule` và `\specialrule`. Tất cả các tham số của các macro này đều được xử lí.
- `\noalign` và tham số của nó có thể được đặt sau `\\`.

Dưới đây là một ví dụ về tam giác Pascal "ngược":

```

1 \begin{spreadtab}{{\tabular}{*5{c}}
2 [0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1] \\\[1em]
3 [0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1] & \\
4 [0,1] & [-1,1]+[0,1] & [-1,1] & & \\
5 [0,1] & [-1,1] & & & \\
6 1 & & & & \\
7 \end{spreadtab}

```



1	4	6	4	1
1	3	3	1	
1	2	1		
1	1			
1				

### 3.2 Ẩn một hàng hay một cột

Thỉnh thoảng, nguyên một hàng hay một cột được sử dụng để chứa các phép tính trung gian mà ta muốn chúng không được hiển thị trong bảng. `spreadtab` sẽ giúp ta thực hiện điều này nhờ các lệnh `\SThiderow` (ẩn hàng) và `\SThidecol` (ẩn cột). Nếu chúng được đặt trong một ô nào đó, thì hàng hay cột chứa ô đó sẽ được ẩn đi.

Sau đây là một ví dụ:

```

1 \begin{spreadtab}{{\tabular}{|r|ccc|}}
2 \hline
3 @Giá trị của $x$ & -1 & 0 & \SThidecol & 2 & 3 & \\\hline
4 @$f(x)=2x-1$ & 2*[0,-1]-1 & 2*[0,-1]-1 & 2*[0,-1]-1 & 2*[0,-1]-1 & 2*[0,-1]-1 & \\
5 @$g(x)=x-10$ \SThiderow & [0,-2]-10 & [0,-2]-10 & [0,-2]-10 & [0,-2]-10 & [0,-2]-10 & \\
6 @$h(x)=1-x$ & 1-[0,-3] & 1-[0,-3] & 1-[0,-3] & 1-[0,-3] & 1-[0,-3] & \\\hline
7 \end{spreadtab}

```

Giá trị của $x$	-1	2	3
$f(x) = 2x - 1$	-3	3	5
$h(x) = 1 - x$	2	-1	-2

Chúng ta đã ẩn đi hàng chứa  $g(x)$  và cột chứa giá trị 0 (cột thứ ba).

Cần nhớ rằng môi trường bảng (trong ví dụ trên là `tabular`) sẽ không đọc được các hàng và các cột ẩn, điều đó giải thích tại sao trong phần định nghĩa môi trường bảng, chúng ta chỉ có 4 cột (`|r|ccc|`), trong khi đó `spreadtab` lại làm việc với 5 cột.

Để thấy sự khác biệt, chúng ta sẽ lấy lại ví dụ trên nhưng không ẩn một hàng hay cột nào:

Giá trị của $x$	-1	0	2	3
$f(x) = 2x - 1$	-3	-1	3	5
$g(x) = x - 10$	-11	-10	-8	-7
$h(x) = 1 - x$	2	1	-1	-2

### 3.3 Lưu giá trị của một ô

Nhiều lúc chúng ta cần giá trị của một ô để sử dụng bên ngoài một công thức hay thậm chí là bên ngoài của bảng. Ta sử dụng cú pháp sau:

`\STsavecell{<lệnh>}{<vị trí tuyệt đối>}`

Cú pháp này cho phép lưu giá trị của ô có <vị trí tuyệt đối> trong <lệnh>. *Chú ý:* `spreadtab` sẽ không kiểm tra là <lệnh> đã được định nghĩa trước đó hay chưa.

Chúng ta chỉ có thể sử dụng vị trí *tuyệt đối* bởi vì cú pháp trên được đặt trong phần tùy chọn của môi trường `spreadtab`.

Ta có thể lưu cùng một lúc nhiều ô, chỉ cần đặt trong phần tùy chọn một số tương ứng lệnh `\STsavecell`.

```

1 \begin{spreadtab}[\STsavecell\hhh{b3}\STsavecell\mmm{c3}\STsavecell\sss{d3}]{\tabular
2 }{|rc|}}\hline
3 @ Vận tốc (km/h) & \SThidecol & \SThidecol & \SThidecol & 35 & \\
4 @ Khoảng cách (km) & & & & 180 & \\\hline
5 @ Thời gian (h min s) & trunc(e2/e1,0) & trunc(60*(e2/e1-b3),0) & trunc(3600*(e2/e1-b3)
6 -60*c3,1) & @\hhh\ h \mmm\ min \sss\ s & \\\hline
7 \end{spreadtab} \par \medskip
8 Phải mất ít nhất \hhh\ giờ

```

Vận tốc (km/h)	35
Khoảng cách (km)	180
Thời gian (h min s)	5 h 8 min 34.2 s

Phải mất ít nhất 5 giờ

### 3.4 Sử dụng `\multicolumn`

Gói `spreadtab` tương thích với cú pháp

`\multicolumn{<n>}{<kiểu>}{<nội dung>}`

Cú pháp này cho phép hợp nhất `<n>` ô thành một ô có kiểu và nội dung được xác định trong phần tùy chọn.

Bảng dưới đây chứa một số ô đã được hợp nhất với nhau, các ô của nó sẽ có vị trí như sau đối với `spreadtab`:

a1	b1	c1	d1	e1	f1	g1
a2	b2		d2	e2	f2	g2
a3			d3	e3		g3

Như vậy, ô ở ngay sau ô hợp nhất sẽ có số thứ tự phụ thuộc vào số ô đã được hợp nhất.

Ở hàng cuối cùng, các ô `a3`, `b3` và `c3` được hợp nhất với nhau, và nếu ô `a3` chứa công thức thì các ô `b3` và `c3` sẽ *không tồn tại* đối với `spreadtab`: chúng ta không thể tham chiếu đến chúng từ một ô nào khác.

Trong ví dụ sau đây, mỗi số ở hàng trên sẽ bằng tích của hai số ngay phía dưới của nó:

```

1 \newcolumntype{K}[1]{@{}>\centering\arraybackslashp{#1cm}@{}}
2 \begin{spreadtab}{\tabular}{*6{K{0.5}}}}
3 \cline{2-5}
4 &\multicolumn{2}{|K{1}|}{:=a2*c2}} & \multicolumn{2}{|K{1}|}{:=c2*e2}} &\\hline
5 \multicolumn{2}{|K{1}|}{:=8}& \multicolumn{2}{|K{1}|}{:=7}& \multicolumn{2}{|K{1}|}{:=6}}\\hline
6 \end{spreadtab}

```

	56	42	
8	7	6	

### 3.5 Gói `fp`

Như đã nói ở trên, tất cả các phép tính đều được thực hiện bởi gói `fp` và macro `\FPeval`<sup>5</sup> của nó. Gói này cung cấp một khả năng tính toán rất mạnh cho `TeX` và còn được trang bị tất cả các hàm số thông dụng. Các phép tính được tính với độ chính xác đến  $10^{-18}$ , do đó kết quả sẽ được hiển thị đến 18 chữ số thập phân nếu nó không tròn ! Tất nhiên trong hầu hết các trường hợp, 18 chữ số là quá nhiều và ta muốn giảm bớt.

Có nhiều cách để giải quyết vấn đề này:

- có thể dùng gói `numprint` để quản lý việc hiển thị kết quả;
- sử dụng hàm `round(kết quả,n)` để làm tròn kết quả đến `n` chữ số thập phân;
- ta cũng có thể sử dụng macro `\STautoround` của `spreadtab`, tham số của nó là số chữ số thập phân cần hiển thị. Nếu không có tham số (mặc định), nó sẽ không làm tròn kết quả.

<sup>5</sup>Cả hai khái niệm "trung tố" (infix) và "hậu tố" (postfix) đều được chấp nhận bởi `\FPeval`, do đó các công thức trong `spreadtab` có thể được viết dưới cả hai dạng trên.

Ví dụ sau là bảng chứa các số nguyên từ 1 đến 7 và các nghịch đảo của chúng, được làm tròn đến  $10^{-6}$ :

```

1 \STautoround{6}
2 \begin{spreadtab}{\tabular}{|l|*7{>{\centering\arraybackslash}m{1.35cm}|}}
3 \hline
4 @x$ & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\\hline
5 @x^{-1}$ & 1/b1 & 1/c1 & 1/d1 & 1/e1 & 1/f1 & 1/g1 & 1/h1 & \\\hline
6 \end{spreadtab}

```

$x$	1	2	3	4	5	6	7
$x^{-1}$	1	0.5	0.333333	0.25	0.2	0.166667	0.142857

### 3.6 Dấu thập phân

Gói `fp` sử dụng dấu chấm là dấu thập phân theo mặc định. Trong việc hiển thị kết quả, chúng ta có thể thay thế dấu chấm bởi một kí tự khác nhờ cú pháp:

`\STsetdecimalsep{<kí tự>}`

Chẳng hạn để thay dấu chấm bởi dấu phẩy ta dùng:

`\STsetdecimalsep{,}`

Lưu ý là trong môi trường toán, dấu phẩy không được hiểu là dấu thập phân (mặc định là dấu chấm), do đó luôn có một khoảng trắng theo sau nó. Để xóa đi khoảng trắng này, ta có thể đặt dấu phẩy giữa hai dấu ngoặc nhọn như trong ví dụ sau:

<pre> 1 3,14 không hiển thị giống \$3,14\$. \par 2 3,14 hiển thị giống \$3{,}14\$ </pre>	<p>3,14 không hiển thị giống 3, 14.</p> <p>3,14 hiển thị giống 3,14</p>
--	---

Khi các ô được hiển thị trong môi trường toán, chúng ta có thể<sup>6</sup> thay dấu chấm bởi dấu phẩy giữa hai dấu ngoặc nhọn với lệnh `\STsetdecimalsep{,}`. Trong hai bảng dưới đây, ta thấy rằng khoảng trắng sau dấu phẩy trong bảng thứ hai đã biến mất:

```

1 \STsetdecimalsep{,}
2 \begin{spreadtab}{\tabular}{|*3{>{\$}r{<{\$}}|}\hline
3 @x & @y & @\text{Trung bình}\hline
4 5 & -4 & (a2+b2)/2\hline
5 -6.1 & -8 & (a3+b3)/2\hline
6 9.85 & 3.7 & (a4+b4)/2\hline
7 \end{spreadtab}\par\smallskip
8 \STsetdecimalsep{,}
9 \begin{spreadtab}{\tabular}{|*3{>{\$}r{<{\$}}|}\hline
10 @x & @y & @\text{Trung bình}\hline
11 5 & -4 & (a2+b2)/2\hline
12 -6.1 & -8 & (a3+b3)/2\hline
13 9.85 & 3.7 & (a4+b4)/2\hline
14 \end{spreadtab}

```

$x$	$y$	Trung bình
5	-4	0,5
-6,1	-8	-7,05
9,85	3,7	6,775

$x$	$y$	Trung bình
5	-4	0,5
-6,1	-8	-7,05
9,85	3,7	6,775

## 4 Macro hàm

Bởi vì tất cả các phép tính đều được thực hiện bởi gói `fp` nên các hàm số trong các công thức phải được "hiểu" bởi `fp`. Tuy số lượng các hàm số cung cấp bởi `fp` là rất lớn, nó vẫn không đủ để thực hiện các phép tính trong bảng. Đó là lí do tại sao cần phải xây dựng các macro hàm. Các macro này được tạo ra nhờ vào các hàm sẵn có của `fp`.

Nhận xét quan trọng: có thể lấy hàm hợp lẫn nhau của các macro hàm, và các tham số của chúng có thể chứa các phép toán cũng như hàm số của `fp`.

<sup>6</sup>Tuy nhiên sẽ tốt hơn nếu ta sử dụng `numprint` để thay đổi cách hiển thị kết quả. Ngoài ra chúng ta cũng có thể thay đổi code của dấu phẩy: `\mathcode'="013B\relax` (chỉ dành cho người dùng thông thạo).

## 4.1 Macro hàm toán học

### 4.1.1 Hàm tổng

Hàm **sum** cho phép lấy tổng của một hay nhiều mảng ô và có cú pháp:

**sum**(<mảng 1>;<mảng 2>;...;<mảng 3 n>)

trong đó một mảng :

- hoặc là một ô riêng biệt, chẳng hạn "a1" hay "[2,1]";
- hoặc là một miền chữ nhật giới hạn bởi một ô phía trên bên trái và một ô phía dưới bên phải, nó được nhập vào như sau: "<ô 1>:<ô 2>", với điều kiện là "<ô 1>" phải ở trước "<ô 2>" khi chúng ta đi từ trên xuống dưới và từ trái sang phải.

Ví dụ về mảng ô: "a2:d5", "[-1,-1]:[2,3]", "b4:[5,1]".

Trong các mảng, những ô trống hoặc chỉ chứa văn bản được xem như chứa giá trị 0. Các ô bị nhập vào bởi **\multicolumn** cũng vậy.

Vị trí tương đối và tuyệt đối có thể được sử dụng cùng nhau. Ví dụ sau đây tính tổng các hệ số của tam giác Pascal:

<pre> 1 \begin{spreadtab}{{\tabular}{*5c}} 2 \multicolumn{5}{c}{Tổng = :={sum(a2:e6)}}\\ 3 [0,1] &amp; [-1,1]+[0,1] &amp; [-1,1]+[0,1] &amp; [-1,1]+[0,1] &amp; [-1,1] &amp; \\ 4 [0,1] &amp; [-1,1]+[0,1] &amp; [-1,1]+[0,1] &amp; [-1,1] &amp; &amp; \\ 5 [0,1] &amp; [-1,1]+[0,1] &amp; [-1,1] &amp; &amp; &amp; \\ 6 [0,1] &amp; [-1,1] &amp; &amp; &amp; &amp; \\ 7 1 &amp; &amp; &amp; &amp; &amp; \\ 8 \end{spreadtab} </pre>	<p>Tổng=31</p> <table border="0"> <tr><td>1</td><td>4</td><td>6</td><td>4</td><td>1</td></tr> <tr><td>1</td><td>3</td><td>3</td><td>1</td><td></td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td>1</td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	4	6	4	1	1	3	3	1		1	2	1			1	1				1				
1	4	6	4	1																						
1	3	3	1																							
1	2	1																								
1	1																									
1																										

### 4.1.2 Hàm giai thừa

Macro hàm **fact**(<n>) cho phép tính giai thừa của tham số n, trong đó n là một số nguyên trong khoảng từ 0 đến 18<sup>7</sup>. n có thể là một tham chiếu đến một ô chứa một số nguyên.

Bảng sau chứa giai thừa của các số từ 0 đến 8:

```

1 \begin{spreadtab}{{\tabular}{*9c}}
2 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\\
3 fact(a1)&fact(b1)&fact(c1)&fact(d1)&fact(e1)&fact(f1)&fact(g1)&fact(h1)&fact(i1)
4 \end{spreadtab}

```

0	1	2	3	4	5	6	7	8
1	1	2	6	24	120	720	5040	40320

### 4.1.3 Macro hàm sumprod

Hàm **sumprod** cho phép lấy tích của các phần tử tương ứng của hai hay nhiều mảng, sau đó tính tổng của các tích này. Cú pháp:

**sumprod**(<mảng 1>;<mảng 2>;...;<mảng n>)

Tất cả các mảng của hàm phải có cùng kích thước.

Ví dụ đơn giản sau đây tính tuổi trung bình của một nhóm trẻ em từ 10 đến 15 tuổi:

<sup>7</sup>Đối với fp, số lớn nhất là 10<sup>18</sup> - 1. Giai thừa của 19 vượt quá số này.

```

1 \begin{spreadtab}{\tabular}{r*6c}}
2 @Tuổi & 10 & 11 & 12 & 13 & 14 & 15 \\
3 @Số lượng & 5 & 8 & 20 & 55 & 9 & 3 \\
4 @Trung bình & \multicolumn{6}{l}{:=\sumprod(b1:g1;b2:g2)/\sum(b2:g2)}}
5 \end{spreadtab}

```

Tuổi	10	11	12	13	14	15
Số lượng	5	8	20	55	9	3
Trung bình	12.64					

Cũng giống như macro hàm **sum**, các ô chứa văn bản hoặc được nhập bởi **\multicolumn** được xem như có giá trị 0.

#### 4.1.4 Số ngẫu nhiên

Macro hàm **randint** và **rand** trả về một số ngẫu nhiên.

Chú ý: giá trị ngẫu nhiên này phụ thuộc vào thời điểm biên dịch của tài liệu. Nếu chúng ta muốn nhận được những giá trị không phụ thuộc vào thời điểm biên dịch, cần định nghĩa lại macro **\ST@seed** và gán cho **\FPseed** một giá trị nguyên:

```

1 \makeatletter
2 \renewcommand\ST@seed{}
3 \makeatletter
4 \FPseed=27% chẳng hạn, hoặc một số nguyên bất kì nào khác

```

Macro hàm **randint** trả về một số nguyên phụ thuộc vào tham số. Cú pháp như sau:

**randint**([<số 1>,<số 2>])

trong đó <số 1> là một tham số nguyên (mặc định bằng 0). Số nguyên ngẫu nhiên được trả về sẽ nằm trong đoạn [<số 1>;<số 2>].

Macro hàm **rand**() trả về một số thập phân ngẫu nhiên giữa 0 và 1:

```

1 \STautoround{6}
2 \begin{spreadtab}{\tabular}{l|l|cccc|}\hline
3 @Số thuộc đoạn [0;1] & & \rand() & \rand() & \rand() & \rand() & \\
4 @Số thuộc đoạn [-5;5] & & \randint(-5,5) & \randint(-5,5) & \randint(-5,5) & \randint(-5,5) & \\
5 @Số thuộc đoạn [0;20] & & \randint(20) & \randint(20) & \randint(20) & \randint(20) & \\
6 \hline
7 \end{spreadtab}

```

Số thuộc đoạn [0;1]	0.482277	0.621581	0.911164	0.938429
Số thuộc đoạn [-5;5]	-3	2	5	-5
Số thuộc đoạn [0;20]	12	13	3	9

## 4.2 Các macro hàm kiểm tra

Có ba macro hàm kiểm tra, chúng có cú pháp như sau:

**ifeq**(<số 1>,<số 2>,<số 3>,<số 4>)

**ifgt**(<số 1>,<số 2>,<số 3>,<số 4>)

**iflt**(<số 1>,<số 2>,<số 3>,<số 4>)

Phép so sánh được thực hiện giữa <số 1> và <số 2>:

- kiểm tra đẳng thức với **ifeq**:  $\langle \text{số } 1 \rangle = \langle \text{số } 2 \rangle$  ?
- kiểm tra tính lớn hơn nghiêm ngặt với **ifgt**:  $\langle \text{số } 1 \rangle > \langle \text{số } 2 \rangle$  ?
- kiểm tra tính nhỏ hơn nghiêm ngặt với **iflt**:  $\langle \text{số } 1 \rangle < \langle \text{số } 2 \rangle$  ?

Nếu phép kiểm tra là đúng thì  $\langle \text{số } 3 \rangle$  được trả về, nếu không thì  $\langle \text{số } 4 \rangle$  được trả về.

Ví dụ sau đây xác định một vài giá trị của hàm số  $f(x) = \begin{cases} 10 & \text{nếu } x < 1 \\ 0 & \text{nếu } x = 1 \\ -10 & \text{nếu } x > 1 \end{cases}$

```

1 \begin{spreadtab}{\tabular{|*2c|}}\hline
2 @x$ & @f(x)$ \\ \hline
3 -0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\
4 [0,-1]+0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\
5 [0,-1]+0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\
6 [0,-1]+0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\
7 [0,-1]+0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\
8 [0,-1]+0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\
9 [0,-1]+0.5 & \iflt([-1,0],1,10,\ifeq([-1,0],1,0,-10))\\ \hline
10 \end{spreadtab}

```

$x$	$f(x)$
-0.5	10
0	10
0.5	10
1	0
1.5	-10
2	-10
2.5	-10

## 5 Một số lưu ý đặc biệt

### 5.1 Định nghĩa lại lệnh vẽ đường kẻ ngang

Chúng ta có thể thử định nghĩa một lệnh để vẽ một đường kẻ ngang kép chẳng hạn:

```
\newcommand\dline{\hline\hline}
```

rồi sau đó sử dụng nó trong một bảng tính số hạng của dãy Fibonacci:

0	1	2	3	4	5	6
1	1	2	3	5	8	13

Nhưng khi nhập vào đoạn mã sau

```

1 \newcommand\dline{\hline\hline}
2 \spreadtab{\begin{tabular}{*7c}}
3 {
4 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \dline
5 1 & 1 & a2+b2 & b2+c2 & c2+d2 & d2+e2 & e2+f2
6 }

```

và biên dịch thì hệ thống báo lỗi:

! Improper alphabetic constant.

Lí do rất đơn giản, đó là **spreadtab** không hiểu được lệnh **\dline** và do đó không xem nó như một đường kẻ ngang, như vậy nó được xem như thuộc ô đầu tiên của hàng tiếp theo. Với **spreadtab**, ô b1 chứa:

**\dline 1**

Do không có **@** hay **:=**{...}, **\FPeval** cố gắng tính giá trị của ô, và tất nhiên là thất bại!

Để có thể biên dịch mà không bị lỗi, cần phải có một công thức trong ô b1:

```

1 \newcommand\dline{\hline\hline}
2 \begin{spreadtab}{{\tabular}{*7c}}
3 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \dline
4 :=\{1\} & 1 & a2+b2 & b2+c2 & c2+d2 & d2+e2 & e2+f2
5 \end{spreadtab}

```

0	1	2	3	4	5	6
1	1	2	3	5	8	13

## 5.2 Sử dụng cùng lúc \multicolumn và \SThidecol

Trước hết, thông thường việc sử dụng cùng lúc \multicolumn và \SThiderow là không thể, và đa số người sử dụng sẽ không rơi vào hoàn cảnh này, và do đó, không cần đọc mục này.

Nhưng nếu bạn đọc nào muốn hiểu rõ vấn đề, chúng ta hãy cùng xem. Một cột được ẩn đi không được chứa một ô có lệnh \multicolumn, nhưng chuyện gì sẽ xảy ra nếu ta rơi vào trường hợp này ?

Nói chung, hệ thống sẽ không báo lỗi, nhưng các vị trí và tham chiếu sẽ bị xáo trộn ở hàng chứa \multicolumn...

Hãy lấy một ví dụ, trong bảng sau, ta sẽ hợp nhất các ô từ b2 đến h2 và ta muốn ẩn đi các cột c, d và f (màu xám):

a1	b1	c1	d1	e1	f1	g1	h1	i1	j1
a2	b2							i2	j2

Có 4 ô ẩn được hợp nhất, do đó ta sẽ nhập vào \multicolumn{4}, bởi vì khi đếm số ô để hợp nhất, ta sẽ trừ ra các ô đã được ẩn.

Bây giờ, nếu đếm 4 chữ cái kể từ chữ cái b ta sẽ đi đến chữ cái e: ta được "đoạn cột b-e". Đoạn này chứa 2 cột ẩn (c và d) và không chứa 1 cột ẩn khác (f). Hai giá trị này là rất quan trọng để hiểu được phần tiếp theo, trong trường hợp tổng quát ta hãy kí hiệu chúng là a và b.

Cần tuân thủ các quy tắc sau :

- phải thêm vào b kí tự "&" sau \multicolumn (con số này là 1 với ví dụ trên);
- dịch chữ cái thứ tự cột của các ô sau \multicolumn a chữ cái về phía đầu của bảng chữ cái. Đối với ví dụ trên, nếu ta muốn tham chiếu đến ô "i2", cần phải nhập vào g2.

Trong ví dụ sau, mỗi số ở hàng dưới sẽ bằng tổng của số ở hàng trên và 1. Bảng này có cấu trúc tương tự như ví dụ trước:  $a = 2$  và  $b = 1$ . Chú ý rằng ta sẽ thêm vào một kí tự "&" bởi vì  $b = 1$ .

```

1 \begin{spreadtab}{{\tabular}{|*7c|}}}
2 \hline
3 1 & 2 & \SThidecol3 & \SThidecol4 & 5& \SThidecol6 & 7& 8& 9 & 10 \\ \hline
4 a1+1& \multicolumn4{1|}{:=\{b1+1\}}& & & & i1+1 & j1+1 \\ \hline
5 a2+1& b2+1 & & & & & & g2+1 & h2+1 \\ \hline
6 \end{spreadtab}

```

1	2	5	7	8	9	10
2	3				10	11
3	4				11	12

Sau đây là một ví dụ khác tương tự, ta sẽ ẩn một cột (cột d):  $a = 1$  và  $b = 0$ .

```

1 \begin{spreadtab}{{\tabular}{|*9c|}}}
2 \hline
3 1 & 2 & 3 & \SThidecol4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
4 a1+1& \multicolumn6{1|}{:=\{b1+1\}} & & i1+1 & j1+1 \\ \hline
5 a2+1& b2+1 & & & & & & h2+1 & i2+1 \\ \hline
6 \end{spreadtab}

```

1	2	3	5	6	7	8	9	10
2	3						10	11
3	4						11	12

### 5.3 Các thông báo lỗi

spreadtab sẽ dùng biên dịch và thông báo lỗi trong các trường hợp sau:

- để tính giá trị của một ô, chúng ta tính các ô mà nó tham chiếu đến; với các ô này chúng ta lại làm tương tự, và nếu có một ô nào đó tham chiếu đến ô ban đầu (tham chiếu vòng), hệ thống sẽ báo lỗi và hiển thị trong tin nhắn báo lỗi vòng tham chiếu này;
- một công thức chứa một tham chiếu đến một ô rỗng hoặc chỉ chứa văn bản;
- một ô tham chiếu đến một ô không xác định (bên ngoài giới hạn của bảng);
- một ô tham chiếu đến một ô bị hợp nhất bởi `\multicolumn`;
- một ví trí tương đối không đúng cú pháp.

spreadtab có thể thông báo lỗi trong tập tin log. Lệnh `\STmessage` với tham số `true` hoặc `false` cho phép thực hiện hay không việc báo lỗi này (`true` theo mặc định).

Chúng ta hãy lấy một ví dụ đơn giản:

```
1 \begin{spreadtab}{{\tabular}{|cccc|c|}}\hline
2 b1+1 & c1+1 & d1+1 & 10 & a1+b1+c1+d1\\\hline
3 \end{spreadtab}
```

13	12	11	10	46
----	----	----	----	----

Việc tiến hành xây dựng bảng ở đây rất đơn giản. Sau đây là một số thông tin cho bởi spreadtab:

```
1 [spreadtab] New spreadtab ({\tabular}{|cccc|c|})
2 * reading tab: ok
3 * computing formulas:
4   cell A1-B1-C1
5   cell B1
6   cell C1
7   cell D1
8   cell E1
9 * building tab: ok
10 [spreadtab] End of spreadtab
```

Môi trường bảng cho bởi người sử dụng được đặt trong các dấu ngoặc (trong trường hợp này là `{\tabular}{|cccc|c|}`). Chúng ta thấy rõ 3 bước cần thiết của spreadtab (bắt đầu bởi các dấu sao) để thực hiện nhiệm vụ của nó: đọc bảng, tính các công thức rồi sau đó xây dựng lại và hiển thị bảng.

Ở bước thứ hai, các ô được đánh giá từ trên xuống dưới, từ trái sang phải: ta thấy rằng spreadtab bắt đầu tính ô A1. Để làm điều đó, nó phải tính B1 trước, và trước đó nữa là C1. Vì C1 chỉ phụ thuộc vào D1 (có giá trị bằng 10) nên nó có thể được tính.

Các hàng tiếp theo chỉ có một ô, nghĩa là spreadtab tính các ô này trong khi chúng đã có sẵn giá trị hoặc tham chiếu đến những ô đã được tính.

### 5.4 Sửa lỗi

Việc sử dụng spreadtab sẽ dễ dàng hơn nhờ chế độ sửa lỗi của nó. Chế độ này được sẽ được kích hoạt khi lệnh `\STdebug` được đặt trong phần tham số tùy chọn của môi trường spreadtab. Lệnh này sẽ khiến cho spreadtab, thay vì hiển thị bảng kết quả, hiển thị một hay nhiều "bảng sửa lỗi". Việc hiển thị này được thực hiện ngay sau khi spreadtab vừa đọc hết các ô của bảng, khi mà chưa một công thức nào được tính. Số lệnh `\STdebug` sẽ quyết định số bảng sửa lỗi được hiển thị. Chỉ có ba tham số được sử dụng:

- `\STdebug{formula}` : hiển thị vùng số của tất cả các ô và các lệnh xuống hàng;



- `\STdebug{text}` : hiển thị vùng chữ của tất cả các ô;
- `\STdebug{code}` : hiển thị mã mà spreadtab gán cho mỗi ô khi đọc nó. Mã này có giá trị bằng:
  - -1 nếu đó là một ô được hợp nhất bởi lệnh `\multicolumn`;
  - 0 nếu đó là một ô trống hoặc là một ô chỉ chứa văn bản;
  - 1 nếu vùng số của ô chứa công thức (sẽ được tính sau này);
  - 2 nếu vùng số chứa một giá trị cụ thể.

Khi chế độ sửa lỗi được kích hoạt, bảng kết quả *sẽ không được hiển thị*. Tuy nhiên chúng ta có thể ép cho spreadtab hiển thị nó bằng cách thêm lệnh `\STdisplaytab` vào tham số tùy chọn.

Chúng ta hãy xem ví dụ dưới đây:

```

1 \begin{spreadtab}{\tabular}{|r|r|r|}\hline
2 @x$ & @y$ & @x+y$\\ \hline
3 22 & 54 & \STcopy{v3}{a2+b2} \\
4 43 & 65 & \\
5 49 & 37 & \\ \hline
6 $Sx:=\{a2+a3+a4\}$ & $Sy:=\{b2+b3+b4\}$ & $Sx+Sy:=\{\}$\\ \hline
7 \multicolumn{2}{|r|}{$Sy-Sx:=\{b5-a5\}$} & @\multicolumn{1c}{\\ \cline{1-2}}
8 \end{spreadtab}

```

$x$	$y$	$x + y$
22	54	76
43	65	108
49	37	86
$Sx = 114$	$Sy = 156$	$Sx + Sy = 270$
$Sy - Sx = 42$		

Chúng ta sẽ nhờ spreadtab hiển thị ba bảng sửa lỗi của bảng trên. Chỉ cần thay đổi dòng thứ nhất của đoạn mã ở trên như sau:

```
\begin{spreadtab}[\STdebug{text}\STdebug{formula}\STdebug{code}]{\tabular}{|rr|r|}\hline
```

	A	B	C
1	$x$	$y$	$x+y$
2	$:=$	$:=$	$:=$
3	$:=$	$:=$	$:=$
4	$:=$	$:=$	$:=$
5	$Sx:=$	$Sy:=$	$Sx+Sy:=$
6	$Sy-Sx:=$		

	A	B	C
1			
2	22	54	$a2+b2$
3	43	65	$a3+b3$
4	49	37	$a4+b4$
5	$a2+a3+a4$	$b2+b3+b4$	$a5+b5$
6	$b5-a5$		

	A	B	C
1	0	0	0
2	2	2	1
3	2	2	1
4	2	2	1
5	1	1	1
6	1	-1	0

Ba bảng sửa lỗi này có thể giúp ta hiểu rõ hơn về sự hoạt động của spreadtab. Chúng ta thấy rằng tất cả các ô chứa vùng số, được thể hiện bởi " $:=$ " (xem bảng đầu tiên), đều có mã 1 hoặc 2 (xem bảng thứ 2 và thứ 3). Vị trí của " $:=$ " chính là vị trí sẽ hiển thị kết quả (của vùng số) một khi được tính.

Trong các bảng ở trên, các ô chứa tọa độ chỉ có thể được tô xám khi ta gọi thêm gói `colortbl`.

## 6 Một số ví dụ

Chúng ta hãy kết thúc bằng một vài ví dụ thú vị !

Để biết được kết quả nào đã được tính, chỉ những số không được tính (cho bởi người dùng) sẽ được in đỏ. Trong các ví dụ này, rất nhiều kĩ năng và gói (chủ yếu là `numprint` và các cột "N" của nó để canh thẳng hàng các dấu phẩy thập phân) đã được sử dụng để nhận được những kết quả, nhiều hay ít, thỏa mãn mỹ quan của chúng ta. Mã nhập vào đôi khi khá nặng nề, nhưng đây không phải là những ví dụ cơ bản, mà là những bảng được "trang điểm" tỉ mỉ. Hơn nữa, để cho "sáng sủa", ta sẽ không sử dụng lệnh `\STcopy`.

### 6.1 Lại là tam giác Pascal

```

1 \begin{spreadtab}{{tabular}{*7r}}
2 [0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & &
   [-1,1]\\
3 [0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1] & & \\
4 [0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1] & & & \\
5 [0,1] & [-1,1]+[0,1] & [-1,1]+[0,1] & [-1,1] & & & & \\
6 [0,1] & [-1,1]+[0,1] & [-1,1] & & & & & \\
7 [0,1] & [-1,1] & & & & & & \\
8 \color{red}:=\{1\}& & & & & & & \\
9 \end{spreadtab}

```

```

1 6 15 20 15 6 1
1 5 10 10 5 1
1 4 6 4 1
1 3 3 1
1 2 1
1 1
1

```

### 6.2 Chuỗi hội tụ

Chuỗi mà chúng ta sẽ xét đến là khai triển giới hạn của hàm số mũ:

$$\forall x \in \mathbf{R} \quad e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Bảng sau minh họa "tốc độ" hội tụ ngày càng tăng theo bậc khai triển giới hạn tại 0,5.

```

1 \STautoround{15}
2 \begin{spreadtab}[\STsavecell\xvalue{a1}]{\tabular}{cN{2}{15}}
3 \multicolumn{2}{c}{Hội tụ tại $x=\color{red}{\numprint{:=\{0.5\}}}$\\[1.5ex]}
4 @n$ & e^a1\SThidecol & \hfill{\@ $\displaystyle e^{\numprint\xvalue}-\sum_{k=0}^n\frac{\numprint\xvalue^k}{k!}$}\hfill\\
5 \color{red}:=\{0\} & a1^{-1,0}/fact(-1,0) & b2-[-1,0] \\
6 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
7 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
8 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
9 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
10 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
11 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
12 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
13 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
14 [0,-1]+1 & a1^{-1,0}/fact(-1,0)+[0,-1] & b2-[-1,0] \\
15 \end{spreadtab}

```

Hội tụ tại  $x = 0,5$

$$n \quad e^{0,5} - \sum_{k=0}^n \frac{0,5^k}{k!}$$

0	0,648 721 270 700 128
1	0,148 721 270 700 128
2	0,023 721 270 700 128
3	0,002 887 937 366 795
4	0,000 283 770 700 128
5	0,000 023 354 033 461
6	0,000 001 652 644 572
7	0,000 000 102 545 366
8	0,000 000 005 664 166
9	0,000 000 000 281 877

### 6.3 Hội tụ đến tỉ số vàng

Sau đây là định nghĩa của dãy số Fibonacci:  $F_0 = 1 \quad F_1 = 1 \quad F_{n+2} = F_{n+1} + F_n$

Chúng ta biết rằng thương của hai số hạng liên tiếp  $F_n$  và  $F_{n-1}$  của dãy tiến đến tỉ số vàng  $\varphi = \frac{1+\sqrt{5}}{2}$  và dãy  $u_n = \varphi - \frac{F_n}{F_{n-1}}$  là một dãy đan dấu và hội tụ về 0.

```

1 \STautoround{9}
2 $
3 \begin{spreadtab}{{matrix}}{}
4 @n & @F_n & @\dfrac{F_n}{F_{n-1}} & @\varphi-\dfrac{F_n}{F_{n-1}}\\[2
   ex]\hline
5 1 & 1 & & \\
6 [0,-1]+1 & 1 & [-1,0]/[-1,-1] & (1+5^0.5)/2-[-1,0] \\
7 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
8 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
9 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
10 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
11 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
12 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
13 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
14 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
15 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
16 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
17 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
18 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
19 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
20 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
21 [0,-1]+1 & [0,-1]+[0,-2] & [-1,0]/[-1,-1] & d3+1-[-1,0] \\
22 \end{spreadtab}
23 $

```

$n$	$F_n$	$\frac{F_n}{F_{n-1}}$	$\varphi - \frac{F_n}{F_{n-1}}$
1	1		
2	1	1	0.618033989
3	2	2	-0.381966011
4	3	1.5	0.118033989
5	5	1.666666667	-0.048632678
6	8	1.6	0.018033989
7	13	1.625	-0.006966011
8	21	1.615384615	0.002649374
9	34	1.619047619	-0.00101363
10	55	1.617647059	0.00038693
11	89	1.618181818	-0.000147829
12	144	1.617977528	0.000056461
13	233	1.618055556	-0.000021567
14	377	1.618025751	0.000008238
15	610	1.618037135	-0.000003146
16	987	1.618032787	0.000001202
17	1597	1.618034448	-0.000000459

## 6.4 Bảng hóa đơn

Trong bảng hóa đơn sau, các dấu chấm thập phân được canh thẳng hàng trong các cột nhờ kiểu hàng "N" của gói numprint.

Bảng này sử dụng môi trường tabularx sao cho chiều rộng của nó chiếm 80% độ rộng văn bản (của tài liệu này). Lệnh \multicolumn được sử dụng nhiều lần để định dạng bảng:

```

1 \nprouddigits2
2 \let\PC\%
3 \begin{spreadtab}{{tabularx}{0.8\linewidth}{|>\rule[-1.2ex]{0pt}{4ex}}X>{\color{red}
4 \hline
5 @Mục&\multicolumn{1}{c}{Giá đơn vị} & @\multicolumn{1}{c}{Số
    lượng} & @\multicolumn{1}{c}{Giá} & @\multicolumn{1}{c}{Giảm giá} & @\textbf{Còn
    lại}\hline
6 @Item 1& 5.99 & 20 & [-2,0]*[-1,0] & $-:={20}\PC$ & [-2,0]*(1-[-1,0]/100)\hline
7 @Item 2& 12 & 7 & [-2,0]*[-1,0] & $-:={10}\PC$ & [-2,0]*(1-[-1,0]/100)\hline
8 @Item 3& 4.50 & 40 & [-2,0]*[-1,0] & $-:={35}\PC$ & [-2,0]*(1-[-1,0]/100)\hline
9 @Item 4& 650 & 2 & [-2,0]*[-1,0] & $-:={15}\PC$ & [-2,0]*(1-[-1,0]/100)\hline
10 @\multicolumn{6}{c}{\vspace{-1.5ex}}\cline{4-6}
11 @\multicolumn{1}{c}{\rule[-1.2ex]{0pt}{4ex}}&\multicolumn{2}{r}{Tổng
    cộng}&\sum(d2:[0,-2]) &\multicolumn{1}{c}{\textbf{\$:=\round((([1,0]/[-1,0]-1)*100,0))\PC$}
    &\fontseries{b}\selectfont:=\sum(f2:[0,-2])}\hline
12 \cline{4-6}
13 \end{spreadtab}

```

Mục	Giá đơn vị	Số lượng	Giá	Giảm giá	Còn lại
Item 1	5,99	20	119,80	-20%	95,84
Item 2	12,00	7	84,00	-10%	75,60
Item 3	4,50	40	180,00	-35%	117,00
Item 4	650,00	2	1 300,00	-15%	1 105,00
Tổng cộng			1 683,80	-17%	1 393,44

## 6.5 Ô vuông ma thuật

```

1 \begin{spreadtab}{{tabular}{|*3>{\hfill\rule[-0.4cm]{0pt}{1cm}}$m{0.7cm}<{\hfill\
    null}|}}

```

```

2 \hline
3 \color{red}:=2 & 5*b2-4*a1 & 3*a1-2*b2 \\\hline
4 2*a1-b2 & \color{red}:=-1 & 3*b2-2*a1 \\\hline
5 4*b2-3*a1 & 4*a1-3*b2 & 2*b2-a1 \\\hline
6 \end{spreadtab}

```

2	-13	8
5	-1	-7
-10	11	-4

## 6.6 Kim tự tháp tổng

Mỗi số hạng là tổng của hai số ngay dưới nó.

```

1 \begin{spreadtab}{\tabular}{*{8}{m{2ex}}}{
2 \cline{4-5}
3 :={\rule{0pt}{2.7ex}}&&&\multicolumn{2}{|c|}{\color{red}:=\{-1,1\}+\{1,1\}}&&&\cline{3-6}
4 :={\rule{0pt}{2.7ex}}&&\multicolumn{2}{|c|}{\color{red}:=\{-1,1\}+\{1,1\}}&&\multicolumn{2}{|c|}{\color{red}:=\{-1,1\}+\{1,1\}}&&\cline{2-7}
5 :={\rule{0pt}{2.7ex}}&&\multicolumn{2}{|c|}{\color{red}:=\{-1,1\}+\{1,1\}}&&\multicolumn{2}{|c|}{\color{red}:=\{-1,1\}+\{1,1\}}&&\hline
6 \multicolumn{2}{|c|}{\rule{0pt}{2.7ex}\color{red}:=\{-5\}}&&\multicolumn{2}{|c|}{\color{red}:=\{-5\}}&&\multicolumn{2}{|c|}{\color{red}:=\{-5\}}&&\hline
7 &&&&&&&&&&
8 \end{spreadtab}

```

		-5		
	-1		-4	
-2		1		-5
-5	3	-2	-3	

★  
★ ★

Đó là tất cả, hi vọng rằng gói này sẽ giúp ích cho các bạn !

Vì mới ra đời nên **spreadtab** không thể không có những sai sót. Xin vui lòng báo cho tác giả qua **email** các lỗi mà các bạn gặp phải, các macro hàm cần bổ sung và tất cả các ý kiến đóng góp *mang tính thực tế*: **spreadtab** cần giữ được nét đơn giản của nó, nó không phải là **excel** hay **calc** và không thể bổ sung tất cả các chức năng nâng cao của các bảng tính này.

Christian TELLECHEA

Nd: Xin vui lòng góp ý cho bản dịch qua **email**. Người dịch xin cảm ơn anh HUỖNH KỲ ANH đã giúp đỡ trong việc hoàn thành tài liệu này !