delimiter **1**

`delimiter.` A *delimiter* is a character that is intended to be used as a visible boundary of a math formula. The essential property of a delimiter is that TEX can adjust its size according to the vertical size (height plus depth) of the subformula. However, TEX performs the adjustment only if the delimiter appears in a "delimiter context", namely, as an argument to one of the commands `\left`, `\right`, `\overwithdelims`, `\atopwithdelims`, or `\abovewithdelims` (see pp. '`\overwithdelims`', '`\left`'). The delimiter contexts also include any argument to a macro that uses the argument in a delimiter context.

For example, the left and right parentheses are delimiters. If you use parentheses in a delimiter context around a formula, TEX makes the parentheses big enough to enclose the box that contains the formula (as long as the fonts you're using have big enough parentheses). For example:

```
$$ \left( a \over b \right) $$
```

yields:

$$\left(\frac{a}{b}\right)$$

Here TEX has made the parentheses big enough to accommodate the fraction. But if you write, instead:

```
$$({a \over b})$$
```

you'll get:

$$(\frac{a}{b})$$

Since the parentheses aren't in a delimiter context, they are *not* enlarged.

Delimiters come in pairs: an opening delimiter at the left of the subformula and a closing delimiter at its right. You can explicitly choose a larger height for a delimiter with the commands `\bigl`, `\bigr`, and their relatives (p. '`\bigl`').[1] For instance, in order to get the displayed formula:

$$\bigl(f(x)-x\bigr)\bigl(f(y)-y\bigr)$$

in which the outer parentheses are a little bigger than the inner ones, you should write:

```
$$\bigl( f(x) - x \bigr) \bigl( f(y) - y \bigr)$$
```

The 22 plain TEX delimiters, shown at their normal size, are:

( ) [ ] { } ⌊ ⌋ ⌈ ⌉ ⟨ ⟩ / \ | ‖ ↑ ↓ ↕ ⇑ ⇓ ⇕

Here they are at the largest size provided explicitly by plain TEX (the `\Biggl`, `\Biggr`, etc., versions):

$$\Big(\Big) \Big[\Big] \Big\{ \Big\} \Big\lfloor \Big\rfloor \Big\lceil \Big\rceil \Big\langle \Big\rangle \Big/ \Big\backslash \Big| \Big\| \Big\uparrow \Big\downarrow \Big\updownarrow \Big\Uparrow \Big\Downarrow \Big\Updownarrow$$

[1] Plain TEX defines the various `\big` commands by using `\left` and `\right` to provide a delimiter context. It sets the size by constructing an empty formula with the desired height.

The delimiters (except for '(', ')', and '/') are among the symbols listed on pages '\lbrace'–'\Uparrow'. They are listed in one place on page 146 of *The TEXbook*.

A delimiter can belong to any class. For a delimiter that you enlarge with \bigl, \bigr, etc., the class is determined by the command: "opener" for l-commands, "closer" for r-commands, "relation" for m-commands, and "ordinary symbol" for g-commands, e.g., \Big.

You can obtain a delimiter in two different ways:

1) You can make a character be a delimiter by assigning it a nonnegative delimiter code (see below) with the \delcode command (p. '\delcode'). Thereafter the character acts as a delimiter whenever you use it in a delimiter context.[2]
2) You can produce a delimiter explicitly with the \delimiter command (p. '\delimiter'), in analogy to the way that you can produce an ordinary character with the \char command or a math character with the \mathchar command. The \delimiter command uses the same delimiter codes that are used in a \delcode table entry, but with an extra digit in front to indicate a class. It's rare to use \delimiter outside of a macro definition.

A delimiter code tells TEX how to search for an appropriate output character to represent a delimiter. The rules for this search are rather complicated (see pages 156 and 442 of *The TEXbook*). A complete understanding of these rules requires knowing about the organization of font metrics files, a topic that is not just beyond the scope of this book but beyond the scope of *The TEXbook* as well.

In essence the search works like this. The delimiter code specifies a "small" output character and a "large" output character by providing a font position and a font family for each (see p. '\delcode'). Using this information, TEX can find (or construct) larger and larger versions of the delimiter. TEX first tries different sizes (from small to large) of the "small" character in the "small" font and then different sizes (also from small to large) of the "large" character in the "large" font, seeking one whose height plus depth is sufficiently large. If none of the characters it finds are large enough, it uses the largest one that it finds. It's possible that the small character, the large character, or both have been left unspecified (indicated by a zero in the appropriate part of the delimiter code). If only one character has been specified, TEX uses that one. If neither has been specified, it replaces the delimiter by a space of width \nulldelimiterspace.

[2] It's possible to use a character with a nonnegative delimiter code in a context where it isn't a delimiter. In this case TEX doesn't perform the search; instead it just uses the character in the ordinary way (see page 156 of *The TEXbook*).