[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06K 9/00

[12]发明专利说明书

[21] ZL 专利号 94109513.4

[45]授权公告日 2002 年 11 月 27 日

[11]授权公告号 CN 1095142C

[22]申请日 1994.8.3 [21]申请号 94109513.4

[30]优先权

[32]1993.8.13 [33]JP [31]201543/93

[73]专利权人 索尼公司

地址 日本东京

[72]发明人 茶谷公之 前地三千儿

审查员 张静海

[74]专利代理机构 中国专利代理(香港)有限公司

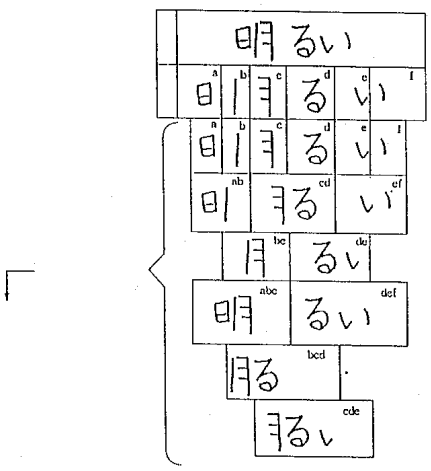
代理人 傅 康 叶恺东

权利要求书 2 页 说明书 28 页 附图 34 页

[54]发明名称 手写文字输入方法及装置

[57]摘要

本发明的目的是为了能够更正确地识别手写输入的文字。当手写输入文字时,把它分解为 a ~ f 的子字符,并且,把子字符 a ~ f 中相邻的子字符组合成指定的数量,识别为 1 个文字。



ISSN1008-4274

1. 一种手写文字输入方法，其特征在于：在检测手写输入的文字各笔划的特征，利用检测的上述特征量识别子字符并将上述子字符组合后识别文字的的写文字输入方法中，备有行切割规则和子字符切割规则，当输入上述笔划时，根据该笔划的特征、上述子字符和上述行切割规则判断输入的行，当输入的上述行与以前输入的行是同一行时，根据上述笔划的特征、上述子字符和上述子字符切割规则，进行上述子字符的判断。

2. 一种手写文字输入方法，其特征在于：在检测手写输入的文字各笔划的特征，利用检测的上述特征量识别子字符并将上述子字符组合后识别文字的手写文字输入方法中，识别由上述笔划构成的上述文字的行，当上述文字的行变更时，一并将上述子字符组合后进行文字的识别。

3. 一种手写文字输入方法，其特征在于：在检测手写输入的文字各笔划的特征，利用检测的上述特征量识别子字符并将上述子字符组合后识别文字的手写输入方法中，每次输入上述笔划，都进行上述子字符的切割，当切割出新的上述子字符时，逐次将上述子字符组合后进行文字的识别，同时，识别由上述笔划构成的上述文字的行，当上述文字的行变更时，将上述字符组合后进行文字的识别。

4. 按权利要求 1，2 和 3 的任何一个所述的手写文字输入方法，其特征是在上述子字符或文字的识别中，至少要使用上述子字符或文字的上述笔划的总数。

5. 按权利要求 1~4 的任一权利要求所述的手写文字输入方法，其特征是：为了进行上述子字符或行的切割，对多个条件预先建立数据库，单独计算各个条件的满足度，并将各个条件的计算结果的总和

与基准进行比较。

6. 一种手写文字输入装置，其特征在于：它使用权利要求 1 所述的方法。

7. 一种手写文字输入装置，其特征在于：它使用权利要求 2 所述的方法。

8. 一种手写文字输入装置，其特征在于：它使用权利要求 3 所述的方法。

9. 一种手写文字输入装置，其特征在于：它使用权利要求 4 所述的方法。

10. 一种手写文字输入装置，其特征在于：它使用权利要求 5 所述的方法。

手写文字输入方法及装置

本发明涉及例如用于笔记本式电子装置及便携式微机等装置的极佳的手写文字输入方法及装置。

最近，例如笔记本式电子装置等能用手写方式输入文字的装置正在普及。这种能用手写方式输入文字的装置和通常的文字处理器等不同，不是用键盘而是用笔把文字写在面板上就可以输入文字，所以，如同用铅笔等把文字写在纸上一样，可以很轻松地输入文字。与使用键盘的情况比较，不必记住键的位置，所以，初学者也可以很容易地输入文字。

这种能用手写方式输入文字的装置有两种，一种设定了输入文字的输入框，另一种不设定输入框，可在任意的位置输入文字。

在设定输入框的装置中，使用者通过手写 1 个字 1 个字地将文字输入到该输入框内。该输入框通常分为多个区域，使用者 1 个文字 1 个文字地输入到各个区域。由于输入的位置有限制，所以，对使用者来说，这种限制给操作带来不便，但是，由于输入各文字的区域是分开的，所以，不必将输入的文字 1 个文字 1 个文字地切割开就能比较准确地认识各文字。

与此相反，在未设定输入框的装置中，使用者用手写方式把文字输入到输入面板（平板）上的任意位置。这时，通过手写方式输入的文字例如按照图 4 8 所示的流程那样进行识别。

即，最初输入文字时，由输入的文字所构成的行在指定的条件下进行识别，并切割出来（识别行的分段）。并且，当切割出行之后，将该行中的文字在指定的条件下1个文字1个文字地切割开。即，确定输入的文字与文字的界限。最后，将切割开的各个文字作为1个文字进行识别，并显示该识别结果。

结果，例如，如图49（a）所示的那样，当使用者输入了“明”这个汉字时，在把“日”和“月”字切割为构成1个文字时，就将输入的文字认作汉字“明”。

但是，如图49（b）所示的那样，若书写得“日”和“月”之间的间隔比较大，将这两个文字分别作为独立的文字切割开时，就会将“日”和“月”字分别认作独立的文字。

这时，就无法判断使用者输入的是“明”字还是“日”和“月”字。

这样，对于先有的未设定输入框的用手写方式输入文字的装置，就存在一个按照文字的切割状态而误认文字的可能性大的问题。

本发明就是鉴于上述问题而研制的，目的在于能更准确地识别手写文字。

本发明所述的手写文字输入方法，其特征在于：在检测手写输入的文字各笔划的笔迹特征，利用检测的特征量识别子字符并将子字符组合后识别文字的手写文字输入方法中，备有行切割规则和子字符切割规则，当输入笔划时，根据该笔划的特征、子字符和行切割规则判断输入的行，当输入的行与以前输入的行是同一行时，根据笔划的特征、子字符和子字符切割规则进行子字符的判断。

本发明所述的手写文字输入方法，其特征在于：在检测手写输入

的文字各笔划的笔迹特征，利用检测的特征量识别子字符并将子字符组合后识别文字的手写文字输入方法中,识别由笔划构成的文字的行，并在变更文字的行时一并将子字符组合后进行文字识别。

另外，本发明所述的手写文字输入方法，其特征在于：在检测手写输入的文字各笔划的笔迹特征，利用检测的特征量识别子字符并将子字符组合后识别文字的手写文字输入方法中，每次输入笔划，进行子字符的切割并切割出新的子字符时，逐次将子字符组合后进行文字识别，同时，识别由笔划构成的文字的行，并在变更文字的行时将子字符组合后进行文字的识别。

在这种子字符或文字的识别中，至少可以使用子字符或文字的笔划的总数。

另外，为了进行子字符或行的切割，可以预先对多个条件建立起数据库，分别计算各个条件的充足度，并将各个条件的计算结果的总和与基准值进行比较。

在本发明所述的手写文字输入方法中，当输入笔划时，根据该笔划的特征、子字符和行切割规则判断输入的行，并在输入的行与以前输入的行为同一行时，根据笔划的特征、子字符和子字符切割规则进行子字符的判断。因此，可以准确地切割子字符。

在本发明所述的手写文字输入方法中，在识别文字的行并变更该行时一并将子字符组合后进行文字的识别。因此，可以准确地识别文字。

另外，在本发明所述的手写文字输入方法中，在变更行时将子字符组合后进行文字识别，同时在切割出新的子字符时逐次进行文字的识别。因此，可以更迅速而准确地识别。

图 1 是应用本发明的手写文字输入方法的装置的一个实施例的结构框图。

图 2 是说明图 1 中的实施例的动作的流程图。

图 3 是说明图 2 中 S 1 3 的行切割判断处理的更详细处理的流程图。

图 4 是说明图 2 中 S 1 5 的子字符切割判断处理的更详细处理的流程图。

图 5 是说明图 2 中 S 1 9 的子字符组合识别处理的更详细处理的流程图。

图 6 是说明行切割的具体例子的图。

图 7 是说明子字符切割的具体例子的图。

图 8 是说明子字符组合识别处理的具体例子的图。

图 9 是说明子字符组合识别处理的其它具体例子的图。

图 1 0 是说明第 1 个行切割规则的流程图。

图 1 1 是说明第 2 个行切割规则的流程图。

图 1 2 是说明第 3 个行切割规则的流程图。

图 1 3 是说明第 4 个行切割规则的流程图。

图 1 4 是说明第 5 个行切割规则的流程图。

图 1 5 是说明第 6 个行切割规则的流程图。

图 1 6 是说明第 7 个行切割规则的流程图。

图 1 7 是说明第 1 个行切割规则的图。

图 1 8 是说明第 2 个行切割规则的图。

图 1 9 是说明第 3 个行切割规则的图。

图 2 0 是说明第 4 个行切割规则的图。

图 2 1 是说明第 5 个行切割规则的图。

图 2 2 是说明第 6 个行切割规则的图。

图 2 3 是说明第 7 个行切割规则的图。

图 2 4 是说明第 1 个子字符切割规则的流程图。

图 2 5 是说明第 2 个子字符切割规则的流程图。

图 2 6 是说明第 3 个子字符切割规则的流程图。

图 2 7 是说明第 4 个子字符切割规则的流程图。

图 2 8 是说明第 5 个子字符切割规则的流程图。

图 2 9 是说明图 2 7 中的子字符切割规则的图。

图 3 0 是说明图 2 8 中的子字符切割规则的图。

图 3 1 是说明图 5 中 S 6 6 的作为文字合成条件的第 1 个子字符合成条件的流程图。

图 3 2 是说明图 5 中 S 6 6 的作为文字合成条件的第 2 个子字符合成条件的流程图。

图 3 3 是说明图 5 中 S 6 6 的作为文字合成条件的第 3 个子字符合成条件的流程图。

图 3 4 是说明笔划特征数据的格式的图。

图 3 5 是说明单个文字数据数据库的格式的图。

图 3 6 是说明行数据数据库的格式的图。

图 3 7 是说明行数据的格式的图。

图 3 8 是说明识别结果数据数据库的格式的图。

图 3 9 是说明识别结果的格式的图。

图 4 0 是说明行切割规则数据库的格式的图。

图 4 1 是说明行切割规则记录格式的图。

图４２是说明子字符切割规则数据库的格式的图。

图４３是说明子字符切割规则记录格式的图。

图４４是说明图１中的实施例的其它动作例子的流程图。

图４５是说明图４４中Ｓ２３２的更详细的处理的图。

图４６是说明图４４中Ｓ２３１的处理的图。

图４７是说明图４４中Ｓ２３１的处理的图。

图４８是说明先有的文字识别处理流程的流程图。

图４９是说明先有的文字识别处理的图。

１－ＣＰＵ

２－ＲＯＭ

３－ＲＡＭ

８－总线

１１－显示控制器

１２－Ｖ－ＲＡＭ

１３－显示器

１４－输入面板接口

１５－输入面板

１６－笔

图１是应用本发明的手写文字输入方法的装置的一个实施例的结构框图。ＣＰＵ１根据ＲＯＭ２存储的程序控制各部分，进行文字输入动作。在ＲＡＭ３中，根据需要存储指定的数据等。备用ＲＡＭ４由电池５经常供给指定的电压，以使在切断装置的电源时也可以存储想存储的信息。定时器６以晶体振子７为基准，由电池５保持其经常动作。

显示控制器１１将通过总线８输入的数据在Ｖ－ＲＡＭ１１上展开，并使在其上展开的图像在显示器１３上进行显示。通过操作笔16，输入面板１５将输入的座标位置数据输给面板接口１４，面板接口14通过总线８将该数据输给ＣＰＵ１或显示控制器１１。

传声器１８将接收到的声音信号输给固体录音处理电路１７，并存储起来。另外，固体录音处理电路１７输出的声音信号通过加法器２０供给扬声器２１进行放音。拨号器(dial)１９根据ＣＰＵ１的控制，通过加法器２０将指定的拨号信号输给扬声器２１。

在切换装置的动作时，操作多个琴键开关群２２，与该操作对应的信号通过总线８从键接口２３输给ＣＰＵ１。另外，通过操作多个琴键开关群２４的指定开关，接通电源电路２５，向各部分供电，可以进行固体录音处理电路１７的录音、放音、停止和音量调整等动作。

插口２６内可以插入图中未示出的集成电路卡（ＩＣ卡），ＩＣ卡接口２７对该ＩＣ卡进行数据的输送与接收。

插头接续器２９可以连接例如调制解调器等，通过扩展接口２８和调制解调器可以进行信号的输送与接收。

另外，输入面板１５由透明的材质构成，形成在显示器１３的上面。这样，显示器１３的显示便可通过输入面板１５而看到。

下面，参照图２的流程图说明上述结构的装置中通过手写方式输入文字的动作。这一处理是按照ＲＯＭ２存储的程序由ＣＰＵ２进行的。使用者用笔１６在输入面板１５上每次输入构成文字的任意的１个笔划，ＣＰＵ１都进行图２中的流程图所示的处理。在该流程图中，表示笔画数的变量之和表示后面所述的子字符数的变量Ｓ在接通电源时都初始化为０。

另外，进行该流程图中的各个处理所需要的数据库存储在ROM 2或ＲＡＭ３内。

开始，在步Ｓ１１，抽取输入笔划的特征。并且将该特征输给笔划特征数据库ＤＢ１进行存储。关于存储在该笔划特征数据库ＤＢ１中的数据的格式，后面将参照图３４进行介绍。

然后，进入Ｓ１２，判断输入的笔迹是否为行数据初始化后最初的笔迹。即，判断变量i是否为０（如后所述，该变量i在Ｓ１４将增加１，所以，在第２笔划以后输入时成为大于１的值）。输入的笔迹是行数据初始化后最初的笔迹时，进入Ｓ２０，进行换行处理。在此生成新的一行数据，并且，变量i和Ｓ分别置为１。

在Ｓ１２，当断定输入的笔迹不是行数据初始化后最初的笔迹时，则进入Ｓ１３，进行行切割判断的处理。在该处理中，使用在Ｓ１１生成的存储在笔划特征数据库ＤＢ１中的笔划特征数据、存储在行数据库ＤＢ４中的行数据和存储在行切割规则数据库ＤＢ２中的行切割规则。在此判断现在输入的笔划是否包含在由行数据表示的行内（是输入到相同的行内还是输入到不同的行内）。关于该行切割判断的详细情况，后面将参照图３进行介绍。

在最新笔划的特征抽取数据中，例如，除了笔划的始点或终点等特征点的座标外，还包含与笔划外接的长方形的数据等。另外，关于行切割规则，将在后面参照图１０～图２３进行介绍，但是，这里要使用笔迹数据形成的行的外接长方形的座标、笔划数、各笔划的外接长方形的座标、各笔划的投影长度和实际长度等。

在行切割判断处理中，只要未断定本次输入的新的输入笔划输入到与迄今输入的笔划不同的行，即断定输入到了与迄今输入的笔划相

同的行内时，便进入Ｓ１４，将本次输入的笔划的笔划特征数据追加到行数据库ＤＢ４的行数据内。并且，使表示行数据内的笔划数的变量i增加１。

然后，进入Ｓ１５，进行子字符切割判断处理。这里，所谓子字符是指构成１个文字的要素的笔迹的集合。即，在本实施例中，１个文字分解为作为要素的子字符，判断各输入笔划是构成与前一个输入的笔划相同的子字符的笔划，还是构成与一个以上的前输入的多个笔划相同的子字符的笔划或者是构成与迄今输入的笔划不同的子字符的笔划。

在该子字符切割的判断处理中，要使用在子字符切割规则数据库ＤＢ３中存储的子字符切割规则和在行数据库ＤＢ４中存储的行数据。该判断处理的详细情况，将在后面参照图４进行介绍，但是，例如是考虑最新笔划笔迹的水平或垂直投影成分与已存在的子字符笔迹集合的水平或垂直投影成分的重叠部分后进行判断的。

在Ｓ１５，当最新笔划的笔迹是构成和前一个子字符（第Ｓ－１个子字符）相同的子字符的笔迹时，进入Ｓ１６，进行追加处理。这里，将最新笔划数据追加到第Ｓ－１个子字符数据中。即，作为构成和前一个子字符相同的子字符的笔迹处理。变量Ｓ保持该值不变。

在Ｓ１５，当断定最新笔划的笔迹与前一个以上的多个子字符数据成为一体，构成１个子字符时，即，断定与从第Ｓ－ｎ到第Ｓ－１的ｎ个子字符一起构成１个子字符时，进入S17，进行最优先处理。即，在此处理中，将从已生成的第Ｓ－ｎ个子字符到第Ｓ－１个子字符的ｎ个子字符作为与本次输入的笔迹一起构成１个子字符处理，从而归纳为１个子字符。并将变量Ｓ置为Ｓ－ｎ。

另外，在Ｓ１５，当断定最新笔划的笔迹构成与迄今生成的子字符不同的新的子字符时，便进入Ｓ１８，进行新字符处理。即，在此处理中，最新笔划数据作为构成新的子字符处理，变量Ｓ增加１。

新输入的笔划在输入到与迄今不同的行内之前，反复进行以上Ｓ１１－Ｓ１８所示的处理。

并且，当在Ｓ１３断定新输入的笔划输入到与迄今不同的行内时，便进入Ｓ１９，通过组合迄今为止输入到行内的子字符，进行文字的识别处理。在该识别处理中，要使用存储在行数据库ＤＢ４中的行数据（子字符数据）。根据识别结果得到的文字数据输给识别结果数据库ＤＢ５，进行存储。

即，在本实施例中，以在新的行进行输入为契机，通过组合迄今为止输入到行内的笔迹的子字符开始进行识别文字的处理。

然后，进入Ｓ２０，生成与本次输入的笔划对应的新的数据，并将变量i和Ｓ置为１。

下面，参照图３中的流程图，说明在Ｓ１３行的行切割判断处理的详细情况，开始，在Ｓ３１将行切割规则序号的计数１ｒ初始化为０。然后，在Ｓ３２，读取存储在行切割规则数据库ＤＢ２中的行切割规则Ｌ＿Rule［１ｒ］。另外，从行数据数据库ＤＢ４读取行数据。

在该行数据数据库ＤＢ４中，存储着根据图２中Ｓ１６～Ｓ１８的处理结果得到的子字符数据和由Ｓ１４供给的笔划数i。将该行数据应用于行切割规则Ｌ＿Rule［１ｅ］，计算用于判断是否进行行切割（是否作为向新的行输入的处理）的判断值ｇ［１ｒ］，（ｇ［１ｒ］＝Ｌ＿Rule［１ｒ］）。计算出的判断值ｇ［１ｒ］输给行切割等级数据库ＤＢ１２进行存储。

然后，进入Ｓ３３，判断是否从行切割规则数据库ＤＢ２读出了所有的行切割规则（是否１ｒ＝ＭＡＸ＿ＬＲ），并且，当存在未读出的行切割规则时，便进入Ｓ３４，使行切割序号的计数1r增加１。然后，返回到Ｓ３２，反复进行同样的处理。即，通过反复进行该处理，可以得到与所有存储在行切割规则数据库ＤＢ２中的多个行切割规则对应的判断值ｇ［1r］。

然后，进入Ｓ３５，从行切割等级数据库ＤＢ１２中读取与以上计算的各个行切割规则对应的判断值ｇ［０］～ｇ［ＭＡＸ＿ＬＲ］的值，并进行合成（加法运算），对变量Ｇ赋值。

接着，进入Ｓ３６，将在Ｓ３５计算出的合成值Ｇ与预先设定的基准值G＿1mL进行比较。当合成值Ｇ大于作为最低的基准值G＿1mL时，则断定本次输入的笔迹输入到与迄今为止的行不同的行内。与此相反，当断定合成值Ｇ等于或小于基准值G＿1mL时，则断定本次输入的笔迹输入到与迄今为止的行相同的行内。

下面，参照图４说明图２中Ｓ１５的子字符切割判断处理的详细情况。开始，在Ｓ４１，分别将表示最优先数的变量ｎ初始化为０，并将表示子字符数计数的变量ｊ置为Ｓ－１。然后，在Ｓ４２，将子字符切割规则序号计数的变量Sr初始化为０。

然后，进入Ｓ４３，读取存储在子字符切割规则数据库ＤＢ３中的子字符切割规则Ｓ＿Rule［Sr］。另外，将存储在行数据库DB４内的行数据中最新笔划的数据应用到现在读出的子字符切割规则Ｓ＿Rule［sr］中，计算用于判断是否进行子字符切割的判断值g［sr］（g［sr］＝S＿Rule［sr］）。并将根据计算结果得到判断值g［sr］输给子字符切割等级数据库DB22进行存储。关于子字符切割规则S＿

Rule[sr]的详细情况，将在后面参照图24～图30进行介绍。

在S44，将S43的处理存储到子字符切割规则数据库DB3内。当断定未得到与所有的子字符切割规则对应的判断值g[sr]时，便进入S45，使变量sr增加1，再次反复进行S43以后的处理。

这样，当断定得到了与存储在子字符切割规则数据库DB3中的所有子字符切割规则对应的判断值g[sr]时，便进入S46，从子字符切割等级数据库DB22中读取该判断值g[sr]，并在S46进行合成（加法运算）。然后，将该合成值赋予变量G。

然后，在S47，将在S46计算的合成值G与预先设定的基准值G_1mS进行比较。当断定该合成值G等于或小于最低的基准值G_1mS时，则断定现在输入的笔划是对相同子字符的输入，接着，便进入S48，将表示最优先数的变量n量为变量S－j。然后，在S49使变量j减1后，进入S50。

在S50，判断第j个子字符[j]和现在输入的笔划是否满足子字符合成条件。或者判断变量j是否小于0。即，在S49通过使变量j减1，注意前一个子字符，当该子字符和输入笔划满足子字符合成条件时或者变量j大于0时，便返回到S42，将变量sr置为0，反复进行其后的处理。

即，通过反复进行以上的动作，对j个所有的子字符应用子字符切割规则，生成判断值g[sr]。

作为在S50的子字符合成条件，例如，可以作为子字符[j－1]和子字符[j]的外接四边形是否具有交点的条件。这时，没有交点时，便返回到S42。

另一方面，在S47，当断定合成值G大于最低的基准值G_1mS

时，则断定现在输入的笔划构成与迄今的子字符不同的子字符，于是，跳过上述S48的处理，反复进行S49，S50以后的处理。即，这时，使变量n保持以前的值不变。这样，便将变量n置为最优先数。

然后，在S50，当断定子字符和输入笔划满足子字符合成条件时或者断定变量j为负值（j＜0）时，便进入S51，判断变量n的值为多大。当变量n为1时，进入S16，进行追加处理（1个最优先处理）；当变量n大于2时，进入S17，进行最优先处理；当变量n为0时，进入S18，进行新字符处理。

下面，参照图5中的流程图说明图2中S19的子字符组合识别处理的详细情况。在本实施例中，对从第i到第j的K个子字符进行处理，开始，在S61，将流程图中开始子字符计数的变量i初始化为0，在S62将结束子字符计数的变量j置为i。

然后，进入S63，从行数据数据库DB4中读取从第i到第j的K个子字符进行组合，生成1个文字数据，并输给单一文字数据数据库DB31进行存储。

接着，在S64进行文字识别处理。即，读取存储在单一文字数据数据库SB31中的单一文字数据，识别该文字，并将根据识别结果得到的数据输给识别结果数据数据库DB5进行存储。

然后，进入S65，使变量j增加1。即，使结束子字符增加1（变更为最近的子字符）。接着，进入S66，判断第i个和第j个子字符是否满足文字合成条件，或者判断变量j是否等于s。

在S65使结束子字符增加1后，当断定第i个和第j个子字符满足文字合成条件时，便返回到S63，将这两个子字符再次生成单一文字数据，反复进行文字识别处理。当变量j不等于s时，即，结束子字符

计数的值j不等于子字符数s时（结束子字符还未达到最后1个子字符时），和上述情况一样，返回到S63，反复进行同样的处理。关于文字合成条件的详细情况，将在后面参照图31～图33进行介绍。

在S66，当断定第i个和第j个子字符不满足文字合成条件时或者断定变量j等于s时（结束子字符不等于第s个子字符时），进入S67，使变量i增加1。即，只将开始子字符变更为最新的。并且在S68断定变量i不等于s时，返回到S62，将变量i置为变量j，反复进行其后的处理。即，在只将开始子字符变更为最新的子字符的状态下，反复进行同样的处理。

并且，在S68在断定开始子字符等于最新的子字符之前，反复进行同样的处理。

上面，对于输入手写文字的情况说明了进行切割和子字符切割以及在行切割后进行从子字符到文字的识别处理，为了加深理解，下面参照图6和图7说明行切割动作和子字符切割动作的具体例子。

图6示出了图2中的S13和图3中的行切割动作的具体例子。在本实施例中，用笔16在输入面板15上写出了占据L1到L4的4行文字。在行L1中写了“ここでは”的字样，在行L2中写下了“行切出しの”的字样。在行L3中写下了“说明を”的字样，在行L4中写下了“します”的字样。

例如，从行L1移向行L2时，是使用者在输入行L1的“ここでは”的字样之后进行行L2的文字“行”字的输入。设行L1的最后的文字“、”和行L2的文字“行”字的前头书写位置之间的距离大于3cm。这时，根据后面参照图10所述的行切割规则L_Rule[0]，则断定为书写出与迄今为止的行L1不同的新的行L2。

另外，行Ｌ３的“説”字的书写位置与行Ｌ２的行框的终点之间的距离L_{23}比行Ｌ２的行框的纵向宽度L_{2H}长，行Ｌ３的书写位置到行Ｌ２的行框底边的距离比行框高度L_{2H}的1／4短，并且行Ｌ２的行框高度L_{2H}和宽度W_{2H}都大于５mm，当满足这个条件时，根据后面参照图１６所述的行切割规则L_Rule[６]，则断定“説”字写到了与迄今为止的行Ｌ２不同的行Ｌ３内。

行Ｌ４的开头文字“し”的高度L_{4H}大于行Ｌ３的行框的高度L_{3H}的２／３，并且行Ｌ３的行框的底边与文字“し”的垂直方向中点的距离大于３mm。这时，根据后面参照图１４所述的行切割规则Ｌ_Rule[４]，则断定为向新的行的输入。

下面，参照图７说明图２中S15和图４中的子字符切割判断处理的具体例子。现在，假定使用者想输入平假名文字“な”。开始。输入第１笔划(Stk1)时，与该第1笔划的输入对应地生成子字符[０]，并将表示子字符的变量Ｓ置为１。

然后，当输入第２笔划时，由于第１笔划与第２笔划的输入交叉，所以，根据后面所述的子字符切割规则，识别为同一子字符。因此，将表示最优先数的变量ｎ置为１，进行追加处理。借此，将第２笔划的数据追加到由第１笔划的输入对应生成的子字符[０]内，由２个笔划生成１个子字符[０]。

当输入第３笔划时，则断定该输入是与迄今为止的子字符不同的新的子字符，并将表示最优先数的变量ｎ置为０，进行新的字符处理。借此，生成与第３笔划的数据对应的子字符[１]。并且将表示子字符数的变量s置为２。

当输入第４笔划时，则断定该子字符与子字符[０]构成同一子

字符，从而生成子字符［０］。并且，进一步断定该子字符［０］和子字符［１］构成同一个子字符，将表示最优先数的变量ｎ置为２。然后，进行最优先处理，将子字符［０］和子字符［１］合成，形成子字符［０］，并将子字符数s置为１。

下面，参照图８和图９说明图２中的S19和图５中的子字符组合识别处理的具体例子。

开始，如图８所示，当输入“明るい”的字样时，如该图所示，在本实施例的情况下，这些文字生成６个子字符ａ～ｆ。并且如图所示，只将这６个子字符ａ～ｆ中邻接的子字符组合成必要的个数。然后，根据各组合识别各个文字。

在图９的实施例中，输入“言”、“川”和、“貝”三个汉字，与此对应地生成ａ～ｅ五个子字符。适当组合这些子字符，可以根据各个组合得到文字识别候选字。

下面，参照图１０～图１６的流程图和与各个流程图对应的图17～图２３，说明在图３的行切割规则数据库DB２中存储的行切割规则Ｌ＿Rule［ｉ］。在这些图中，各个符号分别表示如下数据。

Ｐns：最新笔划的始点。

（由笔划特征数据数据库DB１得到）

Ｐpe：最新子字符的最后笔划的终点

（由行数据数据库DB４得到）

ＮＢx：最新笔划的外接长方形

（由笔划特征数据数据库DB１的笔划特征数据得到）

ＰＢx：最新子字符的最后笔划的外接长方形

（由行数据数据库DB４的行数据得到）

LBx：包围该行的全部笔迹的外接长方形

（由行数据数据库DB4的行数据得到））》

LEN_MN	5mm
LEN0	3cm
LEN1	1．5cm
LEN2	5mm
LEN3	1cm
LEN4	3mm
LEN5	5mm

图10表示行切割规则L_Rule[0]。开始，在S81，判断最新笔划的始点Pns与最新子字符的最后笔划的终点Ppe之间的距离是否大于LEN0（3cm）。当两个点的距离大于3cm时，进入S82，将判断值g[0]置为最大值MAX_GRADE；当两个点的距离小于3cm时，进入S83，将判断值g[0]置为0。在这个规则的情况下，如图17所示，利用最新笔划的始点Pns与最新子字符的最后笔划的终点Ppe之间的距离决定其等级。

判断值g[i]中数值较大的表示满足该条件（本实施例的情况是向新的行输入）的可能性大。

图11表示行切割规则L_Rule[1]。在该规则中，在S91，如图18所示，判断最新笔划的始点Pns与包围该行全部笔迹的外接长方形LBx的距离是否大于1．5cm。两者的距离大于1．5cm时，在S92将判断值g[1]置为最大值；两者的距离小于1．5cm时，在S93将判断值g[1]置为0。

图12表示行切割规则L_Rule[2]。在S101，如图19所示，

判断包围该行所有笔迹的外接长方形ＬＢx的高度和宽度是否都大于５mm（ＬＥＮ＿ＭＮ）。并且，当其长度大于５mm时，进而在S102判断包围该行全部笔迹的外接长方形ＬＢx与最新笔划的外接长方形NBx的垂直方向距离是否大于５mm（ＬＥＮ２）。

当S101和S102中的两个条件都满足时，在S103将判断值ｇ［２］置为最大值，任何一个条件不满足时，便在S104或S105将判断值g[2]置为０。

图１３表示行切割规则Ｌ＿Rule［３］。开始，在S111，如图20所示，判断包围该行的全部笔迹的外接长方形ＬＢx的高度和宽度是否都大于５mm。并且，当其长度大于５mm时，进一步在S112判断最新笔划的外接长方形ＮＢx与最新子字符的最后笔划的终点Ｐpe的距离是否大于１cm（ＬＥＮ３）。

当S111和S112中的两个条件都满足时，在S113将判断值ｇ［３］置为最大值，当任何一个条件不满足时，就在S114或S115将判断值ｇ［３］置为０。

图１４表示行切割规则Ｌ＿Rule［４］。开始，在S121，如图21所示，判断最新笔划的外接长方形ＮＢx的高度是否大于包围该行的全部笔迹的外接长方形ＬＢx的高度的２／３。满足该条件时，便进入S122，进而判断包围该行的全部笔迹的外接长方形LBx与最新笔划的外接长方形NBx的中心之间垂直方向的距离是否大于３mm（LEN４）。

两个条件都满足时，在S123将判断值ｇ［４］置为最大值，任何一个条件不满足时，就在S124或S125将判断值ｇ［４］置为０。

图１５表示行切割规则Ｌ＿Rule［５］。在该规则中，如图２２所示，在S131～S133判断３个条件。在S131，判断包围该行的全部笔

迹的外接长方形LBx的高度和宽度是否都大于5mm（LEN_MN）。在S132，判断最新笔划的外接长方形NBx是否位于距离包围该行的全部笔迹的外接长方形LBx的底边1／4以下。另外，在S133，判断最新笔划的外接长方形NBx是否位于最新子字符的最后笔划的外接长方形PBx左边约5mm（LEN5）处。

这三个条件都满足时，在S134将判断值g[5]置为最大值，任何一个条件不满足时，就在S135～S137将判断值g[5]置为0。

图16表示行切割规则L_Rule[6]。在该规则中，如图23所示，在S141～S143判断三个条件。在S141，判断包围该行的全部笔迹的外接长方形LBx的高度和宽度是否都大于5mm。在S142，判断最新笔划的外接长方形NBx是否位于距离包围该行的全部笔迹的外接长方形LBx的底边1／4以下处。在S143，判断最新笔划的外接长方形NBx是否位于最新子字符的最后笔划的终点Ppe左边包围该行全部笔迹的外接长方形LBx的高度处。

S141～S143的三个条件都满足时，在S144将判断值g[6]置为最大值，任何一个条件不满足时，就在S145～S147将判断值g[6]置为0。

将这样求出的各规则的判断值g[i]合成，并将该合成值G与基准值G_1mL进行比较，合成值大于基准值时，断定为输入新的行内，这和以上所述一样。

下面，参照图24～图30说明存储在图4中的子字符切割规则数据库DB3内的子字符切割规则S_Rule[i]。

图24表示子字符切割规则S_Rule[0]。开始，在S151判断最新笔划的水平投影与已存的子字符的水平投影的重叠是否小于任一

方的水平投影的1／2，如果小于，则进入S512，将判断值g［0］置为最大值；如果不小于，则进入S513，将判断值g［0］置为0。

图25表示子字符切割规则S_Rule［1］。在该规则中，在S514，判断最新笔划是否位于子字符左边。并且，当断定最新笔划不位于子字符左边时，就在S155将判断值g［1］置为最大值；当断定位于左边时，就进入S156，将判断值g［1］置为0。

图26表示子字符切割规则S_Rule［2］。在该规则中，在S157判断最新笔划和子字符的笔划数总和是否大于30，大于30时，就在S158将判断值g［2］置为最大值；等于或小于30时，就进入S159，将判断值g［2］置为0。

图27表示子字符切割规则S_Rule［3］。在该规则中，例如，如图29所示，进行子字符的切割判断。如该图所示，令前一笔划的外接长方形的纵向和横向的长度中大的一方为PB_SIZE，前一笔划的外接长方形的始点为P_SP，最新输入笔划的外接长方形的横向宽度为B_W，最新输入笔划的最左边的点为L_P，行框的纵向宽度为LH。

在S311，判断PB_SIZE是否小于LH／5。即，判断前一笔划的外接长方形的纵向和横向的长度中大的一方的长度PB_SIZE是否小于行框的纵向宽度HL的1／5。

在S312，判断最新输入笔划的外接长方形的横向宽度B_W是否大于行框的纵向宽度LH。并且在S313判断前一笔划的外接长方形的始点P_SP的x坐标P_SPx是否大于最新输入笔划最左侧的点L_P的x坐标L_Px，即，判断始点P_SP是否位于点L_P的右侧。

S311～S313中的三个条件都满足时，进入S314，将判断值g[3]置

为最大值；任何一个条件不满足时，就进入S315，将判断值g[3]置为0。

图28表示子字符切割规则S_Rule[4]。在该规则中，如图30所示，令最新输入笔划的横向宽度为BW，行框的纵向宽度为LH，最新输入笔划的外接长方形的y坐标的最大值为MX_BY，行框（行的外接长方形）的y坐标的最小值为MN_LY。

在S321，判断最新输入笔划的横向宽度BW是否大于3倍行框的纵向宽度LH。在S322，判断最新输入笔划的外接长方形的y坐标的最大值MX_BY是否小于行的外接长方形的y坐标最小值MN_LY与1／3行框的纵向宽度LH之和。

S321和S322中的两个条件都满足时，进入S323，将判断值g[4]置为最大值；任何一个条件不满足时，就进入S324，将判断值g[4]置为0。

该子字符切割规则S_Rule[i]的判断值g[i]越大，满足该条件的可能性也越大。将其合成值G与基准值G_1mS进行比较，大于基准值时，最后便切割出子字符。

图31～图33表示图5中的S66中文字合成条件判断处理的详细情况。

在图31的实施例中，进行根据距离判断从子字符[i]到子字符[j]的子字符是否构成1个文字的处理。在S161，将表示第1个被注意的子字符的变量t置为i，在S162将表示第2个被注意的子字符的变量S置为t+1（因此，开始时s=i+1）。然后，在S163将子字符[t]到子字符[s]之间的距离（开始时，是子字符[i]与子字符[i+1]之间的距离）设定为变量D。

在S164，判断该距离D是否大于2倍的行框纵向宽度。当距离D等于或小于2倍的行框纵向宽度时，则断定这两个子字符应合成（构成1个文字）。并且，进入S165，判断变量S是否等于变量j，即，判断应被注意的子字符中是否已成为最后的子字符[j]。当未达到最后的子字符时，进入S166，使变量S增加1（只把第2个被注意的子字符变更为新的），然后再次返回到S163，反复进行同样的处理。

在S165，当断定第2个被注意的子字符达到了最后的子字符[j]时，便进入S167，判断变量t是否等于变量j。即，判断第1个被注意的子字符[t]是否已达到最后的子字符[j]。当断定未达到最后的子字符[j]时，进入S168，使变量t增加1。即，使第1个被注意的子字符变更为下一个子字符。然后，返回到S162，设定变量S比在S168设定的变量t大1，并反复进行S163以后的处理。

在S167，当断定变量t等于变量j时（第1个被注意的子字符达到最后被注意的子字符时），转移到图32所示的处理。

在S164，当断定距离D大于2倍的行框的纵向宽度时，则断定将子字符合成为1个文字的条件不成立。

这样，在图31的处理中，判断子字符[i]与子字符[j]间的距离是否大于2倍的行框的纵向宽度，如果不大于，则断定将子字符合成为1个文字的条件成立，从而转移到图32所示的处理。

在图32中，开始，在S181将包含子字符[i]～子字符[j]的所有外接长方形的外接长方形设定为Bx。在S182，将在S181设定的外接长方形Bx的横向宽度设定为Bx_W，并在S183将外接长方形Bx的纵向宽度设定为Bx_H。

在S184，判断在S182和S183设定的横向宽度Bx_W与纵向宽度Bx_H之

比B_{xw}／B_{xH}是否小于3。当该比值小于3时，则断定子字符满足合成条件；当该比值大于3时，则断定不满足合成条件。

这样，在S181～S184就是判断包围子字符[i]～子字符[j]的全部笔迹的外接长方形的宽B_{xw}与高B_{xH}之比是否小于3，当小于3时，便进行图33所示的判断。

在图33中，开始，在S191计算子字符[i]到子字符[j]的笔划数之和，并将该值设定为Nk。然后，在S192判断该值Nx是否大于30，当小于30时，则断定子字符满足合成条件；当大于30时，则断定不满足将子字符合成的条件。

这样，在图33所示的处理中就是判断子字符[i]～子字符[j]的笔划数总和是否大于30。

当断定图31～图33所示的三个条件都满足时，最后则断定将子字符合成为1个文字的条件成立，当断定任何一个条件不满足时，则断定将子字符合成为1个文字的条件不成立。

这样，当将子字符合成的条件成立时（真时），便从图5的S66返回到S63；当断定将子字符合成的条件不满足时（伪时），就从S66进入S67。

下面，参照图34～图43说明各数据库中的格式。

图34表示笔划特征数据库DB1中的格式的实施例。在该实施例中，笔划特征等级矢量go～gn配置在其前端。该笔划特征等级矢量gi表示与预先准备的该笔划的1个笔划形状图形（T1，…，Tn）的近似度。

特征点坐标Co～Cm分布在其后，这些特征点坐标Ci表示始点、终点；中点或凸点等的坐标。

接着，排列的是现笔划与下一笔划的始点和终点的方位dir_hh～dir_tt。dir_hh表示从现笔划的始点到下一笔划的始点的方位。dir_ht表示从现笔划的始点到下一笔划的终点的方位。dir_th表示从现笔划的终点到下一笔划的始点的方位。另外，dir_tt表示从现笔划的终点到下一笔划的终点的方位。

然后，分布的是现笔划和下一笔划的始点与终点的距离len_hh～len_tt。len_hh表示从现笔划的始点到下一笔划的始点的距离，len_ht表示从现笔划的始点到下一笔划的终点的距离。len_th表示从现笔划的终点到下一笔划的始点的距离，len_tt表示从现笔划的终点到下一笔划的终点的距离。

再下面，分布的是外接长方形KB的顶点坐标KBx1，KBY1，KBx2，KBY2。

然后，分布的是元素数en，接着就是元素串eo～e_{en-1}。元素ei是使笔迹近似于直线的数据，由始点坐标（xi、yi）、方位Di、元素长度Li和表示笔的升降的标志i构成。

图35表示在单个文字数据数据库DB31中存储的单个文字数据的格式。如图所示，首先排列的是单个文字的外接长方形的顶点坐标SBX1，SBY1，SBX2，SBY2。然后分布的是该单个文字的笔划数Kn，接着便是该单个文字的各笔划的全长K1。最后，分布的是笔划特征数据串Ko，K1，…，K_{Kn-1}。

图36表示行数据库DB4的格式。如图所示，行数据按顺序排列，最后带有文件结束符号。

图37表示图36中各行数据的格式。开始，排列的是表示行ID的Lid，然后是笔划数Lnk。接着，排列子字符数Lns，然后排列外接

长方形的顶点坐标LBX1，LBY1，LBX2，LBY2。最后排列的是子字符数据串S0，S1，…，S_{Lns-1}。

图 3 8 表示识别结果数据库DB 5 的格式。如图所示，在该数据库中，识别结果数据按顺序排列，最后，带有文件结束符号。

并且，各识别结果数据的格式结构示于图 3 9 。如图所示，开始，先排列开始子字符序号StSC，然后排列结束子字符序号EdSc。该开始子字符序号表示合成识别文字的子字符串的最初的子字符是该行的第几个子字符，结束子字符序号表示构成识别文字的子字符串的最后 1 个子字符是该行的第几个子字符。

然后，排列的是文字候选数rn，最后排列文字候选子符串R0，R1，…，R_{rn-1}。该文字候选符Ri由识别结果文字代码Ci和识别等级gi构成。

图 4 0 表示行切割规则数据库DB 2 的格式。如图所示，开始，先排列规则数MAX_LR，然后是按顺序排列规则L_Rule[0]，L_Rule[1]，…,L_Rule[MAX_LR_1]。

图 4 0 中的各个规则L_Rule[i]的格式示于图 4 1 。即，开始先排列指令代码i，然后操作数ni，最后，顺序排列操作 0 ～操作ni-1。

图 4 2 表示子字符切割规则数据库DB 3 的格式。开始，先排列规则数MAX_SR，然后，按顺序排列各规则S_Rule[0]，S_Rule[1]，…，S_Rule[MAX_SR-1]。

另外，各规则S_Rule[i]的记录格式如图 4 3 所示。开始，先排列指令代码i，然后是操作数mi，最后，顺序排列操作0～操作mi-1。

在图 2 的实施例中，在S13断定进行向不同的行输入时，便进入S19，将迄今生成的子字符组合后进行文字的识别处理，但是，该文

字识别处理也可以在每进行 1 个笔划输入后进行。图 4 4 表示这样处理时的流程图。

S211～S220的处理，和图 2 中S11～S20的处理一样。在本实施例中，除了这些处理之外，分别在S217和S218（图 2 中的S17和S18）之后还附加了S231和S232的处理。

即，在本实施例中，当在S215断定已输入了新的子字符和在S218断定进行新字符处理时，便进入S232，进行子字符组合识别处理。

在S232中，作为识别对象的子字符，是从子字符［ 0 ］到在S218中作为新的子字符而附加的子字符［S-1］之前的子字符［S-2］的S-1个子字符。

与此相反，作为在S219进行的识别对象的子字符，由于这里的识别是在向新的行进行输入后进行的识别，所以，是从子字符［ 0 ］到子字符［S-1］的S个子字符。

图 4 5 示出了S232（S219时也一样）的子字符组合识别处理的更详细的处理例子。其基本的处理和图 5 所示的情况相同，但是，在图 4 5 中，开始时，在S261将变量i设定为es。该es表示作为识别对象的子字符群的脚标的最大值。即，子字符［es］表示最新的子字符。

然后，进入S262，将变量j设定为变量i，并进入S263。在S263，将从子字符［i］到子字符［j］的子字符进行合成，生成 1 个文字数据，并将生成的 1 个文字数据输入单个文字数据库DB31进行存储。

在S264，读取存储在单个文字数据库DB31中的单个文字数据，进行文字识别处理，并将根据文字识别的结果得到的文字输给识别结果数据数据库DB 5 进行存储。

然后，进入S265，便变量i减少 1 。即，将识别对象从最新的子

字符转移到前一个（旧的）子字符上。并且，进入S266，判断子字符［i］和子字符［j］是否满足文字合成条件或者判断变量i是否与ss相等。ss表示作为识别对象的子字符群的脚标的最小值。

在S266，当断定子字符［i］和子字符［j］满足文字合成条件时，便返回到S263，将子字符［i］～子字符［j］合成，反复进行生成1个文字数据的处理。作为识别对象的子字符［i］不等于最早的子字符［ss］时也一样，返回到S263，反复进行其后的处理。

当断定子字符［i］和子字符［j］不满足文字合成条件时或者作为识别对象的子字符［i］等于最早的子字符［ss］时，则进入下一步（在S232情况下，则结束，在S219的情况下，则进入S220）。

即，在图所示的实施例中，按照子字符［i］从旧的到新的顺序进行识别处理，在图45的实施例中，则从新的到旧的顺序进行处理。

当然，S219的情况也和S232的情况一样，也可以按照从最新的到旧的顺序进行识别处理。或者反过来，不论哪种情况，都可以按照从旧的到新的顺序进行识别处理。

另一方面，在图44中的S215，当断定现在输入的笔划应该与已经生成的多个子字符进行组合时，便进入S217，进行最优先处理。并且进入S231，使结束子字符为从子字符［S+1］到子字符［S+n］间的子字符（在S217从设定为S=S-n之前的子字符［S-（n-1）］到子字符［s］的子字符）时的识别结果无效。即，由于这些识别结果通过进行最优先处理归结为1个子字符，所以使之无效。

例如，如图46所示，现在设有子字符A，B，C。这时，作为识别结果，有对子字符A的识别结果、对子字符A，B的识别结果、对子字符B的识别结果、对子字符A，B，C的识别结果、对子字符

B，C的识别结果和对子字符C的识别结果。

在此状态下，输入新的笔划并在S215断定最优先数n为2时，由于n是最优先的现存子字符的个数，所以，利用子字符B，C和新输入的笔划生成子字符D（新的子字符B）。这时，不需要与子字符B和C对应的识别结果。即，如图47所示，由于现在的情况是存在（剩下）子字符A和子字符D，所以，作为识别结果，存在对子字符D的识别结果、对子字符A、D的识别结果和对子字符A的识别结果。因此，不需要图46所示的对子字符A的识别结果以外的识别结果（在图46中，在范围E中所示的识别结果）。所以，这些识别结果是无效的。

如图44所示，每次进行新的输入，如果进行子字符的识别处理，则如图2所示的实施例那样，与每当向新的行进行输入时才开始进行识别处理的情况相比，可以迅速地完成识别处理。

按照本发明所述的手写文字输入方法，当输入笔划时，就根据该笔划的特征、子字符和行切割规则判断输入的行，当输入的行与以前输入的行为同一行时，就根据笔划的特征、子字符和子字符切割规则，进行子字符的判断，所以，可以正确地切割子字符。

按照本发明所述的手写文字输入方法，当变更文字的行时，是一并将子字符组合进行文字识别的，所以，可以正确地进行文字识别。

另外，按照本发明所述的手写文字输入方法，以变更文字的行为契机，不仅进行文字的识别而且切割新的子字符时，是逐次将子字符组合后进行文字识别的，所以，可以更迅速地进行识别。

图 1

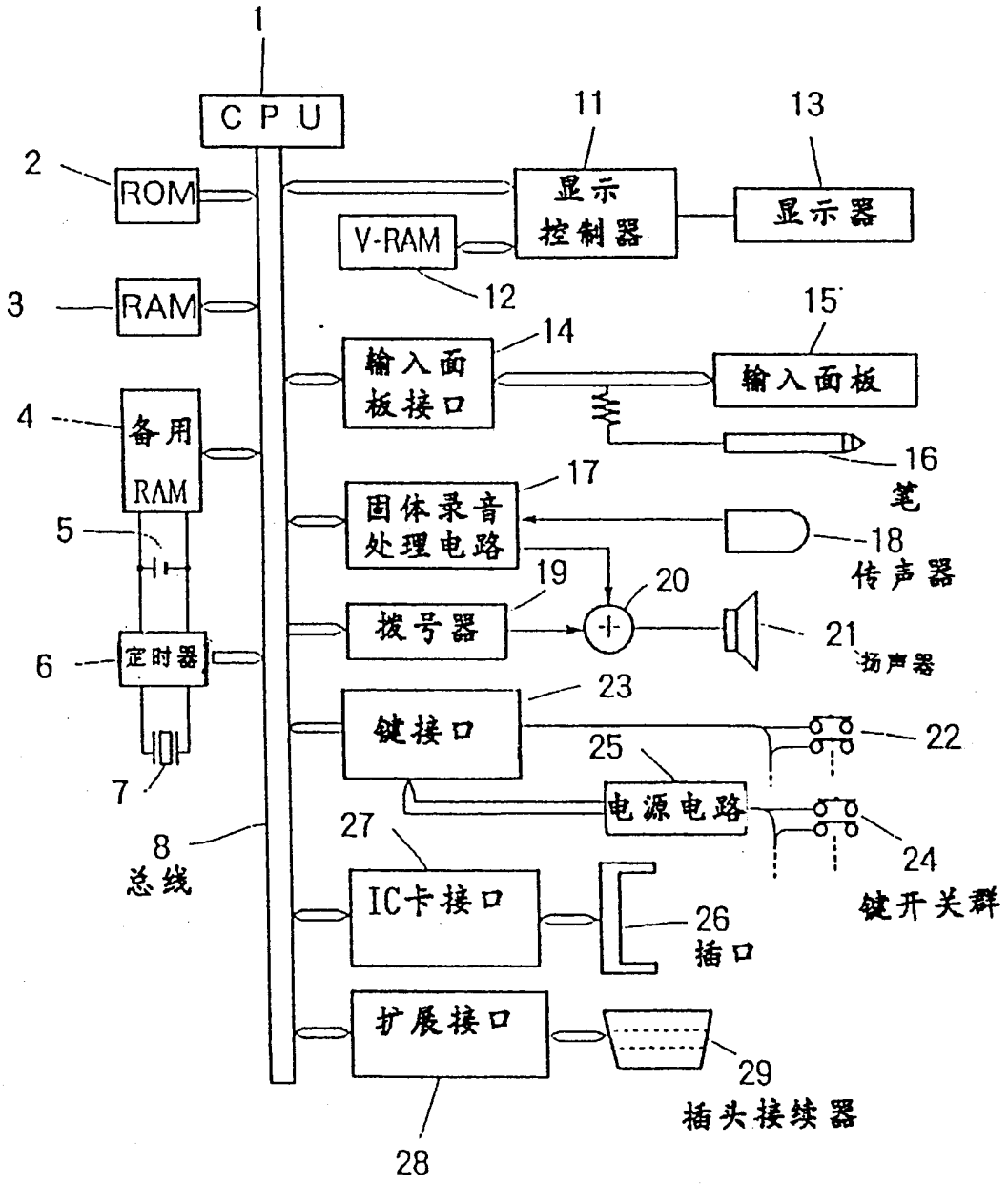

1
CPU
2
ROM
3
RAM
4
备用
RAM
5
6
定时器
7
8
总线
11
显示
控制器
V-RAM
12
13
显示器
14
输入面
板接口
15
输入面板
16
笔
17
固体录音
处理电路
18
传声器
19
拨号器
20
21
扬声器
23
键接口
22
25
电源电路
24
键开关群
27
IC卡接口
26
插口
扩展接口
29
插头接续器
28

图 2

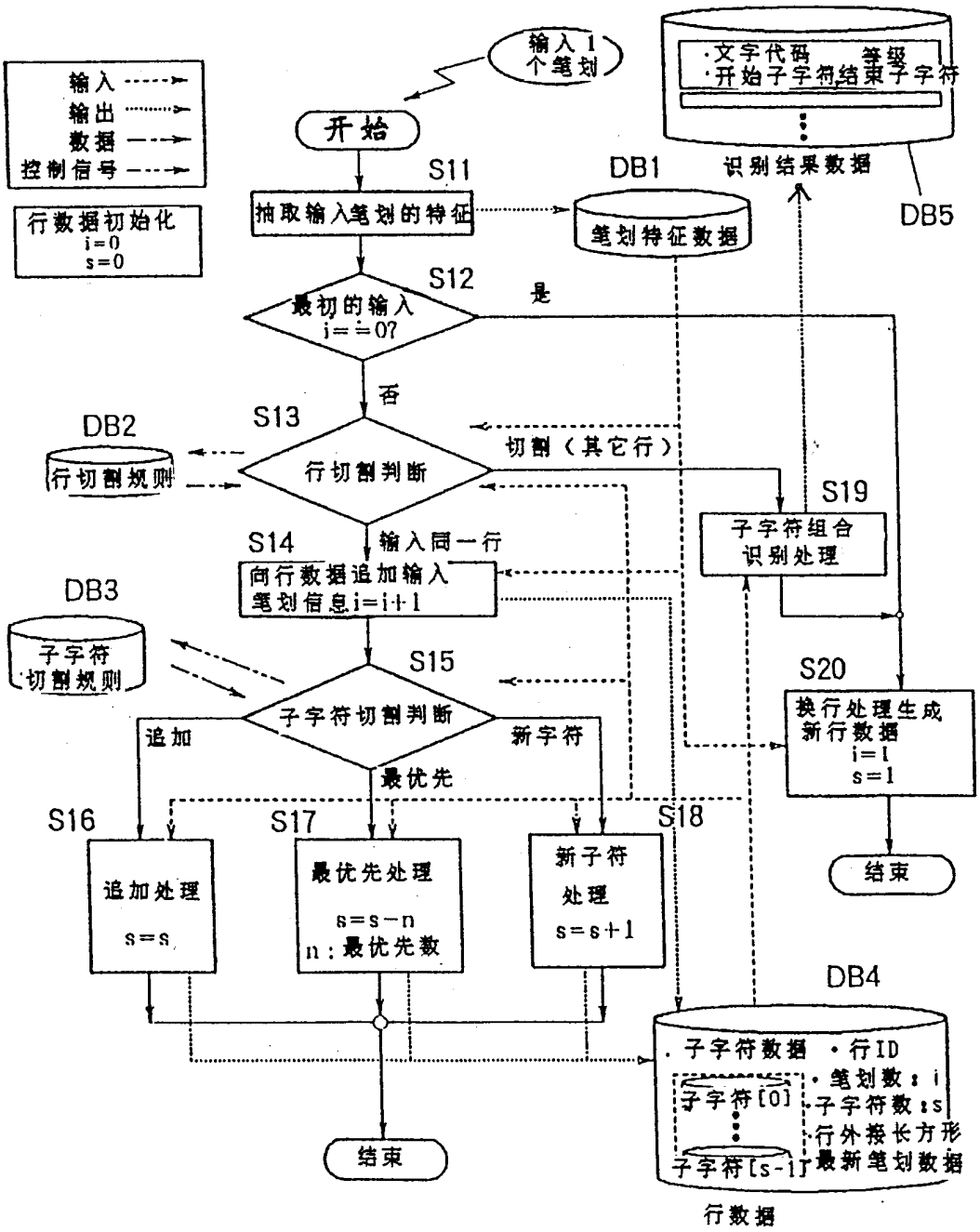

输入
输出
数据
控制信号
行数据初始化
i=0
s=0
输入1个笔划
开始
S11
抽取输入笔划的特征
DB1
笔划特征数据
·文字代码 等级
·开始子字符 结束子字符
识别结果数据
DB5
S12
最初的输入
i==0?
是
否
DB2
行切割规则
S13
行切割判断
切割（其它行）
S19
子字符组合
识别处理
S14
输入同一行
向行数据追加输入
笔划信息 i=i+1
DB3
子字符
切割规则
S15
子字符切割判断
追加
新字符
最优先
S20
换行处理生成
新行数据
i=1
s=1
S16
追加处理
s=s
S17
最优先处理
s=s-n
n：最优先数
S18
新子符
处理
s=s+1
结束
DB4
·子字符数据 ·行ID
·笔划数：i
子字符[0]
子字符数：s
行外接长方形
子字符[s-1]
最新笔划数据
行数据
结束

图 3

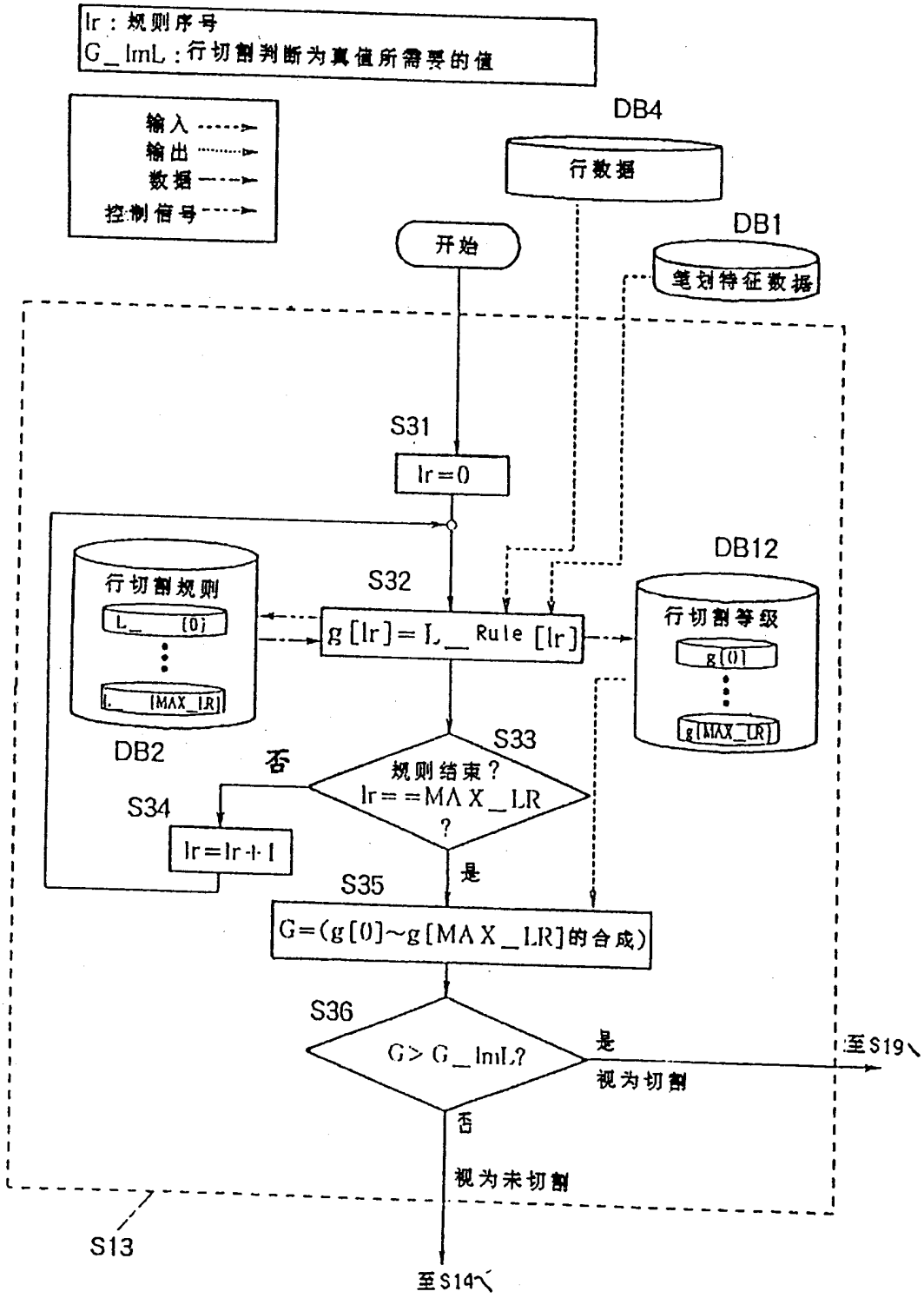

lr：规则序号
G_lmL：行切割判断为真值所需要的值
输入
输出
数据
控制信号
DB4
行数据
DB1
笔划特征数据
开始
S31
lr=0
DB12
行切割规则
L_ [0]
L_ [MAX_LR]
S32
g[lr]=L_ Rule [lr]
行切割等级
g[0]
g[MAX_LR]
DB2
S33
否
规则结束？
lr==MAX_LR
？
S34
lr=lr+1
是
S35
G=(g[0]~g[MAX_LR]的合成)
S36
G>G_lmL?
是
至S19
视为切割
否
视为未切割
S13
至S14

图 4

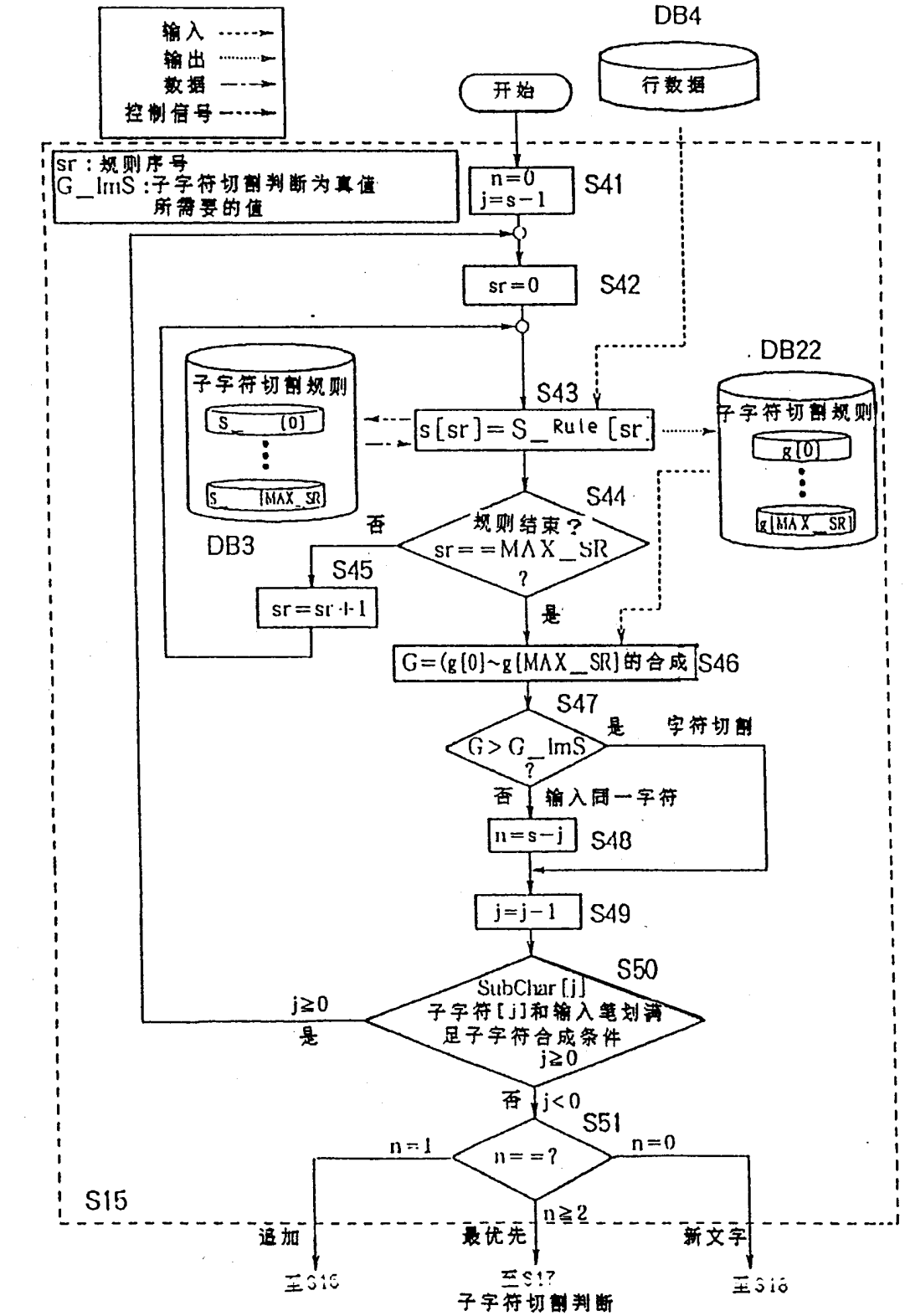

输入
输出
数据
控制信号
DB4
行数据
开始
sr：规则序号
G_ImS：子字符切割判断为真值所需要的值
n=0
j=s-1
S41
sr=0
S42
DB22
子字符切割规则
S43
s[sr]=S_Rule[sr]
子字符切割规则
g[0]
g[MAX_SR]
S_[0]
S_[MAX_SR]
DB3
S44
规则结束？
sr==MAX_SR
？
否
是
S45
sr=sr+1
G=(g[0]~g[MAX_SR]的合成
S46
S47
G>G_ImS
？
是
字符切割
否
输入同一字符
n=s-j
S48
j=j-1
S49
S50
SubChar[j]
子字符[j]和输入笔划满足子字符合成条件
j≥0
j≥0
是
否
j<0
S51
n==？
n=1
n=0
n≧2
S15
追加
最优先
新文字
至S16
至S17
子字符切割判断
至S16

图 5

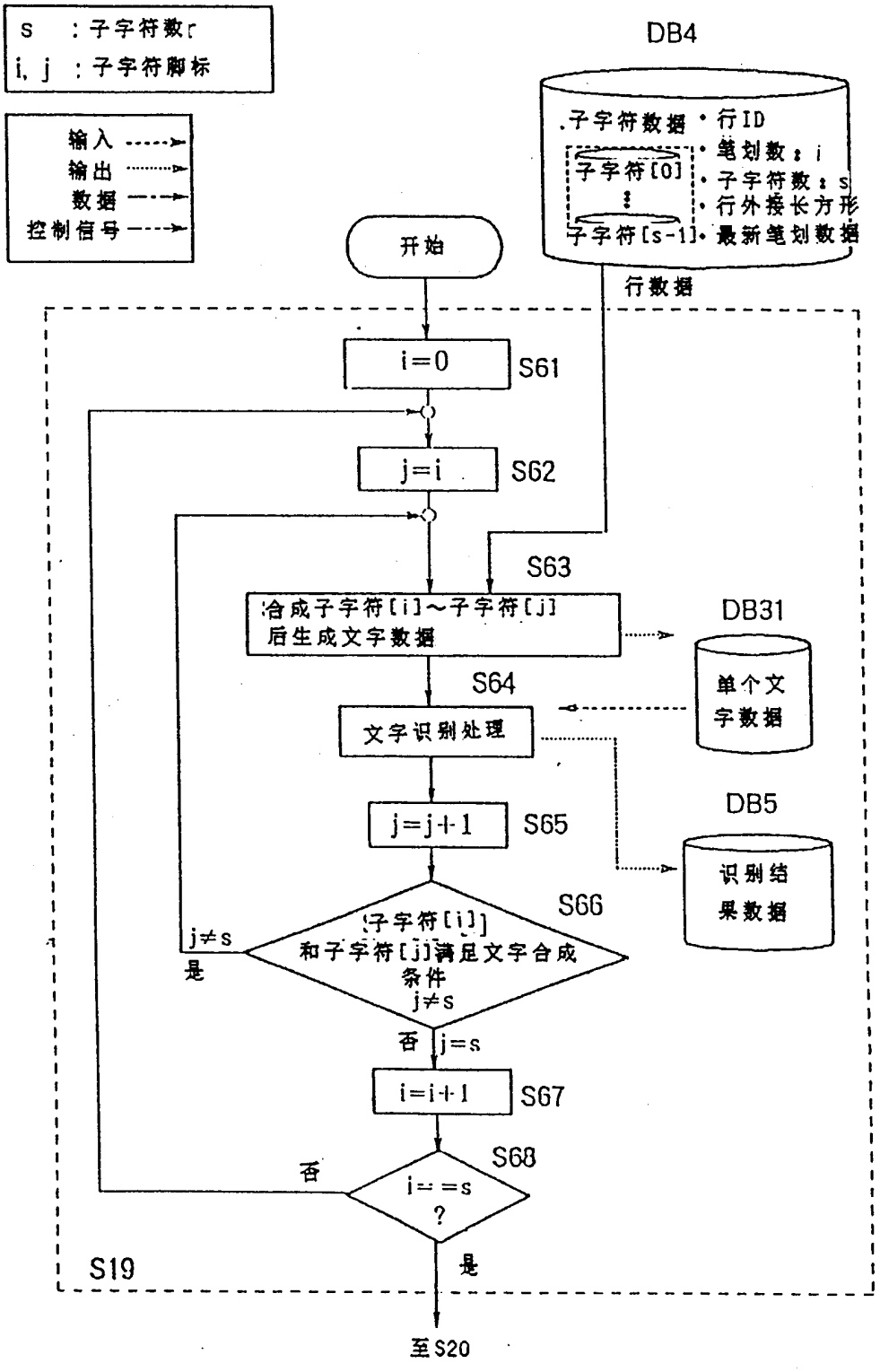
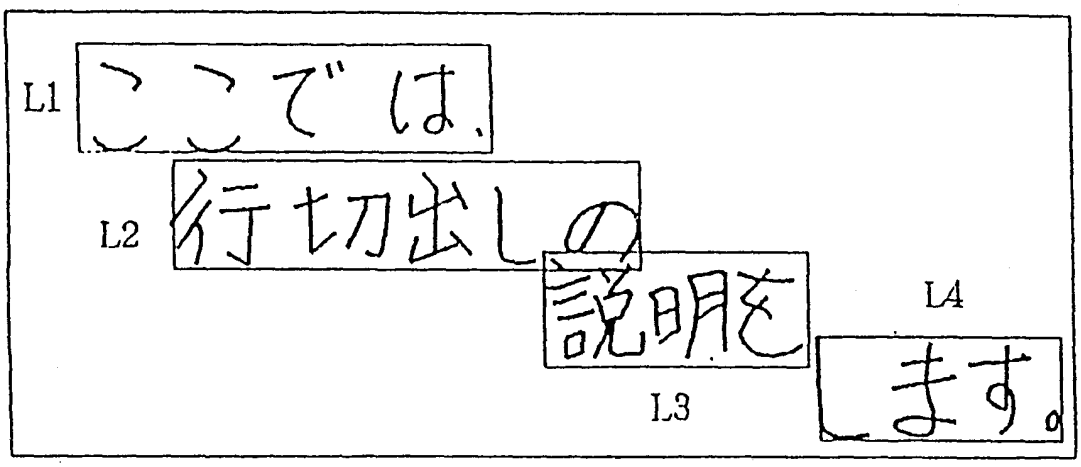
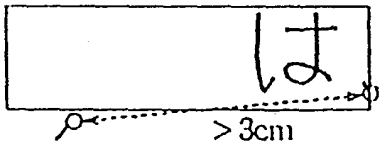

s ：子字符数
i, j ：子字符脚标
输入
输出
数据
控制信号
DB4
子字符数据
子字符[0]
子字符[s-1]
·行ID
·笔划数：i
·子字符数：s
·行外接长方形
·最新笔划数据
开始
行数据
i=0
S61
j=i
S62
S63
合成子字符[i]～子字符[j]后生成文字数据
DB31
单个文字数据
S64
文字识别处理
j=j+1
S65
DB5
识别结果数据
S66
子字符[i]和子字符[j]满足文字合成条件
j≠s
j≠s
是
否
j=s
i=i+1
S67
S68
否
i==s ?
是
S19
至S20

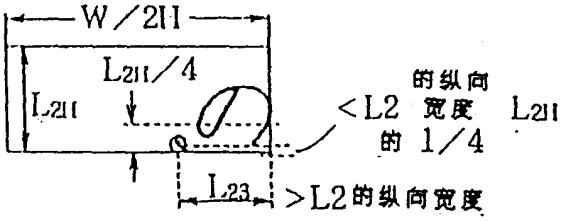
图 6



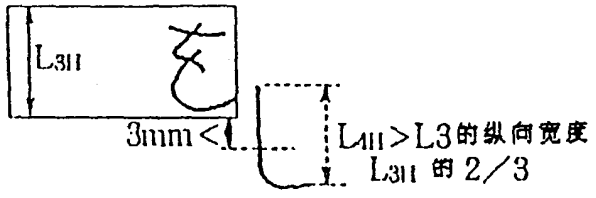
L1, L2 → 按L_Rule[0]切割



L2, L3 → 按L_Rule[6]切割

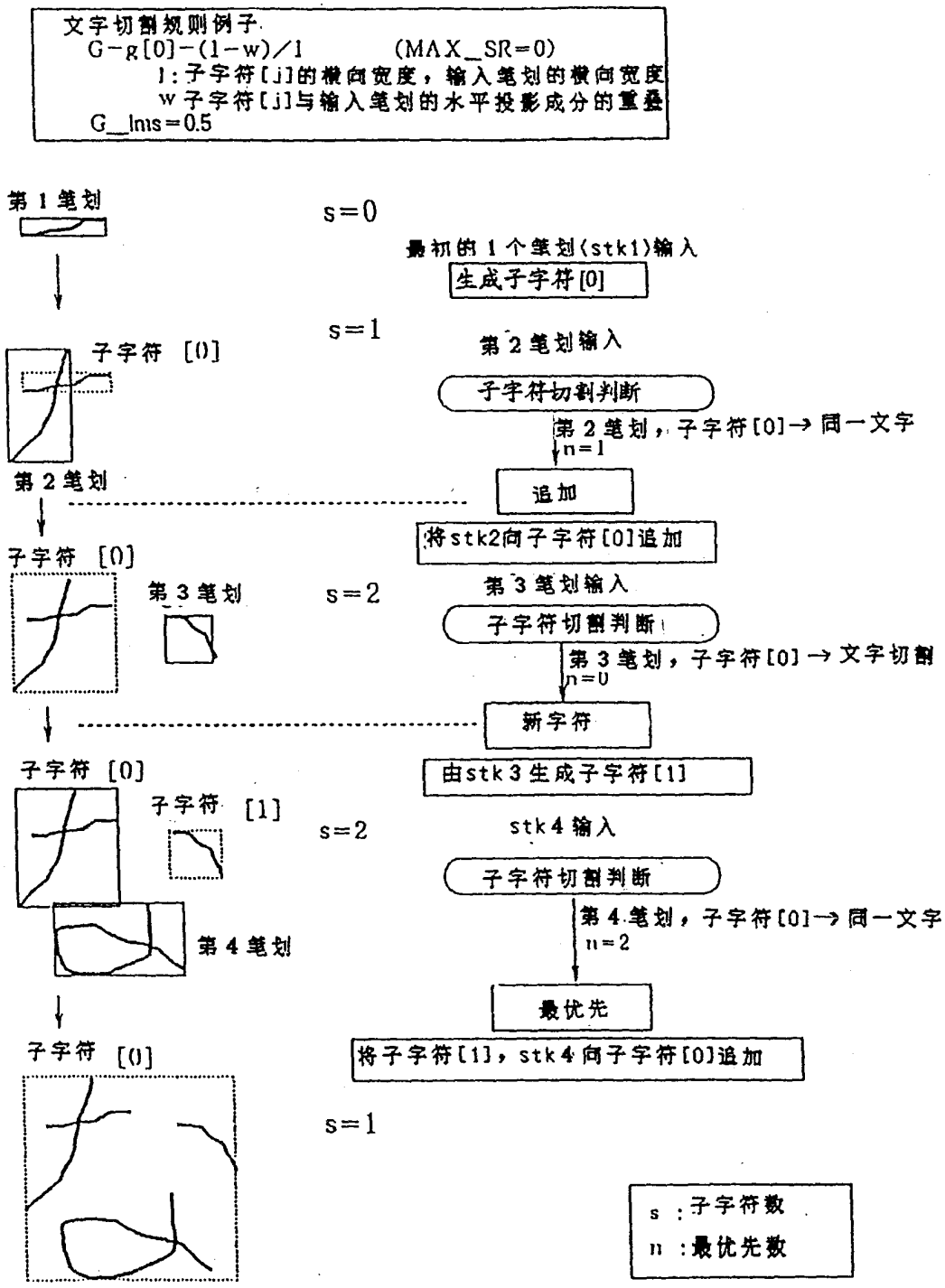


L3, L4 → 按L_Rule[4]切割



行切割动作例子

图 7



子字符切割动作例子

图 8

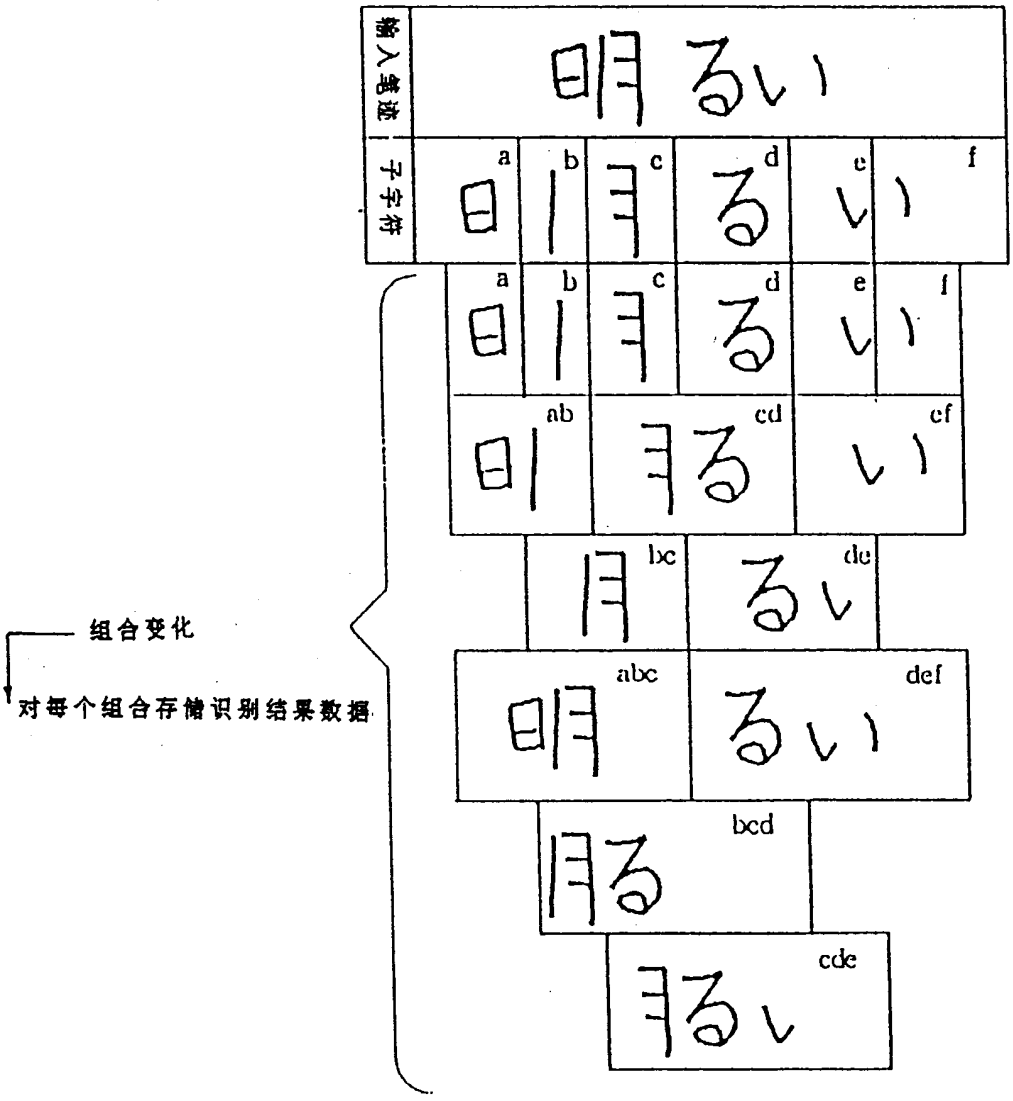

输入笔划
单字符
a
b
c
d
e
f
ab
cd
ef
bc
de
abc
def
bcd
cde
组合变化
对每个组合存储识别结果数据

图 9

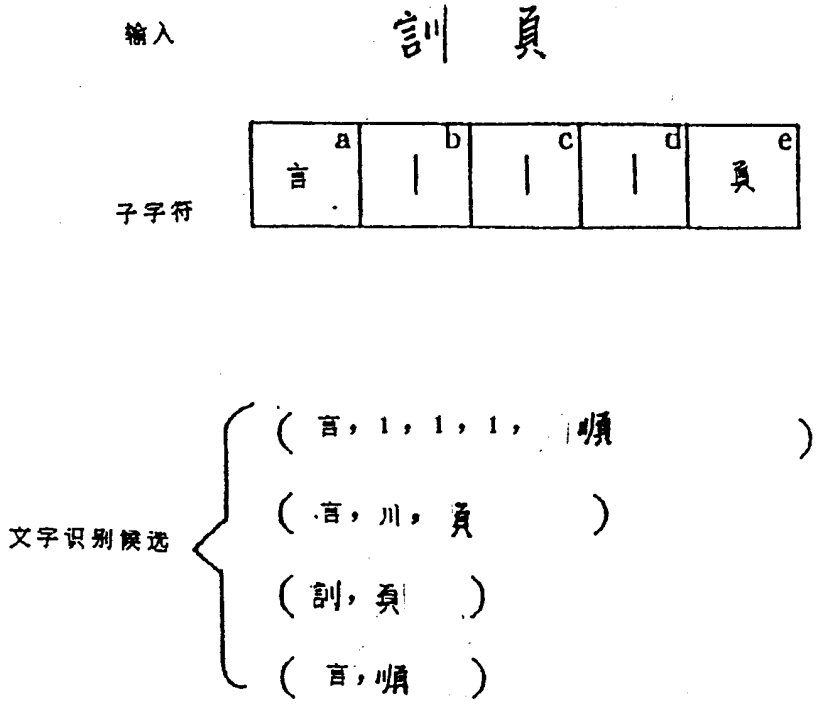


图 10

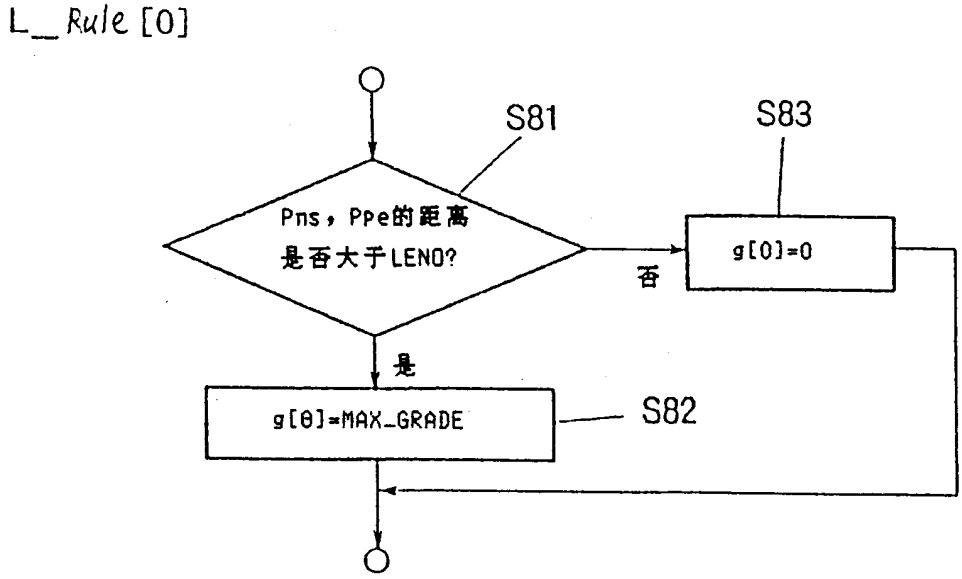

L_Rule[0]
S81
S83
Pns，Ppe的距离
是否大于LEN0?
否
g[0]=0
是
g[0]=MAX_GRADE
S82

图 11

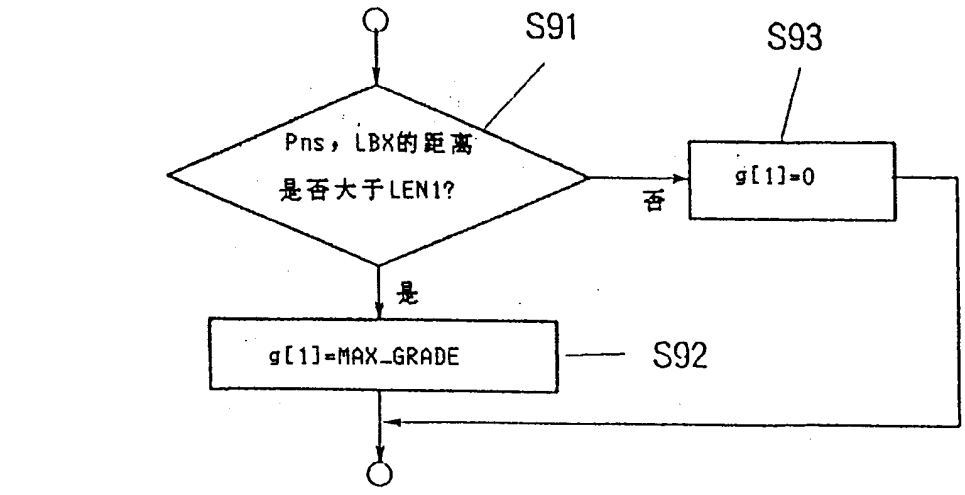

L_ [1]
S91
S93
Pns，LBX的距离
是否大于LEN1?
否
g[1]=0
是
g[1]=MAX_GRADE
S92

图 12

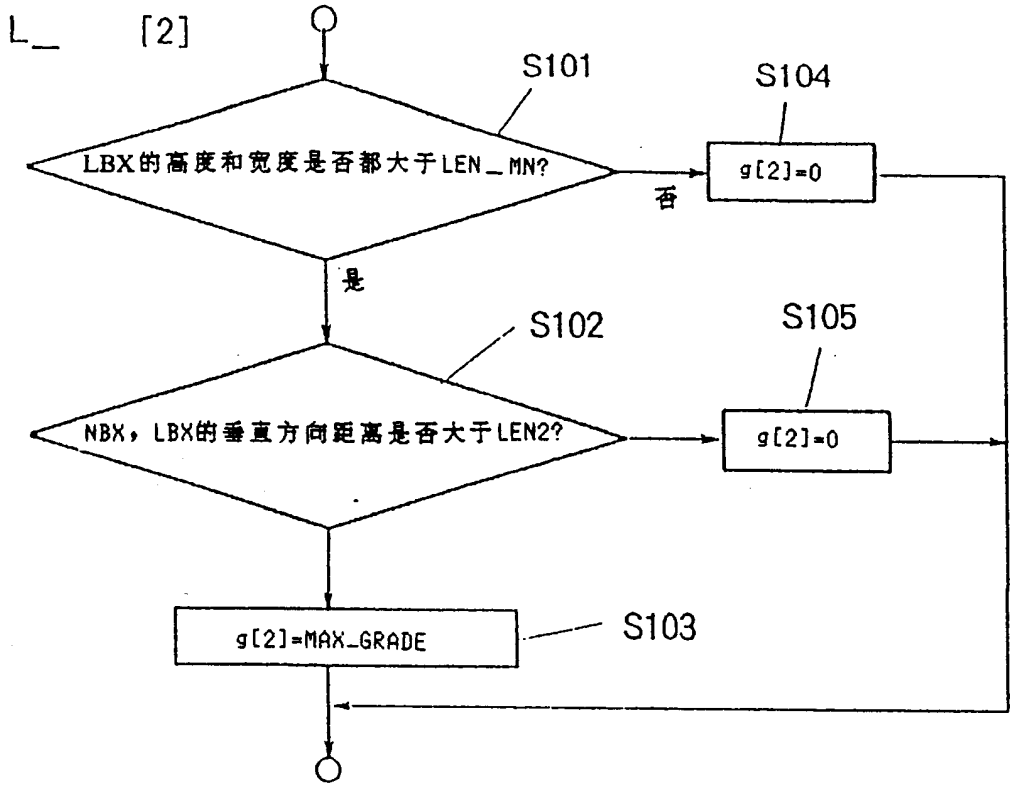

L_ [2]
S101
LBX的高度和宽度是否都大于LEN_MN?
否
S104
g[2]=0
是
S102
NBX，LBX的垂直方向距离是否大于LEN2?
S105
g[2]=0
g[2]=MAX_GRADE
S103

图 13

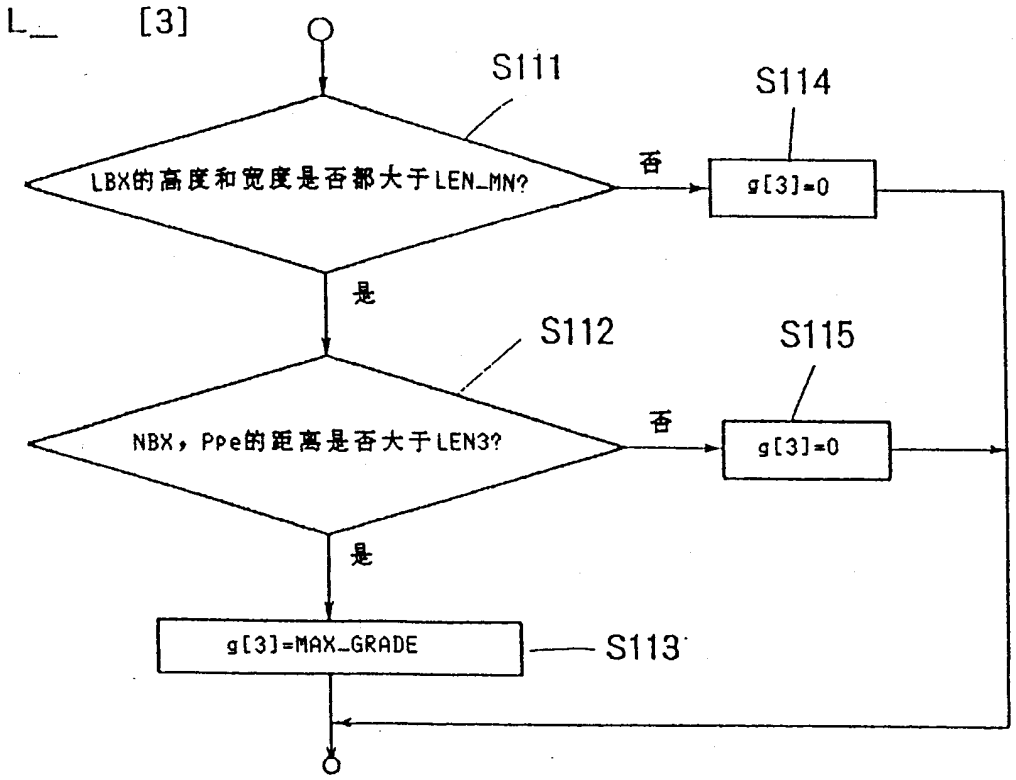

L_ [3]
S111
LBX的高度和宽度是否都大于LEN_MN?
否
S114
g[3]=0
是
S112
NBX，Ppe的距离是否大于LEN3?
否
S115
g[3]=0
是
g[3]=MAX_GRADE
S113

图 14

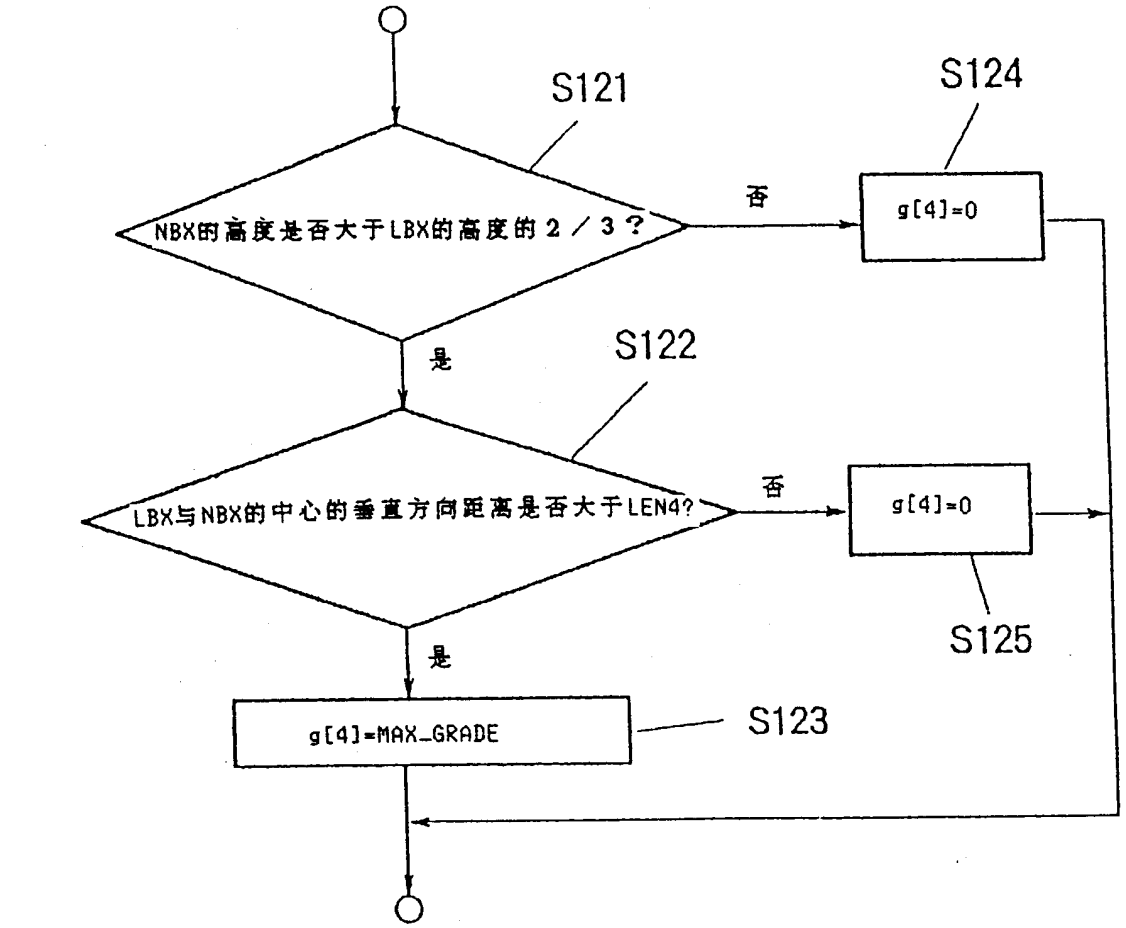

L_ Rule [4]
S121
NBX的高度是否大于LBX的高度的2／3？
否
S124
g[4]=0
是
S122
LBX与NBX的中心的垂直方向距离是否大于LEN4?
否
g[4]=0
S125
是
g[4]=MAX_GRADE
S123

图 15

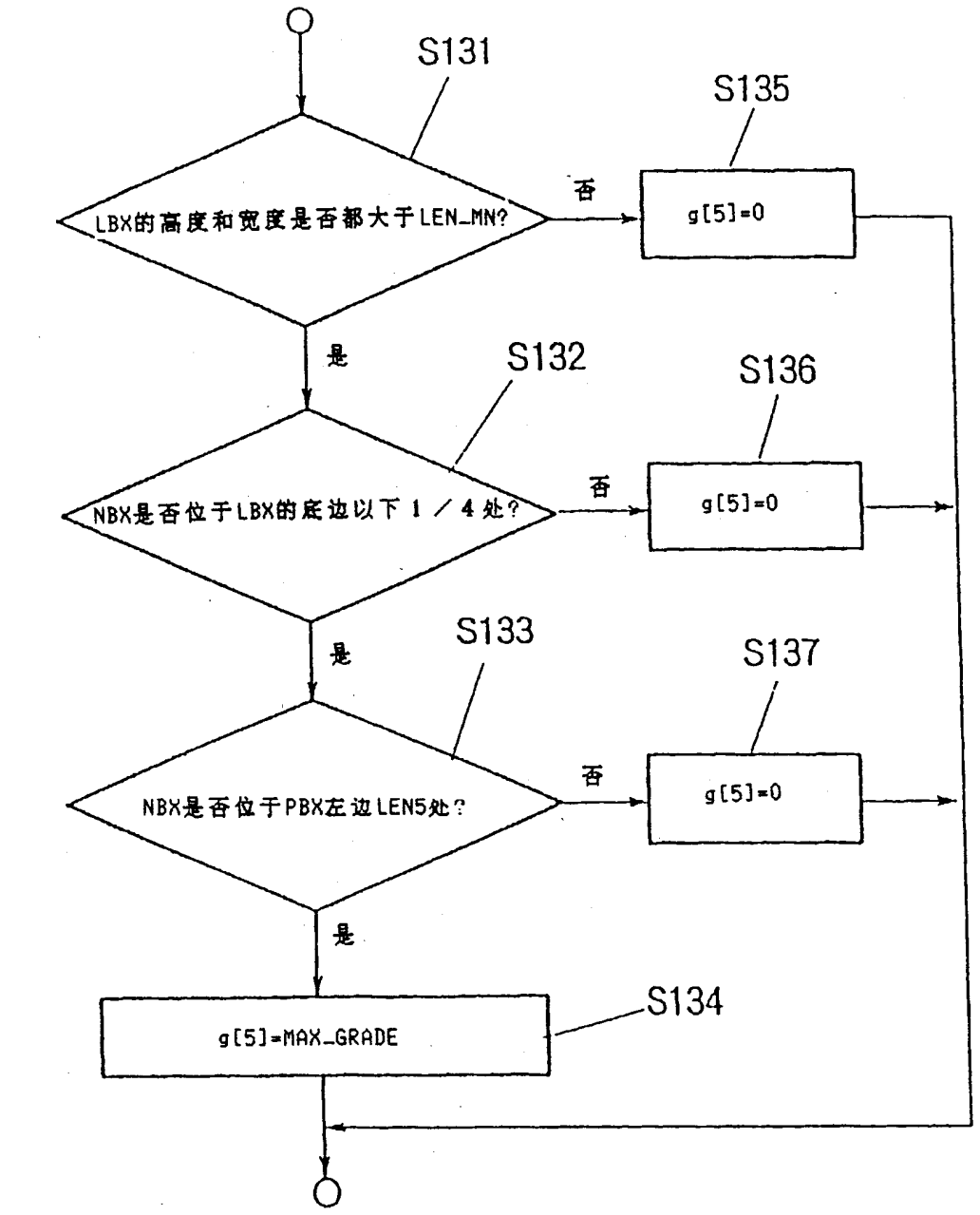

L_ Rule[5]
S131
LBX的高度和宽度是否都大于LEN_MN?
否
S135
g[5]=0
是
S132
NBX是否位于LBX的底边以下1／4处?
否
S136
g[5]=0
是
S133
NBX是否位于PBX左边LEN5处?
否
S137
g[5]=0
是
g[5]=MAX_GRADE
S134

图 16

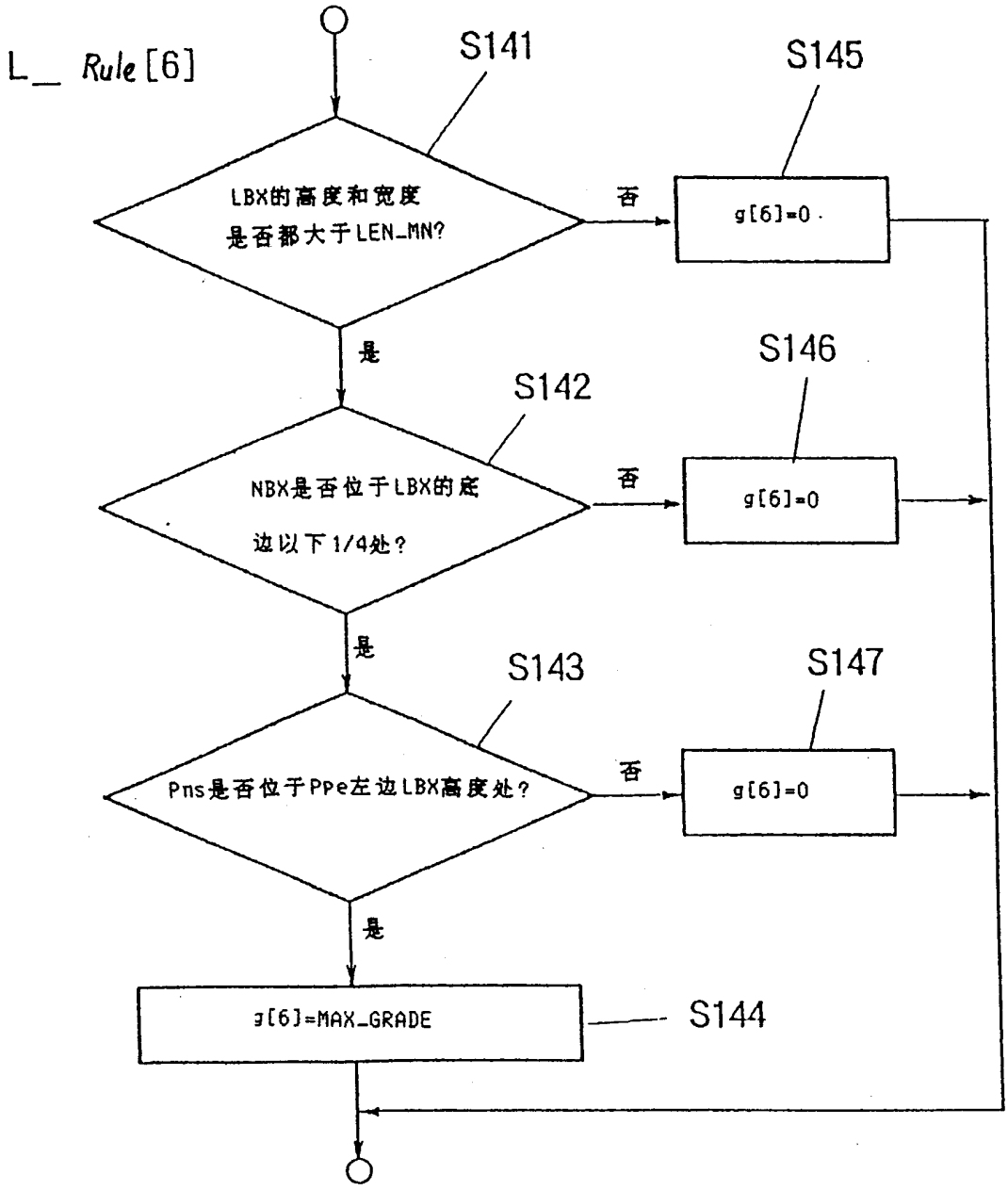

L_ Rule[6]
S141
LBX的高度和宽度
是否都大于LEN_MN?
否
S145
g[6]=0
是
S142
S146
NBX是否位于LBX的底
边以下1/4处?
否
g[6]=0
是
S143
S147
Pns是否位于Ppe左边LBX高度处?
否
g[6]=0
是
g[6]=MAX_GRADE
S144

图 17

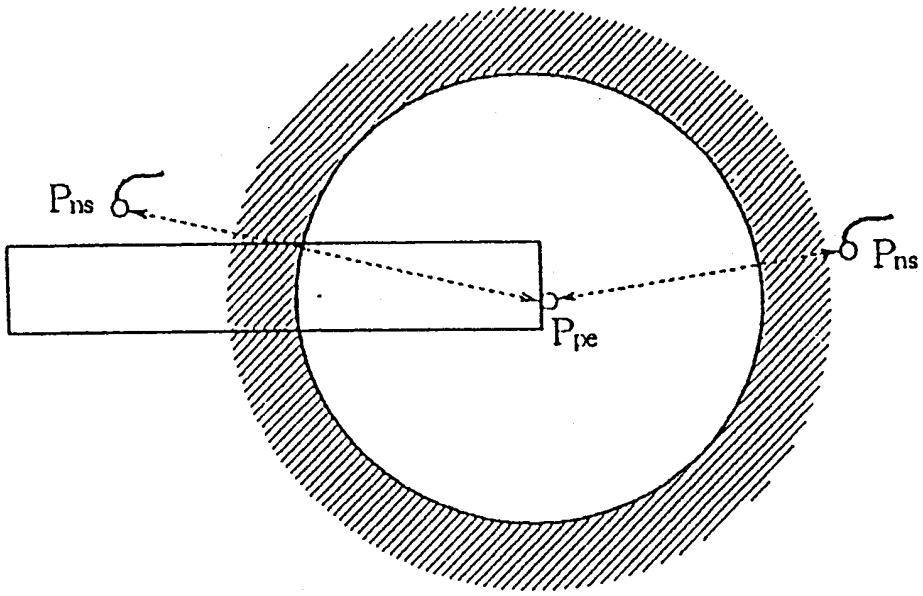


图 18

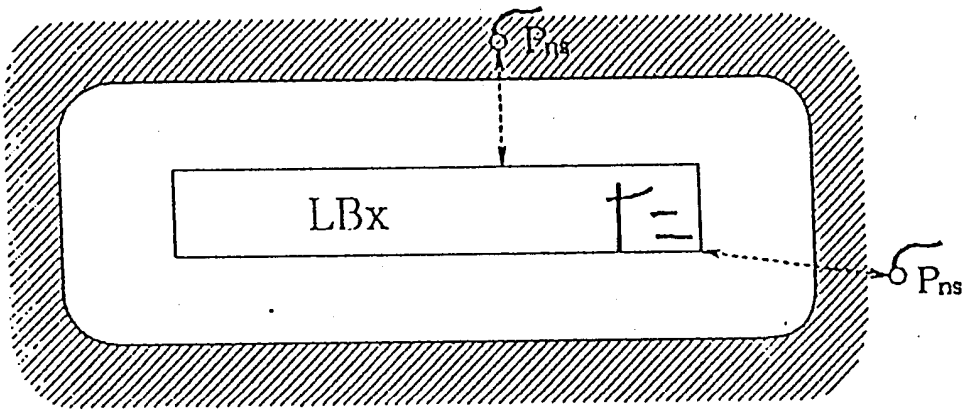


图 19

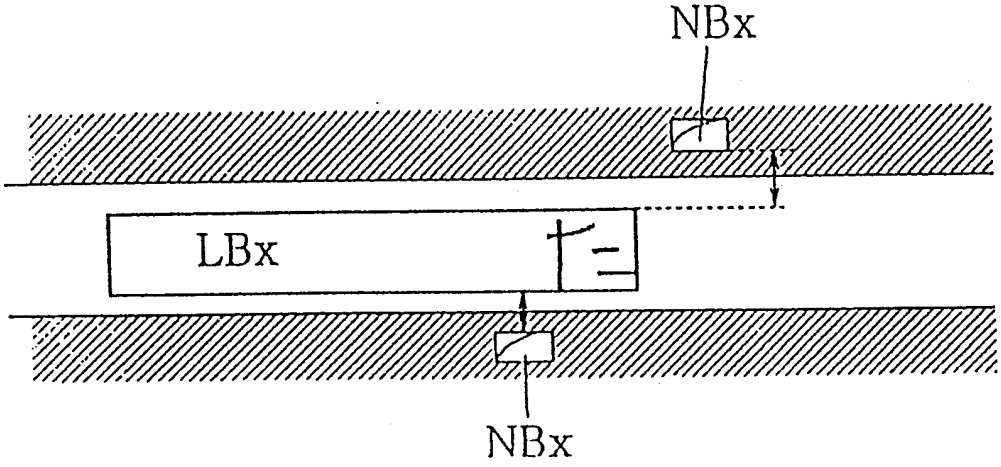


图 20

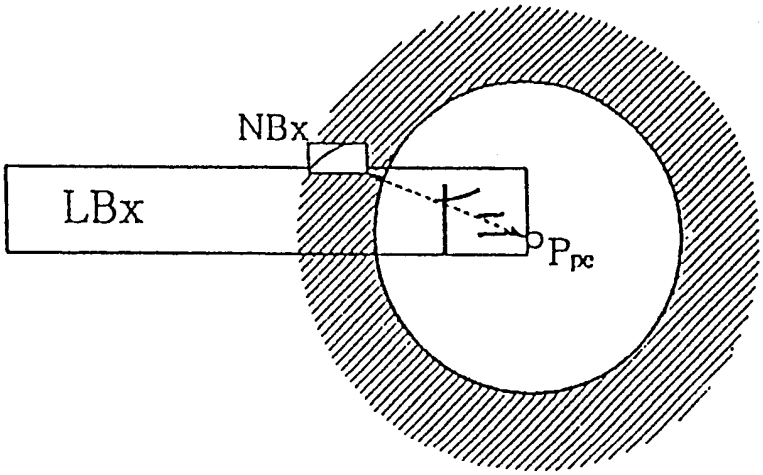


图 21

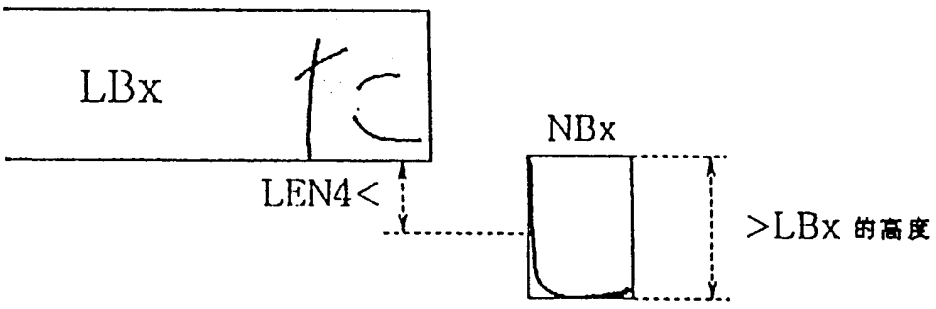


图 22

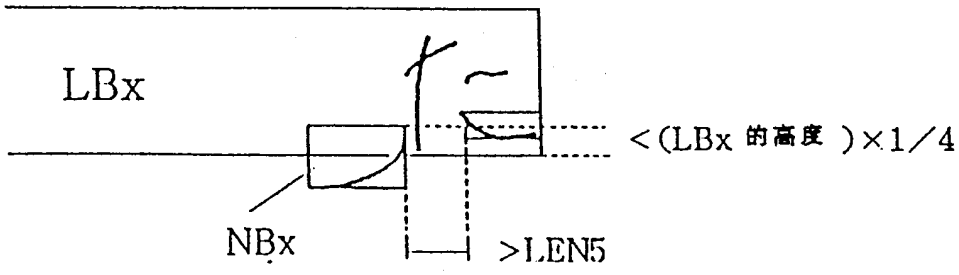


图 23

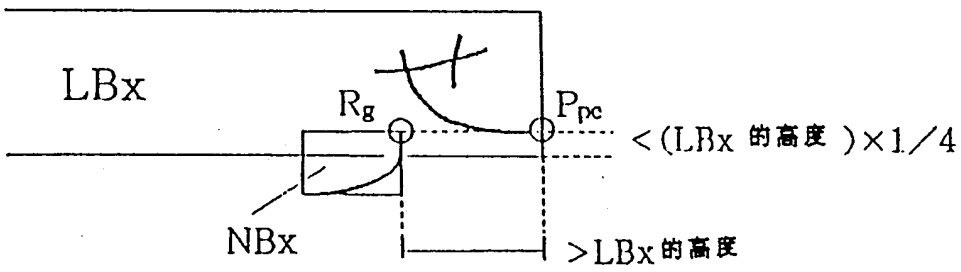


图 24

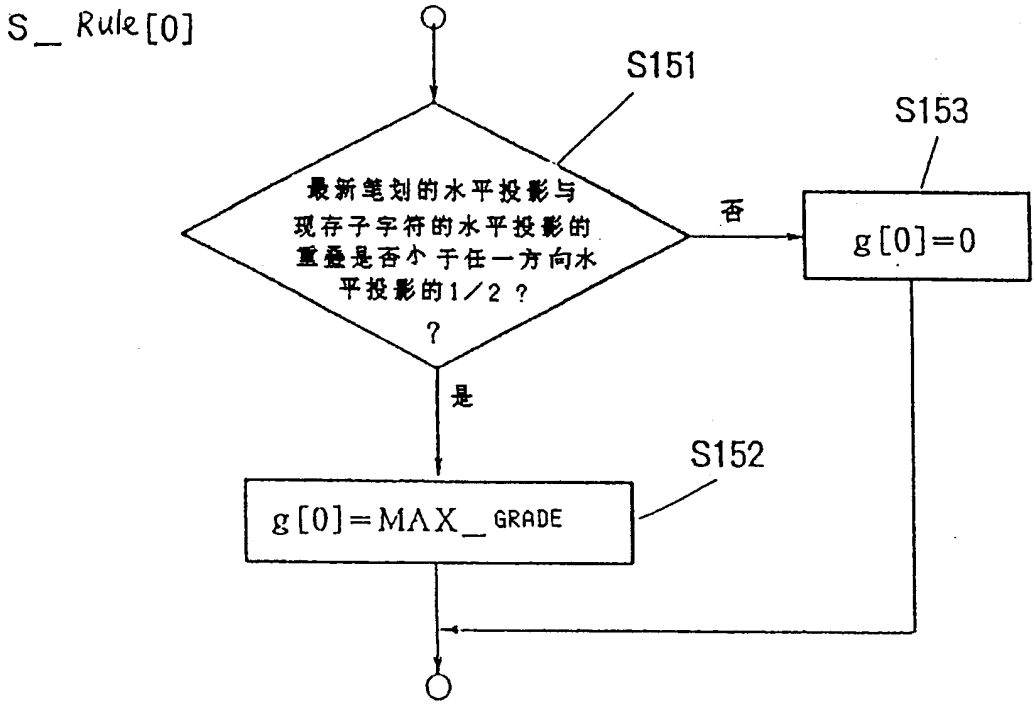

S_ Rule[0]
S151
最新笔划的水平投影与现存子字符的水平投影的重叠是否小于任一方向水平投影的1／2？
?
否
S153
g[0]=0
是
S152
g[0]=MAX_ GRADE

图 25

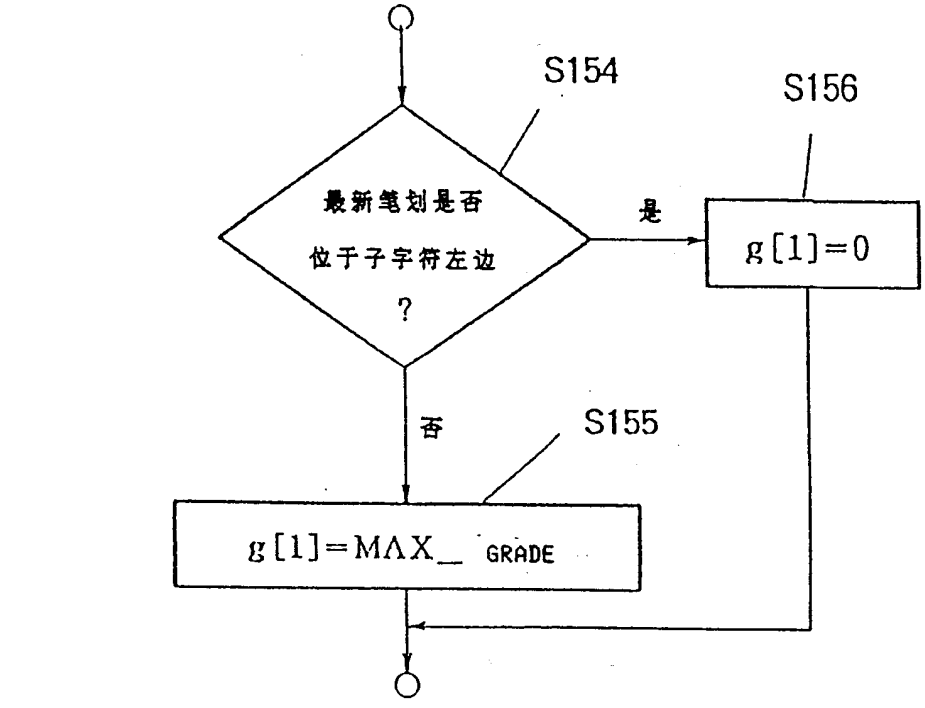

S_ Rule[1]
S154
最新笔划是否位于子字符左边
?
是
S156
g[1]=0
否
S155
g[1]=MAX_ GRADE

图 26

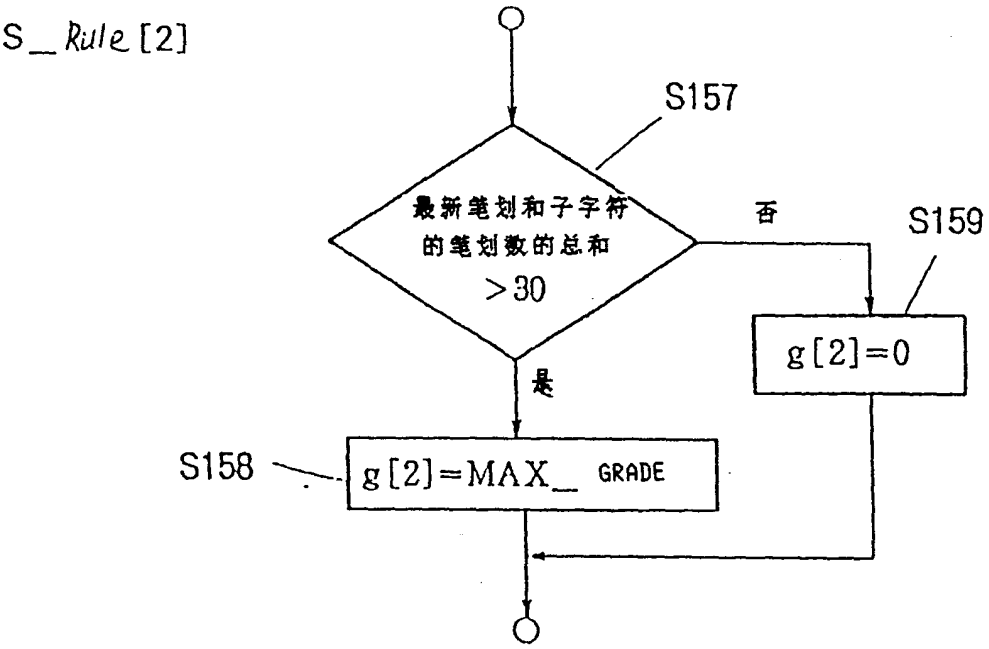

S_Rule[2]
S157
最新笔划和子字符
的笔划数的总和
>30
否
S159
g[2]=0
是
S158
g[2]=MAX_ GRADE

图 27

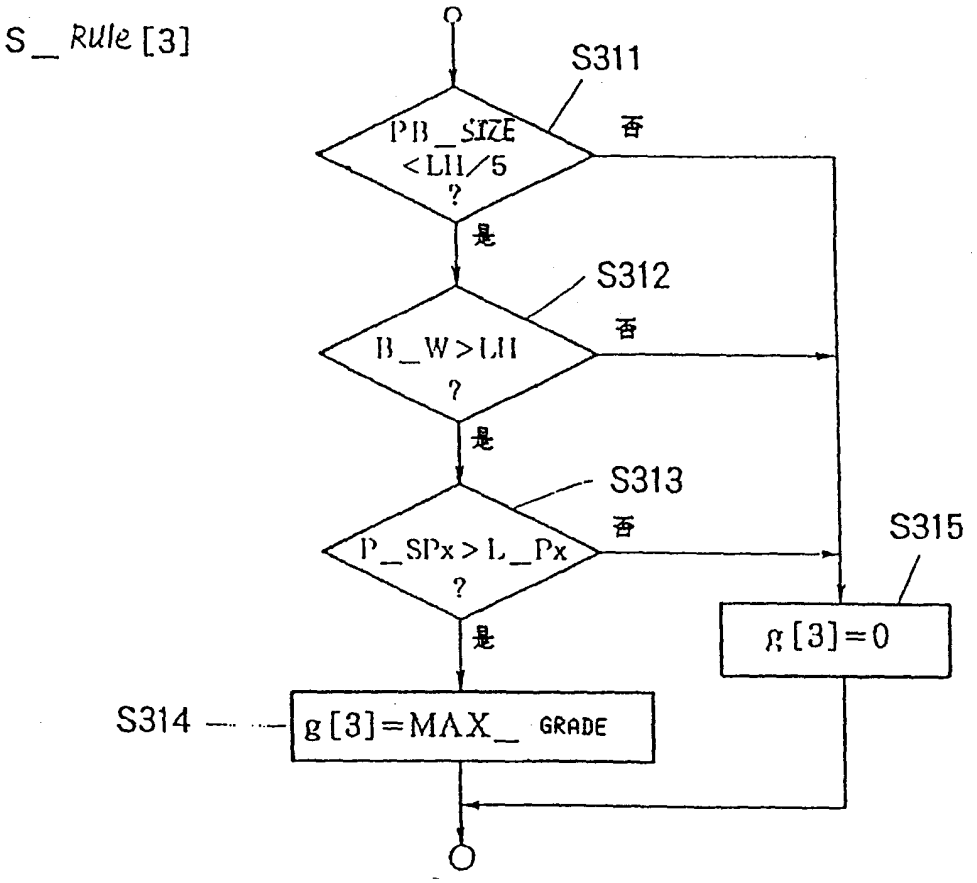

S_Rule[3]
S311
PB_SIZE
<LH/5
?
否
是
S312
B_W>LH
?
否
是
S313
P_SPx>L_Px
?
否
是
S315
g[3]=0
S314
g[3]=MAX_ GRADE

图 28

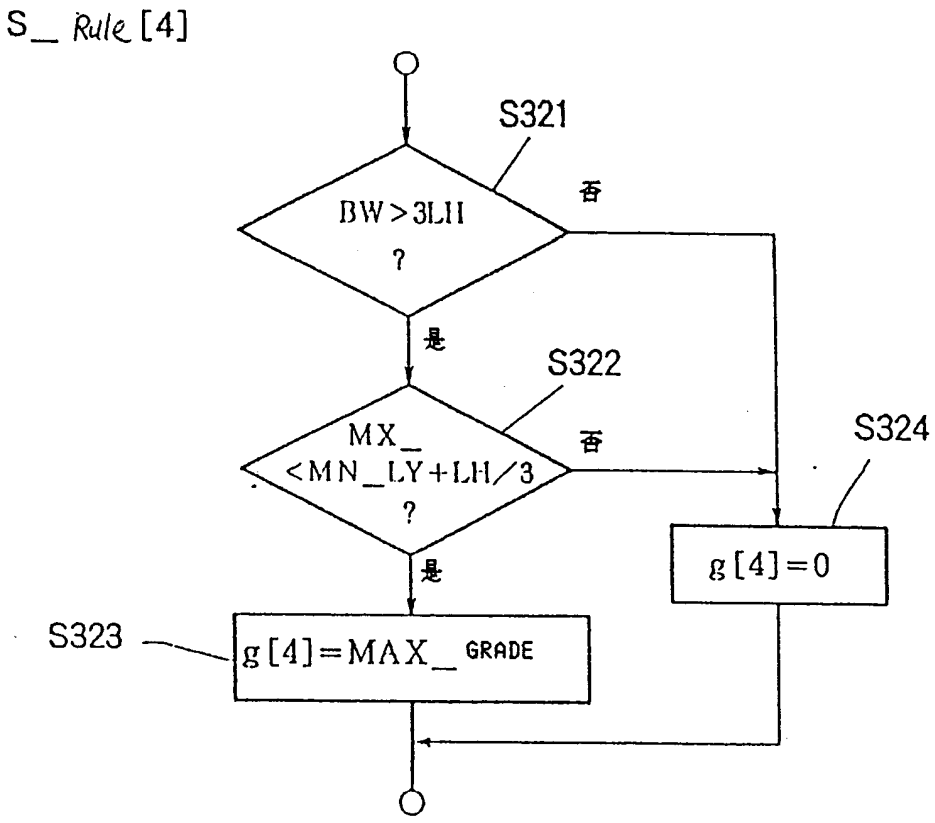

S_ Rule[4]
S321
BW>3LH
?
否
是
S322
MX_
<MN_LY+LH/3
?
否
是
S324
g[4]=0
S323
g[4]=MAX_ GRADE

图 29

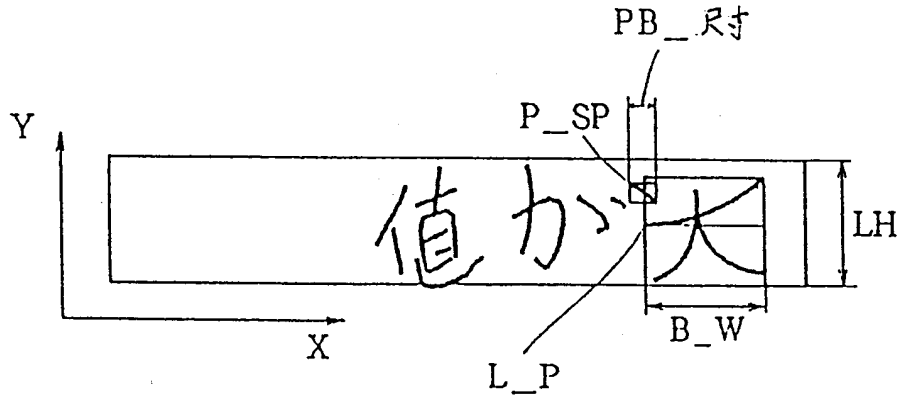

PB_尺寸
P_SP
Y
值が大
LH
X
B_W
L_P

图 30

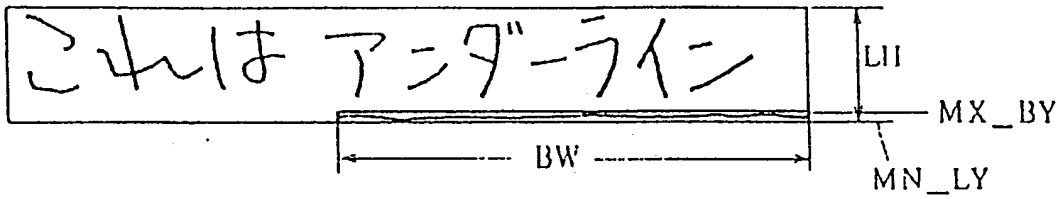

これはアンダーライン
LII
MX_BY
BW
MN_LY

图 31

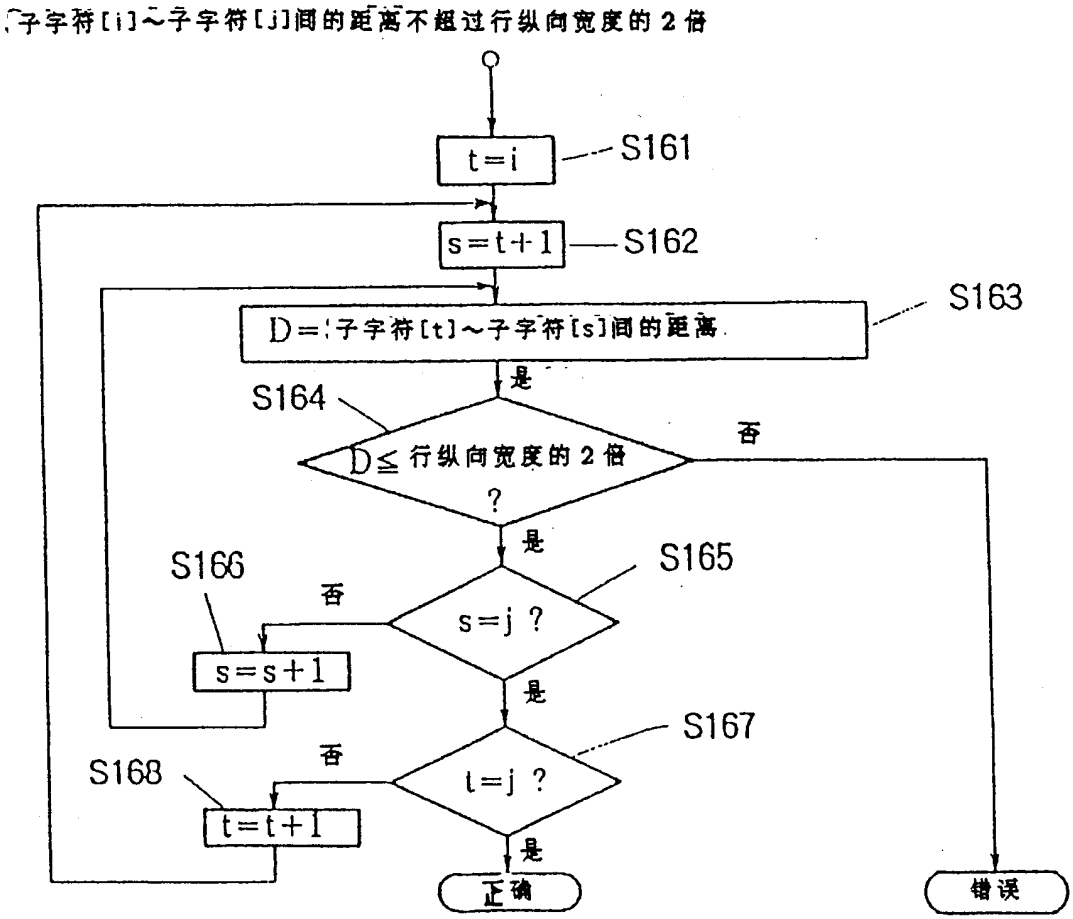

子字符[i]～子字符[j]间的距离不超过行纵向宽度的2倍
t=i
S161
s=t+1
S162
D=|子字符[t]～子字符[s]间的距离|
S163
是
S164
D≦行纵向宽度的2倍
?
否
是
S166
S165
否
s=j ?
s=s+1
是
S167
否
S168
t=j ?
t=t+1
是
正确
错误

图 3 2

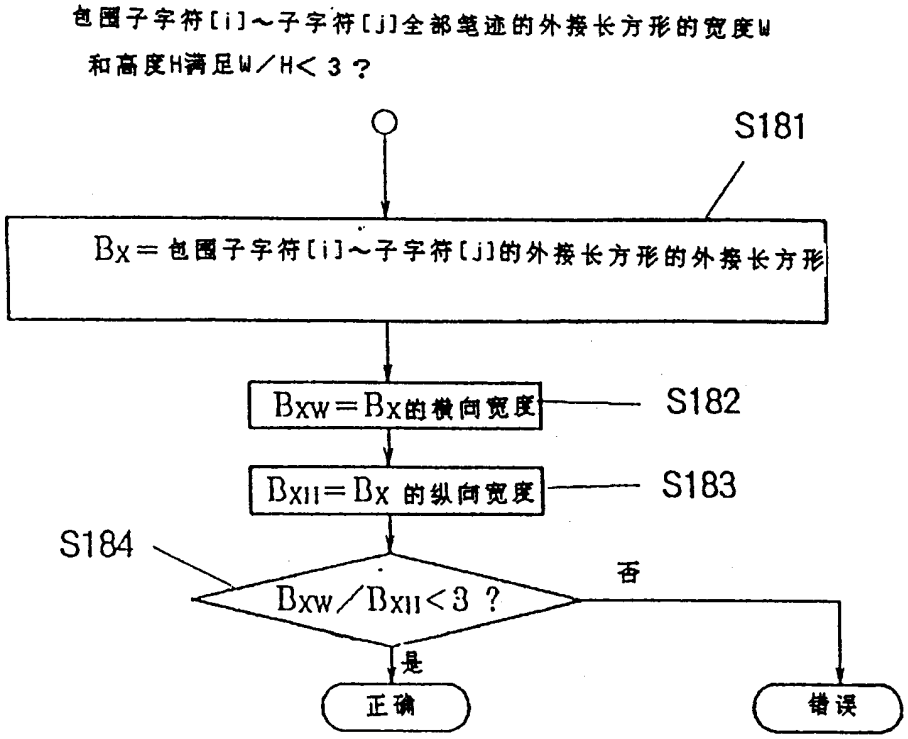

包围子字符[i]～子字符[j]全部笔迹的外接长方形的宽度W
和高度H满足W／H＜３？
S181
Bx＝包围子字符[i]～子字符[j]的外接长方形的外接长方形
Bxw＝Bx的横向宽度
S182
Bxh＝Bx 的纵向宽度
S183
S184
Bxw／Bxh＜3 ？
否
是
正确
错误

图 3 3

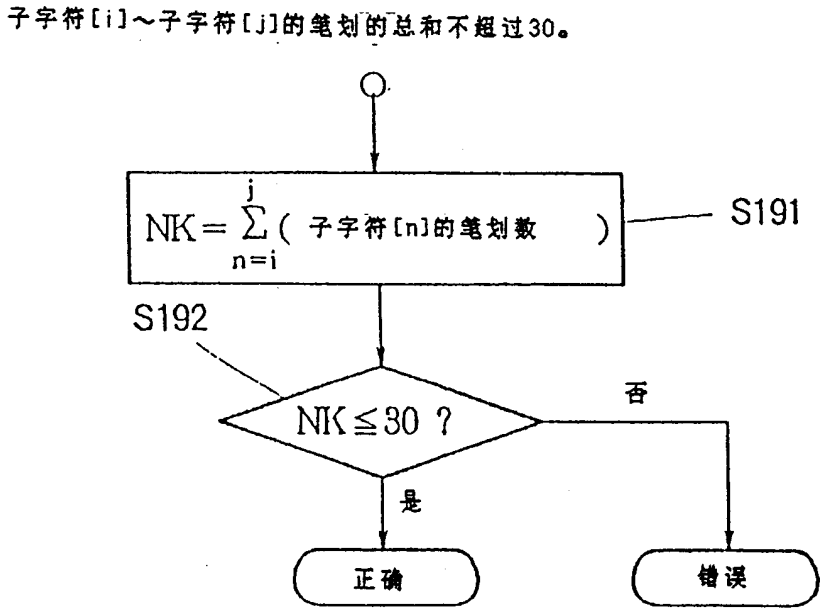
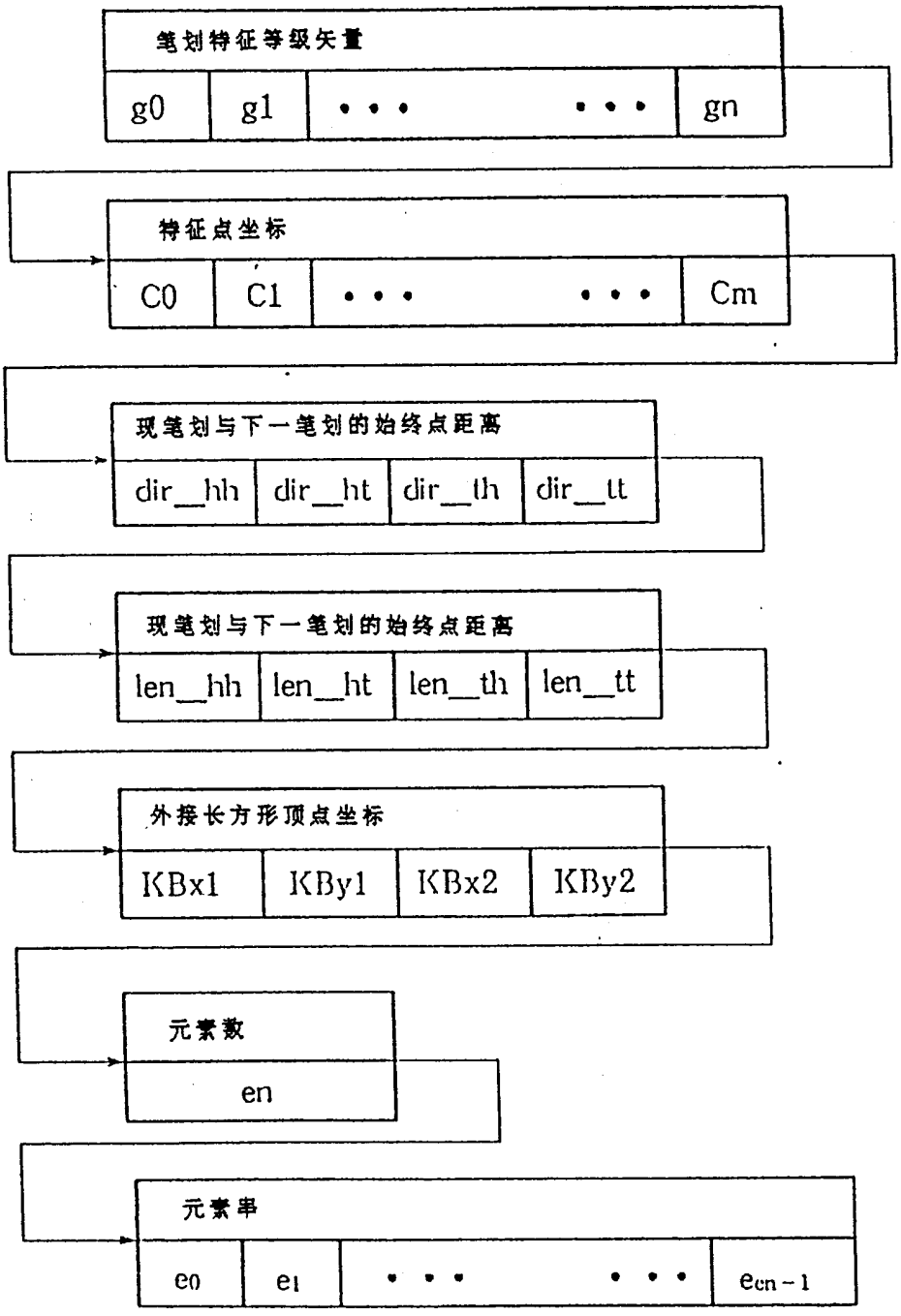
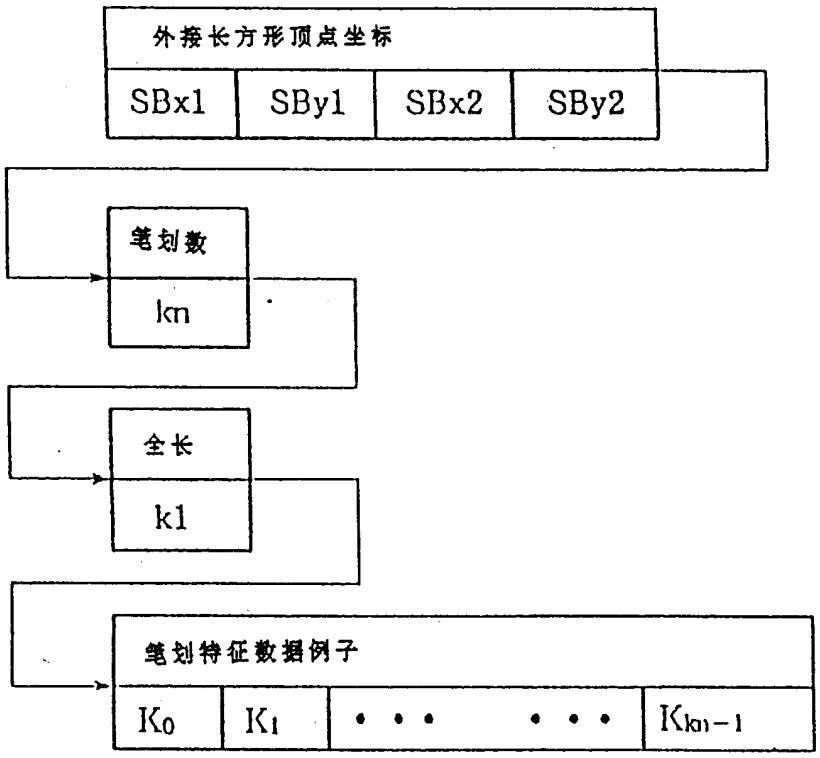

子字符[i]～子字符[j]的笔划的总和不超过30。
NK＝ Σ (子字符[n]的笔划数)
n=i
j
S191
S192
NK≦30 ？
否
是
正确
错误

图 3 4



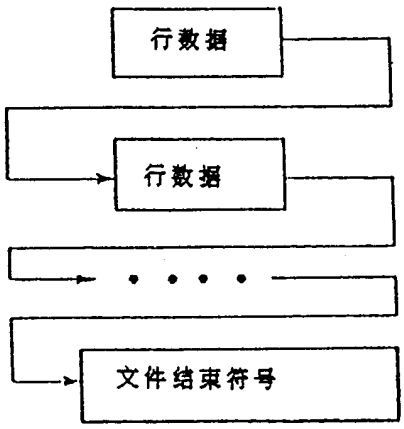
笔划特征数据的逻辑格式例子

图 3.5



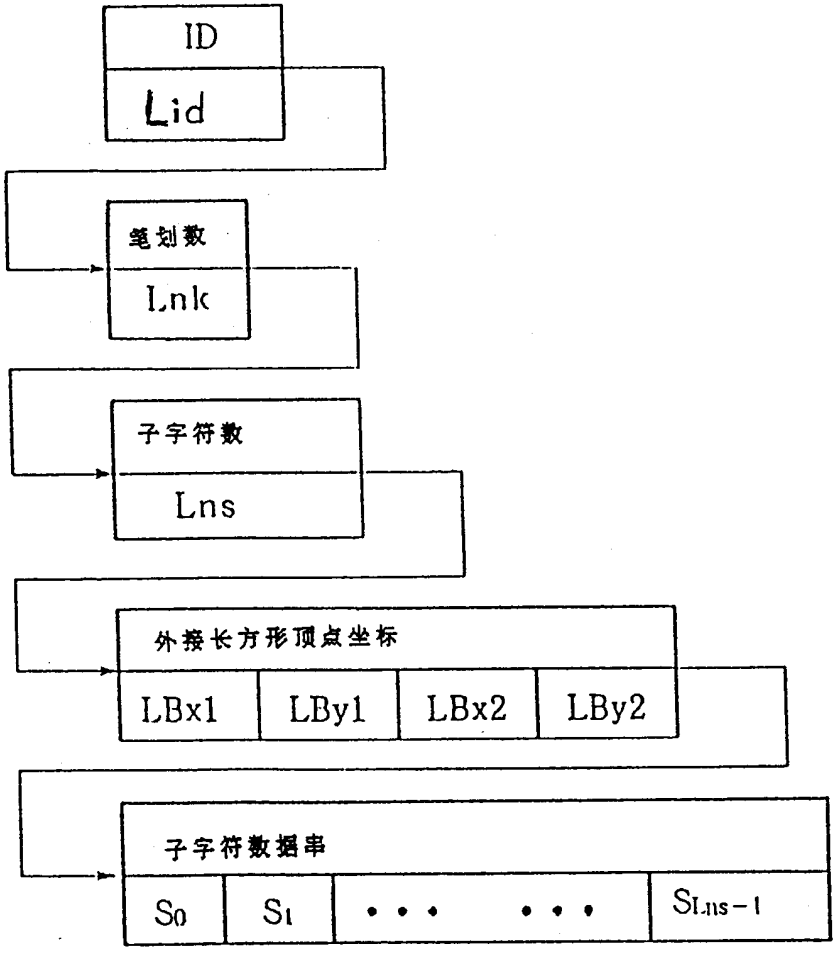
子字符数据（单个文字数据）的逻辑格式例子

图 3 6



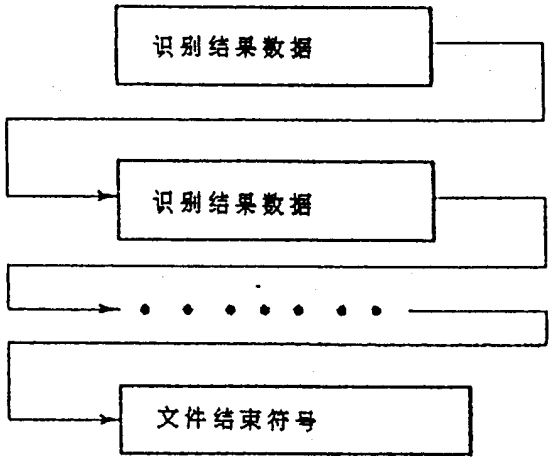
行数据库的逻辑格式例子

图 3 7



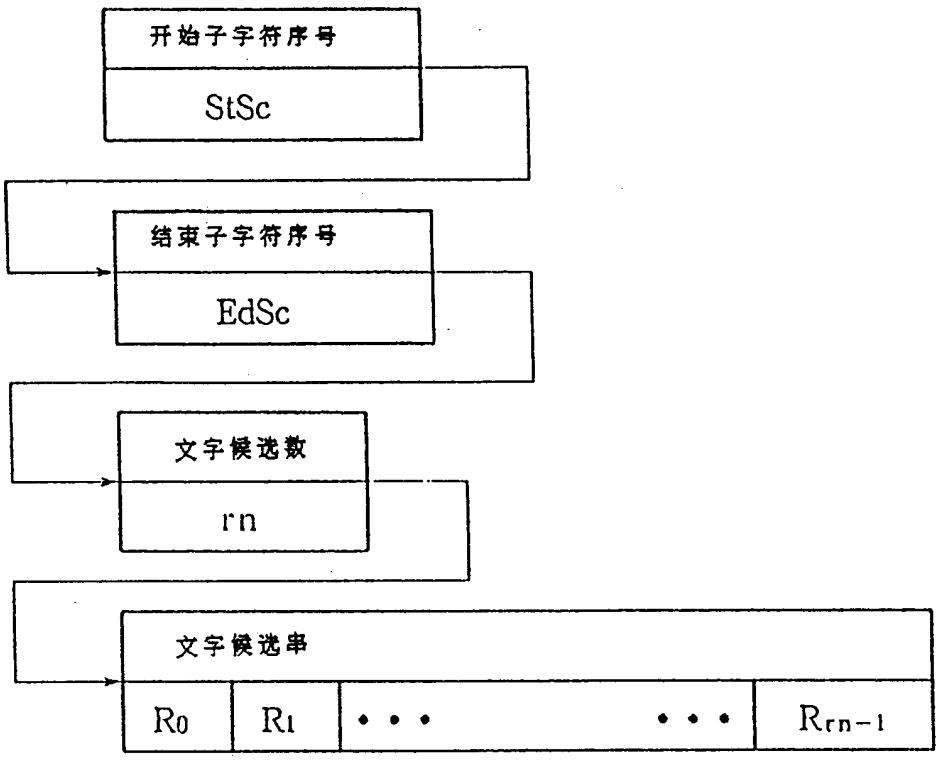
行数据格式例子

图 3 8



识别结果数据库的格式的例子

图 3 9

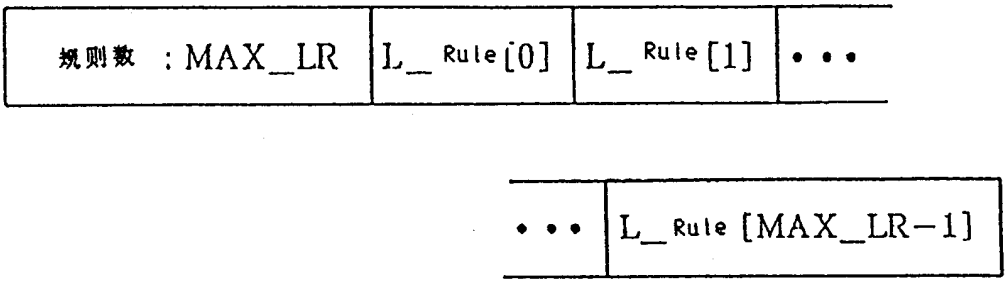


- 开始子字符序号
  构成识别文字的子字符串的最初的子字符位于该行的第几个？
- 结束子字符序号
  构成识别文字的子字符串的最后一个子字符位于该行的第几个？
- 文字候选 Ri(i=0, 1, ---------- rn)

  识别结果文字代码 :Ci, 识别等级：由gi构成

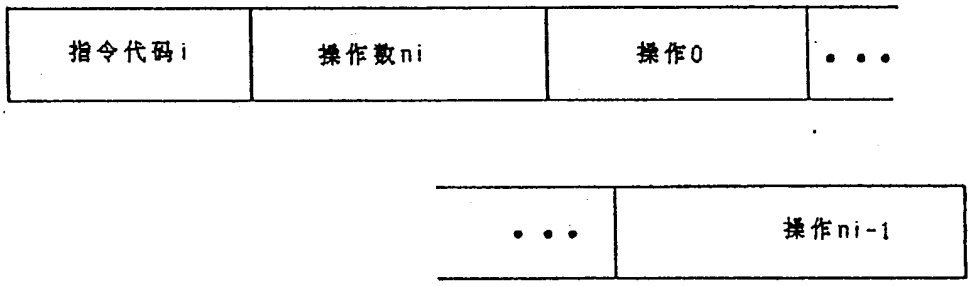
识别结果数据的格式

图 40



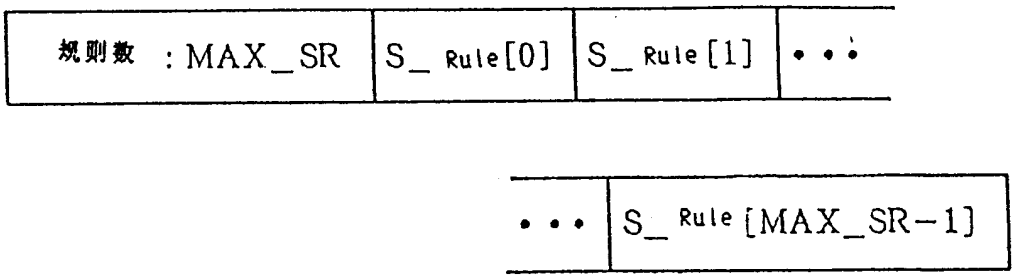
行切割规则数据库的逻辑格式的例子

图 41



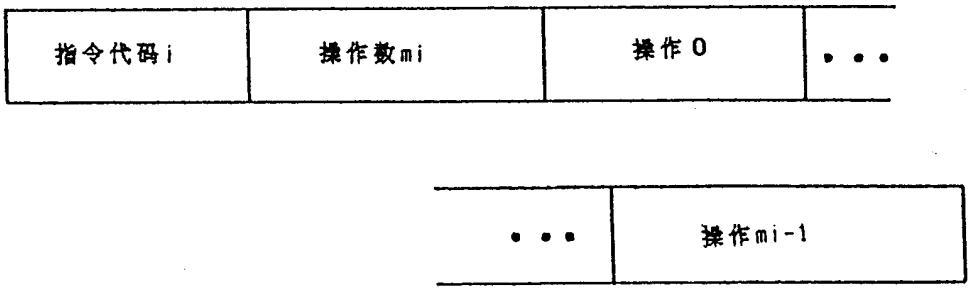
L_Rule[i]记录的格式例子

图 4 2



子字符切割规则数据库的逻辑格式例子

图 4 3



S_Rule[i]记录的格式例子

图 44

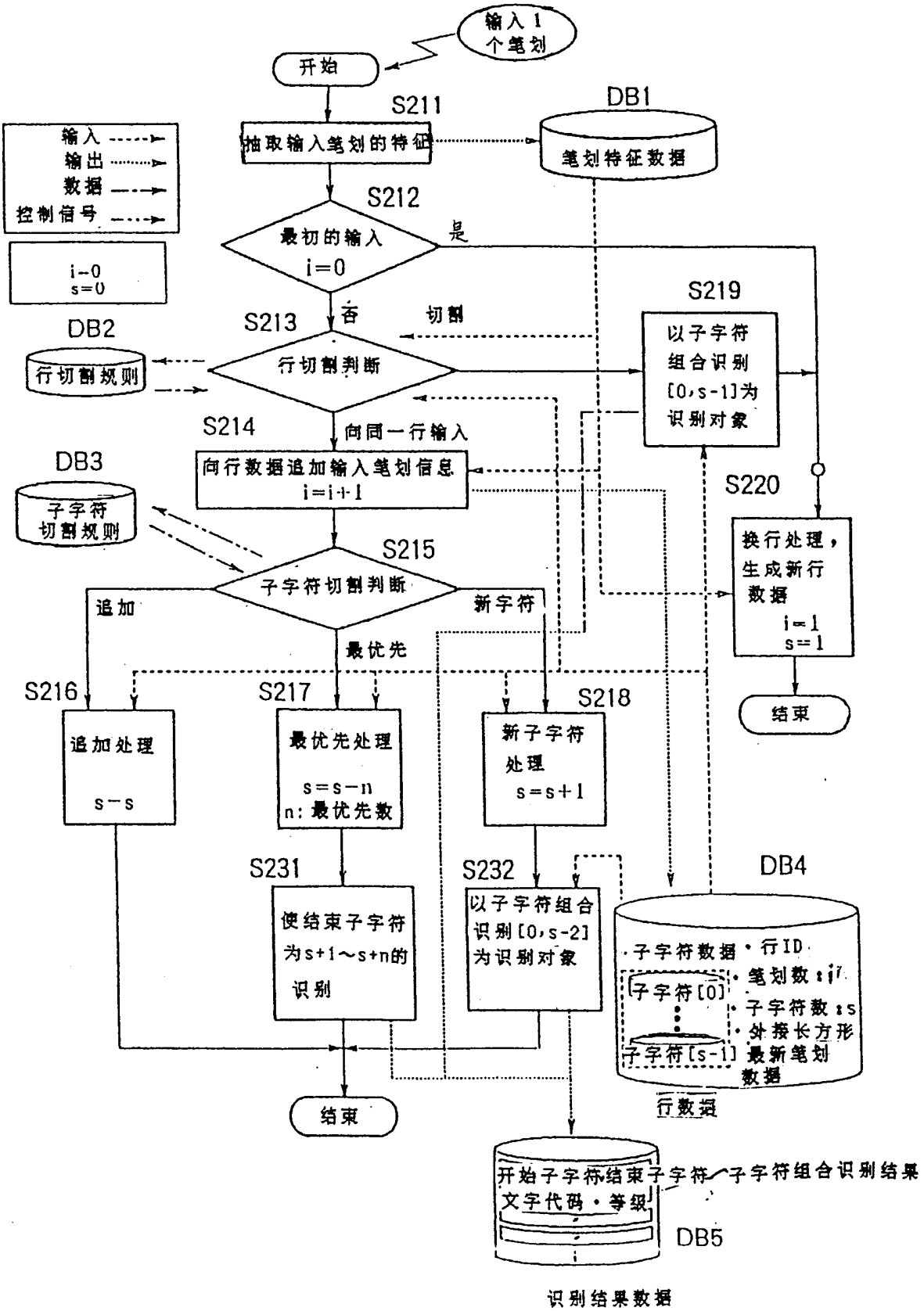

输入1个笔划
开始
S211
抽取输入笔划的特征
DB1
笔划特征数据
输入
输出
数据
控制信号
i=0
s=0
S212
最初的输入
i=0
是
否
切割
S213
行切割判断
DB2
行切割规则
S219
以子字符组合识别[0,s-1]为识别对象
S214
向同一行输入
向行数据追加输入笔划信息
i=i+1
DB3
子字符切割规则
S220
换行处理，生成新行数据
i=1
s=1
结束
S215
子字符切割判断
追加
新字符
最优先
S216
追加处理
s=s
S217
最优先处理
s=s-n
n:最优先数
S218
新子字符处理
s=s+1
S231
使结束子字符为s+1～s+n的识别
S232
以子字符组合识别[0,s-2]为识别对象
DB4
·子字符数据·行ID
·笔划数:i
子字符[0]
·子字符数:s
·外接长方形
子字符[s-1]
最新笔划数据
行数据
结束
开始子字符
结束子字符
子字符组合识别结果
文字代码·等级
DB5
识别结果数据

图 45

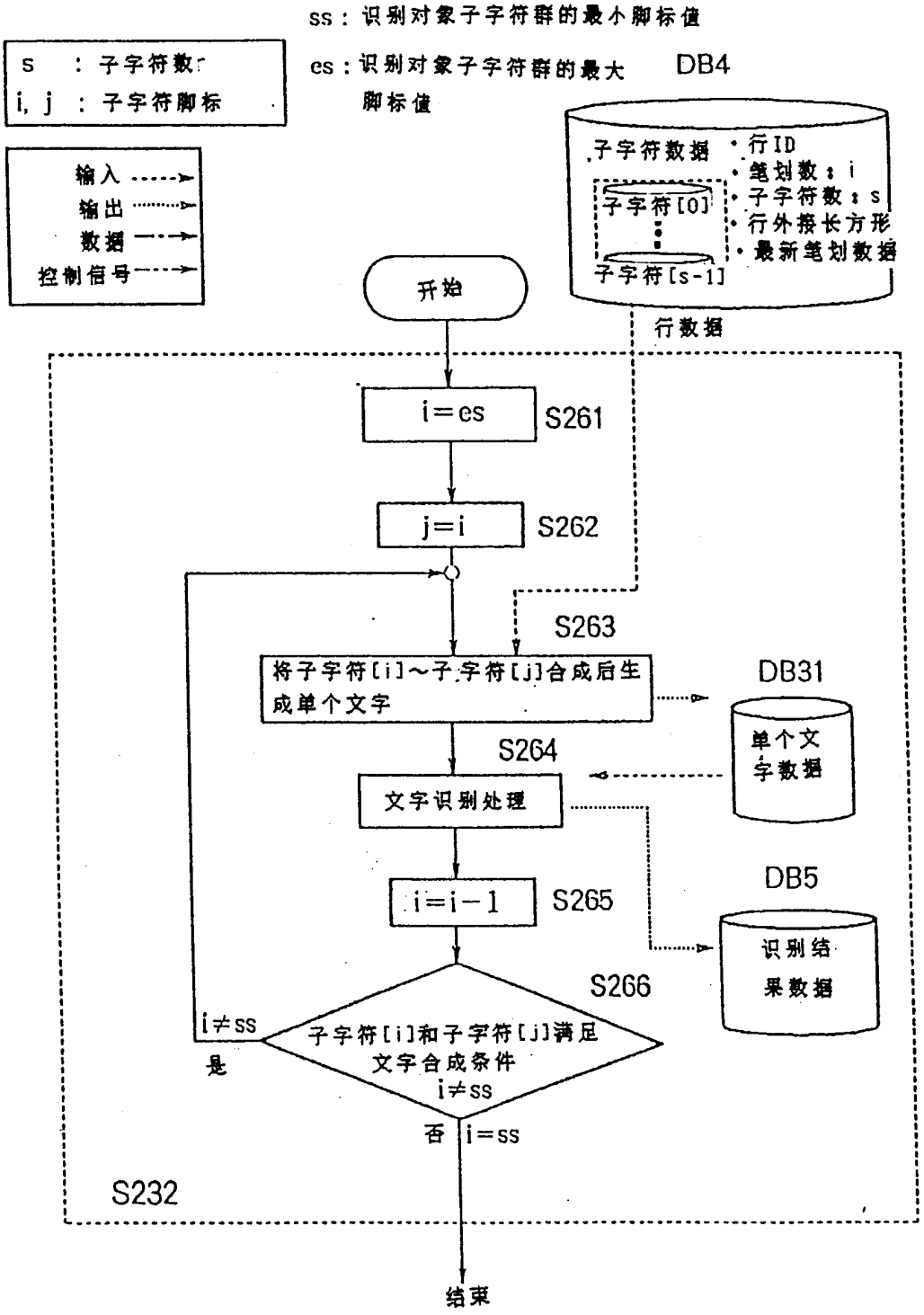

ss：识别对象子字符群的最小脚标值
s ：子字符数
i, j ：子字符脚标
es：识别对象子字符群的最大脚标值
DB4
输入
输出
数据
控制信号
子字符数据
·行ID
·笔划数：i
·子字符数：s
·行外接长方形
·最新笔划数据
子字符[0]
子字符[s-1]
开始
行数据
i=es
S261
j=i
S262
S263
将子字符[i]～子字符[j]合成后生成单个文字
DB31
单个文字数据
S264
文字识别处理
i=i-1
S265
DB5
识别结果数据
S266
i≠ss
是
子字符[i]和子字符[j]满足文字合成条件
i≠ss
否
i=ss
S232
结束

图 4 6

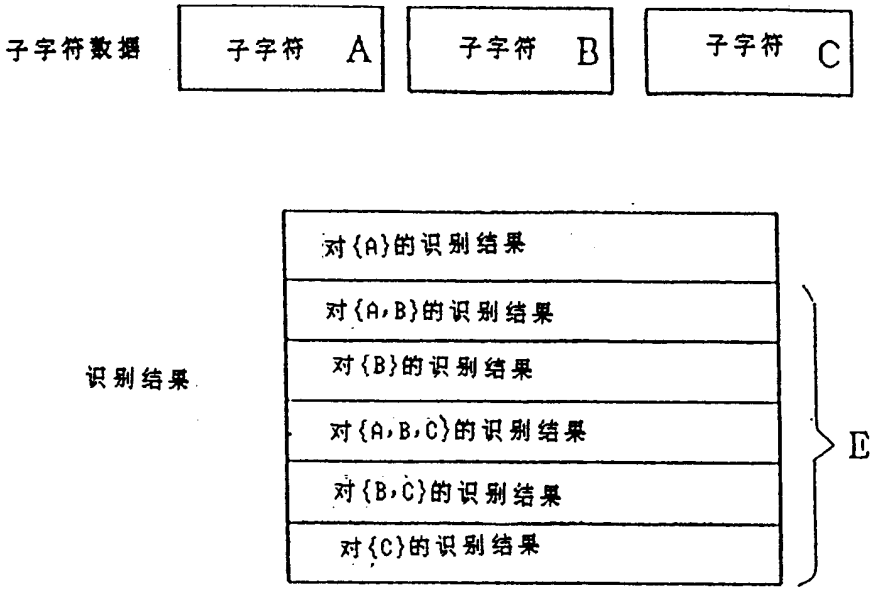

子字符数据
子字符 A
子字符 B
子字符 C
识别结果
对{A}的识别结果
对{A,B}的识别结果
对{B}的识别结果
对{A,B,C}的识别结果
对{B,C}的识别结果
对{C}的识别结果
E

图 4 7

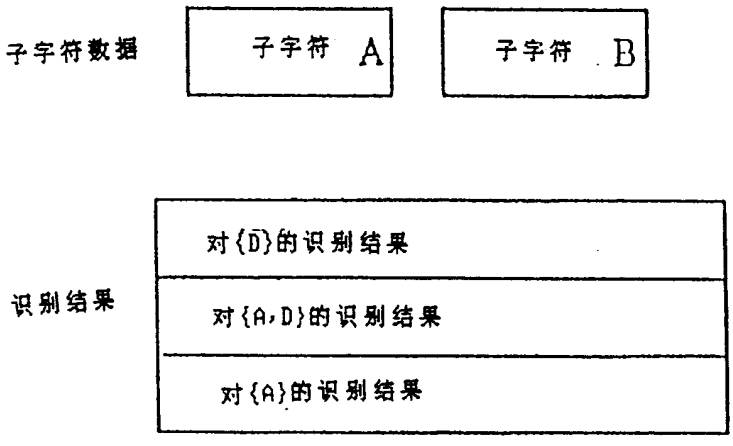

子字符数据
子字符 A
子字符 B
识别结果
对{D}的识别结果
对{A,D}的识别结果
对{A}的识别结果

图 4 8

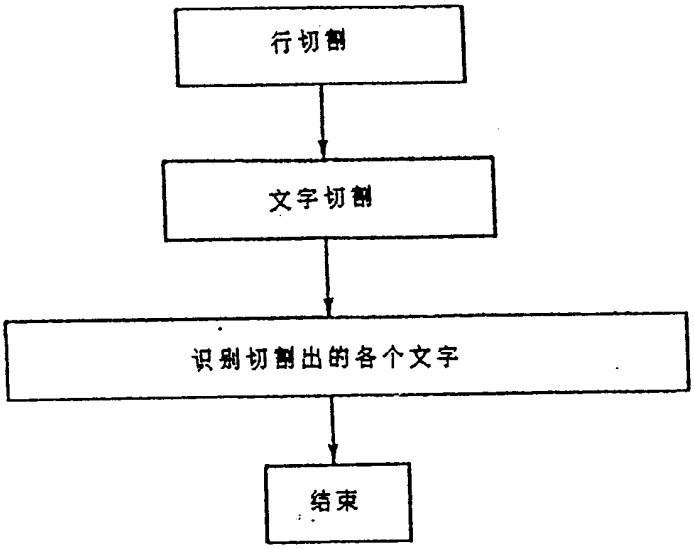


图 4 9

