



(19) 中華民國智慧財產局

(12) 發明說明書公開本

(11) 公開編號：TW 201627888 A

(43) 公開日：中華民國 105 (2016) 年 08 月 01 日

(21) 申請案號：104128417

(22) 申請日：中華民國 104 (2015) 年 08 月 28 日

(51) Int. Cl. : G06F17/30 (2006.01)

G06F11/34 (2006.01)

(30) 優先權：2015/01/22 中國大陸

201510032801.0

(71) 申請人：阿里巴巴集團服務有限公司 (香港地區) ALIBABA GROUP SERVICES LIMITED  
(HK)

香港

(72) 發明人：羅光春 (CN)；孫廷韜 (CN)

(74) 代理人：林志剛

申請實體審查：無 申請專利範圍項數：31 項 圖式數：3 共 35 頁

(54) 名稱

正則表達式的生成方法及系統

(57) 摘要

本發明的實施例公開了一種規則運算式的生成方法及系統。該方法包括：獲得預設字串；回應於觸發指令獲得預設字串中的待收集字串；從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串；從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串；根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。利用本發明的實施例可以生成使用者需要字串的規則運算式。

指定代表圖：

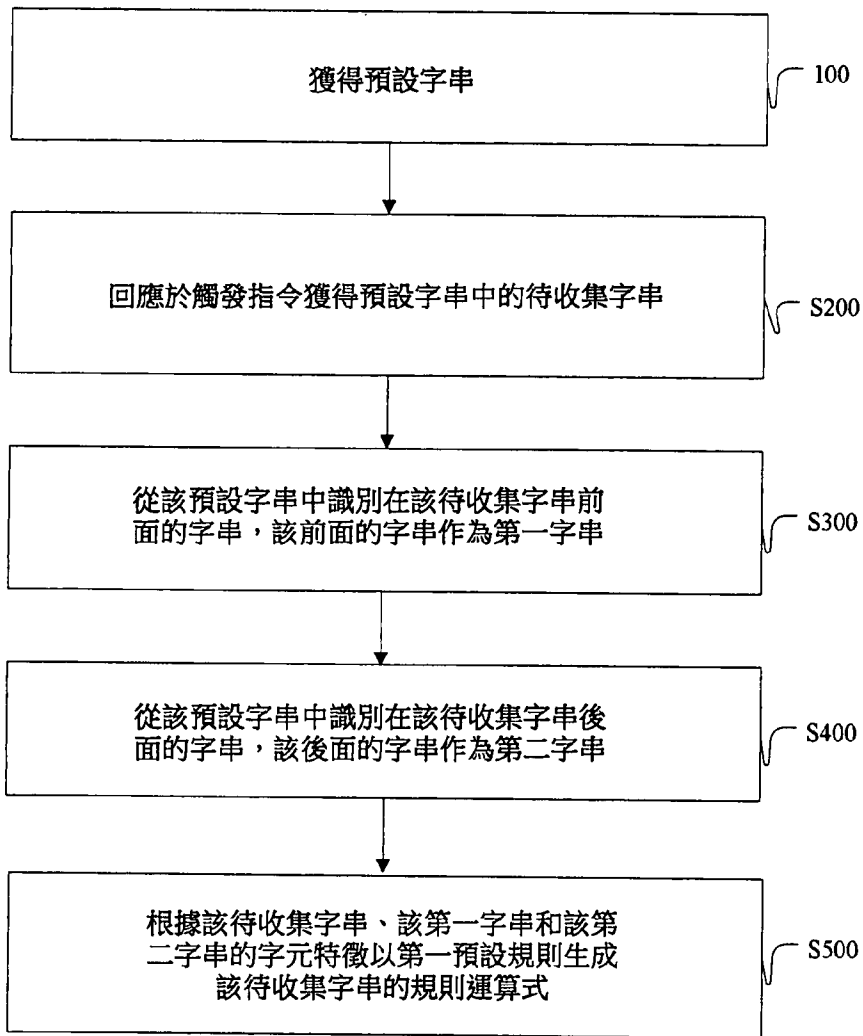


圖 2

# 發明摘要

※申請案號：104128417

※申請日：104年08月28日

※IPC分類：G06F 17/30 (2006.01)

G06F 11/34 (2006.01)

【發明名稱】(中文/英文)

正則表達式的生成方法及系統

## 【中文】

本發明的實施例公開了一種規則運算式的生成方法及系統。該方法包括：獲得預設字串；回應於觸發指令獲得預設字串中的待收集字串；從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串；從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串；根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。利用本發明的實施例可以生成使用者需要字串的規則運算式。

## 【英文】

【代表圖】

【本案指定代表圖】：第(2)圖。

【本代表圖之符號簡單說明】：無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：無

# 發明專利說明書

(本說明書格式、順序，請勿任意更動)

## 【發明名稱】(中文/英文)

正則表達式的生成方法及系統

## 【技術領域】

本發明關於電腦技術領域，特別關於一種規則運算式的生成方法及系統。

## 【先前技術】

規則運算式是電腦科學的一個概念。規則運算式使用單個字串來描述、匹配一系列符合某個句法規則的字串。在很多文字編輯器裡，規則運算式通常被用來檢索、替換那些符合某個模式的文本。

一般可以通過規則運算式生成工具來生成規則運算式。現有的規則運算式生成工具包括Txt2re。Txt2re可以提供多個文本項選定按鈕。通過某個文本項選定按鈕，Txt2re工具可以執行對應規則的處理。這種處理通常包括提取一段字串中具有相應規則的部分字串，並由提取的部分字串生成對應規則運算式的代碼。以利用Txt2re生成一段日誌的規則運算式為例，具體步驟包括：

S1：Txt2re工具通過文字方塊接收輸入的日誌。假設該段日誌為：

```
10.200.98.220 -- [28/Jun/2013:14:53:08 +0800] "POST
/PutData?Category=YunOsAccountOpLog&AccessKeyId=U0UjpekFQOVJW45A&Date=Fri
%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12
XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1" 0.024 18204 200 37 "-"
"aliyun-sdk-java"
```

S2：Txt2re工具遍歷輸入的日誌中的每個字元，識別日誌中具有有一定規則的字串，並按照預設規則生成對應日誌中每個字串的規則運算式標記。

例如，Txt2re工具遍歷輸入的日誌中的每個字元，在遍歷完“10.200.98.220”這些字元時，根據預設規則判斷這一串字元為一IPv4的位址，並生成代表該欄位的規則運算式標記“ipaddress”。

S3：Txt2re工具接收按一下介面中“show matches”按鈕的指令，回應該指令並顯示日誌中的字串及對應的規則運算式標記，並且在該規則運算式標記上提供點擊命令。如圖1所示，通過接收點擊的“show matches”按鈕，Txt2re工具可以顯示一個區域。在該區域中分兩行分別顯示日誌內容和與日誌內容中所識別字串對應的規則運算式標記。並且，在該規則運算式標記上提供可點擊的按鈕。

S4：Txt2re工具接收按一下上述標記為“ipaddress”的按鈕，對對應的字串“10.200.98.220”進行處理並生成對應該字串規則運算式的代碼。

後續，操作人員可以手工或者利用其它工具將該代碼轉化為對應欄位的規則運算式。

由此可見，Txt2re工具可以按預設規則提供多個待收集字串選定按鈕。通過按一下某個待收集字串的選定按

鈕，Txt2re工具就可以生成該待收集字串對應的規則運算式。

在實現本發明的過程中，發明人發現現有技術中至少存在如下問題：

現有技術利用Txt2re工具生成規則運算式時，可以獲取的規則運算式是有限的，也即只能提供符合一定規則字串的規則運算式，不能按照使用者的需要生成規則運算式。比如上述日誌中，Txt2re工具可以提供包括“[28/Jun/2013:14:53:08 +0800]”這個字串對應的選定按鈕。但這個字串中的中括弧並沒有太大意義，所以用戶可以選擇的話，一般會選擇字串“28/Jun/2013:14:53:08+0800”，也即使用者需要去掉括弧後的這個字串的規則運算式。但現有技術並沒有給用戶選擇的機會。因此，利用現有技術生成規則運算式，靈活性差，無法滿足使用者需要。

### 【發明內容】

本發明的實施例的目的是提供一種規則運算式的生成方法及系統，可以生成使用者需要字串的規則運算式。

為解決上述技術問題，本發明的實施例提供一種規則運算式的生成方法及系統是這樣實現的：

一種規則運算式的生成方法，包括：

獲得預設字串；

回應於觸發指令獲得預設字串中的待收集字串；

從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串；

從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串；

根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。

一種規則運算式的生成系統，包括：

預設字串獲得單元，用於獲得預設字串；

待收集字串獲得單元，用於回應於觸發指令獲得預設字串中的待收集字串；

第一字串獲得單元，用於從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串；

第二字串獲得單元，用於從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串；

規則運算式生成單元，用於根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。

由以上本發明的實施例提供的技術方案可見，本發明的實施例通過回應於觸發指令，根據使用者的需要獲取預設字串中待收集的字串，並根據預設規則生成該待收集字串的規則運算式。與現有技術相比，大大提高了用戶使用的靈活性，可以根據使用者需要生成相應字串的規則運算式，進而可以根據該規則運算式解析出對使用者有用的字串。



**【圖式簡單說明】**

為了更清楚地說明本發明的實施例或現有技術中的技術方案，下面將對實施例或現有技術描述中所需要使用的附圖作簡單地介紹，顯而易見地，下面描述中的附圖僅僅是本發明中記載的一些實施例，對於本領域通常知識者來講，在不付出創造性勞動性的前提下，還可以根據這些附圖獲得其他的附圖。

圖1是現有技術中 Txt2re 工具匹配日誌的部分示意圖；

圖2是本發明的一種規則運算式的生成方法一個實施例的流程圖；

圖3是本發明的一種規則運算式的生成系統一個實施例的流程圖。

**【實施方式】**

本發明的實施例提供一種規則運算式的生成方法及系統。

為了使本技術領域的人員更好地理解本發明中的技術方案，下面將結合本發明的實施例中的附圖，對本發明的實施例中的技術方案進行清楚、完整地描述，顯然，所描述的實施例僅僅是本發明的一部分實施例，而不是全部的實施例。基於本發明中的實施例，本領域通常知識者在沒有做出創造性勞動前提下所獲得的所有其他實施例，都應

當屬於本發明保護的範圍。

以下以幾個具體的例子詳細說明本發明的實施例的具體實現。

以下首先介紹本發明的一種規則運算式的生成方法的實施例。結合附圖2，該實施例包括：

S100：獲得預設字串。

本發明的實施例中，獲取預設字串可以通過回應使用者導入的操作，這裡的預設字串可以是一段日誌，比如：

```
"[2015-01-07 18:50:36.432] [INFO] access - 127.0.0.1 - - "GET /user/projects/ali-sls-test HTTP/1.1" 304 - "http://localhost:8888/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36"
```

這是一條存取日誌，記錄的內容包括使用者存取服務時的資訊，比如使用者的IP，用戶請求的統一資源定位符（Uniform Resource Locator，URL），用戶使用的瀏覽器等等。

此外，本發明的實施例該的獲取預設字串的實現方式可以是通過回應使用者導入的操作，在實際中還可以是其他形式，本發明的實施例並不以此為限。

進一步的，本發明的實施例該的預設字串可以是一段日誌，在實際中還可以是其他形式，例如一段符合一定句法規則的字串，本發明的實施例並不以此為限。

S200：回應於觸發指令獲得預設字串中的待收集字串。

本發明的實施例中，該觸發指令具體可以包括使用者根據具體的需要選擇並拖拽預設字串中的部分字串的操

作；還可以包括使用者點擊預設字串中的部分字串的起始位置觸發的第一指令，以及點擊該部分字串的結束位置觸發的第二指令，該第二指令還可以包括使用者按住預設快速鍵並點擊該部分字串的結束位置的操作。

以下述一段日誌片段為例，假設獲取的預設字串為：

```
"[2014-11-02 19:46:03.895] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"
```

假設使用者想要收集的字串，也即待收集的字串為：

“2014-11-02 19:46:03.895”，那麼使用者選擇並拖拽待該字串的操作可以觸發指令，根據觸發指令從該預設字串中識別出該待收集的字串為：“2014-11-02 19:46:03.895”。

進一步的，用戶可以任意選中想要收集的字串，但一般使用者選中的字串是具有一定意義的字串。比如使用者選中的字串可以為“INFO”，也可以是“IN”，但前者是具有一定意義的，代表了日誌級別，而後者則沒有意義，即使選中後生成相應的規則運算式也沒有意義。

此外，本發明的實施例該觸發指令的實現方式可以包括選擇並拖拽預設字串中的部分字串的操作或者點擊預設字串中的部分字串的起始位置第一指令，以及點擊預設字串中的部分字串的結束位置的第二指令，在實際中還可以是其他形式，例如選擇並右擊預設字串中的部分字串所在區域的操作等，本發明的實施例並不以此為限。

S300：從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串。

以步驟 S200 中該的預設字串：“[2014-11-02 19:46:03.895] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"” 為例。

當待收集字串為：“2014-11-02 19:46:03.895”時，從該預設字串中識別在該待收集字串前面的字串為：“[”，也即該第一字串為：“[”。

S400：從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串。

以步驟 S300 中該的例子，那麼預設字串中在該待收集字串後面的字串，也即該第二字串為：“] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"”。

S500：根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。

該第一預設規則具體包括：

這裡要生成待收集規則運算式，首先需要判斷該第一字串和該待收集的字串中的字元特徵，並根據該第一字串和該待收集字串中的字元特徵生成第一部分規則運算式。具體是通過執行下述步驟 S510 至 S530 的操作至確定第一部分規則運算式：

S510：當該第一字串符合第二預設規則時，返回對應結果並作為該第一部分規則運算式。

這裡的第二預設規則具體包括：

當該第一字串不為空，且該第一字串是純數字字元組

成，且該待收集的字串為空或該待收集的字串的第一個字元不是純數字字元時；返回第一預設標識。

該第一預設標識為 "\\d+"，表示該第一字串是由數位組成。

當該第一字串不為空，且該第一字串是包含字母的純單詞字元組成，且該待收集的字串為空或該待收集的字串的第一個字元不是純單詞字元時；返回第二預設標識。

該第二預設標識為 "\\w+"，表示該第一字串是由單詞字元組成。

進一步的，該純單詞字元包括：a-z、A-Z、0-9。

當該第一字串不為空，且該第一字串是非空字元組成，且該待收集的字串為空字串或該待收集的字串的第一個字元是空字元時；返回第三預設標識。

該第三預設標識為 "\\S+"，表示該第一字串是由非空字元組成。

進一步的，這裡的空字元包括：空格、“\t”、“\n”、“\r”、“\v”、“\f”。其中，“\t”是轉義字元，代表水準製表（Tab）；“\n”是轉義字元，代表換行；“\r”是轉義字元，代表回車；“\v”是轉義字元，代表垂直製表；“\f”是轉義字元，代表換頁。因此，只要不是空格、“\t”、“\n”、“\r”、“\v”、“\f”的字元都是非空字元。

當該第一字串為空時；返回第四預設標識。

該第四預設標識為 ""，表示該第一字串為空。

進一步的，當第一字串不符合第二預設規則時，可以判斷該待收集的字串的字元特徵。

S520：當該待收集的字串符合第三預設規則時，返回對應結果並作為該第一部分規則運算式。

這裡的第三預設規則包括：

當該待收集的字串不為空，且該待收集的字串的第一個字元是分隔符號，且該第一字串不包含該待收集的字串的第一個字元時，返回第五預設標識。

該分隔符號包括“ ”（空格），“[”（中括弧），“]”（中括弧），“\”（雙引號），“-”（連字號），“\t”（Tab），“:”（冒號），“,”（逗號），“;”（分號）。

進一步的，該第五預設標識包括“[^”+該待收集的字串的第一個字元+“]”+”，代表該待收集的字串不為空，且該待收集的字串的第一個字元為分隔字元，且該第一字串不包含該待收集的字串的第一個字元。

當該待收集的字串為空時，返回第六預設標識。

該第六預設標識為“.\*”，表示該待收集的字串為空。

進一步的，當該待收集的字串不符合第三預設規則時，可以進一步判斷該第一字串的字元特徵。

S530：當該第一字串符合第四預設規則時，返回對應結果並作為該第一部分規則運算式。

這裡的第四預設規則包括：

定義第一預設變數，依次遍歷該第一字串中的字元，判斷該第一字串中的字元是否為分隔字元。

當該判斷為是時，將該字元追加到第一預設變數。

例如：定義的第一預設變數為 `res1 = ""`，假設該第一字串為“-”，也即在遍歷該第一字串時，該第一字串中的字元為分隔字元，故將該字元追加到第一預設變數後，`res1 = "-"`。

其中，空格，“\t” (Tab) 需要轉義變成“\\s”； “[” (中括弧) 需要轉義變成“\[”； “\” (雙引號) 需要轉義變成“\\”；其他的分隔字元可以直接追加到第一預設變數。

例如：定義的第一預設變數為 `res1 = ""`，假設該第一字串為“- [”，也即在遍歷該第一字串時，該第一字串中的字元為分隔字元，故將該字元追加到第一預設變數後，`res1 = "-\\s\\[”`。

當該判斷為否時，停止遍歷。

判斷是否遍歷全部該第一字串中的字元。

當遍歷全部該第一字串中的字元時，返回第一預設變數。

當前第一預設變數作為該第一部分規則運算式。

當沒有遍歷全部該第一字串中的字元，且當該沒有遍歷的字串中不包含分隔字元時；獲取該第一字串中沒有遍歷的字串，調用步驟 S510。將該第一字串中沒有遍歷的字串作為該第一字串。

進一步的，當當前第一字串符合第二預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

進一步的，當當前第一字串符合第四預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

進一步的，當沒有遍歷全部該第一字串中的字元，且當該沒有遍歷的字串中包含分隔字元時，獲取該第一字串中沒有遍歷的字串。

根據該沒有遍歷的字串中的第一個分隔字元將該沒有遍歷的字串分隔為第一個分隔字元前面的字串、第一個分隔字元和第一個分隔字元後面的字串。

進一步的，將該第一個分隔字元前面的字串作為第一字串，調用步驟 S510。當當前第一字串符合第二預設規則時，將返回的結果追加到第一預設變數。

進一步的，當當前第一字串不符合第二預設規則時，將該第一個分隔字元作為待收集的字串，調用步驟 S520。當當前待收集的字串符合第三預設規則時，將返回的結果追加到第一預設變數。

當當前待收集的字串不符合第三預設規則時，判斷當前第一字串是否符合第四預設規則。當該判斷為是時，將返回的結果追加到第一預設變數。

進一步的，將該第一分隔字元作為第一字串，調用步驟 S510。當當前第一字串符合第二預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

進一步的，將該第一個分隔字元後面的字串作為待收



集的字串，調用步驟 S520。當當前待收集的字串符合第三預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

進一步的，將該第一分隔字元作為第一字串，當當前第一字串符合第四預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

進一步的，該第一預設規則還包括：

需要判斷該待收集的字串和該第二字串中的字元特徵，並根據該待收集字串和該第二字串中的字元特徵生成第二部分規則運算式。將該待收集的字串作為上述確定第一部分規則運算式操作中的第一字串，將該第二字串作為上述確定第一部分規則運算式操作中的待收集的字串，並執行上述操作，將返回的結果作為第二部分規則運算式。具體包括：

S540：將該待收集的字串作為第一字串，調用步驟 S510。當當前第一字串符合第二預設規則時，返回結果作為該第二部分規則運算式。

進一步的，該第一預設規則還可以包括：

S550：將該第二字串作為待收集的字串，調用步驟 S520。當當前待收集的字串符合第三預設規則時，返回結果作為該第二部分規則運算式。

進一步的，該第一預設規則還可以包括：

S560：將該待收集的字串作為第一字串，調用步驟 S530。當當前第一字串符合第四預設規則時，返回結果作

為該第二部分規則運算式。

進一步的，該第一預設規則還可以包括：

S570：根據該第一部分規則運算式和該第二部分規則運算式生成該待收集的規則運算式。

這裡根據該第一部分規則運算式和該第二部分規則運算式生成該待收集的規則運算式，需要先判斷該第二部分規則運算式是否以第五預設標識（“.\*”）結尾；

當該判斷為是時，將該第一部分規則運算式和該第二部分規則運算式以一定規則拼接生成該帶收集的字串的規則運算式。

假設該第一部分規則運算式為：`reg1`，該第二部分規則運算式為：`reg2`，那麼該待收集的字串的規則運算式為：`reg1+“(”+reg2+“)”`。

當該判斷為否時，將該第一部分規則運算式、該第二部分規則運算式和該第五預設標識以一定規則拼接生成該帶收集的字串的規則運算式。

假設該第一部分規則運算式為：`reg1`，該第二部分規則運算式為：`reg2`，那麼該待收集的字串的規則運算式為：`reg1+“(”+reg2+“)+.*`。

進一步的，規則運算式的結尾一般以第五預設標識（“.\*”）結尾。其中“.”表示一個任意字元，“\*”表示0個或多個，“.\*”就是匹配後面有任意多個字元或者沒有字元，由於生成待收集字串的規則運算式不用包括後面的字元的特徵，所以用“.\*”結尾。

需要說明的是，上述生成規則運算式的步驟順序，實際上並不限定必須嚴格按照上述順序。有些步驟也可以在互不依賴的情況下並存執行，比如步驟S510至S530。

以下以上述例子詳細說明規則運算式的獲得方法：

該待收集字串為：“2014-11-02 19:46:03.895”，該第一字串為：“[”，那麼該第二字串為：“] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"”。

先判斷該第一字串 “[”，該第一字串 “[” 不符合步驟S510有關的第二預設規則，進入步驟S520判斷該待收集的字串 “2014-11-02 19:46:03.895”，該待收集的字串 “2014-11-02 19:46:03.895” 不符合步驟S520有關的第三預設規則，進入步驟S530，依次遍歷該第一字串 “[” 中的字元，判斷該第一字串中的字元是否為分隔字元，“[” 是分隔字元且需要轉義，故返回 “\[” 作為第一部分規則運算式。

然後將該待收集的字串 “2014-11-02 19:46:03.895” 作為上述確定第一部分規則運算式操作中的第一字串，將該第二字串 “] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"” 作為上述確定第一部分規則運算式操作中的待收集的字串，並執行上述操作。先判斷字串 “2014-11-02 19:46:03.895”，該字串 “2014-11-02 19:46:03.895” 不符合步驟S510有關的第二預設規則，進入步驟S520開始判斷字串 “] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"”，該字串符合步驟S520該

的第三預設規則中當該待收集的字串不為空，且該待收集的字串的第一個字元是分隔符號，且該第一字串不包含該待收集的字串的第一個字元的規則，也即該字串的第一個字元 "]" 是分隔符號，且此處字串 "2014-11-02 19:46:03.895" 不包含字串 "[INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"" 的第一個字元 "]"，返回 "[^]]+" 作為第二部分規則運算式。

由於第二部分規則運算式為： "[^]]+"，不是以第五預設標識（ ".\*" ）結尾，故當使用者需要收集的字串為： "2014-11-02 19:46:03.895" 時，最終生成的規則運算式為： "\\([^]]+).\*"。

進一步的，當使用者需要獲得輸入的整個日誌片段的規則運算式時，可以通過依次選中具有一定意義的字串，最後以一定規則將每個選中的字串的規則運算式拼接起來。

比如上述日誌片段： "[2014-11-02 19:46:03.895] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK""。

首先，選中 "2014-11-02 19:46:03.895"，該第一字串為： "[ "，該待收集的字串為： "2014-11-02 19:46:03.895"，該第二字串為： "] [INFO] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK""，故返回的結果為 "\\([^]]+).\*"。

然後，選中 "INFO"，該第一字串為： "] ["，該

待收集的字串為：“INFO”，該第二字串為：“] access - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"”，故返回的結果為“]\\\\s\\\\[(\\\\w+).\*”。

接著，選中“access”，該第一字串為：“]”，該待收集的字串為：“access”，該第二字串為：“ - 127.0.0.1 - - "GET / HTTP/1.1" 304 - "OK"”，故返回的結果為“]\\\\s(\\\\w+).\*”。

接著，選中“127.0.0.1”，該第一字串為：“ - ”，該待收集的字串為：“127.0.0.1”，該第二字串為：“ - - "GET / HTTP/1.1" 304 - "OK"”，故返回的結果為“\\\\s-\\\\s(\\\\S+).\*”。

接著，選中“GET / HTTP/1.1”，該第一字串為：“ - - ”，該待收集的字串為：“GET / HTTP/1.1”，該第二字串為：“ " 304 - "OK"”，故返回的結果為“\\\\s-\\\\s\\\\"([^\"]+).\*”。

接著，選中“304”，該第一字串為：“ ”，該待收集的字串為：“304”，該第二字串為：“ - "OK"”，故返回的結果為“\\\\s\\\\s(\\\\d+).\*”。

接著，選中“OK”，該第一字串為：“ - ”，該待收集的字串為：“OK”，該第二字串為：“ ”，故返回的結果為“\\\\s-\\\\s\\\\"(\\\\w+).\*”。

最後，將上述返回的結果拼接起來，除了最後的返回結果，其他返回結果結尾的“.\*”去掉，並對轉義的字元進行恢復，得到最終的規則運算式為：

$\backslash([^\wedge]+)\d+\backslashs\backslash([\w+])\backslashs\backslash(\w+)\backslashw+\backslashs-\backslashs\backslash(s+)\backslashs-\backslashs-\backslashs"([^\wedge]+)\d+" \backslashs\backslash(\d+)\backslashs-\backslashs"(\w+).*$

進一步的，當要獲取多於一個具有一定意義的字串的規則運算式時，該前面的字串選擇的時候是選擇該要收集的字串之前的字串，但要去除已經選擇過的字串及選擇過字串前面的字串。

由此可見，本發明的實施例提供的技術方案通過回應於觸發指令，可以根據使用者的需要獲取預設字串中待收集的字串，並根據預設規則生成該待收集字串的規則運算式。與現有技術相比，大大提高了用戶使用的靈活性，可以根據使用者需要生成相應字串的規則運算式，進而可以根據該規則運算式解析出對使用者有用的字串。

以下介紹本發明的一種規則運算式的生成系統的實施例。結合附圖3，該系統600包括：

預設字串獲得單元610，用於獲得預設字串。

待收集字串獲得單元620，用於回應於觸發指令獲得預設字串中的待收集字串。

第一字串獲得單元630，用於從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串。

第二字串獲得單元640，用於從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串。

規則運算式生成單元650，用於根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。

在一個優選的實施例中，該待收集字串獲得單元620

包括：

第一操作模組，用於回應選擇並拖拽該預設字串中的部分字串的操作指令，獲得待收集字串；或，

第二操作模組，用於回應點擊該預設字串中的部分字串的起始位置的第一指令，以及回應點擊該預設字串中的部分字串的結束位置的第二指令，獲得待收集字串；或，

第三操作模組，用於選擇並回應右擊該預設字串中的部分字串所在區域的操作指令，獲得待收集字串。

由此可見，本發明的實施例提供的一種規則運算式的生成方法及系統，通過回應於觸發指令，可以根據使用者的需要獲取預設字串中待收集的字串，並根據預設規則生成該待收集字串的規則運算式。與現有技術相比，大大提高了用戶使用的靈活性，可以根據使用者需要生成相應字串的規則運算式，進而可以根據該規則運算式解析出對使用者有用的字串。

在20世紀90年代，對於一個技術的改進可以很明顯地區分是硬體上的改進（例如，對二極體、電晶體、開關等電路結構的改進）還是軟體上的改進（對於方法流程的改進）。然而，隨著技術的發展，當今的很多方法流程的改進已經可以視為硬體電路結構的直接改進。設計人員幾乎都通過將改進的方法流程程式設計到硬體電路中來得到相應的硬體電路結構。因此，不能說一個方法流程的改進就不能用硬體實體模組來實現。例如，可程式設計邏輯裝置（Programmable Logic Device, PLD）（例如現場可程式設

計閘陣列 ( Field Programmable Gate Array, FPGA ) ) 就是這樣一種積體電路，其邏輯功能由使用者對裝置程式設計來確定。由設計人員自行程式設計來把一個數位系統“集成”在一片PLD上，而不需要請晶片製造廠商來設計和製作專用的積體電路晶片<sup>2</sup>。而且，如今，取代手工地製作積體電路晶片，這種程式設計也多半改用“邏輯編譯器 ( logic compiler ) ”軟體來實現，它與程式開發撰寫時所用的軟體編譯器相類似，而要編譯之前的原始代碼也得用特定的程式設計語言來撰寫，此稱之為硬體描述語言 ( Hardware Description Language, HDL ) ，而HDL也並非僅有一種，而是有許多種，如ABEL ( Advanced Boolean Expression Language ) 、 AHDL ( Altera Hardware Description Language ) 、 Confluence 、 CUPL ( Cornell University Programming Language ) 、 HDCal 、 JHDL ( Java Hardware Description Language ) 、 Lava 、 Lola 、 MyHDL 、 PALASM 、 RHDL ( Ruby Hardware Description Language ) 等，目前最普遍使用的是VHDL ( Very-High-Speed Integrated Circuit Hardware Description Language ) 與Verilog<sup>2</sup>。所屬技術領域中具有通常知識者也應該清楚，只需要將方法流程用上述幾種硬體描述語言稍作邏輯程式設計並程式設計到積體電路中，就可以很容易得到實現該邏輯方法流程的硬體電路。

控制器可以按任何適當的方式實現，例如，控制器可以採取例如微處理器或處理器以及儲存可由該 ( 微 ) 處理



器執行的電腦可讀程式碼（例如軟體或固件）的電腦可讀媒介、邏輯閘、開關、應用特定積體電路（Application Specific Integrated Circuit, ASIC）、可程式設計邏輯控制器和嵌入微控制器的形式，控制器的例子包括但不限於以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及Silicone Labs C8051F320，記憶體控制器還可以被實現為記憶體的的控制邏輯的一部分。

所屬技術領域中具有通常知識者也知道，除了以純電腦可讀程式碼方式實現控制器以外，完全可以通過將方法步驟進行邏輯程式設計來使得控制器以邏輯閘、開關、應用特定積體電路、可程式設計邏輯控制器和嵌入微控制器等的形式來實現相同功能。因此這種控制器可以被認為是一種硬體部件，而對其內包括的用於實現各種功能的裝置也可以視為硬體部件內的結構。或者甚至，可以將用於實現各種功能的裝置視為既可以是實現方法的軟體模組又可以是硬體部件內的結構。

上述實施例闡明的系統、裝置、模組或單元，具體可以由電腦晶片或實體實現，或者由具有某種功能的產品來實現。

為了描述的方便，描述以上裝置時以功能分為各種單元分別描述。當然，在實施本發明時可以把各單元的功能在同一個或多個軟體和/或硬體中實現。

通過以上的實施方式的描述可知，本領域的技術人員可以清楚地瞭解到本發明可借助軟體加必需的通用硬體平

臺的方式來實現。基於這樣的理解，本發明的技術方案本質上或者說對現有技術做出貢獻的部分可以以軟體產品的形式體現出來，在一個典型的配置中，計算設備包括一個或多個處理器(CPU)、輸入/輸出介面、網路介面和記憶體。該電腦軟體產品可以包括若干指令用以使得一台電腦設備（可以是個人電腦，伺服器，或者網路設備等）執行本發明的各個實施例或者實施例的某些部分該的方法。該電腦軟體產品可以儲存在記憶體中，記憶體可能包括電腦可讀媒介中的非永久性記憶體，隨機存取記憶體(RAM)和/或非易失性記憶體等形式，如唯讀記憶體(ROM)或快閃記憶體(flash RAM)。記憶體是電腦可讀媒介的示例。電腦可讀媒介包括永久性和非永久性、可移動和非可移動媒體可以由任何方法或技術來實現資訊儲存。資訊可以是電腦可讀指令、資料結構、程式的模組或其他資料。電腦的儲存媒介的例子包括，但不限於相變記憶體(PRAM)、靜態隨機存取記憶體(SRAM)、動態隨機存取記憶體(DRAM)、其他類型的隨機存取記憶體(RAM)、唯讀記憶體(ROM)、電可抹除可程式設計唯讀記憶體(EEPROM)、快閃記憶體或其他記憶體技術、唯讀光碟唯讀記憶體(CD-ROM)、數位多功能光碟(DVD)或其他光學儲存、磁盒式磁帶，磁帶磁磁片儲存或其他磁性存放裝置或任何其他非傳輸媒介，可用於儲存可以被計算設備存取的資訊。按照本文中的界定，電腦可讀媒介不包括暫態媒介(transitory media)，如調變的資料信號和載波。

本說明書中的各個實施例均採用遞進的方式描述，各個實施例之間相同相似的部分互相參見即可，每個實施例重點說明的都是與其他實施例的不同之處。尤其，對於系統實施例而言，由於其基本相似於方法實施例，所以描述的比較簡單，相關之處參見方法實施例的部分說明即可。

本發明可用於眾多通用或專用的電腦系統環境或配置中。例如：個人電腦、伺服器電腦、手持設備或可攜式設備、平板型設備、多處理器系統、基於微處理器的系統、置頂盒、可程式設計的消費電子設備、網路PC、小型電腦、大型電腦、包括以上任何系統或設備的分散式運算環境等等。

本發明可以在由電腦執行的電腦可執行指令的一般上下文中描述，例如程式模組。一般地，程式模組包括執行特定任務或實現特定抽象資料類型的常式、程式、物件、元件、資料結構等等。也可以在分散式運算環境中實踐本發明，在這些分散式運算環境中，由通過通信網路而被連接的遠端處理設備來執行任務。在分散式運算環境中，程式模組可以位於包括存放裝置在內的本地和遠端電腦儲存媒介中。

雖然通過實施例描繪了本發明，本領域通常知識者知道，本發明有許多變形和變化而不脫離本發明的精神，希望所附的請求項包括這些變形和變化而不脫離本發明的精神。

**【符號說明】**

610：預設字串獲得單元

620：待收集字串獲得單元

630：第一字串獲得單元

640：第二字串獲得單元

650：規則運算式生成單元

## 申請專利範圍

1.一種規則運算式的生成方法，其特徵在於，包括：  
獲得預設字串；

回應於觸發指令獲得預設字串中的待收集字串；

從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串；

從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串；

根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。

2.根據請求項第 1 項所述的規則運算式的生成方法，其中，該觸發指令包括：

選擇並拖拽該預設字串中的部分字串的操作；或，

點擊該預設字串中的部分字串的起始位置的第一指令，以及點擊該部分字串的結束位置的第二指令；或，

選擇並右擊該預設字串中的部分字串所在區域的操作。

3.根據請求項第 1 項所述的規則運算式的生成方法，其中，該第一預設規則包括：

當該第一字串符合第二預設規則時，返回對應結果並作為該第一部分規則運算式。

4.根據請求項第 3 項所述的規則運算式的生成方法，其中，該第二預設規則包括：

當該第一字串不為空，且該第一字串是純數字字元組

成，且該待收集的字串為空或該待收集的字串的第一個字元不是純數字字元時，返回第一預設標識。

5.根據請求項第 3 項所述的規則運算式的生成方法，其中，該第二預設規則包括：

當該第一字串不為空，且該第一字串是包含字母的純單詞字元組成，且該待收集的字串為空或該待收集的字串的第一個字元不是純單詞字元時，返回第二預設標識。

6.根據請求項第 3 項所述的規則運算式的生成方法，其中，該第二預設規則包括：

當該第一字串不為空，且該第一字串是非空字元組成，且該待收集的字串為空字串或該待收集的字串的第一個字元是空字元時，返回第三預設標識。

7.根據請求項第 3 項所述的規則運算式的生成方法，其中，該第二預設規則包括：

當該第一字串為空時，返回第四預設標識。

8.根據請求項第 1 項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

當該待收集的字串符合第三預設規則時，返回對應結果並作為該第一部分規則運算式。

9.根據請求項第 8 項所述的規則運算式的生成方法，其中，該第三預設規則包括：

當該待收集的字串不為空，且該待收集的字串的第一個字元是分隔符號，且該第一字串不包含該待收集的字串的第一個字元時，返回第五預設標識。

10. 根據請求項第 8 項所述的規則運算式的生成方法，其中，該第三預設規則包括：

當該待收集的字串為空時，返回第六預設標識。

11. 根據請求項第 1 項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

當該第一字串符合第四預設規則時，返回對應結果並作為該第一部分規則運算式。

12. 根據請求項第 11 項所述的規則運算式的生成方法，其中，該第四預設規則包括：

定義第一預設變數，依次遍歷該第一字串中的字元，判斷該第一字串中的字元是否為分隔字元；

當該判斷為是時，將該字元追加到第一預設變數；

當該判斷為否時，停止遍歷；

判斷是否遍歷全部該第一字串中的字元。

13. 根據請求項第 12 項所述的規則運算式的生成方法，其中，當遍歷全部該第一字串中的字元時，該第四預設規則還包括：

返回第一預設變數。

14. 根據請求項第 12 項所述的規則運算式的生成方法，其中，當沒有遍歷全部該第一字串中的字元，且當該沒有遍歷的字串中不包含分隔字元時，該第四預設規則還包括：

獲取該第一字串中沒有遍歷的字串，將該第一字串中沒有遍歷的字串作為該第一字串。

15.根據請求項第 14 項所述的規則運算式的生成方法，其中，當當前第一字串符合第二預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

16.根據請求項第 14 項所述的規則運算式的生成方法，其中，當當前第一字串符合第四預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

17.根據請求項第 12 項所述的規則運算式的生成方法，其中，當沒有遍歷全部該第一字串中的字元，且當該沒有遍歷的字串中包含分隔字元時，該第四預設規則還包括：

獲取該第一字串中沒有遍歷的字串；

根據該沒有遍歷的字串中的第一個分隔字元將該沒有遍歷的字串分隔為第一個分隔字元前面的字串、第一個分隔字元和第一個分隔字元後面的字串。

18.根據請求項第 17 項所述的規則運算式的生成方法，其中，將該第一個分隔字元前面的字串作為第一字串，當當前第一字串符合第二預設規則時，將返回的結果追加到第一預設變數。

19.根據請求項第 17 項所述的規則運算式的生成方法，其中，當當前第一字串不符合第二預設規則時，該第四預設規則還包括：

將該第一個分隔字元作為待收集的字串，當當前待收集的字串符合第三預設規則時，將返回的結果追加到第一預設變數。



20.根據請求項第 17 項所述的規則運算式的生成方法，其中，當當前第一字串符合第四預設規則時，該第四預設規則還包括：

將返回的結果追加到第一預設變數。

21.根據請求項第 18 至 20 項中之任一項所述的規則運算式的生成方法，其中，該第四預設規則還包括：

將該第一分隔字元作為第一字串，當當前第一字串符合第二預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

22.根據請求項第 18 至 20 項中之任一項所述的規則運算式的生成方法，其中，該第四預設規則還包括：

將該第一個分隔字元後面的字串作為待收集的字串，當當前待收集的字串符合第三預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

23.根據請求項第 18 至 20 項中之任一項所述的規則運算式的生成方法，其中，該第四預設規則還包括：

將該第一分隔字元作為第一字串，當當前第一字串符合第四預設規則時，將返回的結果追加到第一預設變數，返回第一預設變數。

24.根據請求項第 3、8 或 11 項中之任一項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

將該待收集的字串作為第一字串，當當前第一字串符合第二預設規則時，返回對應結果並作為該第二部分規則運算式。

25.根據請求項第 24 項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

根據該第一部分規則運算式和該第二部分規則運算式生成該待收集的規則運算式。

26.根據請求項第 3、8 或 11 項中之任一項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

將該第二字串作為待收集的字串，當當前待收集的字串符合第三預設規則時，返回對應結果並作為該第二部分規則運算式。

27.根據請求項第 26 項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

根據該第一部分規則運算式和該第二部分規則運算式生成該待收集的規則運算式。

28.根據請求項第 3、8 或 11 項中之任一項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

將該待收集的字串作為第一字串，當當前第一字串符合第四預設規則時，返回對應結果並作為該第二部分規則運算式。

29.根據請求項第 28 項所述的規則運算式的生成方法，其中，該第一預設規則還包括：

根據該第一部分規則運算式和該第二部分規則運算式生成該待收集的規則運算式。

30.一種規則運算式的生成系統，其特徵在於，包括：

預設字串獲得單元，用於獲得預設字串；

待收集字串獲得單元，用於回應於觸發指令獲得預設字串中的待收集字串；

第一字串獲得單元，用於從該預設字串中識別在該待收集字串前面的字串，該前面的字串作為第一字串；

第二字串獲得單元，用於從該預設字串中識別在該待收集字串後面的字串，該後面的字串作為第二字串；

規則運算式生成單元，用於根據該待收集字串、該第一字串和該第二字串的字元特徵以第一預設規則生成該待收集字串的規則運算式。

31.根據請求項第 30 項所述的規則運算式的生成系統，其中，該待收集字串獲得單元包括：

第一操作模組，用於回應選擇並拖拽該預設字串中的部分字串的操作指令，獲得待收集字串；或，

第二操作模組，用於回應點擊該預設字串中的部分字串的起始位置的第一指令，以及回應點擊該預設字串中的部分字串的結束位置的第二指令，獲得待收集字串；或，

第三操作模組，用於選擇並回應右擊該預設字串中的部分字串所在區域的操作指令，獲得待收集字串。

# 圖式

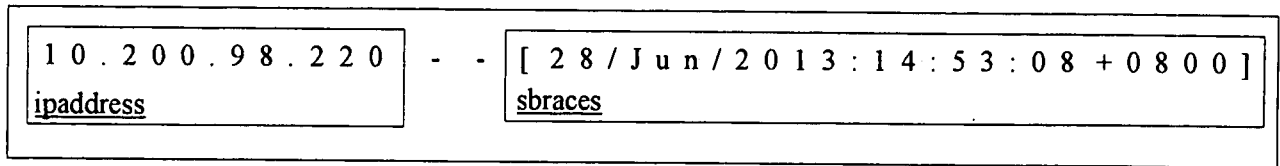


圖 1

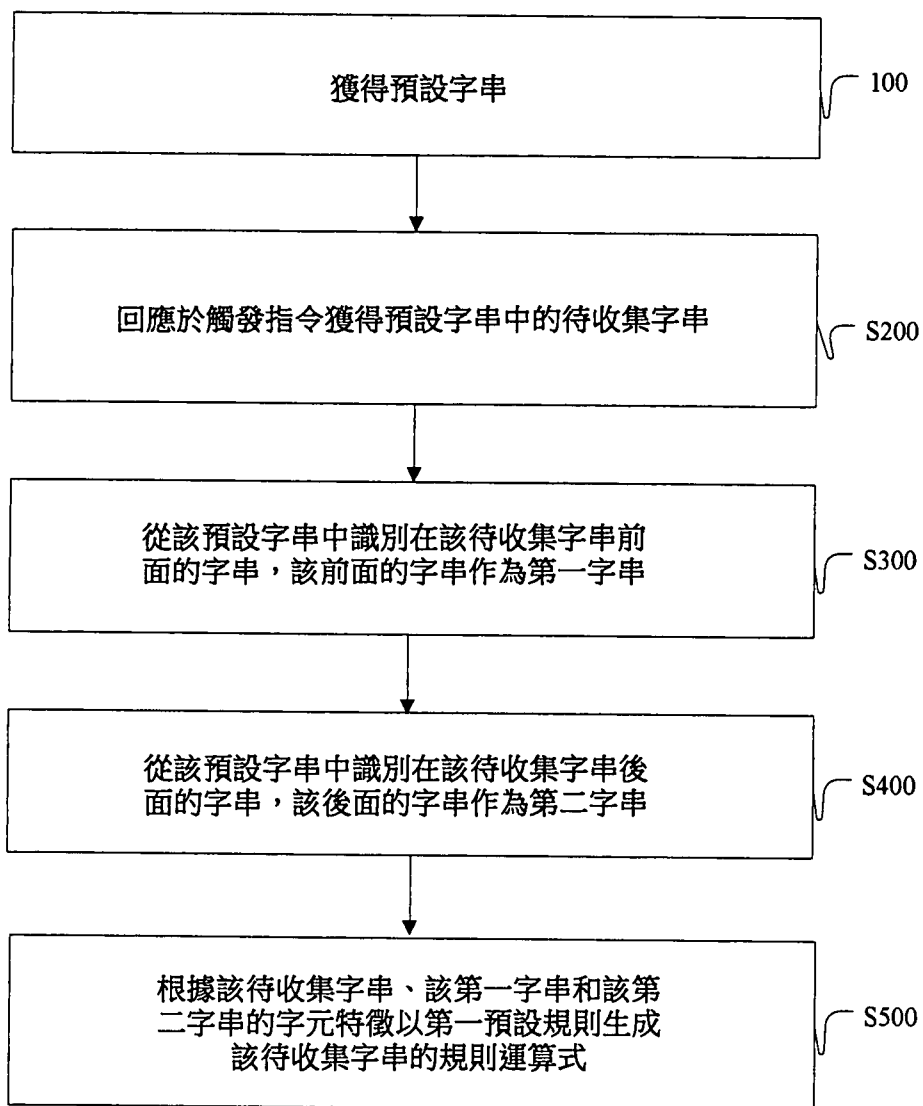


圖 2

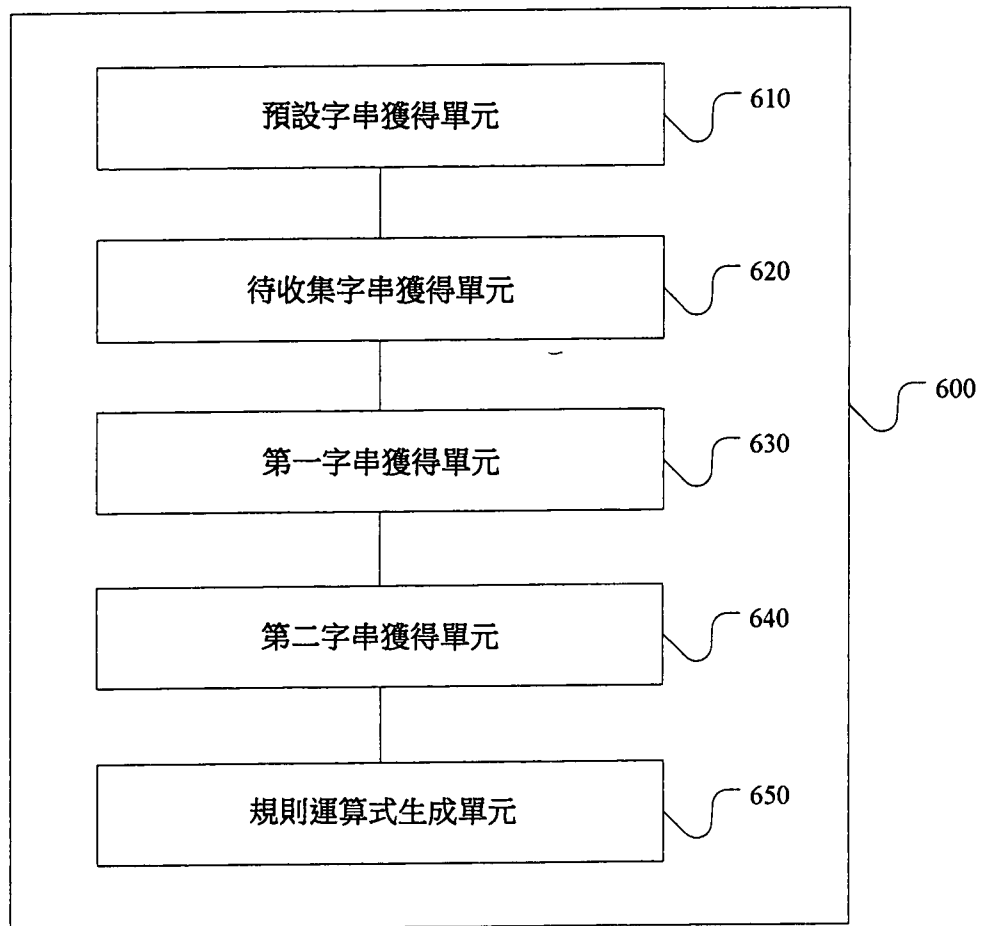


圖 3