International Journal of Information Technology & Decision Making
Vol. 17 (2018)

DOI: 10.1142/S0219622018500347

A Language for Financial Chart Patterns

Jiajun Zhu*, Yuqing Wan† and Yain-Whar Si‡

Department of Computer and Information Science
University of Macau, P. R. China
**june306333131@hotmail.com*
†qinggor@qq.com
‡fstasp@umac.mo

Published 16 July 2018

In stock markets around the world, financial analysts continuously monitor and screen chart patterns (technical patterns) to predict future price trends. Although a plethora of methods have been proposed for classification of these patterns, there is no uniform standard in defining their shapes. To facilitate the classification and discovery of chart patterns in financial time series, we propose a novel domain-specific language called "Financial Chart Pattern Language" (FCPL). The proposed language is formally described in Extended Backus–Naur Form (EBNF). FCPL allows incremental composition of complex shapes from simple basic units called primitive shapes. Hence, patterns defined in FCPL can be reused for composing new chart patterns. FCPL separates the specification of a chart pattern from the mechanism of its implementation. Due to its simplicity, FCPL can be used by stock market experts and end users to describe the patterns without programming expertise. To highlight its capabilities, several representative financial chart patterns are defined in FCPL for illustration. In the experiments, we classify several representative chart patterns from the datasets of HANG SENG INDEX (HSI), NYSE AMEX COMPOSITE INDEX (NYSE), and Dow Jones Industrial Average (DJI).

Keywords: Chart patterns; domain-specific language; primitive shapes; financial time series.

1. Introduction

In financial markets, predicting future price trends is one of the biggest challenges for both individual investors and finance companies. Professional traders use two major types of analysis to make accurate decisions in financial markets: fundamental and technical.[1] Technical analysis was developed by Charles Dow in 1884, whilst its modern definition was proposed by Edwards and Magee in 1997.[2] Technical analysis is a method used to predict future price trends by studying the historical prices of stocks. Even though the efficient market hypothesis introduced by Fama[3] indicates that it is not possible to predict the market price based on past price or volume information, studies performed since the 1980s have overturned this hypothesis.[4,5] For example, in Ref. 6, strategies based on moving average are found to be profitable

‡Corresponding author.

for trading in the stock markets of BRICS countries. When designing the strategies, traders may use indicators such simple moving average (SMA), moving average convergence divergence (MACD), and intraday information.[7]

Market participants buy and sell stocks for a variety of reasons. The forces of supply and demand can be reflected in compact diagrams called chart patterns. The appearance of a chart pattern is a signal to predict the future price movement of stocks. Some chart patterns are formed in a short period, and some need a long time to take shape. Therefore, chart patterns can be used to make short-term and long-term predictions. The charting technique may be operationalized through the trading rules of the form[8]: "If chart pattern X is identified in the previous N trading days, then buy; and sell on the Yth trading day after that." The challenging part of the charting technique is the identification of the chart patterns. Different chart patterns have been proposed by stock experts. Richard Schabacker[9] laid the foundations for modern pattern analysis in 1932. Since then, scholars have developed the art of charting in financial time series. Edwards *et al.*[2] introduced technical theories, different chart patterns, and other indicators in financial time series. Bulkowski also detailed the comprehensive descriptions of 53 chart patterns in the *Encyclopedia of Chart Patterns*.[10] In some of the previous studies, researchers described chart patterns with simple shapes, such as saucer[11] and bull flag.[8,12,13]

To facilitate the classification of chart patterns in financial time series, we propose a domain-specific language called Financial Chart Pattern Language (FCPL). The proposed language is formally defined in Extended Backus–Naur Form[14] (EBNF) and can be used to specify the shape and volume of a chart pattern. The main contributions of this paper can be summarized as follows:

- FCPL is a comprehensive domain-specific language. FCPL can be used to describe any existing chart patterns reported in literature[10] or can be used to define new chart patterns.
- Patterns defined in FCPL can be reused. Specifically, FCPL allows incremental composition of chart patterns. In FCPL, complex patterns can be incrementally composed from simple patterns or shapes. FCPL is based on basic units, called primitive shapes, to construct the specific shapes of a price or volume trend. For instance, several primitive shapes can be used to form a curved shape.
- The syntax of FCPL is intentionally kept simple. Therefore, it is suitable for stock experts or end users with no programming experience to define the shapes of their own chart patterns or existing chart patterns. Patterns defined in FCPL can then be translated into target programming languages.

Section 2 presents related work. In Sec. 3, we propose our FCPL. Several examples of representative chart patterns defined in FCPL are discussed in Sec. 4. In Sec. 5, we classify several representative chart patterns from real datasets. Section 6 concludes the paper with future work.

2. Related Work

Domain-specific languages have been developed to facilitate the construction of models at a level closer to the conceptual model, thereby making model implementation more accessible to domain experts.[15] Many domain-specific languages have been proposed for the construction and maintenance of software systems.[16] Van *et al.*[17] propose a textual language called Feature Description Language (FDL) to describe feature diagrams. A feature diagram is formalised by FDL and operations are defined for manipulating FDL expressions. Córdoba-Sánchez *et al.*[18] propose a modeling language to provide a more expressive and suitable syntactic support for designing the sets of Java annotations and their associated integrity constraints. Marand *et al.*[19] propose a domain-specific modeling language for concurrent programming (DSML4CP). DSML4CP can be used to reduce the complexities of the concurrent programs by providing a higher level of abstraction. In the context of business applications, application-specific functionalities correspond to specific business logic. A domain-specific language called IIS*CFuncLang[20] is proposed to enable complete specification of application-specific functionalities at the platform independent model level.

In financial trading, Anand *et al.*[21] propose a chart pattern language (CPL) to facilitate the pattern recognition process. In CPL, six primitive patterns including "bar", "up", "down", "horizontal", "resistance line" and "support line" are used to construct simple chart patterns. Complex chart patterns can be composed with simple chart patterns using the operators "followed-by" and "overlay". The syntactic constraint of CPL is related to the Shape Description Language (SDL),[22] which allows queries about the shapes found in historical time series. CPL is a domain-specific language embedded in the functional programming language Haskell. Therefore, the syntax of CPL is closely related to Haskel. Although CPL provides programming language-liked capabilities, the complexity of Haskel offsets its usability. In contrast to their approach, the syntax of proposed FCPL is rather simple and can be easily understood by novice users. In FCPL, the overall shape of a chart pattern consists of primitive shapes which are connected with special points. Once we define a chart pattern with primitive shapes and special points, we can simply use the definition in classification.

In technical analysis, analysts usually predict future price trend by observing the time indicators of buys and sells on stock charts. Charting is one of the popular methods for observing the forces of supply and demand in the financial markets. Specifically, chart patterns are often treated as indicators in predicting future price trends. In the *Encyclopedia of chart pattern*,[10] examples about designing trading strategies based on different chart patterns are presented in detail. In Ref. 13, Cervelló–Royo *et al.* developed a trading strategy based on "Flag Pattern". In their approach, a weight matrix is used to identify the "Flag Pattern" from the financial time series. In their strategy, a buying or selling operation is triggered when a bear or bull Flag is detected in the financial time series. In contrast to their approach, in this paper, we focus on the development of a domain-specific language for specifying chart patterns.

The notion of pattern is also used in other problem domains. For instance, in Ref. 23, Wu *et al.* introduced the concept of real estate investment pattern which can be characterized by attributes such as location, period, and value of the property. In contrast to financial trading, real estate investment often involves multi-criteria decision making by a group of experts. One of the well-known methods for multi-criteria decision making is a consensus model based on Analytic Hierarchy Process (AHP).[23] In AHP, a Pairwise Comparison Matrix (PCM)[24] is often used to compute the relative priorities of the alternatives.

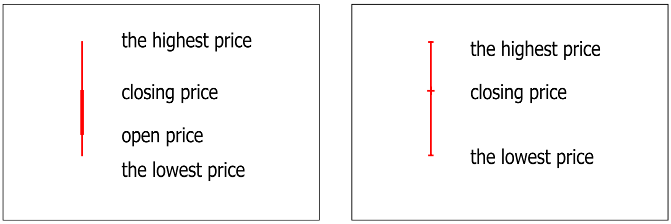
3. Financial Chart Pattern Language

A time series of a stock can be constructed based on the historical price or traded volume. In this paper, "a price time series" is used to denote the time series constructed from the price data and "a volume time series" is used to denote the time series constructed from the volume data. A price time series is usually represented with candlesticks. A daily candlestick is a price stick that consists of "opening price", "highest price", "lowest price" and "closing price" for a day. Figure 1(a) depicts a candlestick which consists of all four price data. A candlestick can also be represented without an opening price as shown in Fig. 1(b). A daily volume is the number of shares or contracts traded in a day.

In Ref. 10, Bulkowski describes the identification guidelines for 53 chart patterns. In these guidelines, a chart pattern can be identified by detecting the change in the "price trend" over a period of time. In some cases, the shape of the "volume trend" is also an important characteristic to identify a chart pattern. In the following sections, we introduce the primitive shapes that can be used as building blocks to construct the shape of a price or volume trend.

3.1. *Primitive shapes*

Figure 2 shows a hierarchy of primitive shapes that are used to construct the shape of a price or volume trend. In FCPL, reserved terms "up", "su", "wn", "se", "dn", "sd"



(a) (b)

Fig. 1. (a) A candlestick with "opening price", "highest price", "lowest price" and "closing price". (b) A candlestick with "highest price", "closing price" and "lowest price".

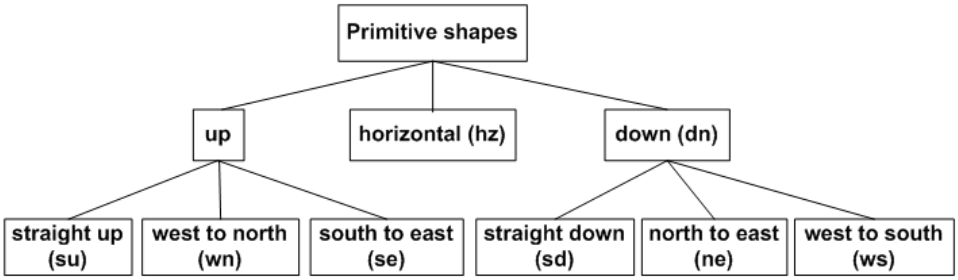


Fig. 2. Primitive shapes that are used to construct the shape of a price or volume trend.

"ne" and "ws" are used to describe these primitive shapes. In the following figures (from Figs. 3–11), each point refers to one of the prices of a candlestick (e.g., "closing price") when we describe the primitive shapes of the price. On the other hand, when we describe the primitive shapes of the volume, the point denotes the traded volume. In the following discussion, we use the term "indicators" to represent one of the prices of a candlestick or the volume of a time point.

3.1.1. *Upward primitive shapes*

(1) **"su" — straight up shapes:** N ($N \in [2, +\infty)$) candlesticks form a straight-up shape if and only if every candlestick's indicator is greater than the indicator of its previous candlesticks, and the slopes of two neighboring line segments are the same (see Fig. 3).
(2) **"wn" — curved shapes from west to north (smooth up):** N ($N \in [3, +\infty)$) candlesticks form a curved shape from west to north if and only if every candlestick's indicator is greater than the indicator of its previous candlesticks, and the slopes of two neighboring line segments are in decreasing order (see Fig. 4).
(3) **"se" — Curved shapes from south to east (sharp up):** N ($N \in [3, +\infty)$) candlesticks form a curved shape from south to east if and only if every candlestick's indicator is greater than the indicators of its previous candlesticks, and the slopes of two neighboring line segments are in increasing order (see Fig. 5).
(4) **"up" — up shapes:** N ($N \in [2, +\infty)$) candlesticks form an up shape if and only if every candlestick's indicator is greater than the indicators of its previous candlesticks (see Fig. 6).

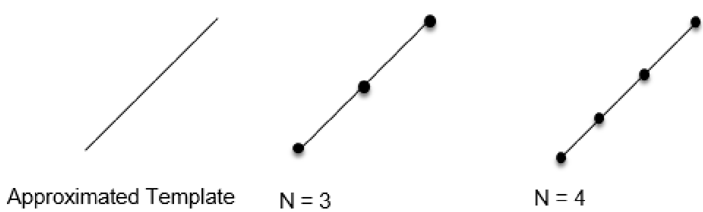


Fig. 3. Straight up shapes.

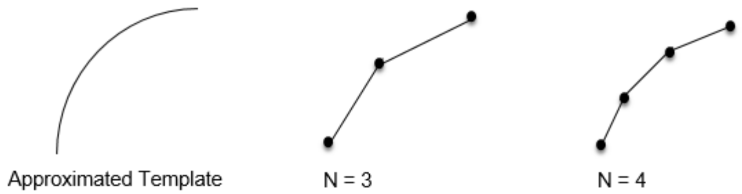


Fig. 4. Curved shapes from west to north.

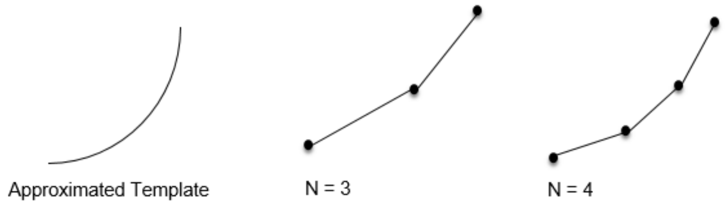


Fig. 5. Curved shapes from south to east.

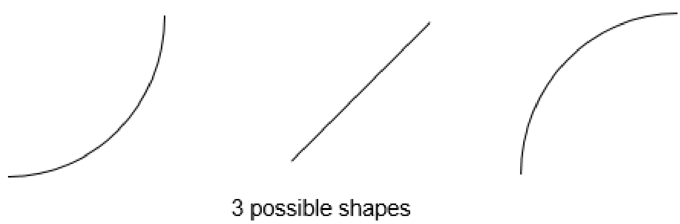


Fig. 6. Up shapes.

3.1.2. *Downward primitive shapes*

(1) **"sd" — straight down Shapes:** N ($N \in [2, +\infty)$) candlesticks form a straight down shape if and only if every candlestick's indicator is less than the indicator of its previous candlesticks, and the slopes of two neighboring line segments are the same (see Fig. 7).

(2) **"ne" — curved shapes from north to east (sharp down):** N ($N \in [3, +\infty)$) candlesticks form a curved shape from north to east if and only if every

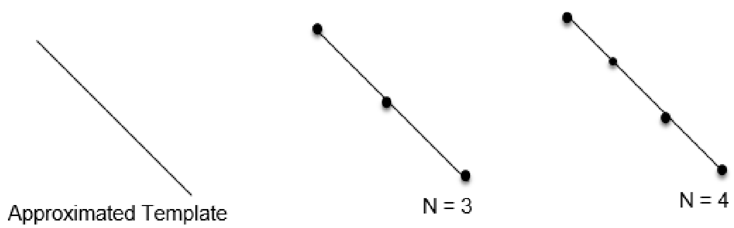


Fig. 7. Straight down shapes.

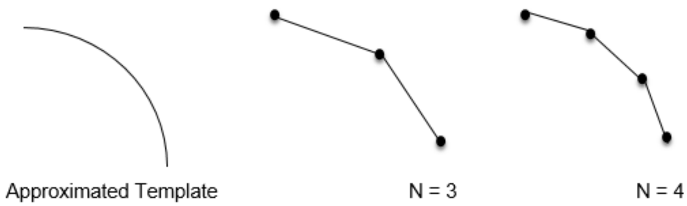


Fig. 8. Curved shapes from north to east.

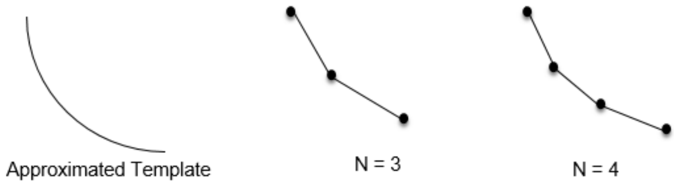


Fig. 9. Curved shapes from west to south.

candlestick's indicator is less than the indicator of its previous candlesticks, and the slopes of two neighbor line segments are in decreasing order (see Fig. 8).

(3) **"ws" — curved shapes from west to south (smooth down):** N ($N \in [3, +\infty)$) candlesticks form a curved shape from west to south if and only if every candlestick's indicator is less than the indicator of its previous candlesticks, and the slopes of two neighbor line segments are in increasing order (see Fig. 9).

(4) **"dn" — down Shapes:** N ($N \in [2, +\infty)$) candlesticks form a down shape if and only if a candlestick's indicator is less than the indicator of its previous candlesticks (see Fig. 10).

3.1.3. *Horizontal primitive shapes*

(1) **"hz" — horizontal shapes:** N ($N \in [2, +\infty)$) candlesticks form a horizontal shape if and only if all candlesticks' indicators are at the same level (see Fig. 11).

3.2. *Extended Bakus–Naur form*

EBNF[14,25] is a syntactic metalanguage that defines the syntax of a language by a set of rules. An EBNF consists of terminal symbols and nonterminal production rules,

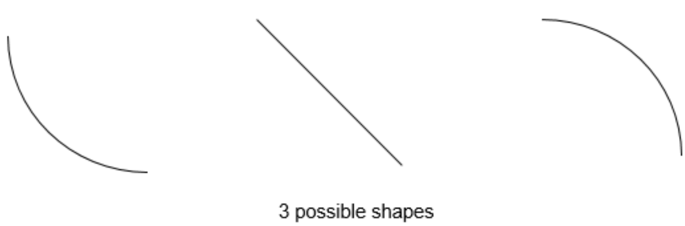


Fig. 10. Down shapes.

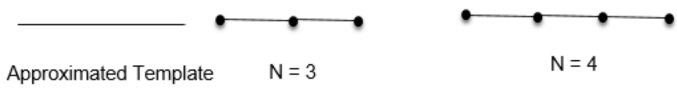


Fig. 11. Horizontal shapes.

Table 1. EBNF.

EBNF	Meaning
=	definition
,	concatenation
;	termination
|	alternation
[...]	option
{...}	repetition
(...)	grouping
"..."	terminal string
'...'	terminal string
(*...*)	comment
?...?	special sequence
−	exception

which are the restrictions that govern how terminal symbols can be combined into a legal sequence. EBNF is commonly used to define a formal description of a language. Table 1 shows the summary of ISO/IEC 14977 EBNF.[26]

EBNF is used to formally describe the syntax of FCPL. An EBNF description is an unordered list of EBNF rules. In this paper, five rules expressed with EBNF are used to define a chart pattern. The first EBNF rule is used to describe the name of a chart pattern. The second EBNF rule is used to describe the shape of a chart pattern with primitive shapes and special points. The third EBNF rule is used to describe the relationship between different special points. In addition, we use two EBNF rules to describe the rules for the volume trend of some of the chart patterns. The fourth EBNF rule is used to describe the shape of a volume trend formed by special points. The fifth EBNF rule is used to describe the relationships between these special points.

The EBNF expression of the terms used in the rules to define a chart pattern are listed in Table 2. The term "shape" in Table 2 denotes the primitive shapes, which are explained in Sec. 3.1. "Sn" denotes the "price special point" which connects two different primitive shapes in a chart pattern. "A price special point" is a point on a selected candlestick. Segmentation methods can be used to extract/select certain candlesticks from an input time series. With segmentation, we can ignore small fluctuations of the original time series.

Perceptual Important Points (PIP)[27] is one of the most commonly used segmentation methods. The PIP method extracts a set of critical points called PIPs from the original subsequence. Special points can be PIPs or other critical points

Table 2. Definitions of terms in EBNF.

<table>
<tr><th>Term</th><th>Explanation</th><th>EBNF expression</th></tr>
<tr><td>Letter</td><td>English letters</td><td>"A" | "B" | "C" | "D" | "E" | "F" | "G" | "H"
| "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P"
| "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X"
| "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f"
| "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n"
| "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v"
| "w" | "x" | "y" | "z" |</td></tr>
<tr><td>shape</td><td>primitive shapes</td><td>"up" | "su" | "wn" | "se" | "dn" | "sd" | "ne" | "ws" | "hz";</td></tr>
<tr><td>Sn</td><td>special price points</td><td>"S0" | "S1" | "S2" | "S3" | "S4" |...| "Sn" |;</td></tr>
<tr><td>Vn</td><td>special volume points</td><td>"V0" | "V1" | "V2" | "V3" | "V4" |...| "Vn" |;</td></tr>
<tr><td>A</td><td>attributes</td><td>"h" | "l" | "c" | "o" | "x" | "v"</td></tr>
<tr><td>TSD</td><td>time series data</td><td>(Sn, ".", A)|(Vn)</td></tr>
<tr><td>difference</td><td>the difference of two values</td><td>"d(", TSD, ",", TSD, ")"</td></tr>
<tr><td>slope</td><td>the slope of two time series data</td><td>"slope(", TSD, ",", TSD, ")"</td></tr>
<tr><td>width</td><td>the width between two time series data</td><td>"w(", TSD, ",", TSD, ")"</td></tr>
<tr><td>operator</td><td>mathematical operator</td><td>"<" | ">" | "=" | ">=" | "<=";</td></tr>
<tr><td>R</td><td>random real number</td><td>R is a random real number</td></tr>
<tr><td>NULL</td><td>empty expression</td><td>""</td></tr>
</table>

extracted with other segmentation methods. For example, Fig. 12 shows the process of extracting five PIPs from a time series of closing price. Other commonly used segmentation methods include the piecewise aggregate approximation (PAA) method,[28] the piecewise linear approximation (PLA) method,[29] the turning points (TP) method,[30] etc.

Note that approaches such as manifold learning algorithms can be used to reduce the dimensionality of high-dimensional financial data. By identifying the intrinsic characteristics and structure of the high-dimensional data, manifold learning algorithms can produce a simplified and nonoverlapping representation of the data. Huang *et al.*[31] proposed a kernel entropy manifold learning algorithm for generating a low-dimensional embedding of the original financial data set. They adopted an information theory-based metric to measure the difference between data points. In Ref. 32, an information metric-based manifold learning algorithm is proposed to extract the intrinsic manifold of a dynamic financial system. The underlying manifold is then used to detect early warning ranges for critical transitions in financial markets. Recall that the time series data considered in this paper only consist of price or volume trends. Therefore, for the purpose of simplification, we use PIP segmentation method[28] for extracting candlesticks from the input time series.

In Table 2, "*A*" denotes the attributes of a candlestick. Table 3 shows the attributes of a candlestick. In a price time series, a candlestick may contain the opening price ("*o*"), the highest price ("*h*"), the lowest price ("*l*") and the closing price ("*c*"). Volume ("*v*") is the number of shares or contracts traded. "*x*" is the time coordinate of a candlestick. We use "*Vn*" in Table 2 to denote the "volume special

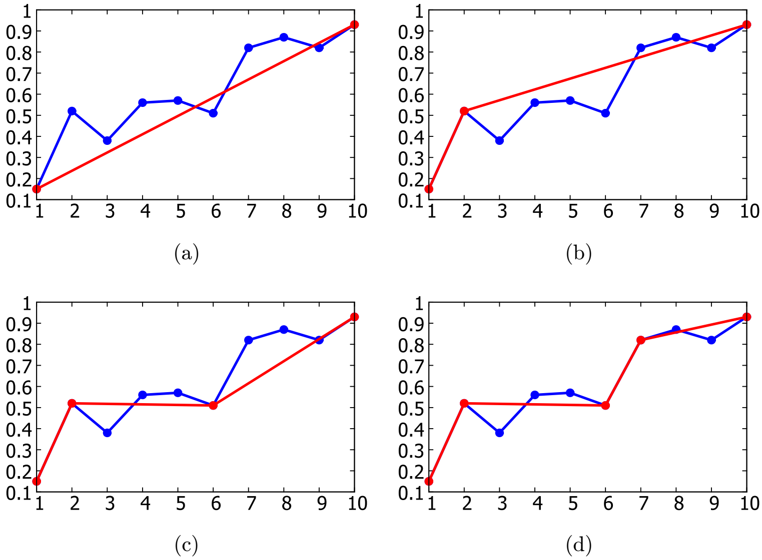


Fig. 12. (Color online) Five PIPs (shown as red points) are selected from a time series formed by closing price (shown as a blue line). The selection order is from (a) to (d). The segmentation result is the red line in (d).

Table 3. Attributes of a candlestick.

Attributes	Description
o	the opening price of a candlestick
c	the closing price of a candlestick
h	the highest price of a candlestick
l	the lowest price of a candlestick
v	the volume of a candlestick
x	time coordinate of a candlestick

points" in the volume time series. Volume special points can be extracted as PIPs from a volume time series. "*TSD*" denotes a data point from the price or volume time series. In Table 2, "difference" (d) denotes the difference between two "*TSD*"s. "slope" denotes the slope of the line connecting two "*TSD*"s. "width" (w) denotes the width between two "*TSD*"s. The functions "difference" and "width" are defined in Eqs. 1 and 2.

$$d(a, b) = \frac{\max(a, b) - \min(a, b)}{\max(a, b)}, \tag{1}$$

$$w(a, b) = \max(a, b) - \min(a, b), \tag{2}$$

where "a" and "b" are "*TSD*"s.

3.3. *EBNF rules for chart patterns*

We define the following rules to describe the shape of a chart pattern. These rules are based on the definitions given in Table 2. The EBNF expressions of these rules are given in Table 4.

(1) *Pattern Name*: The name of a chart pattern name consists of letters and underlines. The first letter should be capital. e.g., *Hill_Valley*.

(2) *Price Trend*: This rule is used to describe the general shape of a price trend. In a "Price Trend", special price points are defined from left to right order. The sequence numbers of the special price points are integers and they are in increasing order.
(Example) Price Trend: $S0$; su; $S1$; sd; $S2$; wn; $S3$
In the above example, special price points ($S0, S1, S2$ and $S3$) are used to connect primitive shapes to form the shape of a price trend. In FCPL, complex chart patterns can be incrementally composed from simple chart patterns. For the purpose of reusability, we can use "Pattern Name" in the price trend. If a "Pattern Name" is written in the "Price Trend", it means that the special price points and primitive shapes from an existing pattern are reused for price trend of the current pattern.
(Example) Price Trend: *Hill*; *Valley*
In the above example, the price trend of the current pattern is formed by the price trends of two exiting patterns called "Hill" and "Valley".

(3) *Price Special Points*: This rule is used to describe the relationships between the special points from "Price Trend".
(Example) Price Special Points: $S1.c < S2.c$; $d(S2.h, S3.h) > d(S0.h, S1.h)$
In the above example, $S1.c$ and $S2.c$ denote the closing prices of price special points $S1$ and $S2$. $S2.h$ and $S3.h$ denote the highest prices of $S2$ and $S3$. $S0.h$ and $S1.h$ denote the highest prices of $S0$ and $S1$.

Table 4. Definitions of five EBNF rules.

Rule name	EBNF expression
1. Pattern Name	{*letters*| "_"}
2. Price Trend	(*Sn* | *Pattern Name*, ";", *shape*|*Pattern Name*, ";", {[*Sn*|*Pattern Name*], *shape*|*Pattern Name*, ";"} |*NULL*, *Sn*|*Pattern Name*, ";")|*NULL*
3. Price Special Points	{((*R*|*TSD*), *operator*, (*R*|*TSD*), ";") | (*R*|*difference*, *operator*, *R*|*difference*, ";") | (*R*|*slope*, *operator*, *R*|*slope*, ";") |*NULL*}
4. Volume Trend	((*Sn*|*Vn*|*Pattern Name*), ";", *shape*|*Pattern Name*, ";", {[*Sn*|*Vn*|*Pattern Name*], *shape*|*Pattern Name*, ";"} |*NULL*, *Sn*|*Vn*|*Pattern Name*, ";")|*NULL*
5. Volume Special Points	{((*R*|*TSD*), *operator*, (*R*|*TSD*), ";") |(*R*|*difference*, *operator*, *R*|*difference*, ";") |(*R*|*slope*, *operator*, *R*|*slope*, ";") |*NULL*}

(4) *Volume Trend*: This rule describes the general shape of a volume trend. The syntax used for the "Volume Trend" is similar to the "Price Trend". The only difference is that in the "Volume Trend", we use the special points from the volume time series instead of the special points from the price time series.
(5) *Volume Special Points*: This rule is used to describe the relationship between the volume special points from "Volume Trend". The syntax for "Volume Special Points" is similar to the "Price Special Points".

4. Examples of Describing Chart Patterns with FCPL

With FCPL, we can describe new chart patterns or existing chart patterns. In Ref. 10, Bulkowski introduces the detailed descriptions of 53 chart patterns. In this paper, we divide these 53 chart patterns into five categories based on their shapes. These categories are numbered from C1 to C5.

(1) $C1$:*Variable fluctuation patterns*: In C1, each pattern has an uncertain number of fluctuations. Therefore, the number of fluctuations cannot be predefined in advance. We select "Wedge Rising" as a representative pattern from this category. There are 20 patterns in this group.
(2) $C2$:*Fixed fluctuation patterns*: Each chart pattern in C2 has an exact number of fluctuations. Therefore, the number of fluctuations of a pattern in this category can be predefined in advance. We select "Head-and-Shoulders Tops" as a representative pattern from this category. There are ten patterns in this group.
(3) $C3$:*Patterns with curved shapes*: C3 is the category of patterns with curved shapes. Fluctuations form when the price changes sharply. In contrast, curves form when the price changes in a gentle way. In this paper, we use "Rounding Bottoms" and "Cup with Handle" patterns as representative patterns from this category. There are 16 patterns in this group.
(4) $C4$:*Patterns with spikes*: C4 is the category of patterns with spikes. Patterns with spikes have two spikes that are remarkably longer than the other spikes around them. In these cases, the highest/lowest prices of two candlesticks are much higher/lower than those of the candlesticks around them. "Pipe Bottoms" is an example. There are four patterns in this group.
(5) $C5$:*Patterns with gaps*: In C5, each pattern has a gap. A gap forms when a candlestick's highest price is lower than the lowest price of the candlesticks around it or when a candlestick's lowest price is higher than the highest price of the candlesticks around it. We select "Gaps" as a representative pattern from this category. There are three patterns in this group.

FCPL can be used to describe any of the 53 chart patterns. To illustrate the capabilities of FCPL, we describe six representative chart patterns for discussion. These six chart patterns are "Wedge Rising", "Head-and-Shoulders Tops", "Rounding Bottoms", "Cup with Handle", "Pipe Bottoms" and "Gaps". In addition, the composition of a chart pattern "Hill_and_Valley" is used as an example to demonstrate the reusability of FCPL.

Financial time series are commonly depicted in candlesticks. Each candlestick is associated with five values: date, opening price, the highest price, the lowest price, and the closing price. In most cases, closing price is used to form the time series. The closing price is an important attribute because it represents the final valuation of the stock made by the market during the day.[2] For "Head-and-Shoulders Tops", "Wedge Rising", " Rounding Bottoms" and "Cup with Handle", we use the closing price of a candlestick to describe the price trend.

4.1. *C*1 — *variable fluctuation patterns:* "*Wedges, Rising*"

A "Wedge, Rising" pattern is depicted in Fig. 13. S_k ($k = 0, 1, 2, 3, 4, 5, 6$) denotes the special points. In this pattern, an upward price spiral is bounded by two intersecting, up-sloping trend lines (two dotted lines in Fig. 13). Both trend lines must have upward slopes and eventually intersect with the bottom trend line producing a steeper slope than that at the top. A "Wedges, Rising" should have at least five touches (three on one side and two on the other). A "Wedge Rising" pattern with seven points and five touches is depicted in Fig. 13 (Top). The volume decreases during the time period of the formation of the "Wedge Rising" (see Fig. 13 (Bottom)). Therefore, the attribute "Volume Trend" describes the downward trend of volume.

- **FCPL definition of "Wedges Rising" pattern**
  Pattern Name: *Wedges_Rising*
  Price Trend: $S0$; *su*; $S1$; *sd*; $S2$; *su*; $S3$; *sd*; $S4$; *su*; $S5$; *sd*; $S6$
  Price Special Points: $S5.c > S3.c$; $S3.c > S1.c$; $S4.c > S2.c$; $w(S6.x, S0.x) > 21$; $slope(S2.c, S4.c) > slope(S1.c, S5.c)$
  Volume Trend: $V0$; *sd*; $V1$
  Volume Special Points: *NULL*

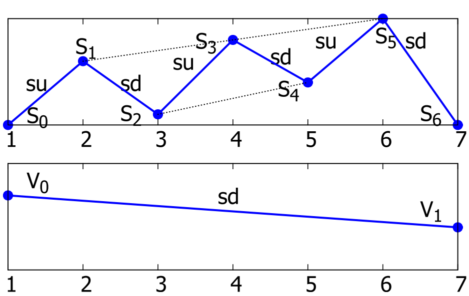


Fig. 13. (Top) A "Wedges, Rising" template. (Bottom) The volume of "Wedges, Rising" patterns.

4.2. *C2 — fixed fluctuation patterns:* "*Head-and-Shoulders Tops*"

Figure 14(Top) illustrates a "Head-and-Shoulders Tops" pattern which is formed by three fluctuations and seven special points. According to the definition, the center special point S_3 should be higher than the two shoulders S_1 and S_5. In addition, the two shoulders should be at a similar price level. All of the special points except the first and last special points are used to connect two different primitive shapes which are defined in Sec. 3.1. In Fig. 14(Bottom), the volume is highest on the left shoulder and lowest on the right shoulder. The head shows a medium volume among the three peaks.

We can easily describe this chart pattern with primitive shapes and special points. The rule "Price Trend" includes primitive shapes ("up" and "down") connected by special points. The rule "Price Special Points" describes the relationship between different special points (closing prices). In "Volume Special Points", we define the relationships of the volumes at three time points (the "left shoulder", the "head" and the "right shoulder"). Since there is no shape for the "Volume Trend", its value is set to "NULL".

- **FCPL definition of "Head-and-Shoulders Tops" pattern**
  Pattern Name: *Head_and_Shoulders_Tops*
  Price Trend: $S0$; su; $S1$; sd; $S2$; su; $S3$; sd; $S4$; su; $S5$; sd; $S6$
  Price Special Points:
  $S3.c > S1.c$; $S3.c > S5.c$; $S6.c \leq S4.c$; $d(w(S3.x, S1.x), w(S3.x, S5.x)) \leq 10\%$; $d(S1.c, S5.c) \leq 10\%$; $d(S2.c, S4.c) \leq 10\%$
  Volume Trend: *NULL*
  Volume Special Points: $S1.v > S3.v$; $S3.v > S5.v$

4.3. *C3 — patterns with curved shapes:* "*Rounding Bottoms*" *and* "*Cup with Handle*"

A "Rounding Bottoms" pattern is a bowl shape. We use primitive shapes "ws" and "se" to form the curved bowl. In Fig. 15(Top), special point S_1 is used to connect two different primitive shapes "ws" and "se". The two rims (special points S_0 and S_2) of

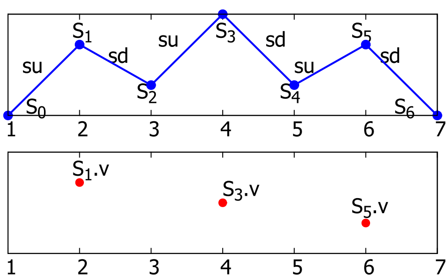


Fig. 14. (Top) A "Head-and-Shoulders Tops" template. (Bottom) The volume of "Head-and-Shoulders Tops" patterns.

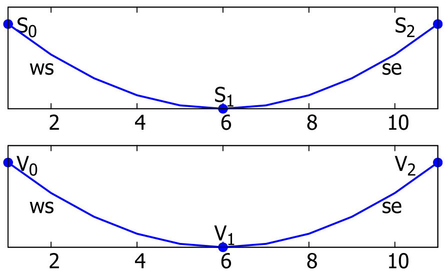


Fig. 15. (Top) A "Rounding Bottoms" template. (Bottom) The volume of "Rounding Bottoms" patterns.

the pattern should be at the same price level. In Fig. 15(Bottom), the shape of the "Volume Trend" is similar to the "Price Trend".

- **FCPL definition of "Rounding Bottoms" pattern**
  Pattern Name: *Rounding_Bottoms*
  Price Trend: $S0$; ws; $S1$; se; $S2$
  Price Special Points: $d(S0.c, S2.c) \leq 10\%$
  Volume Trend: $V0$; ws; $V1$; se; $V2$
  Volume Special Points: *NULL*

"Cup with Handle" is a U-shaped cup with a handle on the right-hand side. Figure 16 shows a "Cup with Handle". The cup lips (S_0 and S_2 are at same price level. The handle (S_3) should form in upper half of the cup. Since there is no common definition about the volume of "Cup with Handle", therefore the rules "Volume Trend" and "Volume Special Points" are set to "NULL".

- **FCPL definition of "Cup with Handle" pattern**
  Pattern Name: *Cup_with_Handle*
  Price Trend: $S0$; ws; $S1$; se; $S2$; sd; $S3$; su; $S4$
  Price Special Points: $d(S0.c, S2.c) \leq 10\%$; $d(S3.c, S2.c) < 0.5 \times d(S1.c, S2.c)$; $w(S2.x, S0.x) \geq 49$; $w(S2.x, S0.x) \leq 455$; $w(S4.x, S2.x) \geq 7$
  Volume Trend: *NULL*
  Volume Special Points: *NULL*

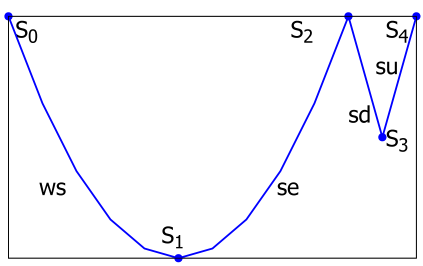


Fig. 16. A "Cup with Handle" template.

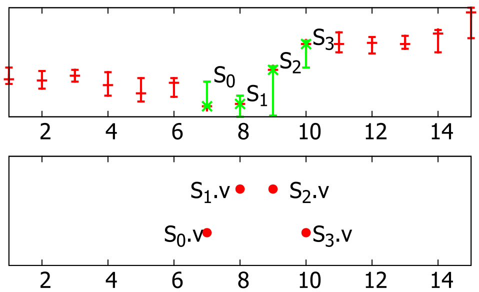


Fig. 17. (Top) A "Pipe Bottoms" template. (Bottom) The volume of "Pipe Bottoms" patterns.

4.4. *C4 — patterns with spikes:* "*Pipe Bottoms*" *pattern*

A "Pipe Bottoms" pattern is formed with four candlesticks. The relative positions of the low prices of the four candlesticks are used to describe the "Pipe Bottoms". In Fig. 17(Top), a "Pipe Bottoms" is formed with four green candlesticks in a time series of red candlesticks. We describe the relative positions of the lowest prices of four candlesticks with the rule "Price Special Points". The rule "Price Trend" is set to "NULL".

- **FCPL definition of "Pipe Bottoms" pattern**
  Pattern Name: *Pipe_Bottoms*
  Price Trend: *NULL*
  Price Special Points: $S0.l > S1.l$; $d(S1.l, S2.l) \leq 10\%$; $S2.l < S3.l$
  Volume Trend: *NULL*
  Volume Special Points: $S0.v < S1.v$; $S0.v < S2.v$; $S3.v < S1.v$; $S3.v < S2.v$

4.5. *C5 — patterns with gaps:* "*Gaps*" *pattern*

A gap appears when yesterday's daily highest price is below today's lowest price or when yesterday's lowest price is above today's highest price. In Fig. 18, a gap appears when the first candlestick's (S_0) lowest price is above the second candlestick's (S_1) highest price in a time series of red candlesticks.

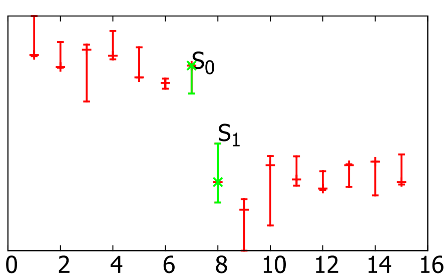


Fig. 18. A "Gaps" template.

- **FCPL definition of "Gaps" pattern**
  Pattern Name: *Gaps*
  Price Trend: *NULL*
  Price Special Points: $S0.l > S1.h$
  Volume Trend: *NULL*
  Volume Special Points: *NULL*

4.6. *An example of composing a complex pattern from simple patterns*

In FCPL, we can compose complex chart patterns by reusing simple patterns. To illustrate the reusability of FCPL specifications, two simple chart patterns "Hill" (see Fig. 19) and "Valley" (see Fig. 20) are defined as follows:

- **FCPL definition of "Hill" pattern**
  Pattern Name: *Hill*
  Price Trend: $S0$; su; $S1$; sd; $S2$
  Price Special Points: $d(S0.c, S1.c) > 0.2$; $d(S1.c, S2.c) \leq 0.3$
  Volume Trend: *NULL*
  Volume Special Points: *NULL*

- **FCPL definition of "Valley" pattern**
  Pattern Name: *Valley*
  Price Trend: $S0$; sd; $S1$; su; $S2$
  Price Special Points: $d(S0.c, S1.c) > 0.2$; $d(S1.c, S2.c) \leq 0.3$

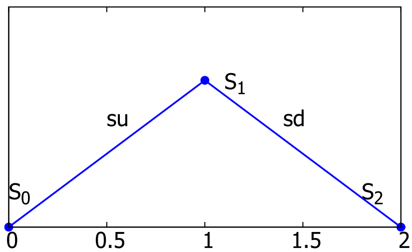


Fig. 19. A "Hill" template.

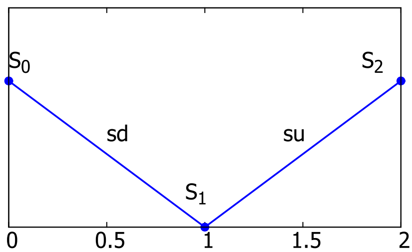


Fig. 20. A "Valley" template.

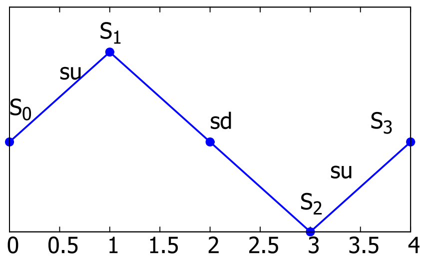


Fig. 21. A "Hill_and_Valley" template.

Volume Trend: *NULL*
Volume Special Points: *NULL*

Next, we construct a new chart pattern "Hill_and_Valley" (see Fig. 21) with the two simple chart patterns "Hill" and "Valley" defined from the previous step.

- **FCPL definition of "Hill_and_Valley" pattern**
  Pattern Name: *Hill_and_Valley*
  Price Trend: *Hill*; *Valley*
  Price Special Points: *NULL*
  Volume Trend: *NULL*
  Volume Special Points: *NULL*

According to the above definition, the new chart pattern's name is "Hill_and_Valley". Its "Price Trend" is defined as "Hill; Valley;" which means that the combination of Hill's price trend followed by Valley's price trend is used for the current price trend. The "Price Special Point" of "Hill_and_Valley" is set to "Null" since the new pattern inherits the "Price Special Points" from its constituent patterns.

5. Experiments on Real Datasets

In this experiment, we classify "Wedges, Rising" (see definitions in Sec. 4.1), "Head-and-Shoulders Tops" (see definitions in Sec. 4.2), "Rounding Bottoms" (see definitions in Sec. 4.3) and "Pipe Bottoms" patterns (see definitions in Sec. 4.4) from the following real datasets with FCPL definitions proposed in this paper. The real datasets used in the experiments were downloaded from Yahoo Finance.[33]

(1) The historical daily prices of the Hang Seng Index (HSI) from 2 January 2003 to 31 December 2012, which contains 2,506 points,
(2) The NYSE AMEX Composite Index (NYSE) from 1 February 2005 to 31 December 2015, which contains 2,769 points, and
(3) The Dow Jones Industrial Average (DJI) from 1 February 2005 to 31 December 2015, which contains 2,769 points.

Algorithm 5.1. Pseudocode of the PIP identification

Function: PIP Identification (T, Q)
Input: sequence T of $Len(T) = m$, template Q of $Len(Q) = n$
Output: pattern SP of $Len(SP) = n$
Set $SP_1 = P_1$, $SP_n = P_m$
repeat
Select point T_j with maximum distance to the adjacent points in SP (SP_1 and SP_n initially)
Add T_j to SP
until all SP are all filled
return SP

For the patterns formed by points such as "Wedges, Rising", "Head-and-Shoulders Tops", and "Rounding Bottoms", we used a sliding window to obtain a subsequence from the historical price data. Next, we selected a number of PIPs from the subsequence. Finally, we examined the resulting segmented subsequence. The window sizes are set to 60 in the experiments. For the patterns formed by candlesticks such as "Pipe Bottoms", we traversed the whole time series to identify the patterns without performing segmentation. Algorithm 5.1 is the pseudocode of the PIP identification[34] used in the experiments.

Tables 5–7 show the number of chart patterns found in HSI, NYSE and DJI, respectively. Figures 22(a)–22(c) show the examples of the chart patterns found in HSI. We did not find any "Rounding Bottoms" in HSI, NYSE and DJI datasets. We also observed that the conditions regarding the trading volume of the "Rounding Bottoms" pattern was found to be too strict for positively identifying any patterns from the data sets. To clarify this issue, an additional experiment was performed to

Table 5. The number of chart patterns found in HSI.

Name	The number of chart patterns
Wedges, Rising	33
Head-and-Shoulders Tops	1
Rounding Bottoms	0
Pipe Bottoms	114

Table 6. The number of chart patterns found in NYSE.

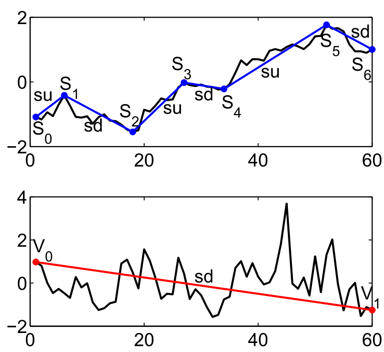
Name	The number of chart patterns
Wedges, Rising	19
Head-and-Shoulders Tops	0
Rounding Bottoms	0
Pipe Bottoms	74

Table 7. The number of chart patterns found in DJI.

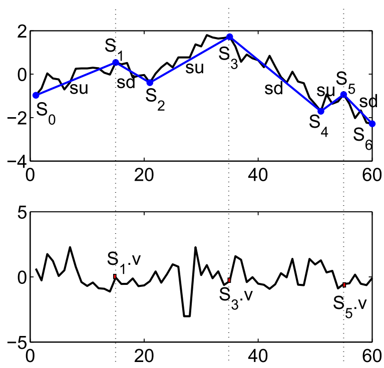
Name	The number of chart patterns
Wedges, Rising	18
Head-and-Shoulders Tops	0
Rounding Bottoms	0
Pipe Bottoms	175

Table 8. The start dates and end dates of the chart patterns from Figs. 22 and 23.

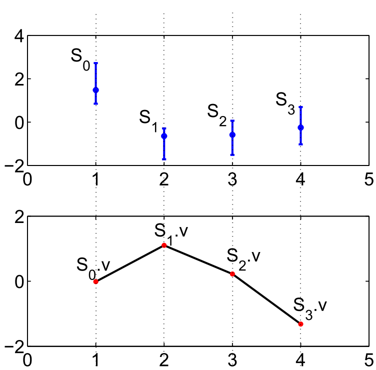
Stock	Chart pattern	Start date	End date
HSI	"Wedges, Rising": Fig. 22(a)	March-31-2003	June-27-2003
HSI	"Head-and-Shoulders Tops": Fig. 22(b)	February-25-2010	May-20-2010
HSI	"Pipe Bottoms": Fig. 22(c)	October-5-2005	October-10-2005
Amazon	"Rounding Bottoms": Fig. 23	April-24-2014	July-18-2014



(a) A "Wedges, Rising" (Top) and its volume (Bottom)



(b) A "Head-and-Shoulders Tops" (Top) and its volume (Bottom)



(c) A "Pipe Bottoms"(Top) and its volume (Bottom)

Fig. 22. The chart patterns found in HSI.

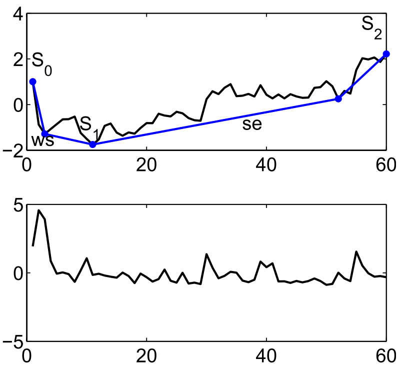


Fig. 23. A "Rounding Bottoms" found in Amazon (top) and its volume (bottom).

classify "Rounding Bottoms" with a real dataset of Amazon (AMZN) from 3 January 2005 to 31 December 2015, which contains 2,769 points. In this experiment, we relax the restrictions about the volume and as a result, we found one "Rounding Bottoms" in AMZN with window size of 60 (see Fig. 23). To better illustrate the chart patterns found in the experiments, all time series in Figs. 22 and 23 are normalized. Table 8 shows the exact start dates and end dates of the chart patterns from Figs. 22 and 23.

6. Conclusions

In this paper, we propose a novel domain-specific language called "FCPL" for defining technical patterns based on five rules. The syntax of the proposed language is formally described in EBNF. Five rules expressed in EBNF are used to describe the characteristics of a chart pattern. The first rule is used to describe the name of a chart pattern. The second rule is used to describe the shape of a chart pattern. The third rule is used to describe the relationship between different special points that are used to connect the primitive shapes of a chart pattern. The other two rules are used to describe the shape of volume for a chart pattern. The main objective of FCPL is to become a simple, comprehensive, expressive, and reusable domain-specific language for defining chart patterns in financial trading.

FCPL is a comprehensive language for defining chart patterns since it can be used to describe all 53 chart patterns in Ref. 10. The comprehensiveness of FCPL is demonstrated in the examples for representative patterns from five categories. In FCPL, primitive shapes are defined as building blocks that are used to construct the whole shape of a chart pattern. Patterns defined in FCPL are reusable since we can reuse the defined chart patterns to construct more complex chart patterns. The syntax of FCPL is intentionally kept simple. Therefore, stock experts or novice users without programming expertise can use the language to define existing or new patterns. Although, FCPL is simple, it is powerful enough to construct complex patterns. This expressive power is demonstrated in the example about incremental composition of a pattern called "Hill_and_Valley".

The FCPL definitions of chart patterns can be translated into a programming language. With FCPL, the specification of a chart pattern is separated from the mechanism of its implementation. To evaluate the usefulness of FCPL, we classified chart patterns from Hang Seng Index (HSI), NYSE AMEX Composite Index (NYSE), and Dow Jones Industrial Average (DJI). In addition, to aid the users in using FCPL for specifying chart patterns, we have developed an online application called FCPL Generator. In this application, user can simply use an Internet browser to draw a chart pattern and specify the relevant constraints. FCPL Generator developed for this paper will be released as a freeware under the LGPL License. Latest version of FCPL Generator is available on http://www.cis.umac.mo/∼fstasp/FCPLGenerator.html. As for the future work, we are planning to design a rule-based expert system based on FCPL. In this system, users can input their FCPL definitions of the chart patterns for searching from a given time series.

Acknowledgement

This research was funded by the Research Committee of the University of Macau, Grants MYRG2016-00148-FST and MYRG2017-00029-FST.

References

1. A. Bagheri, H. M. Peyhani and M. Akbari, Financial forecasting using anfis networks with quantum-behaved particle swarm optimization, *Expert Systems with Applications* **41**(14) (2014) 6235–6250.
2. R. D. Edwards, J. Magee and W. C. Bassetti, *Technical Analysis of Stock Trends* (CRC Press, 2007).
3. E. F. Fama, Efficient capital markets: A review of theory and empirical work, *The Journal of Finance* **25**(2) (1970) 383–417.
4. W. F. Bondt and R. Thaler, Does the stock market overreact? *The Journal of Finance* **40**(3) (1985) 793–805.
5. N. Jegadeesh and S. Titman, Returns to buying winners and selling losers: Implications for stock market efficiency, *The Journal of Finance* **48**(1) (1993) 65–91.
6. M. J. S. de Souza, D. G. F. Ramos, M. G. Pena, V. A. Sobreiro and H. Kimura, Examination of the profitability of technical analysis based on moving average strategies in BRICS, *Financial Innovation* **4**(1) (2018) 3.
7. C. G. da Silva, Modeling intraday information in financial markets with the scatter search metaheuristic, *International Journal of Financial Engineering* **2**(2) (2015) 1–22.
8. W. Leigh, N. Modani, R. Purvis and T. Roberts, Stock market trading rule discovery using technical charting heuristics, *Expert Systems with Applications* **23**(2) (2002) 155–159.
9. R. Schabacker, *Technical Analysis and Stock Market Profits* (Harriman House Limited, 2005).
10. T. N. Bulkowski, *Encyclopedia of Chart Patterns*, 2nd edn. (John Wiley & Sons, 2011).
11. J.-L. Wang and S.-H. Chan, Trading rule discovery in the us stock market: An empirical study, *Expert Systems with Applications* **36**(3) (2009) 5450–5455.

12. J.-L. Wang and S.-H. Chan, Stock market trading rule discovery using pattern recognition and technical analysis, *Expert Systems with Applications* **33**(2) (2007) 304–315.
13. R. Cervelló-Royo, F. Guijarro and K. Michniuk, Stock market trading rule based on pattern recognition and technical analysis: Forecasting the djia index with intraday data, *Expert Systems with Applications* **42**(14) (2015) 5963–5975.
14. R. S. Scowen, Extended BNF-a generic base standard, Technical report, ISO/IEC 14977, URL http://www.cl.cam.ac.uk/mgk25/iso-14977.pdf (1998).
15. A. Fall and J. Fall, A domain-specific language for models of landscape dynamics, *Ecological Modelling* **141**(1) (2001) 1–18.
16. A. Van Deursen, P. Klint and J. Visser, Domain-specific languages: An annotated bibliography, *ACM SIGPLAN Notice* **35**(6) (2000) 26–36.
17. A. Van Deursen and P. Klint, Domain-specific language design requires feature descriptions, *Journal of Computing and Information Technology* **10**(1) (2002) 1–17.
18. I. Córdoba-Sánchez and J. de Lara, Ann: A domain-specific language for the effective design and validation of java annotations, *Computer Languages, Systems & Structures* **45** (2016) 164–190.
19. E. A. Marand, E. A. Marand and M. Challenger, Dsml4cp: A domain-specific modeling language for concurrent programming, *Computer Languages, Systems & Structures* **44** (2015) 319–341.
20. A. Popovic, I. Lukovic, V. Dimitrieski and V. Djukic, A dsl for modeling application-specific functionalities of business applications, *Computer Languages, Systems & Structures* **43** (2015) 69–95.
21. S. Anand, W.-N. Chin and S.-C. Khoo, Charting patterns on price history, in *Proceedings of the Sixth ACM SIGPLAN International Conference on Functional Programming (ICFP '01)* (ACM, New York, NY, USA, 2001), pp. 134–145.
22. R. Agrawal, G. Psaila, E. L. Wimmers and M. Zaït. Querying shapes of histories, in *Proceedings of the 21th International Conference on Very Large Data Bases (VLDB '95)*, (Eds.) U. Dayal, P. M. D. Gray and S. Nishio (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995), pp. 502–514.
23. W. Wu and G. Kou, A group consensus model for evaluating real estate investment alternatives, *Financial Innovation* **2**(1) (2016) 8.
24. G. Kou, D. Ergu, C. Lin and Y. Chen, Pairwise comparison matrix in multiple criteria decision making, *Technological and Economic Development of Economy* **22**(5) (2016) 738–765.
25. N. Wirth, What can we do about the unnecessary diversity of notation for syntactic definitions? *Communications of the ACM* **20**(11) (1977) 822–823.
26. E. S. S. Standard, EBNF: ISO/IEC 14977: 1996 (e), URL http://www.cl.cam.ac.uk/mgk25/iso-14977.pdf70.
27. F.-L. Chung, T.-C. Fu, R. Luk and V. Ng, Flexible time series pattern matching based on perceptually important points, *Int. Joint Conf. Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*, Seattle, Washington, USA, 2001, pp. 1–7.
28. E. J. Keogh and M. J. Pazzani, A simple dimensionality reduction technique for fast similarity search in large time series databases, in *Knowledge Discovery and Data Mining: Current Issues and New Applications* (Springer, 2000), pp. 122–133.
29. E. Keogh, S. Chu, D. Hart and M. Pazzani, An online algorithm for segmenting time series, in *Proc. IEEE Int. Conf. Data Mining (ICDM)* (IEEE, 2001), pp. 289–296.
30. Y.-W. Si and J. Yin, OBST-based segmentation approach to financial time series, *Engineering Applications of Artificial Intelligence* **26**(10) (2013) 2581–2596.

31. Y. Huang and G. Kou, A kernel entropy manifold learning approach for financial data analysis, *Decision Support Systems* **64** (2014) 31–42.
32. Y. Huang, G. Kou and Y. Peng, Nonlinear manifold learning for early warnings in financial markets, *European Journal of Operational Research* **258**(2) (2017) 692–702.
33. Yahoo, Yahoo Finance (2017), URL https://ca.finance.yahoo.com/q/hp?s=YHOO.
34. T.-C. Fu, F.-L. Chung, R. Luk and C.-M. Ng, Stock time series pattern matching: Template-based vs. rule-based approaches, *Engineering Applications of Artificial Intelligence* **20**(3) (2007) 347–364.