

2012

TradeScript™

Скриптовый язык для создания механических торговых систем в терминале SmartX™

Описание языка и примеры построения торговых систем



Оглавление

Глава 1. Введение в TradeScript™	7
Предпосылки	7
Как организовано руководство	7
Язык программирования TradeScript™	7
Введение: Основные концепции	7
Булева логика	8
Структура программы	8
Функции	9
Векторное программирование	9
Функция REF	10
Функция TREND	10
Ценовые разрывы и волатильность	11
Технический анализ	12
Crossovers	12
Скрипт «Ключевая разворотная точка»	13
Глава 2. Функции и операторы	14
Примитивы	14
Условная функция “IF”	14
Функция LOOP и симуляция управляющей структуры	14
COUNTIF	16
LASTIF	16
SUMIF	16
SUM	16
AVG	16
MAX	17
MIN	17
MAXOF	17
MINOF	17
REF	17
TREND	18
CROSSOVER	18
Математические функции	18

ABS	19
SIN	19
COS	19
TAN	19
ATN	19
EXP	19
LOG	19
LOG10	20
RND	20
Операторы	20
Равно (=)	20
Больше чем (>)	20
Меньше чем (<)	20
Больше чем, или равно (>=)	20
Меньше чем, или равно (<=)	21
Не Равно (<> или !=)	21
Логическое И (AND)	21
Логическое ИЛИ (OR)	21
Логическое исключение (XOR)	22
Логическое отрицание (NOT)	22
Логическое сравнение (EQV)	22
MOD	22
Глава 3. Функции технического анализа	23
Скользящие средние	23
Простая скользящая средняя	23
Экспоненциальная скользящая средняя	24
Скользящая средняя «Time Series»	24
Переменная скользящая средняя	24
Триангулярная скользящая средняя	25
Взвешенная скользящая средняя	25
Сглаживание по Уэллсу Уайлдеру (Скользящая средняя)	25
Динамическая средняя индекса волатильности [Volatility Index Dynamic Average - VIDYA (Скользящая средняя)]	26
Функции линейной регрессии	26

R ² (R-квадрат).....	26
Slope (Наклон).....	27
Forecast (Прогноз).....	27
Intercept (Точка пересечения).....	27
Функции диапазонов (конверты).....	27
Границы Боллинджера (Bollinger Bands).....	27
Конверты (Moving Average Envelope).....	28
Границы простых чисел (Prime Number Bands).....	28
Осцилляторы	29
Осциллятор момента (Momentum Oscillator)	29
Осциллятор момента Чанда (Chande Momentum Oscillator).....	29
Осциллятор Объема (Volume Oscillator)	30
Осциллятор Цены (Price Oscillator).....	30
Детрендовый ценовой осциллятор (Detrended Price Oscillator)	30
Осциллятор простых чисел (Prime Number Oscillator)	31
Осциллятор фрактального хаоса (Fractal Chaos Oscillator)	31
Радужный осциллятор (Rainbow Oscillator)	32
TRIX	32
Вертикально горизонтальный фильтр (Vertical Horizontal Filter)	32
Свобода движения (Ease Of Movement)	33
Система направленного движения Уайлдера (Wilder's Directional Movement System)	33
Истинный диапазон (True Range).....	34
Williams %R	34
Накопление/Распределение Вильямса (Williams' Accumulation/Distribution).....	34
Волатильность Чайкина (Chaikin's Volatility)	34
Схождение/расхождение скользящих средних (Moving Average Convergence/Divergence -MACD)	35
Диапазон цены (High Minus Low)	36
Стохастический осциллятор (Stochastic Oscillator)	36
Индексы.....	36
Индекс относительной силы (Relative Strength Index).....	36
Индекс массы (Mass Index).....	37
Индекс исторической волатильности (Historical Volatility Index)	37
Индекс денежного потока (Money Flow Index).....	38

Индекс денежного потока Чайкина (Chaikin's Money Flow Index).....	38
Сравнительный индекс относительной силы (Comparative Relative Strength Index)	38
Объемно-ценовой тренд (Price Volume Trend)	39
Индекс положительного объема (Positive Volume Index).....	39
Индекс отрицательного объема (Negative Volume Index)	39
Баланс объемов (On Balance Volume)	40
Индекс продуктивности (Performance Index).....	40
Индекс торгового объема (Trade Volume Index)	40
Свинг-индекс (Swing Index).....	41
Аккумуляционный свинг-индекс (Accumulative Swing Index).....	41
Индекс товарного канала [Commodity Channel Index (CCI)].....	42
Параболическая система [Parabolic Stop and Reversal (Parabolic SAR)]	42
Стохастический индекс момента (Stochastic Momentum Index).....	42
Общие «индикаторные» функции	43
Медиана цены (Median Price)	43
Типичная цена (Typical Price).....	43
Взвешенная цена закрытия (Weighted Close)	44
Изменение цены (Price Rate of Change)	44
Изменение объема (Volume Rate of Change)	44
Значение максимального пика (Highest High Value).....	45
Значение минимального донышка (Lowest Low Value).....	45
Стандартное отклонение (Standard Deviations)	45
Корреляционный анализ (Correlation Analysis).....	46
Фигуры японских свечей	46
Что такое свеча?	46
Графическое представление	46
Идентификация подсвечниковых паттернов.....	47
Глава 4. Примеры торговых систем	48
Торговые системы	48
Система на основе пересечения скользящих средних (Moving Average Crossover System)	48
Система «Price Gap System»	49
Price Gap Script.....	50
Система «Bollinger Bands System».....	50
Bollinger Bands Script.....	50

Историческая волатильность и тренд.....	51
Historical Volatility and Trend Script.....	51
Система Parabolic SAR/MA System.....	52
Parabolic SAR / MA Script.....	52
Система MACD Momentum System	52
MACD Momentum Script	53
Система Narrow Trading Range Breakout	53
Narrow Trading Range Script.....	54
Система Outside Day System.....	54
Outside Day Script.....	54
Система бычьего и медвежьего поглощения (Japanese Candlestick Engulfing Line System)	55
Глава 5. Прimitives переменные, константы и флаги.....	56
Прimitives типы	56
Вектора цен	56
Основные константы	56
Флаги исторического тестирования	56
Константы скользящих средних	57
Константы функции Trend	57
Константы Points и Percents (используются в индикаторах)	57
Константы свечных фигур.....	57
Глава 6. Решение проблем	61

Глава 1. Введение в TradeScript™

Предпосылки

Для понимания материала данного руководства достаточно только знания основ технического анализа.

TradeScript™ является зарегистрированным товарным знаком, принадлежащем компании Modulus Financial Engineering, Inc., и лицензирован OEM лицензией для ОАО «ИК Ай Ти Инвест» - поставщика торгового терминала SmartX™.

Как организовано руководство

Первая часть руководства содержит небольшие примеры, демонстрирующие выполнение общих задач: идентификация инструментов в специфическом диапазоне цен, повышение волатильности, пересечение индикаторов и т.п. Вы можете скопировать эти примеры в собственные торговые системы.

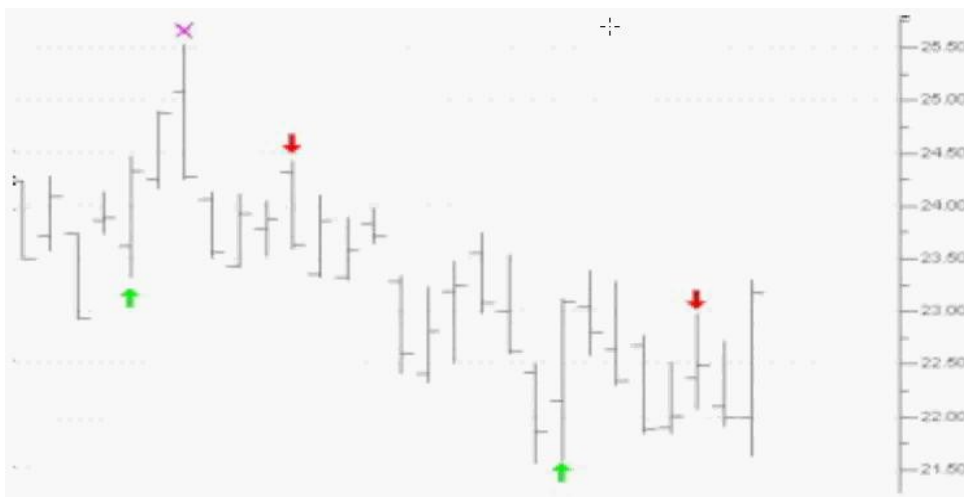
Последняя часть содержит описание функций, свойств и констант, которые поддерживает TradeScript™, а также примеры реальных МТС (механическая торговая система).

Такой подход позволяет начинающим программистам увидеть результаты работы сразу во время обучения.

Язык программирования TradeScript™

TradeScript™ - это «движок», который позволяет использовать скриптовый язык в биржевых торговых приложениях. Это непроцедурный векторный язык программирования, который был создан специально для разработки МТС.

Скрипт – это набор инструкций, который выдается «движку» (TradeScript™), чтобы он выполнил некое полезное действие. Например, активировал *алерт*, если цена акции достигла нового максимума, пересекла вверх скользящую среднюю, или упала на определенный процент. Существует множество вариантов.



Введение: Основные концепции

TradeScript™ - это мощный и гибкий язык программирования для трейдеров. Он предлагает средства для поэтапной разработки изоциренных торговых программ и при этом не требует значительных усилий или существенного опыта программирования.

Ниже показан пример очень простого скрипта, который идентифицирует рынки, которые торгуются на уровнях выше цены открытия:

LAST > OPEN

Без всяких слов понятно, что цель этого скрипта – идентифицировать сигнал, когда цена последней сделки выше цены открытия... это просто как дважды два.

Точно также как разговорный язык позволяет многими способами выразить наши идеи, также и язык программирования TradeScript™ предлагает широкий набор способов программирования торговой системы. Скрипты могут быть очень простыми, типа того что был показан выше, или очень сложными, содержащими сотни строк инструкций. Впрочем, большинство систем обычно содержат несколько строк кода.

Примеры, приведенные в первой части руководства, - относительно короткие и простые. Вместе с тем, они предлагают основу для разработки более сложных скриптов.

Булева логика

Скрипты можно связывать вместе при помощи логических операторов **AND** или **OR**, например...

Скрипт 1 возвратит значение Истина, если цена последней сделки выше цены открытия:

LAST > OPEN

Скрипт 2 возвратит значение Истина, если объем текущего дня (бара) в два раза больше объема предыдущего дня (бара):

VOLUME > REF(VOLUME, 1) * 2

Вы можете объединить эти скрипты. Теперь Истина будет возвращена, если будут соблюдены два условия: текущая цена выше цены открытия и (**AND**) текущий объем в два раза выше объема предыдущего бара:

LAST > OPEN AND VOLUME > REF(VOLUME, 1) * 2

Мы также можем поменять оператор **AND** на **OR**. В этом случае выражение возвратит Истину, если или цена последней сделки выше цены открытия, или текущий объем в два раза выше предыдущего:

LAST > OPEN OR VOLUME > REF(VOLUME, 1) * 2

Как Вы видите, инструкции очень похожи на инструкции обычного языка. Операторы булевой логики **AND** и **OR** являются очень важной концепцией, которая очень часто применяется при программировании на TradeScript™.

Структура программы

Вы можете расположить Ваш код на нескольких строках, зачастую это улучшает его восприятие. Код в следующем примере аналогичен предыдущему скрипту, однако, более понятен при чтении:

LAST > OPEN OR

VOLUME > REF(VOLUME, 1) * 2

Профессиональный подход к структуре скрипта предполагает, чтобы он оставался интуитивно понятными при повторном обращении к его коду даже через длительное время. В некоторых случаях может помочь *комментирование* сложных скриптов. *Комментарий* – поясняющая ремарка в теле скрипта.

Комментарий формируется при помощи знака решётка (**#**), размещенного в начала строки. Интерпретатор скрипта будет игнорировать все слова строки, стоящие после этого знака. Этот текст выполняет только сервисную функцию, чтобы сделать скрипт более понятным:

**# Возвращает Истина, если цена последней сделки
выше цены открытия, или текущий объем
в 2 раза больше предыдущего объема:**

LAST > OPEN OR
VOLUME > REF(VOLUME, 1) * 2

Функции

TradeScript™ имеет множество встроенных функций, облегчающих программирование. Если функция встроена в ядро языка программирования, ее называют *примитивом*. Например, примитивом является функция TREND:

TREND(CLOSE, 30) = UP

В данном примере скрипт вернет *Истина*, если имеет место восходящий тренд, рассчитанный за последние 30 дней по ценам закрытия.

Значения, которые заключены в скобки функции, называются *аргументами*. В функции TREND мы имеем два аргумента. Первый аргумент – источник цен для расчета (в приведенном примере – это цена закрытия периода LAST),

Второй аргумент – это число периодов по которым производится расчет знака коэффициента регрессии, с которым и ассоциируется тренд. Число периодов может находиться в диапазоне от 2 до 30.

В случае неправильного использования функции возможны две ситуации. TradeScript™ может автоматически исправить ошибку и скрипт будет запущен, в противном случае, будет выдано сообщение об ошибке. Вы можете исправить ошибку и попытаться запустить скрипт снова.

Другими словами, ошибки пользователя никогда не станут причиной того, что TradeScript™ прервет работу или вернет неправильный результат без предварительного предупреждения о возможных проблемах.

Давайте уберем аргумент CLOSE из функции TREND:

TREND(30) = UP

В данном случае появится следующее сообщение:

Error: argument of 'TREND' function not optional. [Click here for help.](#)

Вам предлагается исправить ошибку и запустить скрипт снова. Кроме того предлагается ссылка на помощь по данной функции, где приводится список необходимых аргументов.

Векторное программирование

Языки векторного программирования ориентированы на операции с векторами, чтобы их было легко применять к векторам, матрицам и многомерным массивам.

Фундаментальная идея векторного программирования заключается в том, что каждая операция применяется сразу ко всему набору значений (вектору или полю). Это позволяет мыслить и оперировать категориями агрегатов данных, без необходимости использовать циклы или индивидуальные скалярные операции.

Например, чтобы рассчитать простую скользящую среднюю «серединной» цены акций за последние 30 периодов при помощи обычного языка программирования типа BASIC, нужно написать что-то типа:

```
For each symbol
  For bar = 30 to max
    Average = 0
    For n = bar - 30 to bar
      median = (CLOSE + OPEN) / 2
      Average = Average + median
    Next
    MedianAverages(bar) = Average / 30
  Next bar
```

Next symbol

Как Вы видите, чтобы создать вектор “MedianAverages”, требуется 9-10 строк кода. При помощи TradeScript™ та же задача решается при помощи одной строки:

SET MedianAverage = SimpleMovingAverage((CLOSE + OPEN) / 2, 30)

В результате этой операции мы создали реальный новый вектор «MedianAverage», который в каждой точке содержит значение простого 30-периодного «мувинга» срединной цены за период.

Функция REF

Давайте более подробно рассмотрим функции “REF” и “TREND”. Это два очень полезных примитива, встроенных в TradeScript™.

Функция REF используется для ссылки на значение любой точки вектора. Предположим, что вектор MedianAverage содержит усредненные медианные цены акции. Для того чтобы получить доступ к определенному элементу вектора при помощи обычного языка программирования, нужно использовать следующий синтаксис:

SET A = MedianAverage[n]

На TradeScript™ та же операция будет выглядеть следующим образом:

SET A = REF(MedianAverage, n)

Главное различие между этими ссылками в том, что в обычных языках отсчет индекса ссылки начинается с начала вектора (массива), т.е. первый элемент будет иметь индекс 1 или 0. В TradeScript™ отсчет начинается с конца вектора и индекс 0 будет иметь последний элемент вектора. При разработке MTC такой подход более удобен, т.к. все отсчеты идут от последнего значения, которое является наиболее важным. Чтобы получить самое последнее значение вектора MedianAverage нужно написать:

SET A = REF(MedianAverage, 0)

Для ссылки на последнее значение вектора можно обойтись без функции REF. Такой ссылкой является имя самого вектора и это более предпочтительный способ. Поэтому, чтобы получить значение последней точки вектора лучше писать:

SET A = MedianAverage

Чтобы получить значение предыдущего бара, нужно написать:

SET A = REF(MedianAverage, 1)

А если нужно значение на два бара в прошлое, то пишем:

SET A = REF(MedianAverage, 2)

Функция TREND

Трейдерам постоянно оценивают состояние рынка на наличие тренда. Тренд это ситуация, когда цена акции повышается (восходящий тренд) или снижается (нисходящий тренд) в течение нескольких дней, недель, месяцев или лет. Типичный инвестор или трейдер не станет открывать длинную позицию по акции, которая в течение многих месяцев находится в нисходящем тренде.

TradeScript™ имеет специальную функцию (примитив), которая оценивает наличие тренда в ценах акции, объеме или значениях индикатора. Это функция – TREND:

TREND(CLOSE, N) = UP

Этот скрипт идентифицирует наличие восходящего тренда в ценах закрытия за последние N периодов ($2 \leq N \leq 30$). Аналогичным способом при помощи этой функции можно определять наличие тренда в значениях объема или технических индикаторов:

Объем, по 10-ти последним периодам
находится в нисходящем тренде:

TREND(VOLUME, 10) = DOWN

Индикатор СМО с периодом 14 находится indicator
в восходящем тренде по 20-ти периодам:

TREND(CMO(CLOSE, 14), 20) = UP

Функция TREND может оказаться полезной для подтверждения торговых сигналов. Предположим мы имеем МТС, которая покупает, если цена закрытия пересекает вверх собственную 20-дневную простую скользящую среднюю. Скрипт этой МТС может выглядеть так:

Дает сигнал «Buy», если цена закрытия пересекла снизу-вверх 20-дневную SMA

CROSSOVER(CLOSE, SimpleMovingAverage(CLOSE, 20)) = TRUE

Было бы полезно ограничить этот скрипт акциями, которые в это время находились в нисходящем тренде. С этой целью мы можем добавить в скрипт следующую строку:

AND TREND(CLOSE, 40) = DOWN

Функция TREND говорит нам о наличии восходящего, нисходящего или бокового тренда, однако, ничего не говорит о выраженности (силе) этого тренда. Для этих целей можно воспользоваться функцией REF. Эта функция поможет определить выраженность изменения значений, произошедших во время тренда. Чтобы определить абсолютное изменение цены за последние 40 баров, нужно написать следующую строку:

SET A = LAST - REF(CLOSE, 40)

Ценовые разрывы и волатильность

Хотя функция TREND позволяет нам идентифицировать тренд, а функция REF может быть использована для определения силы тренда, часто очень полезно идентифицировать ранние признаки изменения тренда. Такие признаки могут быть идентифицированы с помощью ценовых разрывов (гэпов) и экстремальных изменений объема. Мы можем попытаться сделать это следующим образом:

Возвращает Истина, если был ценовой разрыв вверх

LOW > REF(HIGH, 1)

Или:

Возвращает Истина, если был ценовой разрыв вниз

HIGH < REF(LOW, 1)

Вы также можете установить минимальный процент ценового разрыва:

Возвращает Истина, если цена имела гэп не менее 1%

LOW > REF(HIGH, 1) * 1.01

Мы также можем установить фильтр в виде существенного изменения объема:

Рост объема 1000%

VOLUME > REF(VOLUME, 1) * 10

То же самое, но с усредненными значениями:

Текущий объем на 1000% выше своего среднего значения

VOLUME > SimpleMovingAverage(VOLUME, 30) * 10



Кроме того Вы можете измерить волатильность цен или объема при помощи одного из встроенных индикаторов: Volume Oscillator, Chaikin Volatility Index, Coefficient of Determination, Price Rate of Change, Historical Volatility Index и т.д. Эти индикаторы описываются в третьей главе.

Технический анализ

TradeScript™ имеет множество встроенных функций технического анализа. Только одна строка кода позволяет применять такие функции как скользящие средние, границы Боллинджера, японские свечи и другие. Полный список «технических» функций будет приведен в третьей главе.

Следующий скрипт показывает пример использования наиболее популярной функции – простой скользящей средней:

LAST > SimpleMovingAverage(CLOSE, 20)

Скрип возвратит *Истина*, если цена последней сделки будет выше 20-дневной скользящей средней цен закрытия. Переменная CLOSE является вектором цен закрытия, а не просто самой последней ценой закрытия. Вы можете использовать векторы OPEN, HIGH, LOW, CLOSE и VOLUME, а также создавать собственные векторы при помощи ключевого слова SET:

SET Median = (CLOSE + OPEN) / 2

Этот код создает вектор, содержащий среднюю цену каждого торгового дня. Теперь вектор Median можно использовать с любой функцией, которая требует аргумента в виде вектора:

LAST > SimpleMovingAverage(Median, 20)

Этот скрипт вернет *Истина*, если цена последней сделки больше 20-дневной скользящей средней среднедневной цены (**Median**).

Функции возвращают векторы. Поэтому функции можно вкладывать в другие функции. Другими словами, функции можно использовать в качестве аргументов других функций:

LAST >

SimpleMovingAverage(SimpleMovingAverage(CLOSE, 30), 20)

Истина будет возвращена, если цена последней сделки больше 20-дневной скользящей средней, взятой от 30-дневной скользящей средней цен закрытия.

Crossovers

Вы очевидно знакомы с термином “crossover”. Этот термин обозначает «пересечение» двух кривых.

Многие технические индикаторы, например, MACD, имеют «сигнальную линию». Сигналы на покупку или продажу генерируются, когда сигнальная линия пересекает снизу-вверх или сверху вниз линию индикатора.

Функция CROSSOVER помогает нам узнать, когда одна кривая пересекла снизу-вверх другую кривую.

Например, мы можем точно определить точку во времени, когда одна скользящая средняя пересекла другую скользящую среднюю:

SET MA1 = SimpleMovingAverage(CLOSE, 28)
SET MA2 = SimpleMovingAverage(CLOSE, 14)
CROSSOVER(MA1, MA2) = TRUE

Расположенный выше скрипт вернет Истина, если вектор MA1 пересечет снизу-вверх вектор MA2. Мы можем поменять местами аргументы, и тогда скрипт вернет Истина, если вектор MA1 пересечет MA2 сверху вниз:

CROSSOVER(MA2, MA1) = TRUE

Скрипт «Ключевая разворотная точка»

В заключении этой главы, прежде чем приступить к технической части этого руководства, давайте создадим скрипт, который идентифицирует «ключевые разворотные точки» (Key Reversals). Это делается, чтобы Вы могли из первых рук получить представление, как можно использовать TradeScript™ для создания МТС, основанных на сложных правилах.

Определение «Key Reversal» - текущий тренд - восходящий, цена открытия выше предыдущего закрытия, последний бар должен сделать новый пик, а последняя цена должна быть ниже предыдущей минимальной цены:

```
# сначала убеждаемся, что имеет место восходящий тренд
TREND(CLOSE, 30) = UP

# цена открытия должны быть выше вчерашнего закрытия
AND OPEN > REF(CLOSE, 1)

# сегодня должен быть сделан новый пик
AND HIGH >= ALL_TIME_HIGH

# и цена последней сделки должна быть ниже вчерашнего минимума
AND LAST < REF(LOW, 1)
```

Ради шутки можно отметить, что сам скрипт без комментариев намного короче, чем описание торговой системы. Ключевые разворотные точки не частое явление, однако, этот сигнал очень надежен. Вы можете поэкспериментировать, удалив строку **AND HIGH >= ALL_TIME_HIGH**, или заменив ее другим критерием.

Следующий скрипт реализует ту же стратегию, но в обратном направлении:

```
# сначала убеждаемся, что имеет место нисходящий тренд
TREND(CLOSE, 30) = DOWN

# цена открытия должна быть ниже вчерашнего закрытия
AND OPEN < REF(CLOSE, 1)

# сегодня должно быть сделано новое донышко
AND LOW <= ALL_TIME_LOW

# и цена последней сделки должна быть выше вчерашнего максимума
AND LAST > REF(HIGH, 1)
```

Еще раз повторим, что сигналы этой системы возникают редко, однако, очень надежны.

Глава 2. Функции и операторы

Примитивы

В этой главе описываются встроенные функции TradeScript™, также известные как *примитивы*. Эти важные функции определены в языке программирования TradeScript™ и обеспечивают основу, необходимую для построения сложных торговых систем с нуля.

Без преувеличения можно сказать, что при помощи TradeScript™ можно разработать МТС любого типа с минимальными усилиями. Если МТС может быть выражена математическими формулами или может быть запрограммирована в каком-либо процедурном языке программирования, например, C++, VB или Java, Вы можете быть уверены, что эта МТС также может быть запрограммирована при помощи TradeScript™.

Некоторые формулы, используемые в техническом анализе, могут быть очень сложными. Например, некоторые формулы требуют рекурсивных вычислений и усложнены IF-THEN-ELSE структурами. Эти сложные торговые системы обычно разрабатываются на низкоуровневых языках программирования.

В этой главе обсуждается, как эти задачи можно выполнить более простым путем с использованием векторных операций языка TradeScript™.

Условная функция "IF"

IF(Condition, True part, False part)

Условная функция "IF" позволяет создавать сложные логические фильтры. Эта функция возвращает определенные значения в зависимости от выполнения условия:

```
SET A = IF(CLOSE > OPEN, 1, -1)
```

Первый аргумент функции "IF" – это логическое выражение. Второй аргумент – это значение, которое вернет функция, если значения логического выражения TRUE. Третий аргумент – это возвращаемое значение, если логическое выражение равно FALSE.

Логическим выражением может быть любое значение или выражение, которое можно оценить как TRUE или FALSE. Например, CLOSE > OPEN является логическим выражением. Если цена закрытия больше цены открытия, то значение этого выражения равно TRUE и функция вернет значение 1. В противном случае, выражение равно FALSE и функция вернет значение -1.

Функция LOOP и симуляция управляющей структуры

LOOP(Vector1, Vector2, Offset1, Offset2, Operator)

Функция LOOP обеспечивает симуляцию управляющей структуры посредством однократного вызова функции

Рассмотрим следующий код:

```
SET X = CLOSE  
SET X = REF(X, 1) + X
```

Этот скрипт просто суммирует предыдущую и текущую цены закрытия. REF(X, 1) оценивается один раз. Это ожидаемое поведение для векторного языка. Векторы рассчитываются независимо в пошаговых режимах и не являются рекурсивными.

Теперь заменим CLOSE на 0. Логично ожидать, что X будет равно значению предыдущего X плюс 1, и, следовательно, можно ожидать, что функция REF(X, 1) будет вычисляться один раз для каждой записи вектора:

```
SET X = 0  
SET X = REF(X, 1) + X
```

Несмотря на то, что внешне формулы не отличаются, благодаря тому, что **X** инициализирован как скалярная величина и не связан ни с каким вектором, можно ожидать, что **X** будет рассчитан как ряд: 1,2,3,4,5,6,...n

Сейчас мы столкнулись с ограничениями векторного программирования, которые требуют использования управляющей структуры.

Всякий раз, когда мы присваиваем переменной значение, являющееся ссылкой на саму переменную, например, **SET X = F(X)**, мы предполагаем, что **F(X)** будет рекурсивной. В первом примере мы написали **SET X = CLOSE**. CLOSE является переменной (не функцией) и никак не связана с **X**. Наши предположения изменяются, если мы инициализируем **X** значением, не являющимся существующим вектором.

Функция LOOP преодолевает эти внутренние ограничения, симулируя конструкцию структурного программирования, которая позволяет выполнять циклические итерации:

LOOP(Vector1, Vector2, Offset1, Offset2, Operator)

Vector1 – вектор, из которого берутся данные для калькуляции.

Offset1 – сдвиг, чтобы организовать ссылку на значение в векторе **Vector1** для калькуляций с приращением,

Offset2 – сдвиг, чтобы организовать ссылку на значение в векторе **Vector2**.

Operator – оператор, который определяет действие, которое нужно произвести над **Vector1** чтобы получить **Vector2**.

В функции могут использоваться следующие операторы: **ADD**, **SUBTRACT**, **MULTIPLY** и **DIVIDE**

Пример 1:

X (Vector1) – серия значений от 5.25 до 11.25.

Если мы напишем **LOOP(X, 2, 1, 0, MULTIPLY)**, то функция вернет вектор, который будет содержать значения, начинающиеся со значения вектора **X** с индексом 1, последовательно умножаемые на 2:

X	LOOP
5.25	5.25
6.25	10.5
7.25	21
8.25	42
9.25	84
10.25	168
11.25	336

Пример 2:

В случае **SET X = REF(X, 1)**, Vector1 – X, а Vector2 - 1. Поскольку в следующем примере мы добавляем скалярную величину 1 (не вектор) к X, то Offset2 присваивается 0:

SET X = LOOP(X, 1, 1, 0, ADD)

И в итоге X содержит серию 1,2,3,4,5,6,...n

Пример 3:

SET X = REF(CLOSE,1)
SET Y = (REF(Y, 3) - X) * 2

Поскольку Y требует использования управляющей структуры, мы должны написать:

SET X = REF(CLOSE,1)
SET Y = LOOP(Y, X, 3, 0, SUBTRACT) * 2

Предыдущие выражения можно сократить:

SET Y = LOOP(Y, CLOSE, 3, 1, SUBTRACT) * 2

COUNTIF

COUNTIF(Condition)

Возвращает вектор, показывающий, сколько раз выполнялось заданное условие (**Condition=TRUE**).

Пример:

COUNTIF(CROSSOVER(SimpleMovingAverage(CLOSE, 14), CLOSE))

Этот скрипт возвращает вектор, содержащий возрастающие значения, показывающие сколько раз 14-дневная простая скользящая средняя пересекала снизу-вверх цену закрытия.

LASTIF

LASTIF(Condition)

Возвращает число дней, прошедших с момента, когда выполнялось заданное условие. В день выполнения условия, «счетчик дней» сбрасывается на 0..

Пример:

LASTIF(CLOSE < REF(CLOSE, 1))

Этот скрипт возвращает вектор, содержащий возрастающие значения для каждого бара, если цена закрытия не меньше предыдущей цены закрытия. Если заданное условие оценивается как Истина (цена закрытия меньше предыдущей цены закрытия), то в вектор записывается значение 0 и приращение числа дней начинается заново.

SUMIF

SUMIF(Condition, Vector)

Функция возвращает текущую сумму всех значений вектора **Vector** для баров, при выполнении условия **Condition=TRUE**.

Например, если мы хотим, чтобы вектор содержал сумму объемов всех дней, когда цена закрытия была на 5% выше предыдущей цены закрытия, нужно написать:

SUMIF(CLOSE > REF(CLOSE,1) * 1.05, VOLUME)

В результате мы получим вектор, содержащий текущую сумму для каждого дня, когда выполнялось заданное условие.

SUM

SUM(Vector, Periods)

Функция возвращает вектор, содержащий текущие суммы значений вектора **Vector** за период определенный параметром **Periods**.

Пример:

SUM(CLOSE, 10)

Скрипт возвращает вектор с суммами, рассчитанными в 10-дневном окне.

AVG

AVERAGE(Vector, Periods)

AVG(Vector, Periods)

Возвращает вектор, содержащий текущие средние значения вектора **Vector**, рассчитанные за период **Periods**. Существует сокращенный синтаксис этой функции - AVG.

Пример:

AVERAGE(CLOSE, 10)
AVG(CLOSE, 10)

Оба скрипта возвращают векторы со средними значениями, рассчитанными в 10-дневном окне.

MAX

MAX(Vector, Periods)

Возвращает вектор, содержащий текущие максимальные значения вектора **Vector**, за период **Periods**.

Пример:

MAX(CLOSE, 10)

Возвращает вектор максимальных значений в 10-дневном окне.

MIN

MIN(Vector, Periods)

Возвращает вектор, содержащий текущие минимальные значения вектора **Vector**, за период **Periods**.

Например:

MIN(CLOSE, 10)

Возвращает вектор минимальных значений в 10-дневном окне.

MAXOF

MAXOF(Vector1, Vector2, [Vector3]...[Vector8])

Возвращает вектор, содержащий максимальные значения, специфицированных векторов. Можно использовать до 8 векторов. **Vector1** и **Vector2** являются обязательными аргументами. Вектора 3-8 - необязательны.

Пример:

MAXOF(CLOSE, OPEN)

Возвращает вектор, содержащий максимальное значение каждого бара, которое является либо ценой закрытия, либо ценой открытия этого бара.

MINOF

MINOF(Vector1, Vector2, [Vector3]...[Vector8])

Возвращает вектор, содержащий минимальные значения, специфицированных векторов. Можно использовать до 8 векторов. **Vector1** и **Vector2** являются обязательными аргументами. Векторы 3-8 - необязательны.

Пример:

MINOF(CLOSE, OPEN)

Возвращает вектор, содержащий минимальное значение каждого бара, которое является либо ценой закрытия, либо ценой открытия.

REF

REF(Vector, Periods)

По умолчанию во всех расчетах используется последнее значение вектора. Например, следующий скрипт оценивает меньше цена открытия последнего бара, чем 30\$, или нет.

OPEN < 30

Константа OPEN подразумевает цену открытия *текущего бара* по умолчанию. В то же время при помощи функции REF Вы можете сослаться на предыдущие значения этого вектора:

REF(OPEN, 1) < 30

В этом скрипте сравнивается цена открытия *предпоследнего бара с числом 30*. Число 1 (второй аргумент функции) указывает, что сослаться следует на один (1) бар в прошлое. Чтобы сослаться на 2 бара в прошлое, то единицу нужно просто заменить на число 2. Допустимый диапазон значений для аргумента **Periods** составляет 1 - 250 за исключением некоторых случаев, которые будут описаны позже.

TREND**TREND(Vector,N)**

Функция TREND используется для идентификации типа тренда данных вектора на N периодах ($2 \leq N \leq 30$). Тренд может быть восходящим, нисходящим или боковым. Эта функция может быть использована с ценами (open, high, low, close), объемом, или любым другим вектором. Функция TREND возвращает одно из трех константных значений: **UP**, **DOWN** или **SIDEWAYS**.

Пример:

TREND(CLOSE,N) = UP AND TREND(VOLUME,N) = DOWN

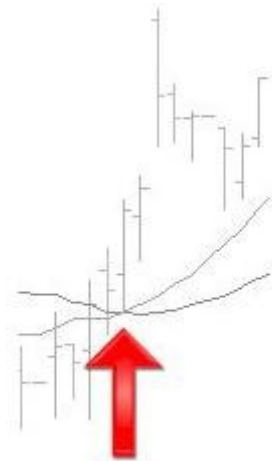
Функция TREND часто используется в качестве фильтра акций, имеющих нужное направление тренда.

CROSSOVER

Многие технические индикаторы, например MACD, имеют "сигнальную линию". Обычно сигналы покупки или продажи генерируются, когда сигнальная линия пересекает снизу-вверх или сверху вниз линию индикатора.

Функция CROSSOVER индицирует момент, когда одна серия данных пересекает другую серию. Например, мы можем точно определить точку во времени, когда одна скользящая средняя пересекает другую скользящую среднюю:

```
SET MA1 = SimpleMovingAverage(CLOSE, 28)
SET MA2 = SimpleMovingAverage(CLOSE, 14)
CROSSOVER(MA1, MA2) = TRUE
```



Приведенный выше скрипт возвратит *Истина*, если вектор MA1 пересечет снизу вверх вектор MA2. Обратное направление пересечения мы можем идентифицировать при помощи следующего скрипта:

CROSSOVER(MA2, MA1) = TRUE**Математические функции**

Обратите внимание, что все математические функции возвращают вектор. Например, **ABS(CLOSE - OPEN)** возвратит вектор, содержащий абсолютные значения разностей **CLOSE - OPEN** (одна запись на бар). Функция RND возвратит вектор случайных значений (одно значение на бар) и т.д.

ABS

Возвращает абсолютное значение числа. Отрицательные значения становятся положительными, а положительные не изменяются.

Пример:

ABS(CLOSE - OPEN)

Это скрипт всегда возвращает положительные числа, независимо от того больше **CLOSE** чем **OPEN**, или нет.

SIN

Возвращает синус числа (угла в радианах).

Пример:

SIN(1.570796327) = 1

COS

Возвращает косинус числа (угла в радианах).

Пример:

COS(3.141592654) = - 1

5)

TAN

Возвращает тангенс числа (угла в радианах).

Пример:

TAN(0.785398163) = 1

ATN

Возвращает арктангенс числа.

Пример:

ATN(1)

Скрипт возвратит 0.785398163

EXP

Возвращает экспоненту числа (e^x). Обратной функцией является LOG.

Например:

EXP(3.26)

Скрипт возвратит 26.28

LOG

Возвращает натуральный логарифм положительного числа. Обратной функцией является EXP. См. также LOG10.

Пример:

LOG(26.28)

Скрипт возвратит 3.26

LOG10

Возвращает десятичный логарифм положительного числа. Также см. LOG.

Пример:

```
LOG10(26.28)
```

Скрипт возвратит 1.42

RND

Возвращает случайное число от 0 до определенного максимального значения.

Пример:

```
RND(100)
```

Возвратит случайное число от 0 до 100.

Операторы

Равно (=)

Оператор равенства используется для присвоения значений переменным или векторам, или для сравнения значений.

Когда этот оператор используется для присвоения значений, то одиночной переменной или вектору, расположенным слева от «=», будет присвоено значение, определенное в одной или более переменных, векторах, и/или выражениях, расположенных справа от знака равенства. Кроме того, перед «левой» переменной должно стоять ключевое слово **SET**:

```
SET A = 123  
SET B = 123  
A = B = TRUE
```

Больше чем (>)

Оператор «>» определяет, больше ли первое выражение, чем второе.

Пример:

```
SET A = 124  
SET B = 123  
A > B = TRUE
```

Меньше чем (<)

Оператор «<» определяет, меньше ли первое выражение, чем второе.

Пример:

```
SET A = 123  
SET B = 124  
A < B = TRUE
```

Больше чем, или равно (>=)

Оператор «>=» определяет, больше или равно первое выражение, по отношению ко второму выражению.

Пример:

```
SET A = 123  
SET B = 123  
A >= B = TRUE
```

И:

```
SET A = 124
SET B = 123
A >= B = TRUE
```

Меньше чем, или равно (<=)

Оператор «<=» определяет, меньше или равно первое выражение, по отношению ко второму выражению.

Пример:

```
SET A = 123
SET B = 123
A <= B = TRUE
```

И:

```
SET A = 123
SET B = 124
A <= B = TRUE
```

Не Равно (<> или !=)

Операторы «!=» или «<>» определяют, не равно ли первое выражение, второму выражению.

Пример:

```
SET A = 123
SET B = 124
A != B = TRUE
```

Логическое И (AND)

Оператор используется для логического умножения двух выражений, если выражения относятся к подтипам «Null» или булевым и их можно оценивать при помощи значений True или False.

AND также можно использовать в качестве "побитового оператора" для побитового сравнения двух целочисленных величин. Если каждый бит равен 1, то выражение возвращает 1, в противном случае возвращается 0.

Порядок выражений внутри булевского выражения при использовании AND не имеет значения.

Пример:

```
(TRUE = TRUE AND FALSE = FALSE) = TRUE
```

И:

```
(TRUE = TRUE AND FALSE = TRUE) = FALSE
```

Логическое ИЛИ (OR)

Оператор используется для логической дизъюнкции двух выражений, если выражения относятся к подтипам «Null» или булевым и их можно оценивать при помощи значений True или False.

OR также можно использовать в качестве "побитового оператора" для побитового сравнения двух целочисленных величин. Если один или оба бита равны 1, то выражение возвращает 1, в противном случае возвращается 0.

Порядок выражений внутри булевского выражения при использовании OR имеет значения.

Пример:

(TRUE = TRUE OR TRUE = FALSE) = TRUE

И:

(FALSE = TRUE OR TRUE = FALSE) = FALSE

Логическое исключение (XOR)

Оператор используется для логического исключения из двух выражений, если выражения относятся к подтипам «Null» или булевским и их можно оценивать при помощи значений True или False.

XOR также можно использовать в качестве "побитового оператора" для побитового сравнения двух целочисленных величин. Если оба бита имеют одинаковое значение (оба 0 или 1), то выражение возвращает 0, в противном случае возвращается 1.

Пример:

(TRUE XOR FALSE) = TRUE

И:

(FALSE XOR FALSE) = FALSE

Логическое отрицание (NOT)

Оператор используется для логического отрицания выражения. Выражение должно относиться к подтипам «Null» или булевским и его можно оценивать при помощи значений True или False. Этот оператор изменяет значение выражения на True, если выражение = False, и наоборот, изменяет значение на False, если выражение = True.

Пример:

NOT (TRUE = FALSE) = TRUE

И:

NOT (TRUE = TRUE) = FALSE

Логическое сравнение (EQV)

Оператор используется для логического сравнения двух выражений, если выражения относятся к подтипам «Null» или булевским и их можно оценивать при помощи значений True или False.

EQV также можно использовать в качестве "побитового оператора" для побитового сравнения двух целочисленных величин. Если оба бита имеют одинаковое значения (оба 0 или 1), то выражение возвращает 1, в противном случае возвращается 0.

Порядок выражений следования выражений при сравнении не имеет значения.

Пример:

TRUE EQV TRUE = TRUE

И:

TRUE EQV FALSE = FALSE

MOD

Возвращает остаток деления.

Пример:

21 MOD 5 = 1

И: **22 MOD 5 = 2**

Глава 3. Функции технического анализа

Для анализа фондового рынка используется фундаментальный и/или технический анализ. Фундаментальный анализ подразумевает изучение таких данных как прибыль на акцию, доход, размер дивидендов и т.п. В то же время, технический анализ предполагает поиск «технических» паттернов на графиках при помощи различных калькуляций, которые объединяются понятием «технические индикаторы».

Технический анализ – вид анализа рынков, при котором исследуется спрос и предложение акций на основе поведения цен и объемов. Технические аналитики пытаются идентифицировать ценовые тренды рынков, используя один или несколько технических индикаторов.

Существует множество различных технических индикаторов, и большинство из них встроено в TradeScript™ в виде *примитивов*.

TradeScript™ также позволяет программировать собственные технические индикаторы, посредством комбинирования функций, описанных во второй главе.

В главе 4 будут рассмотрены примеры и техника построения пользовательских индикаторов и торговых систем.

В этой главе приводится описание всех *примитивных функций технического анализа*, которые поддерживает TradeScript™.

Обратите внимание, многие технические индикаторы имеют длинные и короткие имена. Последние сокращают размер кода и делают его более читабельным.

Длинное имя:

SimpleMovingAverage(CLOSE, 30)

Короткое имя:

SMA(CLOSE, 30)

Обе функции абсолютно идентичны.

Скользящие средние

Скользящие средние являются основой технического анализа. Эти функции вычисляют средние или вариации средних от соответствующего аргумента, который является вектором. Сглаживание является частью методик вычисления многих технических индикаторов.

Например, индикатор Moving Average Convergence/Divergence (MACD) в TradeScript™ позволяет Вам определить тип скользящей средней, используемой для калькуляции «сигнальной линии».

В этом разделе рассказывается о простой скользящей средней, которая является обычной среднеарифметической значений за определенный период времени; экспоненциальной скользящей средней, в которой больший вес придается более поздним значениям; а также о нескольких других типах скользящих средних.

С каждым типом скользящей средней связана специальная константа-идентификатор (MA Type Argument ID). Этот идентификатор может быть использован в качестве аргумента функции для указания типа скользящей средней, который будет использоваться при вычислении технического индикатора.

Простая скользящая средняя

SimpleMovingAverage(Vector, Periods)

SMA(Vector, Periods)

MA Type Argument ID: **SIMPLE**

Обзор

Simple Moving Average – простое среднее значений за определенный период.

Интерпретация

Используется для сглаживания значений цены или индикатора.

Пример

CLOSE > SMA(CLOSE, 30)

Возвращает *Истина*, если цена закрытия больше 30-дневной SMA.

Экспоненциальная скользящая средняя

ExponentialMovingAverage(Vector, Periods)

EMA(Vector, Periods)

MA Type Argument ID: **EXPONENTIAL**

Обзор

Экспоненциальная скользящая средняя похожа на простую, но при расчете **EMA** больший вес присваивается более поздним значениям.

Интерпретация

Та же, что для простой скользящей средней.

Пример

CLOSE > EMA(CLOSE, 30)

Возвращает *Истина*, если цена закрытия больше 30-дневной EMA.

Скользящая средняя «Time Series»

TimeSeriesMovingAverage(Vector, Periods)

TSMA(Vector, Periods)

MA Type Argument ID: **TIME_SERIES**

Обзор

«Time Series» скользящая средняя похожа на простую, за исключением того, что значениями вектора являются прогнозные величины, полученные при помощи линейной регрессии.

Интерпретация

Та же, что для простой скользящей средней.

Пример

CLOSE > TSMA(CLOSE, 30)

Возвращает *Истина*, если цена закрытия больше 30-дневной TSMA.

Переменная скользящая средняя

VariableMovingAverage(Vector, Periods)

VMA(Vector, Periods)

MA Type Argument ID: **VARIABLE**

Обзор

Переменная скользящая средняя похожа на экспоненциальную скользящую среднюю за исключением того, что она «подстраивается» под волатильность вектора.

Интерпретация

Та же, что для простой скользящей средней.

Пример

CLOSE > VMA(CLOSE, 30)

Возвращает *Истина*, если цена закрытия больше 30-дневной VMA.

Триангулярная скользящая средняя

TriangularMovingAverage(Vector, Periods)

TMA(Vector, Periods)

MA Type Argument ID: **TRIANGULAR**

Обзор

Триангулярная скользящая средняя похожа на простую, за исключением того, что больший вес присваивается значениям, лежащим в средней части периода.

Интерпретация

Та же, что для простой скользящей средней.

Пример

CLOSE > TMA(CLOSE, 30)

Возвращает *Истина*, если цена закрытия больше 30-дневной TMA.

Взвешенная скользящая средняя

WeightedMovingAverage(Vector, Periods)

WMA(Vector, Periods)

MA Type Argument ID: **WEIGHTED**

Обзор

При расчете взвешенной скользящей средней большие веса отдаются более поздним значениям и меньшие - более старым значениям.

Интерпретация

Та же, что для простой скользящей средней.

Пример

CLOSE > WMA(CLOSE, 30)

Возвращает *Истина*, если цена закрытия больше 30-дневной WMA.

Сглаживание по Уэллсу Уайлдеру (Скользящая средняя)

WellesWilderSmoothing(Vector, Periods)

WWS(Vector, Periods)

MA Type Argument ID: **WILDER**

Обзор

Сглаживание по Уэллсу Уайлдеру похоже на экспоненциальную скользящую среднюю. Этот индикатор не использует стандартную формулу экспоненциального сглаживания. Для расчета используется $1/14$ текущего значения + $13/14$ вчерашней скользящей средней (14-дневной EMA).

Интерпретация

Та же, что для простой скользящей средней.

Пример**CLOSE > WWS(CLOSE, 30)**

Возвращает *Истина*, если цена закрытия больше 30-дневной WWS.

Динамическая средняя индекса волатильности [Volatility Index Dynamic Average - VIDYA (Скользящая средняя)]
VIDYA(Vector, Periods, R2Scale)

MA Type Argument ID: **VIDYA**

Обзор

Динамическая средняя индекса волатильности, разработанная Тушаром Чандом¹, рассчитывается на базе критерия R-квадрат, используемого в оценке линейной регрессии.

Интерпретация

Та же, что для простой скользящей средней. В связи с тем, что VIDYA является производной линейной регрессии, она быстро адаптируется к волатильности.

Параметры

R2Scale – является значением R-квадрат, которое используется в калькуляции линейной регрессии. Т.Чанд рекомендует использовать значение в пределах 0.5 - 0.8 (по умолчанию 0.65).

Пример**CLOSE > VIDYA(CLOSE, 30, 0.65)**

Возвращает *Истина*, если цена закрытия больше 30-дневной VIDYA с R2 = 0.65.

Функции линейной регрессии

Классическая задача статистики – попытаться определить взаимосвязи между двумя случайными переменными X и Y, например, между ценами закрытия, смещенными во времени. Линейная регрессия предполагает построение прямой, наиболее тесно прилегающей к данным. Регрессионная модель постулирует, что

$$Y = a + bX + e,$$

где параметр **e** является случайной величиной, среднее значение которой стремиться к 0. Коэффициенты **a** и **b** определяются условием, что сумма квадратов остатков должна быть минимальной. Индикаторы, описанные в этом разделе, построены на основе этой модели.

R² (R-квадрат)**RSquared(Vector, Periods)**
R2(Vector, Periods)**Обзор**

R-квадрат – коэффициент детерминации, который поддерживает вектор **Vector** за период **Periods**. Значение лежит в пределах от 0 до 1.

Пример**R2(CLOSE, 30) < 0.1**

Возвращает *Истина*, если коэффициент детерминации меньше 0.1.

¹ Mr. Tuschar Chande

Slope (Наклон)

Slope(Vector, Periods)

Обзор

Возвращает значение наклона линейной регрессии вектора **Vector** за период **Periods**. Соответствует коэффициенту *b*. Значение колеблется между положительными и отрицательными величинами.

Пример

SLOPE(CLOSE, 30) > 0.3

Возвращает *Истина*, если наклон больше 0.3.

Forecast (Прогноз)

Forecast(Vector, Periods)

Обзор

Возвращает прогнозное значение для следующего периода, рассчитанное при помощи линейной регрессии за период **Periods** на векторе **Vector**.

Пример

Forecast(CLOSE, 30) > REF(CLOSE,1)

Возвращает *Истина*, если прогнозное значение больше, чем предыдущая цена закрытия.

Intercept (Точка пересечения)

Intercept(Vector, Periods)

Обзор

Возвращает значение точки, в которой регрессионная прямая пересекается с осью Y. Регрессия рассчитывается для вектора **Vector** за период **Periods**. Соответствует коэффициенту *a*.

Пример

Intercept(CLOSE, 30) > REF(CLOSE,1)

Возвращает *Истина*, если значение точки пересечения больше, чем предыдущая цена закрытия.

Функции диапазонов (конверты)

Некоторые технические индикаторы разработаны как некий диапазон (конверт), который накладывается на график цены подлежащей акции. Это позволяет определить изменение тренда, если подлежащая цена прорывает одну из границ диапазона или возвращается обратно после короткого прикосновения к границе. Наиболее популярным индикатором данного типа являются границы Боллинджера (Bollinger Bands), разработанные Джоном Боллинджером² в начале 1980-х годов.

Границы Боллинджера (Bollinger Bands)

BollingerBandsTop(Vector, Periods, Standard Deviations, MA Type)

BBT(Vector, Periods, Standard Deviations, MA Type)

BollingerBandsMiddle(Vector, Periods, Standard Deviations, MA Type)

BBM(Vector, Periods, Standard Deviations, MA Type)

BollingerBandsBottom(Vector, Periods, Standard Deviations, MA Type)

BBB(Vector, Periods, Standard Deviations, MA Type)

Обзор

² John Bollinger

В основе границ Боллинджера лежит расчет стандартного отклонения цен акций от своего среднего за период. Если акции становятся более волатильными, то границы Боллинджера расширяются (уходят дальше от средней). Напротив, если рынок становится менее волатильным, границы Боллинджера сужаются (прижимаются к средней). Сужение границ часто используют как ранний предвестник скорого повышения волатильности акций.

Интерпретация

Границы Боллинджера (как и другие «диапазонные» индикаторы) могут быть наложены на график цены или график другого индикатора. Если цены растут выше верхней границы, или падают ниже нижней границы индикатора, можно предположить изменение тренда, когда цена пенетрирует границу после небольшого разворота в противоположную сторону.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **20**

Standard Deviations: **2**

MA Type: **EXPONENTIAL**

Пример

CLOSE > BBT(CLOSE, 20, 2, EXPONENTIAL)

Возвращает *Истина*, если цена закрытия выше 20-дневной верхней границы Боллинджера, рассчитанной с 2 стандартными отклонениями с использованием экспоненциальной скользящей средней.

Конверты (Moving Average Envelope)

MovingAverageEnvelopeTop(Periods, MA Type, Shift)

MAET(Periods, MA Type, Shift)

MovingAverageEnvelopeBottom(Periods, MA Type, Shift)

MAEB(Periods, MA Type, Shift)

Обзор

Конверты состоят из скользящих средних, рассчитанных на базе цены подлежащей акции, которые сдвинуты вверх и вниз на определенный процент.

Интерпретация

Аналогично интерпретации границ Боллинджера.

Рекомендованные параметры

Periods: **20**

MA Type: **SIMPLE**

Shift: **5**

Пример

CLOSE > MAET(20, SIMPLE, 5)

Возвращает *Истина*, если цена закрытия выше 20-дневной верхней границы конверта, рассчитанного как 5-процентное отклонение от простой скользящей средней.

Границы простых чисел (Prime Number Bands)

PrimeNumberBandsTop()

PNBT()

PrimeNumberBandsBottom()

PNBB()

Обзор

Этот современный индикатор идентифицирует ближайшие к текущему максимуму и минимуму простые числа, и строит две граничные линии.

Пример

CLOSE > PNBT()

Возвращает *Истина*, если цена закрытия больше Prime Number Bands Top.

Осцилляторы

В этом разделе описываются индикаторы, значения которых колеблется в определенном диапазоне. Наиболее часто осцилляторы измеряют скорость направленного движения цены или объема. Для интерпретации этих индикаторов часто используются понятия зон перекупленности и перепроданности. При нахождении осцилляторов в этих зонах возможна реакция или разворот тренда. Наклон осциллятора обычно пропорционален скорости движения цены. Кроме того, диапазон движения значения осциллятора обычно пропорционален диапазону движения цены. К осцилляторам относится достаточно большое количество индикаторов, поэтому в этом разделе описана только небольшая часть.

Осциллятор момента (Momentum Oscillator)

MomentumOscillator(Vector, Periods)

MO(Vector, Periods)

Обзор

Осциллятор момента рассчитывает изменение цены за определенный промежуток времени в виде отношения.

Интерпретация

Последовательно повышающиеся значения индикатора указывают на восходящий тренд. Этот индикатор тесно связан с MACD и Price Rate of Change (ROC).

Рекомендованные параметры

Vector: **CLOSE**

Periods: **14**

Пример

MO(CLOSE, 14) > 90

Возвращает *Истина*, если осциллятор момента выше 90.

Осциллятор момента Чанда (Chande Momentum Oscillator)

ChandeMomentumOscillator(Vector, Periods)

CMO(Vector, Periods)

Обзор

Осциллятор момента Чанда (Chande Momentum Oscillator - CMO), разработанный Тушаром Чандом, является более продвинутым осциллятором момента, который вычисляется на основе линейной регрессии. Этот индикатор был опубликован в его книге "New Concepts in Technical Trading" в середине 90-х годов.

Интерпретация

CMO входит в зону перекупленности при значениях выше +50, и в зону перепроданности при значениях ниже -50. Вы также можете создать триггеры купить/продать на основе скользящей средней CMO.

Кроме того, последовательность новых максимумов индикатора CMO может говорить о восходящем тренде подлежащей акции, и наоборот, последовательность новых минимумов может свидетельствовать о нисходящем тренде.

Рекомендованные параметрыVector: **CLOSE**Periods: **14****Пример****СМО(CLOSE, 14) > 48**Возвращает *Истина*, если СМО цены закрытия находится в зоне перекупленности.**Осциллятор Объема (Volume Oscillator)****VolumeOscillator(Short Term Periods, Long Term Periods, MA Type, Points or Percent)****VO(Short Term Periods, Long Term Periods, MA Type, Points or Percent)****Обзор**

Осциллятор Объема показывает спред между двумя различными скользящими средними объема за определенный период.

Интерпретация

Предлагает новый взгляд на поведение (повышение/снижение) объема.

Рекомендованные параметрыShort Term Periods: **9**Long Term Periods: **21**MA Type: **SIMPLE**Points or Percent: **PERCENT****Пример****VO(9, 21, SIMPLE, PERCENT) > 0****Осциллятор Цены (Price Oscillator)****PriceOscillator(Vector, Short Term Periods, Long Term Periods, MA Type)****PO(Vector, Short Term Periods, Long Term Periods, MA Type)****Обзор**

Похож на Осциллятор Объема. Рассчитывается как спред двух скользящих средних.

Интерпретация

В основе Осциллятора Цены лежит спред скользящих средних. Сигнал на покупку возникает, когда значение осциллятора растет, а на продажу – когда падает.

Рекомендованные параметрыVector: **CLOSE**Short Term Periods: **9**Long Term Periods: **14**MA Type: **SIMPLE****Пример****PO(CLOSE, 9, 14, SIMPLE) > 0**Возвращает *Истина*, если значение осциллятора положительное число.**Детрендовый ценовой осциллятор (Detrended Price Oscillator)****DetrendedPriceOscillator(Vector, Periods, MA Type)****DPO(Vector, Periods, MA Type)****Обзор**

Похож на Осциллятор Цены. Обычно DPO используется, когда долгосрочные тренды или выбросы затрудняют анализ цены подлежащей акции.

Интерпретация

Покупаем, когда значение осциллятора растет, и продаем, когда падает.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **20**

MA Type: **SIMPLE**

Пример

DPO(CLOSE, 20, SIMPLE) > 0

Возвращает *Истина*, если значение осциллятора положительное число.

Осциллятор простых чисел (Prime Number Oscillator)

PrimeNumberOscillator(Vector)

PNO(Vector)

Обзор

Находит ближайшее простое целое число около текущего пика или донышка вектора и вычисляет разность между этим значением и значениями вектора.

Интерпретация

Этот индикатор может использоваться для идентификации разворотных точек. Если осциллятор остается в одной и той же максимальной точке в течение двух последовательных периодов, находясь при этом в положительном диапазоне, то рассматривается вопрос о продаже. И наоборот, если индикатор остается в нижней точке в течение двух последовательных периодов, находясь в отрицательной зоне, то рассматриваем покупку.

Рекомендованные параметры

Vector: **CLOSE**

Пример

**PNO(CLOSE) = REF(PNO(CLOSE), 1)
AND REF(PNO(CLOSE), 2) != PNO(CLOSE)**

Осциллятор фрактального хаоса (Fractal Chaos Oscillator)

FractalChaosOscillator(Periods)

FCO(Periods)

Обзор

Хаотическая природа движения рынка акций объясняет, почему иногда трудно отличить дневные графики от месячных, если мы не видим шкалу времени. Ценовые модели очень похожи и не связаны с «временным» разрешением. Подобно отсекам Наутилуса, каждый уровень похож на предыдущий, и отличается только размерами. Чтобы определить, что происходит на текущем уровне разрешения, может быть использован осциллятор фрактального хаоса.

Интерпретация

Сигнал на покупку возникает, если осциллятор достигает верхних значений, а на продажу, если индикатор достигает нижних значений.

Рекомендованные параметры

Periods: **21**

Пример

FCO(21) > REF(FCO(21),1)

Радужный осциллятор (Rainbow Oscillator)

RainbowOscillator(Vector, Levels, MA Type)

RBO(Vector, Levels, MA Type)

Обзор

Радужный осциллятор вычисляется на основе нескольких скользящих средних с разными тайм-фреймами.

Интерпретация

Тренд может внезапно развернуться, если значение индикатора остается выше 0.80 или ниже 0.20 два дня подряд.

Рекомендованные параметры

Vector: **CLOSE**

Levels: **3**

MA Type: **SIMPLE**

Пример

SET R = RBO(CLOSE, 3, SIMPLE)
R > 0.8 AND REF(R, 1) > 0.8

Возвращает *Истина*, если осциллятор выше 0.8 как минимум два дня подряд.

TRIX

TRIX(Vector, Periods)

Обзор

TRIX – это осциллятор момента, который показывает скорость изменения экспоненциальной скользящей средней цены закрытия.

Интерпретация

Покупаем, когда значение TRIX растёт, и продаем, если - падает.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **9**

Пример

TRIX(CLOSE, 9) > 0.9

Возвращает *Истина*, если TRIX находится в зоне перекупленности.

Вертикально горизонтальный фильтр (Vertical Horizontal Filter)

VerticalHorizontalFilter(Vector, Periods)

VHF(Vector, Periods)

Обзор

Вертикально горизонтальный фильтр (VHF) идентифицирует находится ли рынок в трендовом, или в «пилообразном» состоянии.

Интерпретация

VHF наиболее часто используется как индикатор волатильности рынка. Кроме того, он часто входит в состав других индикаторов.

Рекомендованные параметрыVector: **CLOSE**Periods: **21****Пример****VHF(CLOSE, 21) < 0.2****Свобода движения (Ease Of Movement)****EaseOfMovement(Vector, Periods)****EOM(Vector, Periods)****Обзор**

Этот индикатор отображает уникальные взаимоотношения между изменениями цены и объема.

Интерпретация

Значение осциллятора растет, если цены имеют восходящий тренд на низком объеме, и наоборот, осциллятор падает, если цены имеют нисходящий, также на низком объеме.

Рекомендованные параметрыVector: **CLOSE**Periods: **21****Пример****EOM(CLOSE, 21) > 0**

Возвращает истина, если значение осциллятора лежит в положительной зоне.

Система направленного движения Уайлдера (Wilder's Directional Movement System)**ADX(Periods), ADXR(Periods), DIP(Periods), DIN(Periods), TRSUM(Periods), DX(Periods)****Обзор**

Система направленного движения содержит пять индикаторов: ADX, DI+, DI-, DX и ADXR.

ADX (Average Directional Movement Index) – индикатор, показывающий степень трендовости рынка. Чем выше значение ADX, тем с большей уверенностью можно говорить о наличии тренда и тем трендследящие системы становятся более предпочтительными.

DI+ и DI- - первый индикатор измеряет восходящий тренд, а второй - нисходящий.

Детальную информацию о методика расчета и интерпретации данного индикатора можно найти в книге Уэлса Уайлдера³ "New Concepts in Technical Trading Systems". Стандартный индикатор строит 14-периодные DI+ и DI- на одном графике. Иногда на этот же график добавляют ADX.

Интерпретация

Сигнал на покупку возникает, когда DI+ пересекает снизу-вверх DI-, а сигнал на продажу при обратном направлении пересечения.

Рекомендованные параметрыPeriods: **21****Пример****DIP(14) > 60**

³ Welles Wilder

Истинный диапазон (True Range)

TrueRange()

TR()

Обзор

Истинный диапазон это компонент системы направленного движения Уайлдера.

Пример

TR() > 1.95

Williams %R

WilliamsPctR(Periods)

WPR(Periods)

Обзор

Индикатор разработан Лари Вильямсом. Он определяет уровни перекупленности перепроданности. Похож на Стохастический осциллятор (Stochastic Oscillator). Значения индикатора колеблются между 0 и -100.

Интерпретация

Рынок считается перепроданным, если %R находится в диапазоне от 0 до -20, и перепроданным, при значениях от -80 до -100.

Рекомендованные параметры

Periods: **14**

Пример

WPR(14) < -80

Возвращает *Истина*, если индикатор Williams' %R перепродан.

Накопление/Распределение Вильямса (Williams' Accumulation/Distribution)

WilliamsAccumulationDistribution()

WAD()

Обзор

Этот индикатор также разработан Лари Вильямсом. Осциллятор Аккумуляции/Дистрибуции показывает взаимосвязи цены и объема.

Интерпретация

Если индикатор растет, то говорят об аккумуляции ценной бумаги, если падает, то говорят о ее дистрибуции. Тренд может развернуться, если индикатор конвергирует с ценой.

Пример

WAD() < 1

Возвращает *Истина*, если индикатор ниже 1.

Волатильность Чайкина (Chaikin's Volatility)

ChaikinVolatility(Periods, Rate of Change, MA Type)

CV(Periods, Rate of Change, MA Type)

Обзор

Осциллятор Волатильности Чайкина является сглаженным индексом Аккумуляции/Дистрибуции. Этот индикатор оценивает волатильность как ширину диапазона между максимальной и минимальной ценами.

Интерпретация

Этот осциллятор показывает волатильность, не связанную с долгосрочным поведением цены акции. Наиболее популярная интерпретация - это продавать, когда индикатор находится на максимумах, и покупать, если индикатор на минимумах.

Рекомендованные параметры

Periods: **10**

Rate of Change: **10**

MA Type: **SIMPLE**

Пример

CV(10, 10, SIMPLE) < -25

Aroon

AroonUp(Periods)

AroonDown(Periods)

Обзор

Индикатор Aroon был разработан Тушаром Чандом в середине 1990-х годов. Этот индикатор часто используется для выявления тренда и степени стабильности этого тренда.

Интерпретация

Тренд подтверждается, если обе линии индикатора (Aroon up и Aroon down) находятся выше 80, о нестабильности цен говорят, если обе линии ниже 20.

Рекомендованные параметры

Periods: **25**

Пример

AroonUp(25) > 80 AND AroonDown(25) > 80

Схождение/расхождение скользящих средних (Moving Average Convergence/Divergence - MACD)

MACD(Short Cycle, Long Cycle, Signal Periods, MA Type)

MACDSignal(Short Cycle, Long Cycle, Signal Periods, MA Type)

Обзор

MACD является осциллятором скользящей средней, который показывает потенциальные фазы перекупленности/перепроданности. MACD рассчитывается на основе двух скользящих средних цены подлежащей акции/индикатора.

Интерпретация

Интерпретация базируется на пересечениях с сигнальной линией (функция MACDSignal), перекупленности/перепроданности уровней MACD и дивергенцией между MACD и подлежащей ценой.

Рекомендованные параметры

Long Cycle: **26**

Short Cycle: **13**

Signal Periods: **9**

MA Type: **SIMPLE**

Пример

SET A = MACDSignal(13, 26, 9, SIMPLE)

SET B = MACD(13, 26, 9, SIMPLE)

CROSSOVER(A, B) = TRUE

Возвращает *Истина*, если сигнальная линия пересекла снизу-вверх линию MACD.

Диапазон цены (High Minus Low)

HighMinusLow()

HML()

Обзор

Возвращает разность между максимумом и минимумом каждого бара.

Интерпретация

Часто используется как компонент других индикаторов. Вместе с тем, индикатор может быть использован отдельно в сглаженном виде, чтобы показать изменения поведения цены за определенное время.

Пример

```
SET A = SMA(HML(), 14)
A > REF(A, 10)
```

Возвращает *Истина*, если средняя *высота* баров за определенное время увеличилась.

Стохастический осциллятор (Stochastic Oscillator)

SOPK(%K Periods, %K Slowing Periods, %D Periods, MA Type)

SOPD(%K Periods, %K Slowing Periods, %D Periods, MA Type)

Обзор

Стохастический осциллятор (стохастик) – это популярный индикатор, который рассчитывает диапазон цены закрытия за определенное время и показывает относительную позицию текущей цены закрытия в этом диапазоне.

Интерпретация

Стохастик имеет два компонента: %K (функция SOPK) и %D (функция SOPD). %K обычно отображается на графике в виде сплошной линии, а %D в виде пунктира. Один из способов интерпретации это покупать, если любой из компонентов выше 80, или продавать, если любой из компонентов ниже 20. Другой способ, это покупать, если %K пересекает снизу-вверх %D, и наоборот, продавать, если %K пересекает %D сверху вниз.

Рекомендованные параметры

% K Periods: **9**

% K Slowing Periods: **3**

% D Periods: **9**

MA Type: **SIMPLE**

Пример

```
SOPK(9, 3, 9, SIMPLE) > 80 OR SOPD(9, 3, 9, SIMPLE) > 80
```

Возвращает *Истина*, если осциллятор находится в зоне перекупленности.

Индексы

В этом разделе рассказывается о технических индикаторах, известных как *индексы*, типа Индекс относительной силы, Индекс исторической волатильности и многих других.

Индекс относительной силы (Relative Strength Index)

RelativeStrengthIndex(Vector, Periods)

RSI(Vector, Periods)

Обзор

RSI – популярный индикатор, разработанный Уэлсом Уайлдером.

Интерпретация

В интерпретации наиболее широко используется выявление дивергенции цены и индикатора, а также исследуют уровни поддержки/сопротивления и графические формации RSI.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **14**

Пример

RSI(CLOSE, 14) > 55

Индекс массы (Mass Index)

MassIndex(Periods)

MI(Periods)

Обзор

Индекс массы идентифицирует изменения цены, посредством индексирования сужения и расширения диапазона между максимальными и минимальными ценами.

Интерпретация

Согласно данным разработчика этого индикатора развороты могут возникать, если 25-периодный выше 27 или падает ниже 26.5.

Рекомендованные параметры

Periods: **25**

Пример

MI(25) > 27

Индекс исторической волатильности (Historical Volatility Index)

HistoricalVolatilityIndex(Vector, Periods, Bar History, Standard Deviations)

HVI(Vector, Periods, Bar History, Standard Deviations)

Обзор

Историческая волатильность рассчитывается как среднеквадратичное отклонение доходностей и соответственно определяется на базе логнормального стандартного отклонения цен.

Формула 30-дневного индекса исторической волатильности:

Stdev(LOG(Close / Close Yesterday), 30) * Sqrt(365)

Индекс колеблется между 0 и 1. Некоторые трейдеры в функции **Sqrt()** используют 252 вместо 365. Функция **LOG(X)** – это натуральный логарифм числа X.

Интерпретация

Высокие значения HVI указывают на высокую волатильность акций. В то же время, низкие значения говорят, что акции лежат во флэте или постоянном тренде.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **15**

Bar History: **30**

Standard Deviations: **2**

Пример

HVI(CLOSE, 15, 30, 2) < 0.01

Индекс денежного потока (Money Flow Index)

MoneyFlowIndex(Periods)

MFI(Periods)

Обзор

Индекс денежного потока измеряет денежный поток ценной бумаги при помощи объема и цены.

Интерпретация

Рыночные минимумы можно идентифицировать при значениях индикатора ниже 20, а пики, при значениях выше 80. Также можно использовать дивергенцию цены и индикатора.

Рекомендованные параметры

Periods: **15**

Пример

MFI(15) < 20

Индекс денежного потока Чайкина (Chaikin's Money Flow Index)

ChaikinMoneyFlow (Periods)

CMF (Periods)

Обзор

Индекс денежного потока Чайкина является индикатором момента, который выявляет покупку или распродажу акций при помощи цены и объема. Этот индикатор базируется на Аккумуляции/Дистрибуции, которая в свою очередь базируется на предположении, что если цена закрытия выше средней точки $[(high + low) / 2]$ торгового дня, то в этот день происходила аккумуляция акций, а если цена закрытия ниже средней точки, то это был день дистрибуции.

Интерпретация

Сигнал на покупку возникает, если индикатор растет в положительной зоне. Сигнал на продажу, если индикатор падает в отрицательной зоне.

Рекомендованные параметры

Periods: **15**

Пример

CMF(15) > 20 AND REF(CMF(15), 1) > 20

Возвращает *Истина*, если **CMF** бычий.

Сравнительный индекс относительной силы (Comparative Relative Strength Index)

ComparativeRelativeStrength(Vector1, Vector2)

CRSI (Vector1, Vector2)

Обзор

Этот индекс сравнивает один вектор с другим.

Интерпретация

Вектор **Vector1** «сильнее» чем **Vector2**, если индикатор имеет восходящий тренд.

Рекомендованные параметры

Vector1: **CLOSE**

Vector2: **[Any]**

Пример**CRSI(CLOSE, VOLUME) > 1**

Возвращает *Истина*, если тренд цены закрытия опережает тренд объема.

Объемно-ценовой тренд (Price Volume Trend)**PriceVolumeTrend(Vector)****PVT(Vector)****Обзор**

Индикатор также известен, как *Volume Price Trend*. Этот индикатор вычисляется рекуррентным образом, когда к предыдущему значению индикатора добавляется или вычитается объем текущего периода, умноженный на изменение цены в процентах. PVT используется для определения баланса между спросом и предложением акции. Этот индикатор похож на индекс *On Balance Volume*.

Интерпретация

Относится к разряду опережающих индикаторов, т.к. его изменения обычно предшествуют движению цены. Предполагается, что хорошо информированные инвесторы покупают, когда индекс растет, а не информированные инвесторы покупают, когда индекс падает.

Рекомендованные параметры

Vector: **CLOSE**

Пример**TREND(PVT(CLOSE)) = UP**

Возвращает *Истина*, если PVT имеет восходящий тренд.

Индекс положительного объема (Positive Volume Index)**PositiveVolumeIndex(Vector)****PVI(Vector)****Обзор**

Индекс положительного объема фокусируется на периодах, когда объем вырос относительно предыдущего периода.

Интерпретация

Интерпретация заключается в том, что большинство инвесторов покупает, когда индекс растет, и продает, когда индекс падает.

Рекомендованные параметры

Vector: **CLOSE**

Пример**TREND(PVI(CLOSE),30) = UP**

Возвращает *Истина*, если PVI имеет восходящий тренд на 30-ти периодах.

Индекс отрицательного объема (Negative Volume Index)**NegativeVolumeIndex(Vector)****NVI(Vector)****Обзор**

Индекс отрицательного объема похож на предыдущий индекс, за исключением того, что фокус переносится на периоды, когда объем снизился относительно предыдущего периода.

Интерпретация

Интерпретация заключается в том, что хорошо информированные инвесторы покупают, когда индекс падает, и продают, когда индекс растет.

Рекомендованные параметры

Vector: **CLOSE**

Пример

TREND(NVI(CLOSE),30) = UP

Возвращает *Истина*, если NVI имеет восходящий тренд на 30-ти периодах.

Баланс объемов (On Balance Volume)

OnBalanceVolume(Vector)

OBV(Vector)

Обзор

Этот индекс показывает отношения цены и объема в форме индекса момента.

Интерпретация

Изменения Баланса объемов обычно предшествует реальному движению цены. Предполагается, что хорошо информированные инвесторы покупают, когда индекс растет, а не информированные инвесторы покупают, когда индекс падает.

Рекомендованные параметры

Vector: **CLOSE**

Пример

TREND(OBV(CLOSE),20) = UP

Возвращает *Истина*, если OBV имеет восходящий тренд на 20-ти периодах.

Индекс продуктивности (Performance Index)

PerformanceIndex(Vector)

PFI(Vector)

Обзор

Индекс продуктивности показывает цену акции в процентах относительно начала расчета. Точка начала расчета соответствует 0%.

Интерпретация

Это индикатор показывает нормированное значение цены акции. Значение индикатора равное 50 означает, что цена подлежащей акции с момента начала расчета индикатора выросла на 50%. Напротив, значение «-50», говорит, что цена акции упала на 50%.

Рекомендованные параметры

Vector: **CLOSE**

Пример

PFI(CLOSE) > 45

Возвращает *Истина*, если значение индекса больше 45%

Индекс торгового объема (Trade Volume Index)

TradeVolumeIndex(Vector, Minimum Tick Value)

TVI(Vector, Minimum Tick Value)

Обзор

Индекс торгового объема показывает, находится ли акция в периоде аккумуляции или периоде дистрибуции (похож на Индекс Аккумуляции/Дистрибуции).

Интерпретация

Если индекс растет, говорят о накоплении акций в портфелях инвесторов. Напротив, если индекс падает, говорят о распродаже (распределении) акций. Цены могут развернуться, если движение индикатора коррелирует с движением цены.

Рекомендованные параметры

Vector: **CLOSE**

Minimum Tick Value: **0.25**

Пример

$TVI(CLOSE, 0.25) > 0$

Возвращает *Истина*, если индекс находится в положительной территории.

Свинг-индекс (Swing Index)

SwingIndex(Limit Move Value)

SI(Limit Move Value)

Обзор

Свинг-индекс (Wilder) является популярным индикатором, который показывает сравнительную силу отдельной акции, сравнивая текущие цены Open, High, Low и Close с предыдущими ценами.

Интерпретация

Свинг-индекс является компонентом Аккумуляционного свинг-индекса (Accumulation Swing Index).

Рекомендованные параметры

Limit Move Value: **1**

Пример

$SI(1) > 0$

Возвращает *Истина*, если индекс находится в положительной территории.

Аккумуляционный свинг-индекс (Accumulative Swing Index)

AccumulativeSwingIndex(Limit Move Value)

ASI(Limit Move Value)

Обзор

Аккумуляционный свинг-индекс (Wilder) является куммулятой свинг-индекса, который показывает сравнительную силу отдельной акции, сравнивая текущие цены Open, High, Low и Close с предыдущими ценами.

Интерпретация

Аккумуляционный свинг-индекс можно анализировать при помощи технических индикаторов, аналитических линий и графических моделей, как альтернатива анализа графика цены.

Рекомендованные параметры

Limit Move Value: **1**

Пример

TREND(ASI(1),10) = UP

Возвращает *Истина*, если индекс имеет восходящий тренд на 10-ти периодах.

Индекс товарного канала [Commodity Channel Index (CCI)]

CommodityChannelIndex(Periods, MA Type)

CCI(Periods, MA Type)

Обзор

Этот индикатор разработал Дональд Ламберт⁴. Хотя цель этого индикатора заключалась в исследовании циклических разворотов цен товаров, она часто используется для анализа акций.

Интерпретация

Этот индикатор осциллирует между зонами перекупленности и перепроданности и лучше всего работает на «боковых» рынках.

Рекомендованные параметры

Periods: **21**

MA Type: **SIMPLE**

Пример

CCI(12, SIMPLE) > 0 AND REF(CCI(12, SIMPLE), 1) < 0

Возвращает *Истина*, если CCI только что перешел в положительную зону.

Параболическая система [Parabolic Stop and Reversal (Parabolic SAR)]

ParabolicSAR(Min AF, Max AF)

PSAR(Min AF, Max AF)

Обзор

Параболическую систему разработал Уэллс Уайлдер. Этот индикатор всегда в рынке (если какая-то позиция закрыта, то сразу открывается противоположная). Наиболее часто индикатор используется для выставления скользящих «стопов».

Интерпретация

Событие «остановиться и развернуться» возникает, когда цена пенетрирует уровень Parabolic SAR.

Рекомендованные параметры

Min AF (Accumulation Factor): **0.02**

Max AF (Accumulation Factor): **0.2**

Пример

CROSSOVER(CLOSE, PSAR(0.02, 0.2)) = TRUE

Возвращает *Истина*, если цена закрытия пересекла Parabolic SAR.

Стохастический индекс момента (Stochastic Momentum Index)

SMIK(%K Periods, %K Smooth, %K Double Periods, %D Periods, MA Type, %D MA Type)

SMID(%K Periods, %K Smooth, %K Double Periods, %D Periods, MA Type, %D MA Type)

Обзор

Стохастический индекс момента, разработанный Уильямом Блау⁵, впервые был обнаружен в январском выпуске журнала «Stocks & Commodities magazine» за 1993 год. Этот

⁴ Donald Lambert

⁵ William Blau

индикатор выявляет степень близости цены к средней точке диапазона максимум/минимум.

Интерпретация

Стохастический индекс момента имеет два компонента: %K (SMIK) и %D (SMID).

Стохастик имеет два компонента: %K (функция SOPK) и %D (функция SOPD). %K обычно отображается на графике в виде сплошной линии, а %D в виде пунктира. Один из способов интерпретации это покупать, если любой из компонентов выше 40, или продавать, если любой из компонентов ниже 40. Другой способ, это покупать, если %K пересекает снизу-вверх %D, и наоборот, продавать, если %K пересекает %D сверху вниз.

Рекомендованные параметры

%K Periods: **14**

%K Smoothing: **2**

%K Double Periods: **3**

%D Periods: **9**

MA Type: **SIMPLE**

%D MA Type: **SIMPLE**

Пример

SMID(14, 2, 3, 9, SIMPLE, SIMPLE) > 40 OR
SMIK(14, 2, 3, 9, SIMPLE, SIMPLE) > 40

Возвращает *Истина*, если индекс находится в зоне перепроданности.

Общие «индикаторные» функции

Медиана цены (Median Price)

MEDIANPRICE()

MP()

Обзор

Медиана цены – средняя максимального и минимального значений бара.

Интерпретация

Может использоваться как альтернатива цены акции, а также применяется при калькуляции других индикаторов.

Пример

CROSSOVER(CLOSE, SMA(MP(), 14))

Возвращает *Истина*, если цена закрытия пересекла снизу-вверх 14-дневную SMA медианы цены.

Типичная цена (Typical Price)

TypicalPrice()

TP()

Обзор

Типичная цена – средняя цен High, Low и Close одного бара.

Интерпретация

Используется при расчете других индикаторов.

Пример

CROSSOVER(CLOSE, SMA(TP(), 14))

Возвращает *Истина*, если цена закрытия пересекла снизу-вверх 14-дневную SMA типичной цены.

Взвешенная цена закрытия (Weighted Close)

WeightedClose()

WC()

Обзор

Взвешенная цена закрытия является средней цен Open, High, Low и Close, где больший вес присвоен Close.

Интерпретация

Этот индикатор предлагает упрощенный взгляд на рыночные цены.

Пример

WC() > REF(WC(), 1)

Возвращает *Истина*, если текущее значение индикатора выше предыдущего.

Изменение цены (Price Rate of Change)

PriceRateOfChange(Vector, Periods)

PROC(Vector, Periods)

Обзор

Индикатор показывает разницу между текущей ценой и ценой на определенное число баров в прошлом.

Интерпретация

12-дневный Price ROC наиболее часто используется как индикатор перекупленности/перепроданности.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **12**

Пример

PROC(CLOSE, 12) > 0 AND REF(PROC(CLOSE, 12), 1) < 0

Возвращает *Истина*, если индикатор только что перешел в положительную зону.

Изменение объема (Volume Rate of Change)

VolumeRateOfChange(Vector, Periods)

VROC(Vector, Periods)

Обзор

Этот индикатор показывает имеется ли тренд объема в том или ином направлении.

Интерпретация

Резкое повышение значения индикатора может быть сигналом ценового прорыва.

Рекомендованные параметры

Vector: **VOLUME**

Periods: **12**

Пример

VROC(VOLUME, 12) > 0 AND REF(VROC(VOLUME, 12), 1) < 0

Возвращает *Истина*, если индикатор только что перешел в положительную зону.

Значение максимального пика (Highest High Value)**HighestHighValue(Periods)****HHV(Periods)****Обзор**

Возвращает значение самого большого максимума за определенное число периодов.

Интерпретация

Используется как компонент других индикаторов.

Рекомендованные параметры

Periods: **21**

Пример**HIGH = HHV(21)**

Возвращает *Истина*, если максимум текущего бара равен самому большому максимуму за последние 21 бар.

Значение минимального доньшка (Lowest Low Value)**LowestLowValue(Periods)****LLV(Periods)****Обзор**

Возвращает значение самого низкого минимума за определенное число периодов.

Интерпретация

Используется как компонент других индикаторов.

Рекомендованные параметры

Periods: **21**

Пример**LOW = LLV(21)**

Возвращает *Истина*, если минимум текущего бара равен самому низкому минимуму за последние 21 бар.

Стандартное отклонение (Standard Deviations)**StandardDeviations(Vector, Periods, Standard Deviations, MA Type)****SDV(Vector, Periods, Standard Deviations, MA Type)****Обзор**

Стандартное отклонение — это статистический показатель, который часто используется для измерения волатильности. Стандартное отклонение используется во многих технических индикаторах.

Интерпретация

Большие максимумы и минимумы часто сопровождаются экстремальной волатильностью. Высокие значения стандартного отклонения указывают, что цена или индикатор имеют волатильность выше обычной.

Рекомендованные параметры

Vector: **CLOSE**

Periods: **21**

Standard Deviations: **2**

MA Type: **SIMPLE**

Пример

`SDV(CLOSE, 21, 2, SIMPLE) > REF(SDV(CLOSE, 21, 2, SIMPLE), 10)`

Возвращает *Истина*, если стандартное отклонение за 21-период, больше чем 10 дней назад.

Корреляционный анализ (Correlation Analysis)

`CorrelationAnalysis(Vector1, Vector2)`

`CA(Vector1, Vector2)`

Обзор

Корреляционный анализ используется, чтобы выявить степень связи двух векторов.

Интерпретация

Функция возвращает значение, показывающее степень связи двух векторов. Векторами могут быть цена, значения индикаторов или другие значения.

Рекомендованные параметры

Vector1: **[Any Vector]**

Vector2: **[Any Vector]**

Пример

`CA(CLOSE, SMA(CLOSE, 14)) > 0.99`

Возвращает *Истина*, если цена закрытия высоко коррелирует со своей 14-дневной простой скользящей средней.

Фигуры японских свечей

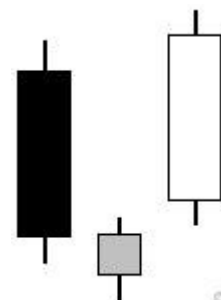
В настоящее время каждый трейдер хорошо знаком с японскими свечами, которые были популяризированы Стивом Нисоном⁶, автором книги "Japanese Candlestick Charting Techniques". Многие трейдеры работают с графиками в виде японских свечей.

Что такое свеча?

Главная особенность свечи заключается в том, что окраска области между ценой открытия и закрытия зависит от направления движения цены. Обычно, черные свечи строятся, если цена закрытия ниже цены открытия, а белые, если цена закрытия выше цены открытия. Максимальную и минимальную цены бара называют тени (хвосты, фитили). Свечи не подразумевают никаких вычислений, скорее они предлагают другую перспективу при взгляде на поведение цен. В основе интерпретации графика в виде свечей лежит анализ фигур, которые формируются на графике. Например, Вы могли слышать термины «Три черные вороны» (Three Black Crows), «Утренняя звезда» (Morning Star) или «Покрытие черным облаком» (Dark Cloud Cover). Все это свечные фигуры, которые формируются двумя или более свечами.

Графическое представление

Свечные фигуры являются группой ценовых баров, как это показано на рисунке. Обычно мы можем увидеть черные свечи в дни, когда цена закрытия ниже цены открытия, или белые свечи в дни, когда цена закрытия выше цены открытия. Иногда в описаниях можно увидеть серые бары, которые означают, что бар может быть или белым, или черным.



⁶ Steve Nison

Идентификация подсвечниковых паттернов

Несмотря на то, что Вы могли бы написать собственные скрипты, идентифицирующие свечные фигуры, TradeScript™ предлагает простые встроенные функции, которые помогают идентифицировать до двух дюжин паттернов:

CandlestickPattern()

CSP()

Обзор

Функция *CandlestickPattern()* идентифицирует свечные фигуры автоматически. Эта функция не принимает аргументов и возвращает константу-идентификатор одного из определенных паттернов.

Пример

CSP() = MORNING_STAR

Возвращает *Истина*, если фигура является «Утренней звездой».

Фигуры

Существуют следующие фигуры:

LONG_BODY
DOJI
HAMMER
HARAMI
STAR
DOJI_STAR
MORNING_STAR
EVENING_STAR
PIERCING_LINE
ENGULFING_LINE
HANGING_MAN
DARK_CLOUD_COVER
BEARISH_ENGULFING_LINE
BEARISH_DOJI_STAR
BEARISH_SHOOTING_STAR
SPINNING_TOPS
HARAMI_CROSS
BULLISH_TRISTAR
THREE_WHITE_SOLDIERS
THREE_BLACK_CROWS
ABANDONED_BABY
BULLISH_UPSIDE_GAP
BULLISH_HAMMER
BULLISH_KICKING
BEARISH_KICKING
BEARISH_BELT_HOLD
BULLISH_BELT_HOLD
BEARISH_TWO_CROWS
BULLISH_MATCHING_LOW

Глава 4. Примеры торговых систем

Торговые системы

Торговая система (МТС) является набором правил, которые определяют точки входа (и выхода) в позицию для любой данной акции. Трейдеры часто эти точки называют *торговыми сигналами*.

МТС объективна и механистична. Цель работы по построению МТС является создание стратегии, которая генерирует прибыли больше, чем убытков. Эта глава предлагает руководство по переводу правил МТС в скрипты на основе примеров реальных торговых систем.

Торговые системы обычно включают один или более технических индикаторов. Например, система, основанная на пересечении скользящих средних (*Moving Average Crossover*) будет покупать, если короткая скользящая средняя пересекает снизу-вверх длинную скользящую среднюю, и продавать, если пересечение идет в обратном направлении.

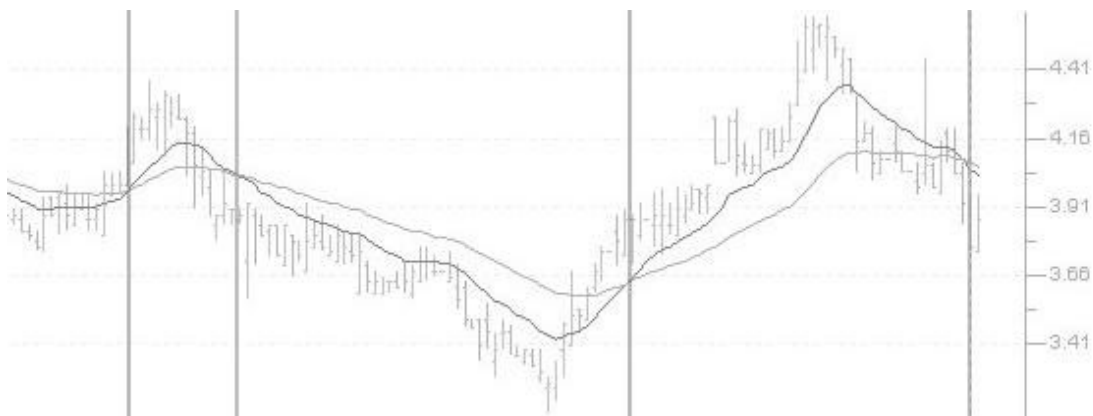
МТС могут иметь любое число правил, типа “не покупай, пока объем не будет иметь восходящий тренд”, или “выходи, если Parabolic SAR пересекла цену закрытия” и т.п.

Трейдерам много времени тратят на оптимизацию своих МТС, чтобы повысить их прибыльность и снизить риски. В случае системы *Moving Average Crossover*, это означает модификацию параметров самих скользящих средних.

Треjder может оптимизировать торговую систему при помощи *исторического тестирования (back testing)*. Функции тестирования TradeScript™ позволяют проводить историческое тестирование МТС и модифицировать их параметры, что позволяет достичь максимальной прибыли и снизить риски.

Система на основе пересечения скользящих средних (Moving Average Crossover System)

Возможно, что системы на основе пересечения скользящих средних относятся к самым простым МТС. В этой системе для генерации сигналов используются две скользящие средние (МА). Сигнал на покупку возникает, если короткая МА пересекает снизу-вверх длинную МА, а сигнал на продажу, если короткая МА пересекает длинную сверху вниз.



Число сигналов, генерируемых этой торговой системой, пропорционально длине и типу используемой скользящей средней. Более короткие МА генерируют больше сигналов, чем более длинные.

К сожалению «кроссоверные» системы со слишком короткими МА генерируют намного больше ложных сигналов, чем системы с длинными МА. В то же время, системы со слишком длинными МА хотя и будут генерировать мало ложных сигналов, но при этом будут иметь большой процент упущенной прибыли. Необходимость нахождения баланса между

ложными сигналами и упущенной прибылью относится практически к каждой торговой системе и является центром обсуждения в многочисленных книгах по техническому анализу.

Одно из решений этой проблемы заключается в использовании второго индикатора, который фильтрует сигналы входа/выхода. Для генерации сигналов выхода из позиции часто используется параболическая система (Parabolic SAR). В следующем скрипте для входа используется 20/60 EMA, а для выхода параболическая система.

Скпунт «Moving Average Crossover System»

Buy Signals

20-периодная EMA пересекает снизу-вверх 60-периодную EMA
CROSSOVER(EMA(CLOSE, 20), EMA(CLOSE, 60))

Sell Signals

20-периодная EMA пересекает сверху вниз 60-периодную EMA
CROSSOVER(EMA(CLOSE, 60), EMA(CLOSE, 20))

Exit Long

Цена закрытия пересекает снизу-вверх Parabolic SAR
CROSSOVER(CLOSE, PSAR(CLOSE, 0.02, 0.2))

Exit Short

Цена закрытия пересекает сверху вниз Parabolic SAR
CROSSOVER(PSAR(CLOSE, 0.02, 0.2), CLOSE)

Система «Price Gap System»

Ценовой разрыв вверх возникает, когда акция открывается существенно выше максимума предыдущего дня. Такая ситуация обычно возникает после неожиданного выхода положительных новостей, отчета намного лучше ожидаемого и т.п.

На открытии рынка приходит большое количество ордеров на покупку. В это время акция может быть существенно переоценена, т.к. инвесторы могут покупать акцию только потому, что она показала свою силу на открытии. В такой ситуации достаточно часто цены могут вернуться назад, чтобы закрыть разрыв, как только ордера на покупку перестают поступать и спрос на акцию слабеет. Ключевой момент этой торговой системы в том, что разворот обычно возникает в течение первого часа торгов. Другими словами, если разрыв не будет закрыт в течение первого часа, то можно предполагать, что покупка скорее всего продолжится.



Эта торговая система часто более успешна, если объем как минимум в два раза выше среднего объема последних 5 дней.

Пример: Скрипт возвращает акции, которые имели гэп не менее 2% и закрылись близко к максимуму. Стратегией на следующий день может быть покупка после первого часа торгов, если акция остается сильной.

Стоп-лосс следует установить на минимуме дня. Консервативной целью прибыли может быть $\frac{1}{2}$ гэпа - 1% в нашем случае.

Price Gap Script

Buy Signals

Имеет место не менее чем 2% гэп вверх на высоком объеме
LOW > REF(HIGH,1) * 1.02 AND
VOLUME > SMA(VOLUME, 5) * 2

Sell Signals

Имеет место не менее чем 2% гэп вниз на высоком объеме
HIGH < REF(LOW,1) * 0.98 AND
VOLUME > SMA(VOLUME, 5) * 2

Exit Long

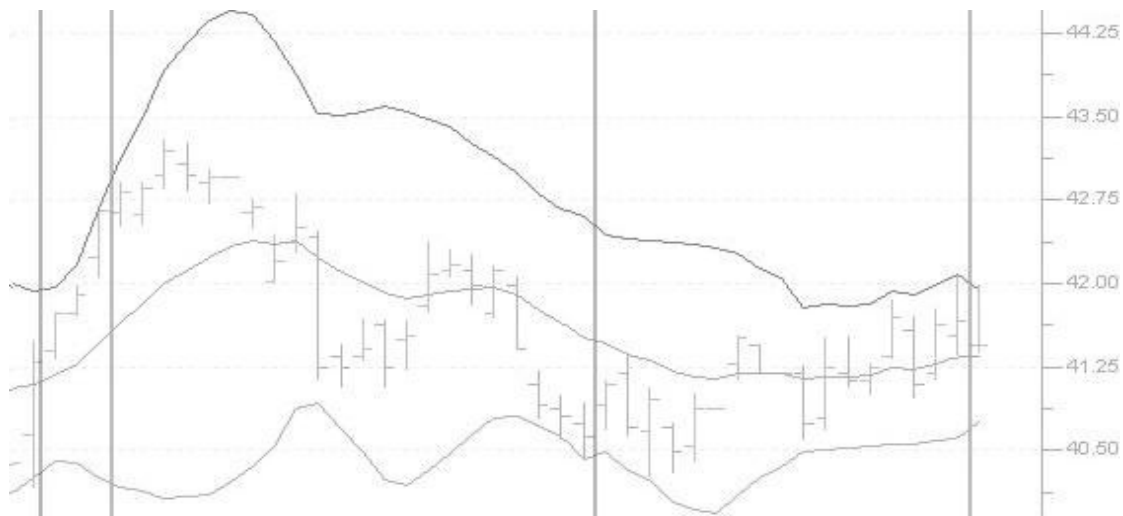
Используем цель прибыли приблизительно $\frac{1}{2}$ от величины гэпа, а также стоп-лосс.

Exit Short

Используем цель прибыли приблизительно $\frac{1}{2}$ от величины гэпа, а также стоп-лосс.

Система «Bollinger Bands System»

Полосы Боллинджера – это временные серии, сдвинутые от скользящей средней вверх и вниз на определенное число стандартных отклонений. Эти полосы формируют границы колебаний цены. Полосы Боллинджера расширяются или сужаются в зависимости от рыночной волатильности.



Обычно цены остаются внутри полос Боллинджера. Одна из стратегий заключается в покупке или продаже после того как цены коснувшись границы разворачиваются обратно. Движение, которое началось у одной полосы, обычно имеет тенденцию продолжиться до другой полосы.

Другая стратегия заключается в покупке или продаже, если цена пробила соответствующую полосу. В этом случае рынок обычно продолжает свое движение в этом направлении определенное время.

Стратегия, описанная в следующем примере, использует комбинацию этих стратегий. Система покупает, если предыдущий бар коснулся нижней полосы, а текущий бар лежит внутри диапазона (между полосами). Кроме того, система покупает, если максимум текущего бара больше верхней полосы на определенный процент. Продажа осуществляется по противоположным правилам.

Bollinger Bands Script

Buy Signals

```
# Покупаем, если предыдущее значение было ниже нижней полосы, а сейчас выше
SET Bottom = BBB(CLOSE, 20, 2, EXPONENTIAL)
SET Top = BBT(CLOSE, 20, 2, EXPONENTIAL)
((REF(CLOSE, 1) < REF(Bottom, 1)) AND
CLOSE > Bottom) OR
# Также покупаем, если цена закрытия выше верхней полосы на 2%
CLOSE > Top * 1.02
```

Sell Signals

```
# Продаем, если предыдущее значение было выше верхней полосы, а сейчас ниже
SET Bottom = BBB(CLOSE, 20, 2, EXPONENTIAL)
SET Top = BBT(CLOSE, 20, 2, EXPONENTIAL)
((REF(CLOSE, 1) > REF(Top, 1)) AND
CLOSE < Top) OR
# Также продаем, если цена закрытия ниже нижней полосы на 2%
CLOSE < Bottom * 0.98
```

Историческая волатильность и тренд

Эта торговая система покупает или продает на растущем объеме и снижающейся волатильности. Идея заключается в том, что тренд более стабилен, если волатильность снижается, а объем за многие дни имеет тенденцию к повышению.



Объем является важным компонентом этой торговой системы, т.к. практически каждая важная разворотная точка сопровождается повышением объема. Ключевой момент этой системы – продолжительность первичного ценового тренда. Чем продолжительнее тренд, тем более надежен сигнал.

Мы также попытаемся поэкспериментировать с этой системой, используя функцию TREND для объема с функцией Volume Oscillator или Volume Rate of Change.

Historical Volatility and Trend Script

Buy Signals

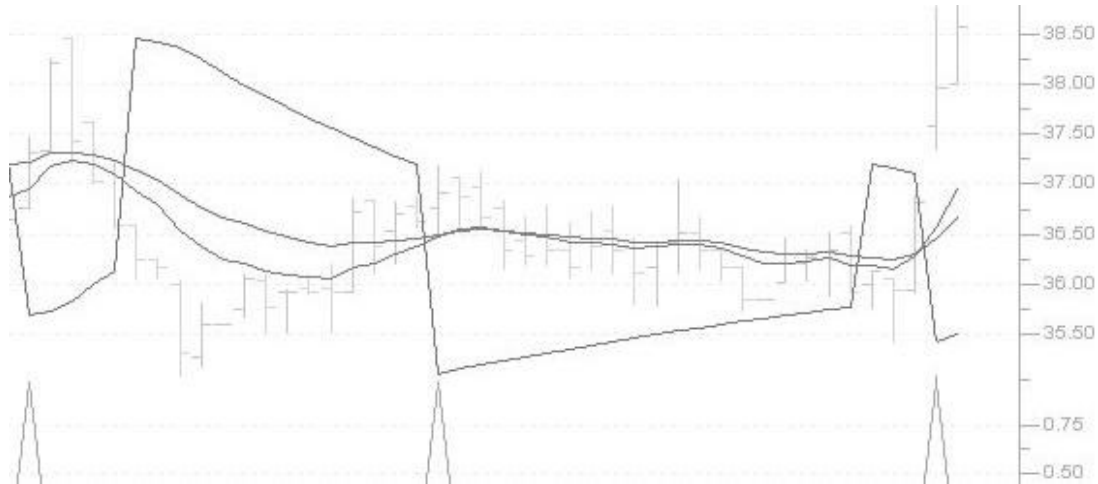
```
# Покупаем, если волатильность снижается, а объем и цена имеют восходящий тренд
HistoricalVolatility(CLOSE, 15, 252, 2) < REF(HistoricalVolatility(CLOSE, 15, 365, 2), 15)
AND
TREND(VOLUME, 5) = UP AND TREND(CLOSE, 40) = UP
```

Sell Signals

```
# Продаем, если волатильность снижается, а объем и цена имеют нисходящий тренд
HistoricalVolatility(CLOSE, 15, 252, 2) < REF(HistoricalVolatility(CLOSE, 15, 365, 2), 15)
AND
TREND(VOLUME, 5) = UP AND TREND(CLOSE, 40) = DOWN
```

Система Parabolic SAR/MA System

Эта система является вариантом стандартной системы пересечения скользящих средних. Обычно параболическая система используется только для получения сигналов выхода из позиции. В этой МТС, однако, мы используем пересечение двух ЕМА чтобы решить, должны ли мы покупать (продавать) всякий раз, когда Parabolic SAR пересекает снизу вверх цену закрытия.



После открытия позиции Parabolic SAR можно использовать в обычном контексте. Прибыль должна фиксироваться, когда цена закрытия пересекает Parabolic SAR.

Этот пример показывает, как использовать булевскую логику, чтобы найти акции, которые соответствуют условиям или текущей торговой сессии или предыдущего торгового дня.

Parabolic SAR / MA Script

Buy Signals

```
# Покупаем, если скользящие средние пересеклись сегодня или вчера и
# если PSAR пересеклись сегодня или вчера
(CROSSOVER(CLOSE, PSAR(0.02, 0.2)) OR
CROSSOVER(REF(CLOSE,1), PSAR(0.02, 0.2)))
AND
(CROSSOVER(EMA(CLOSE, 10), EMA(CLOSE, 20)) OR
CROSSOVER(REF(EMA(CLOSE, 10),1), REF(EMA(CLOSE, 20),1)))
```

Sell Signals

```
# Продаем, если скользящие средние пересеклись сегодня или вчера и
# если PSAR пересеклись сегодня или вчера
(CROSSOVER(PSAR(0.02, 0.2), CLOSE) OR
CROSSOVER(PSAR(0.02, 0.2), REF(CLOSE,1)))
AND
(CROSSOVER(EMA(CLOSE, 20), EMA(CLOSE, 10)) OR
CROSSOVER(REF(EMA(CLOSE, 20),1), REF(EMA(CLOSE, 10),1)))
```

Система MACD Momentum System

В этой торговой системе мы используем ЕМА и функцию TREND, чтобы идентифицировать рыночную инерцию, а также индикатор MACD, чтобы определить рыночный момент.



Как Вы, возможно, знаете, MACD отражает изменения силы быков и медведей. Если тренд индикатора MACD направлен вверх, это говорит, что на рынке преобладают быки, если тренд направлен вниз, это свидетельствует о преобладании медведей. Это известно как рыночный момент.

Эта система покупает, если и инерция (тренд EMA) и момент (MACD) идут в направлении повышающихся цен. Система продает, если при противоположной ситуации.

Сигнал на выход возникает, если тренд инерции и момента меняется на противоположный.

MACD Momentum Script

Buy Signals

Покупаем, если момент и инерция имеют однонаправленный тренд

**TREND(EMA(CLOSE, 20), 15) = UP AND
TREND(MACD(13, 26, 9, SIMPLE), 5) = UP**

Sell Signals

Продаем, если момент и инерция имеют однонаправленный тренд

**TREND(EMA(CLOSE, 20), 15) = DOWN AND
TREND(MACD(13, 26, 9, SIMPLE), 5) = DOWN**

Exit Long Signal

Выходим, если тренд инерции и момента имеет противоположное направления

**TREND(EMA(CLOSE, 20), 15) = DOWN OR
TREND(MACD(13, 26, 9, SIMPLE), 5) = DOWN**

Exit Short Signal

Выходим, если тренд инерции и момента имеет противоположное направления

**TREND(EMA(CLOSE, 20), 15) = UP OR
TREND(MACD(13, 26, 9, SIMPLE), 5) = UP**

Система Narrow Trading Range Breakout

Перевести название этой системы можно, как «прорыв суженного ценового диапазона». Акции, цены которых определенное время колебались в узком ценовом диапазоне, при прорыве диапазона часто продолжают движение в направлении этого прорыва. Например, если акции длительное время находились в диапазоне между \$40 и \$45, то при прорыве выше \$50 очень вероятно, что в обозримом будущем они продолжат свое движение вверх. Объяснением такого поведения может служить то, что чем дольше акции остаются в сжатом диапазоне, тем труднее его прорвать. В тоже время при возникновении прорыва, восходящий тренд должен продолжиться.



Narrow Trading Range Script

Находим акции, заключенные в 2% торговый диапазон за последние 50 дней

**MAX(CLOSE, 50) < CLOSE * 1.01 AND
MIN(CLOSE, 50) > CLOSE * 0.98 AND**

Индицируем прорыв

**CLOSE != REF(CLOSE, 1) AND
REF(CLOSE,1) != REF(CLOSE, 2) AND
REF(CLOSE,2) != REF(CLOSE, 3)**

Система Outside Day System

Внешний день (Outside Day) возникает, когда максимум текущего бара выше максимума предыдущего бара, а минимум текущего бара ниже минимума предыдущего бара. Цена закрытия должна быть противоположна текущему тренду (если тренд восходящий, то Close должен быть меньше, чем Open). *Внешние дни* возникают часто и могут использоваться как часть краткосрочной торговой стратегии.



Внешние дни, возникающие после продолжительного восходящего тренда, как показано на предыдущем рисунке, говорят о нерешительности рынка, и могут быть сигналом разворота или временной коррекции. В зависимости от направления рынка *внешние дни* могут быть или бычьими или медвежьими. Если признаки разворота возникают на уровнях сопротивления их можно расценить как медвежьи. Если они появляются на уровнях поддержки, то их можно отнести к бычьим.

Outside Day Script

Buy Signals

```
# Находим outside days
LOW < REF(LOW, 1) AND
HIGH > REF(HIGH, 1) AND
HIGH > REF(HIGH, 1) AND
CLOSE < OPEN AND
# Outside days более значимы, если
# предыдущий бар короче текущего
HIGH - LOW > (REF(HIGH, 1) - REF(LOW, 1)) * 1.5 AND
# Тренд должен быть восходящим
TREND(CLOSE, 30) = UP
```

Sell Signals

```
# Находим outside days
LOW < REF(LOW, 1) AND
HIGH > REF(HIGH, 1) AND
HIGH > REF(HIGH, 1) AND
CLOSE < OPEN AND
HIGH - LOW > (REF(HIGH, 1) - REF(LOW, 1)) * 1.5 AND
# Тренд должен быть нисходящим
TREND(CLOSE, 30) = DOWN
```

Система бычьего и медвежьего поглощения (Japanese Candlestick Engulfing Line System)

Бычье и медвежье поглощения совместно с анализом объема могут быть частью эффективной краткосрочной стратегии. Эти фигуры могут быть более надежными, если имеет место резкий рост объема. Фигуры поглощения часто сигнализируют о развороте тренда. Сама фигура поглощения состоит из короткой свечи, за которой следует свеча с длинным телом, которое полностью «поглощает» предыдущую (короткую) свечу.



Бычье поглощение указывает на потенциальный разворот нисходящего тренда, в то время как, Медвежье поглощение указывает на возможный разворот бычьего тренда.

Паттерны обладают большей предсказательной силой, если возникли на фоне значительного тренда.

Signals

```
# Бычий паттерн
CANDLESTICKPATTERN() = BULLISH_ENGULFING_LINE AND TREND(CLOSE, 30) =
DOWN AND VOLUME > REF(VOLUME, 1)

# Медвежий паттерн
CANDLESTICKPATTERN() = BEARISH_ENGULFING_LINE AND TREND(CLOSE, 30) = UP
AND VOLUME > REF(VOLUME, 1)
```

В главе 5 приведен полный список паттернов.

Глава 5. Примитивные переменные, константы и флаги

Примитивные типы

Эта глава является справочником по «примитивным» переменным, константам и флагам, таким как определение фундаментальных данных, свечным фигурам, флагам технических индикаторов, и констант секторов и отраслей промышленности.

Вектора цен

Базовые ценовые данные: Open, High, Low, Close и Volume могут быть использованы в качестве вектора в любой функции или выражения.

OPEN	цена открытия бара
HIGH	максимальная цена бара
LOW	минимальная цена бара
CLOSE	цена закрытия бара
LAST	цена последней сделки (вне торговой сессии эквивалентна CLOSE)
VOLUME	объем бара

Основные константы

TRUE	1
FALSE	0
PI	3.1415926535897932384626433832795
NULL	0 (пустой вектор)
ADD	1 (используется в функции LOOP)
SUBTRACT	2 (используется в функции LOOP)
MULTIPLY	3 (используется в функции LOOP)
DIVIDE	4 (используется в функции LOOP)

Флаги исторического тестирования

MAX_POSITION_OPEN Флаг используется при историческом тестировании. Значение флагу можно присвоить при помощи ключевого флага **SET**. Все позиции будут закрыты через специфицированное число дней независимо от наличия торговых сигналов: buy, sell или exit.

Например:

Позиция никогда не будет оставаться открытой более 20 дней

SET MAX_POSITION_OPEN = 20

Константы скользящих средних

SIMPLE	1
EXPONENTIAL	2
TIME_SERIES	3
VARIABLE	4
TRIANGULAR	5
WEIGHTED	6
VOLATILITY	7
WILDER	8

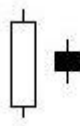
Константы функции Trend

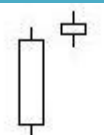
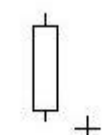
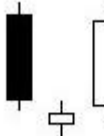
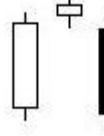
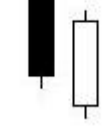
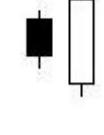
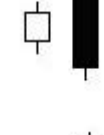
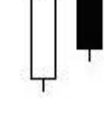
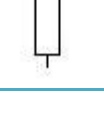
UP	1
DOWN	2
SIDEWAYS	3

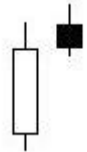
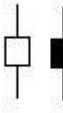
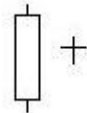
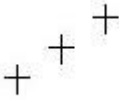
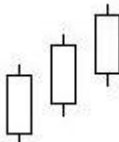
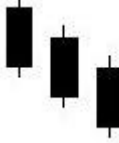
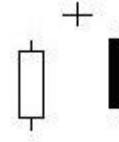
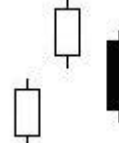
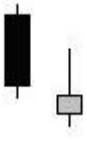
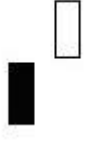
Константы Points и Percents (используются в индикаторах)

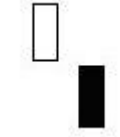
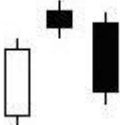
POINTS	1 (выходные значения измеряются в пунктах)
PERCENT	2 (выходные значения измеряются в процентах)

Константы свечных фигур

Константа (Русское наименование)		Вид
LONG_BODY (Длинная свеча)		
DOJI (Доджи)		
HAMMER (молот, см. висельник)		
HARAMI (харами)		

STAR (звезда)		
DOJI_STAR (звезда доджи)		
MORNING_STAR (утренняя звезда)		
EVENING_STAR (вечерняя звезда)		
PIERCING_LINE (проникающие линии)		
BULLISH_ENGULFING_LINE (бычье поглощение)		
BEARISH_ENGULFING_LINE (медвежье поглощение)		
DARK_CLOUD_COVER (завеса из облаков)		
HANGING_MAN (висельник, см. молот))		
BEARISH_DOJI_STAR		

BEARISH_SHOOTING_STAR (падающая звезда)		
SPINNING_TOPS		
HARAMI_CROSS (харами крест)		
BULLISH_TRISTAR (три птицы)		
THREE_WHITE_SOLDIERS (три белые вороны)		
THREE_BLACK_CROWS (три чёрные вороны)		
ABANDONED_BABY		
BULLISH_UPSIDE_GAP		
BULLISH_HAMMER (бычий молот)		
BULLISH_KICKING		

BEARISH_KICKING		
BEARISH_BELT_HOLD (медвежье удержание)		
BULLISH_BELT_HOLD (бычье удержание)		
BEARISH_TWO_CROWS		
BULLISH_MATCHING_LOW		

Глава 6. Решение проблем

Проблема

TradeScript™ возвращает ошибку:

“The variable name ---- is not recognized” or “The argument of ---- function is not optional”.

Решение

Чтобы исправить ошибку в скрипте, следуйте инструкциям на экране.

Проблема

TradeScript™ сообщает, что истекло время выполнения скрипта (script timeout).

Решение

Или сократите размеры Вашего скрипта, или обратитесь к поставщику программного обеспечения.

Проблема

Я хочу использовать TradeScript™ DLL для разработки торговой системы, используя стандартный язык программирования, типа VB6, VB.NET, C++ или Excel.

Решение

Посетите <http://www.modulusfe.com/tradescript/>, где можно получить информацию о лицензировании.