

海龟交易系统 系列讲座 上（一）

徐峰

内容提要

上篇：

❖ 海龟交易系统简介

❖ 海龟交易系统的规则与实现

试图简述海龟交易法则的背景，描述海龟交易交易系统的规则要点；
试图通过一步一步实现海龟交易系统，示范编写交易系统方法。

内容提要

中篇：

❖ 海龟交易系统的应用与分析

试图通过对海龟交易系统的应用操作，示范交易系统测试、优化与应用的方法；

试图分析原版海龟交易法则全貌，并通过对海龟交易系统的分析，体现海龟交易法则的核心要点，及其优点与不足。

内容提要

下篇：

❖ 海龟交易系统的创新与引申

试图通过对海龟交易系统的创新与引申，探索交易系统设计的思路及对交易理念与方法的思考。

目录

- ❖ 海龟交易系统简介
- ❖ 海龟交易系统的规则与实现

海龟实验

- ❖ 1983年年中，理查德·丹尼斯与他的老友比尔·埃克哈特进行了一场辩论，这场辩论是关于伟大的交易员是天生造就还是后天培养的。理查德相信，他可以教会人们成为伟大的交易员。比尔则认为遗传和天性才是决定因素。
- ❖ 为了解决这一问题，理查德建议招募并培训一些交易员，给他们提供真实的帐户进行交易，看看两个人中谁是正确的。
- ❖ 他们在《巴伦氏》、《华尔街期刊》和《纽约时报》上刊登了大幅广告，招聘交易学员。广告中称，在一个短暂的培训会后，新手将被提供一个帐户进行交易。
- ❖ 因为里克（理查德的昵称）或许是当时世界上最著名的交易员，所以，有1000多位申请人前来投奔他。他会见了其中的80位。

——《海龟交易法则》

海龟实验

- ❖ 这一群人精选出10个人，后来这个名单变成13个人，所增加的3个人里克以前就认识。1983年12月底，他们被邀请到芝加哥进行两周的培训，到1984年1月初，他们开始用小帐户进行交易。到了2月初，在他们证明了自己的能力之后，丹尼斯给他们中的大多数人提供了50万至200万美元的资金帐户。
- ❖ “学员们被称为‘海龟’（丹尼斯先生说这项计划开始时他刚刚从亚洲回来，他解释了自己向别人说过的话，‘我们正在成长为交易员，就象在新加坡他们正在成长为海龟一样’）。”
——斯坦利·W·安格瑞斯特，《华尔街期刊》，1989年9月5日
- ❖ 海龟成为交易史上最著名的实验。

——《海龟交易法则》

海龟交易法则的由来

- ❖ 在随后的4年中，海龟们取得了年均复利80%的收益。丹尼斯证明了用一套简单的系统和法则，可以使仅有很少或根本没有交易经验的人成为优秀的交易员。
- ❖ 当时，海龟们认为应对理查德·丹尼斯负责，商定甚至在他们议定的10年保密协定于1993年终止后也不泄漏这些法则。但是，有个别海龟在网站上出售海龟交易法则而谋取钱财。两个原版海龟柯蒂斯·费思和阿瑟·马多克为了阻止个别海龟对知识产权的偷盗和出售海龟交易法则而赚钱的行为，决定在网站上将海龟交易法则免费公之于众。我们现在能看到的海龟交易法则即是由此而来。

——《海龟交易法则》

目录

- ❖ 海龟交易系统简介
- ❖ 海龟交易系统的规则与实现

海龟交易系统规则提纲

注：为了由简到繁、由浅入深逐步描述规则并编写系统，顺序与原版不同

- 1.交易对象
- 2.入场
- 3.离场
- 4.N
- 5.止损
- 6.加仓
- 7.头寸规模

1.交易对象

- ❖ 多品种组合交易
- ❖ 首选流动性强的品种
- ❖ 不选容量小、趋势性不强的品种
- ❖ 不选有价格操纵隐患等人为风险的品种
- ❖ 一经不选，就不再选

原版海龟交易对象的范围

❖ 芝加哥期货交易所（CBOT）

30年期美国长期国债（Treasury Bond）、10年期美国中期国库券（Treasury Note）；

❖ 纽约咖啡可可与原糖交易所（NYCSC）

咖啡、可可、原糖、棉花；

❖ 芝加哥商品交易所（CME）

瑞士法郎、德国马克、英镑、法国法郎、日圆、加拿大元、标准普尔500股票指数、欧洲美元、90天美国短期国库券（Treasury Bill）；

❖ 纽约商品期货交易所（COMEX）

黄金、白银、铜；

❖ 纽约商业期货交易所（NYMEX）

原油、燃油、无铅汽油。

（注：原版海龟可以自行选择品种组合，但在交易中出于风险控制的考虑，会根据品种的⁴²相关性限制头寸规模，因而会影响到实际纳入组合的品种。在中篇另述。）

2.入场

系统1：以20日突破为基础的短期系统

系统2：以55日突破为基础的长期系统

❖ 系统1：

价格超越20日最高或最低价一个最小变动单位入场；

❖ 过滤条件：

若前一次价格的实际突破是“赢利性突破”，则不入场；无论前次“赢利性突破”是否有入场；

若前一次价格的实际突破是“亏损性突破”，则入场。

❖ 过滤后的入场条件：

若前一次突破被过滤，则以系统2长期系统的入场条件入场。

❖ 系统2：

无过滤。

2.入场

❖ 其它:

以指定价入场，非收盘价或次日开盘价；

指定价为突破价+一个最小变动单位；

若开盘跳过突破价，以开盘价入场。

将入场条件编为指标

可使用的函数与语法：

- ❖ HighestFC
- ❖ LowestFC
- ❖ PlotNumeric

HighestFC

附录D - 用户函数 - HighestFC

HighestFC

说明	求最高值(快速计算版本)
语法	Numeric HighestFC(NumericSeries Price, Numeric Length)
参数	Price 用于求最高值的值，必须是数值型序列值； Length 是需要计算的周期数，为整型，必须是常量。
备注	该函数计算指定周期内的数值型序列值的最高值，返回值为浮点数。
示例	HighestFC (Close, 12); 计算12周期以来的收盘价的最高值； HighestFC ((Close + High + Low)/ 3, 10); 计算10周期以来高低收价格的平均值的最高值。

LowestFC

附录D - 用户函数 - LowestFC

LowestFC

说明	求最低(快速计算版本)
语法	Numeric LowestFC (NumericSeries Price, Numeric Length)
参数	Price 用于求最低值的值，必须是数值型序列值； Length 是需要计算的周期数，为整型，必须是常量。
备注	该函数计算指定周期内的数值型序列值的最低值，返回值为浮点数。
示例	LowestFC (Close, 12); 计算12周期以来的收盘价的最低值； LowestFC ((Close + High + Low)/ 3, 10); 计算10周期以来高低收价格的平均值的最低值。

PlotNumeric

附录B - 系统函数 - PlotNumeric

PlotNumeric

说明	在当前Bar输出一个数值。
语法	Numeric PlotNumeric (String Name, Numeric Number, Numeric Locator=0, Integer Color=-1, Integer BarsBack=0)
参数	<p>Name 输出值的名称，不区分大小写；</p> <p>Number 输出的数值；</p> <p>Locator 输出值的定位点，默认时输出单点，否则输出连接两个值线段，用法请看例3；</p> <p>Color 输出值的显示颜色，默认表示使用属性设置框中的颜色；</p> <p>BarsBack 从当前Bar向前回溯的Bar数，默认值为当前Bar。</p>
备注	在当前Bar输出一个数值，输出的值用于在上层调用模块显示。返回数值型，即输入的Number。
示例	<p>例1: PlotNumeric ("RSI",RSIValue); 输出RSI的值。</p> <p>例2: If (PlotNumeric ("AR",ARValue)>180) Alert("AR超买"); 输出AR的值，并且当ARValue>180条件满足时，进行报警。</p> <p>例3: PlotNumeric ("OpenToClose",open,close); 输出开盘价与收盘价的连线。注意：公式输出线形选择柱状图。</p>

将入场条件编为指标

```
1  //-----
2  // 简称: TT_1
3  // 名称: 入场条件指标
4  // 类别: 公式应用
5  // 类型: 用户应用
6  // 输出:
7  //-----
8  Params
9      Numeric boLength(20);                // 短周期 BreakOut Length
10
11  Vars
12      NumericSeries DonchianHi;             // 唐奇安通道上轨, 延后1个Bar, 短周期
13      NumericSeries DonchianLo;            // 唐奇安通道下轨, 延后1个Bar, 短周期
14
15  Begin
16      DonchianHi = HighestFC(High[1],boLength);
17      DonchianLo = LowestFC(Low[1],boLength);
18
19      PlotNumeric("DonchianHi",DonchianHi);
20      PlotNumeric("DonchianLo",DonchianLo);
21
22  End
23  //-----
```

实现不加过滤的入场

可使用的函数与语法：

- ❖ MinMove
- ❖ PriceScale
- ❖ Min
- ❖ Max
- ❖ IIF
- ❖ Buy
- ❖ SellShort

MinMove

附录B - 系统函数 - MinMove

MinMove

说明	当前公式应用商品的最小变动量。
语法	Integer MinMove()
参数	无
备注	当前公式应用商品的最小变动量，返回值为整型。 $\text{MinMove} = \text{最小变动值} / \text{PriceScale}$ 。
示例	上海A股的最小变动值0.01元，其PriceScale = 0.01，因此其MinMove等于1； 沪铝的最小变动值为5，其PriceScale = 1，因此其MinMove等于5。

PriceScale

附录B - 系统函数 - PriceScale

PriceScale

说明	当前公式应用商品的计数单位。
语法	Numeric PriceScale()
参数	无
备注	当前公式应用商品的计数单位，返回值为浮点数。
示例	深圳A股报价精确到小数点2位，因此其PriceScale等于1/100； 上海B股报价精确到小数点3位，因此其PriceScale等于1/1000。

Min, Max

附录D - 用户函数 - Min

Min

说明	求最小值
语法	Numeric Min (Numeric Value1,Numeric Value2)
参数	Value1,Value2是数值型值。
备注	计算两个数之间的最小值。
示例	Min (open,open[1]);计算open和open[1]之间的最小值。

附录D - 用户函数 - Max

Max

说明	求最大值
语法	Numeric Max (Numeric Value1,Numeric Value2)
参数	Value1,Value2是数值型值
备注	计算两个数之间的最大值
示例	Max (close,close[1]);计算close和close[1]之间的最大值

IIF

附录B - 系统函数 - IIF

IIF

说明	执行真假值判断，根据逻辑测试的真假值返回不同的数值。
语法	Numeric IIF(Bool Conditon,Numeric TrueValue,Numeric FalseValue)
参数	Conditon 条件表达式； TrueValue 条件为True时的返回值； FalseValue 条件为False时的返回值。
备注	执行真假值判断，根据逻辑测试的真假值返回不同的数值，TrueValue/FalseValue都是浮点数，该函数返回值也为浮点数。
示例	IIF(Close>Open,Close,Open);

Buy

附录B - 系统函数 - Buy

Buy

说明	产生一个多头建仓操作。
语法	Bool Buy (Numeric Share=0, Numeric Price=0)
参数	Share 买入数量，为整型值，默认为使用系统设置参数； Price 买入价格，为浮点数，默认=0时为使用现价(非最后Bar为Close)。
备注	<p>产生一个多头建仓操作，返回值为布尔型，执行成功返回True，否则返回False。</p> <p>该函数仅用于多头建仓，其处理规则如下：</p> <p>如果当前持仓状态为持平，即MarketPosition = 0 时，该函数按照参数进行多头建仓。</p> <p>如果当前持仓状态为空仓，即MarketPosition = -1 时，该函数平掉所有空仓，同时按照参数进行多头建仓，两个动作同时发出。</p> <p>如果当前持仓状态为多仓，即MarketPosition = 1 时，该函数将继续建仓，但具体是否能够成功建仓要取决于系统中关于连续建仓的设置，以及资金，最大持仓量等限制。</p> <p>当委托价格超出k线的有效范围，在历史数据上，将会取最接近的有效价格发单；在实盘中，将会按照实际委托价格发单。</p> <p>例如：当前k线有效价格为50-100，用buy(1,10)发单，委托价将以50发单。</p>
示例	<p>在MarketPosition=0的情况下：</p> <p>Buy(50,10.2) 表示用10.2的价格买入50张合约。</p> <p>Buy(10,Close) 表示用当前Bar收盘价买入10张合约，马上发送委托。</p> <p>Buy(5,0) 表示用现价买入5张合约，马上发送委托。</p> <p>Buy(0,0) 表示用现价按交易设置中的设置,马上发送委托。</p>

SellShort

附录B - 系统函数 - SellShort

SellShort

说明	产生一个空头建仓操作。
语法	Bool SellShort (Numeric Share=0, Numeric Price=0)
参数	Share 卖出数量，为整型值，默认为使用系统设置参数； Price 卖出价格，为浮点数，默认=0时为使用现价(非最后Bar为Close)。
备注	<p>产生一个空头建仓操作，返回值为布尔型，执行成功返回True，否则返回False。</p> <p>该函数仅用于空头建仓，其处理规则如下：</p> <p>如果当前持仓状态为持平，即MarketPosition = 0 时，该函数按照参数进行空头建仓。</p> <p>如果当前持仓状态为多仓，即MarketPosition = 1 时，该函数平掉所有多仓，同时按照参数进行空头建仓，两个动作同时发出。</p> <p>如果当前持仓状态为空仓，即MarketPosition = -1 时，该函数将继续建仓，但具体是否能够成功建仓要取决于系统中关于连续建仓的设置，以及资金，最大持仓量等限制。</p> <p>当委托价格超出k线的有效范围，在历史数据上，将会取最接近的有效价格发单；在实盘中，将会按照实际委托价格发单。</p> <p>例如：当前k线有效价格为50-100，用SellShort(1,10)发单，委托价将以50发单。</p>
示例	<p>在MarketPosition=0的情况下：</p> <p>SellShort(50,10.2) 表示用10.2的价格空头卖出50张合约。</p> <p>SellShort(10,Close) 表示用当前Bar收盘价空头卖出10张合约，马上发送委托。</p> <p>SellShort(5,0) 表示用现价空头卖出5张合约，马上发送委托。</p> <p>SellShort(0,0) 表示用现价按交易设置中的设置，马上发送委托。</p>

实现不加过滤的入场

```
7 //-----
8 Params
9     Numeric boLength(20);                // 短周期 BreakOut Length
10
11 Vars
12     NumericSeries DonchianHi;            // 唐奇安通道上轨, 延后1个Bar, 短周期
13     NumericSeries DonchianLo;           // 唐奇安通道下轨, 延后1个Bar, 短周期
14     Numeric MinPoint;                   // 最小变动单位
15     Numeric myEntryPrice;               // 开仓价格
16
17 Begin
18     DonchianHi = HighestFC(High[1],boLength);
19     DonchianLo = LowestFC(Low[1],boLength);
20     PlotNumeric("DonchianHi",DonchianHi);
21     PlotNumeric("DonchianLo",DonchianLo);
22     MinPoint = MinMove*PriceScale;
23     If(High > DonchianHi )
24     {
25         // 开仓价格取突破上轨+一个价位和最高价之间的较小值, 这样能更接近真实情况, 并能尽量保证成交
26         myEntryPrice = min(high,DonchianHi + MinPoint);
27         myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 大跳空的时候用开盘价代替
28         Buy(1,myEntryPrice);
29     }
30     If(Low < DonchianLo )
31     {
32         // 开仓价格取突破下轨-一个价位和最低价之间的较大值, 这样能更接近真实情况, 并能尽量保证成交
33         myEntryPrice = max(low,DonchianLo - MinPoint);
34         myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 大跳空的时候用开盘价代替
35         SellShort(1,myEntryPrice);
36     }
37 End
```

3. 离场

- ❖ 系统1：短期系统，采用10日突破退出；
- ❖ 系统2：长期系统，采用20日突破退出。

实现不加过滤的入场出场

- ❖ 可使用的函数与语法：
- ❖ MarketPosition
- ❖ Commentary
- ❖ IF-Else
- ❖ IF-Else的嵌套

MarketPosition

附录B - 系统函数 - MarketPosition

MarketPosition

说明	获得当前持仓状态。
语法	Integer <code>MarketPosition()</code>
参数	无
备注	获得当前持仓状态，返回值为整型。 返回值定义如下： -1 当前位置为持空仓 0 当前位置为持平 1 当前位置为持多仓
示例	<code>if(MarketPosition==1)</code> 判断当前是否持多仓 <code>if(MarketPosition!=0)</code> 判断当前是否有持仓，无论持空仓或多仓

Commentary

附录B - 系统函数 - Commentary

Commentary

说明	在超级图表当前Bar添加一行注释信息。
语法	<code>Commentary(String strTip)</code>
参数	strTip 提示信息。
备注	在超级图表当前Bar添加一行注释信息，该函数无返回值。 作为系统调试的辅助工具，您可以通过在超级图表中选中某一个Bar，按下鼠标左键，察看注释信息。 启动显示提示框按钮，请查看 超级图表工具栏 。
示例	<code>Commentary("当前买入10手，价位1345");</code> <code>Commentary("开仓价格:"+Text(myEntryPrice))。</code>

IF-Else

If-Else

If-Else语句是对指定条件进行判断，如果条件满足执行If后的语句。否则执行Else后面的语句。

语法如下：

```
If (Condition)
{
    TradeBlazer公式语句1;
}Else
{
    TradeBlazer公式语句2;
}
```

Condition是一个逻辑表达式，当Condition为True的时候，TradeBlazer公式语句1将会被执行；Condition为False时，TradeBlazer公式语句2将会被执行。Condition可以是多个条件表达式的逻辑组合，Condition必须用()括起来。

TradeBlazer公式语句是一些语句的组合，如果TradeBlazer公式语句是单条，您可以省略{}，二条或者二条以上的语句必须使用{}。

IF-Else的嵌套

If-Else的嵌套

If-Else的嵌套是在If-Else的执行语句中包含新的条件语句，即一个条件被包含在另一个条件中。

语法如下：

```
If (Condition1)
{
    If (Condition2)
    {
        TradeBlazer公式语句1;
    }Else
    {
        TradeBlazer公式语句2;
    }
}Else
{
    If (Condition3)
    {
        TradeBlazer公式语句3;
    }Else
    {
        TradeBlazer公式语句4;
    }
}
```

实现不加过滤的入场出场（1）

```
9  Params
10      Numeric boLength(20);           // 短周期 BreakOut Length
11      Numeric teLength(10);          // 离市周期 Trailing Exit Length
12
13  Vars
14      NumericSeries DonchianHi;       // 唐奇安通道上轨, 延后1个Bar, 短周期
15      NumericSeries DonchianLo;       // 唐奇安通道下轨, 延后1个Bar, 短周期
16      Numeric MinPoint;               // 最小变动单位
17      Numeric myEntryPrice;           // 开仓价格
18      Numeric myExitPrice;            // 平仓价格
19      Numeric ExitHighestPrice;        // 离市时判断需要的N周期最高价
20      Numeric ExitLowestPrice;         // 离市时判断需要的N周期最低价
21
22  Begin
23      DonchianHi = HighestFC(High[1],boLength);
24      DonchianLo = LowestFC(Low[1],boLength);
25      PlotNumeric("DonchianHi",DonchianHi);
26      PlotNumeric("DonchianLo",DonchianLo);
27      MinPoint = MinMove*PriceScale;
28      ExitLowestPrice = LowestFC(Low[1],teLength);
29      ExitHighestPrice = HighestFC(High[1],teLength);
30
```

实现不加过滤的入场出场（2）

```
31  If(MarketPosition == 0) //未开仓情况下
32  {
33      If(High > DonchianHi )
34      {
35          // 开仓价格取突破上轨+一个价位和最高价之间的较小值，这样能更接近真实情况，并能尽量保证成交
36          myEntryPrice = min(high,DonchianHi + MinPoint);
37          myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 大跳空的时候用开盘价代替
38          Buy(1,myEntryPrice);
39      }
40      If(Low < DonchianLo )
41      {
42          // 开仓价格取突破下轨-一个价位和最低价之间的较大值，这样能更接近真实情况，并能尽量保证成交
43          myEntryPrice = max(low,DonchianLo - MinPoint);
44          myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 大跳空的时候用开盘价代替
45          SellShort(1,myEntryPrice);
46      }
47  }
48
```

实现不加过滤的入场出场（3）

```
49 If(MarketPosition == 1) //有多仓情况下
50 {
51     // 求出持多仓时离市的条件比较值
52     Commentary("ExitLowestPrice="+Text(ExitLowestPrice));
53     If(Low < ExitLowestPrice)
54     {
55         myExitPrice = max(Low,ExitLowestPrice - MinPoint);
56         myExitPrice = IIF(myExitPrice > Open, Open,myExitPrice); // 大跳空的时候用开盘价代替
57         Sell(0,myExitPrice); // 数量用0的情况下将全部平仓
58     }
59 }Else{
60     If(MarketPosition ==-1) // 有空仓的情况
61     {
62         // 求出持空仓时离市的条件比较值
63         Commentary("ExitHighestPrice="+Text(ExitHighestPrice));
64         If(High > ExitHighestPrice)
65         {
66             myExitPrice = Min(High,ExitHighestPrice + MinPoint);
67             myExitPrice = IIF(myExitPrice < Open, Open,myExitPrice); // 大跳空的时候用开盘价代替
68             BuyToCover(0,myExitPrice); // 数量用0的情况下将全部平仓
69         }
70     }
71 }
72 End
```


4.N

- ❖ N=真实波动幅度的20日指数移动平均值，即ATR。
- ❖ 真实波动幅度=市场在一天内的真实波动幅度，包括开盘跳空的情况在内。即， $\text{Max}(\text{High}-\text{Low}, \text{High}-\text{Close}[1], \text{Close}[1]-\text{Low})$

TrueRange

附录D - 用户函数 - TrueRange

TrueRange

说明	求真实范围
语法	Numeric TrueRange()
参数	无
备注	该函数计算当前周期的真实高点 and 真实低点的差值，返回值为浮点数。
示例	TrueRange(); 计算当前周期的真实高点 and 真实低点的差值。

XAverage

附录D - 用户函数 - XAverage

XAverage

说明	求指数平均
语法	Numeric XAverage (NumericSeries Price, Numeric Length)
参数	Price 用于求指数平均的值，必须是数值型序列值； Length 是需要计算的周期数，为整型。
备注	该函数计算指定周期内的数值型序列值的指数平均值，返回值为浮点数； 当序列值的CurrentBar小于Length时，该函数返回无效值。
示例	XAverage (Close, 12); 计算12周期以来的收盘价的指数平均值； XAverage ((Close + High + Low)/ 3, 10); 计算10周期以来高低收价格的平均值的指数平均值。

4.N

- ❖ N =真实波动幅度的20日指数移动平均值，即ATR。
- ❖ 真实波动幅度=市场在一天内的真实波动幅度，包括开盘跳空的情况在内。即， $\text{Max}(\text{High}-\text{Low}, \text{High}-\text{Close}[1], \text{Close}[1]-\text{Low})$
- ❖ $\text{TrueRange} = \text{TrueHigh} - \text{TrueLow}$
- ❖ $N = \text{AvgTR} = \text{XAverage}(\text{TrueRange}, \text{Length})$
- ❖ 由于TB公式中的ATR（指标）= $\text{Average}(\text{TrueRange}, \text{Length})$ ，即真实波动幅度一定周期的简单算术平均，为了与之区别，可将N命名为AvgTR

5.头寸规模

- ❖ 头寸规模=账户净值的1%/价值量波动性
- ❖ 价值量波动性=N*每点价值量

Portfolio_CurrentCapital

附录B - 系统函数 - Portfolio_CurrentCapital

Portfolio_CurrentCapital

说明	获得按当前Bar开盘价计算的可用资金。
语法	Numeric Portfolio_CurrentCapital()
参数	无
备注	获得当前的可用资金，已考虑交易费用，返回值为浮点数。
示例	无

Portfolio_UsedMargin

附录B - 系统函数 - Portfolio_UsedMargin

Portfolio_UsedMargin

说明	获得当前的持仓保证金。
语法	Integer Portfolio_UsedMargin()
参数	无
备注	无
示例	无

5.头寸规模

- ❖ 头寸规模=账户净值的1%/价值量波动性
- ❖ 价值量波动性=N*每点价值量
- ❖ 账户净值TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin()

ContractUnit

附录B - 系统函数 - ContractUnit

ContractUnit

说明	当前公式应用商品的每张合约包含的基本单位数量。
语法	Integer ContractUnit()
参数	无
备注	当前公式应用商品的每张合约包含的基本单位数量，返回值为整型。对于股票来说，是1手包含多少股，对于铜铝商品期货来说是1张合约包含多少吨标底物。
示例	无

BigPointValue

附录B - 系统函数 - BigPointValue

BigPointValue

说明	当前公式应用商品数据的一个整数点的价值。
语法	Numeric BigPointValue()
参数	无
备注	当前公式应用商品数据的一个整数点的价值，默认值为1。
示例	深圳证券交易所A股一个点的价值都为1人民币，因此其BigPointValue等于1； 郑州期货交易所期货品种一个点的价值为1人民币，因此其BigPointValue等于1； 香港联交所恒指期货一个点的价值为50港币，因此其BigPointValue等于50； 芝加哥商业交易所E-MINI一个点的价值为50美元，因此其BigPointValue等于50； 上海证券交易所上证指数为不可交易品种，因此其BigPointValue也等于1。

5.头寸规模

- ❖ 头寸规模=账户净值的1%/价值量波动性
- ❖ 价值量波动性=N*每点价值量
- ❖ 账户净值TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin()
- ❖ 价值量波动性VN=N * ContractUnit()*BigPointValue()

IntPart

附录B - 系统函数 - IntPart

IntPart

说明	返回实数舍入后的整数值。
语法	Integer IntPart(Numeric Number)
参数	Number 需要进行取整处理的实数。
备注	返回实数舍入后的整数值。
示例	<code>IntPart (8.9) = 8;</code> <code>IntPart (-8.9) = -9。</code>

FileAppend

附录B - 系统函数 - FileAppend

FileAppend

说明	在指定文件中追加一行字符串。
语法	Bool FileAppend(String strPath,String strText)
参数	strPath 指定文件的路径，请使用全路径表示，并使用\\做路径分割符，否则会执行失败 strText 输出的字符串内容
备注	在指定文件中追加一行字符串，返回值为布尔型。 执行成功返回True，执行失败返回False。
示例	FileAppend("C:\\Formula.log","Close = "+Text(Close));

5.头寸规模

- ❖ 头寸规模=账户净值的1%/价值量波动性
- ❖ 价值量波动性=N*每点价值量
- ❖ 账户净值TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin()
- ❖ 价值量波动性VN=N * ContractUnit()*BigPointValue()
- ❖ 头寸规模TurtleUnits = (TotalEquity*RiskRatio/100) /VN)
- ❖ 取整，头寸规模TurtleUnits = IntPart(TurtleUnits)
- ❖ FileAppend("C:\\Formula.log","TurtleUnits="+Text(TurtleUnits))

实现带有头寸规模的无过滤入场出场（1）

```

9  Params
10     Numeric boLength(20);           // 短周期 BreakOut Length
11     Numeric teLength(10);          // 离市周期 Trailing Exit Length
12     Numeric ATRLength(20);         // 平均波动周期 ATR Length
13     Numeric RiskRatio(1);          // % Risk Per N ( 0 - 100)
14
15  Vars
16     NumericSeries DonchianHi;       // 唐奇安通道上轨, 延后1个Bar, 短周期
17     NumericSeries DonchianLo;       // 唐奇安通道下轨, 延后1个Bar, 短周期
18     Numeric MinPoint;               // 最小变动单位
19     Numeric myEntryPrice;            // 开仓价格
20     Numeric myExitPrice;             // 平仓价格
21     Numeric ExitHighestPrice;        // 离市时判断需要的N周期最高价
22     Numeric ExitLowestPrice;         // 离市时判断需要的N周期最低价
23     NumericSeries AvgTR;             // ATR
24     Numeric N;                       // N 值
25     Numeric VN;                      // 价值量波动性
26     Numeric TotalEquity;             // 按最新收盘价计算出的总资产
27     Numeric TurtleUnits;            // 交易单位
28
29  Begin
30     DonchianHi = HighestFC(High[1],boLength);
31     DonchianLo = LowestFC(Low[1],boLength);
32     PlotNumeric("DonchianHi",DonchianHi);
33     PlotNumeric("DonchianLo",DonchianLo);
34     MinPoint = MinMove*PriceScale;
35     ExitLowestPrice = LowestFC(Low[1],teLength);
36     ExitHighestPrice = HighestFC(High[1],teLength);
37
38     AvgTR = XAverage(TrueRange,ATRLength);
39     N = AvgTR[1];

```


实现带有头寸规模的无过滤入场出场（2）

```
40 UN = N * ContractUnit()*BigPointValue();
41 TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin();
42 TurtleUnits = ((TotalEquity * RiskRatio/100) / UN );
43 TurtleUnits = IntPart(TurtleUnits); // 对小数取整
44 FileAppend("C:\\Formula.log","TurtleUnits="+Text(TurtleUnits));
45
46 If(MarketPosition == 0) //未开仓情况下
47 {
48     If(High > DonchianHi && TurtleUnits >= 1)
49     {
50         // 开仓价格取突破上轨+一个价位和最高价之间的较小值，这样能更接近真实情况，并能尽量保证成交
51         myEntryPrice = min(high,DonchianHi + MinPoint);
52         myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 大跳空的时候用开盘价代替
53         Buy(TurtleUnits,myEntryPrice);
54     }
55     If(Low < DonchianLo && TurtleUnits >= 1)
56     {
57         // 开仓价格取突破下轨-一个价位和最低价之间的较大值，这样能更接近真实情况，并能尽量保证成交
58         myEntryPrice = max(low,DonchianLo - MinPoint);
59         myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 大跳空的时候用开盘价代替
60         SellShort(TurtleUnits,myEntryPrice);
61     }
62 }
63
```

实现带有头寸规模的无过滤入场出场（3）

```
64 If(MarketPosition == 1) //有多仓情况下
65 {
66     // 求出持多仓时离市的条件比较值
67     Commentary("ExitLowestPrice="+Text(ExitLowestPrice));
68     If(Low < ExitLowestPrice)
69     {
70         myExitPrice = max(Low,ExitLowestPrice - MinPoint);
71         myExitPrice = IIF(myExitPrice > Open, Open,myExitPrice); // 大跳空的时候用开盘价代替
72         Sell(0,myExitPrice); // 数量用0的情况下将全部平仓
73     }
74 }Else{
75     If(MarketPosition ==-1) // 有空仓的情况
76     {
77         // 求出持空仓时离市的条件比较值
78         Commentary("ExitHighestPrice="+Text(ExitHighestPrice));
79         If(High > ExitHighestPrice)
80         {
81             myExitPrice = Min(High,ExitHighestPrice + MinPoint);
82             myExitPrice = IIF(myExitPrice < Open, Open,myExitPrice); // 大跳空的时候用开盘价代替
83             BuyToCover(0,myExitPrice); // 数量用0的情况下将全部平仓
84         }
85     }
86 }
87 End
88
```

6.加仓

- ❖ 首先在突破点建立1个单位的头寸，然后按0.5N的价格间隔一步一步扩大头寸。
- ❖ 这个0.5N的间隔以实际成交价为基础。
- ❖ 1天内允许加满4个头寸单位。

BarStatus

附录B - 系统函数 - BarStatus

BarStatus

说明	当前公式应用商品当前Bar的状态值。
语法	Integer BarStatus()
参数	无
备注	当前公式应用商品当前Bar的状态值，返回值0表示为第一个Bar，返回值为1表示为中间的普通Bar，返回值为2表示最后一个Bar。
示例	<code>BarStatus==0</code> 表示第一个Bar <code>BarStatus==2</code> 表示最后一个Bar <code>BarStatus==1</code> 表示第一个Bar和最后一个Bar之间的所有Bar

while

While

While语句在条件为真的时候重复执行某一项操作。即，只要条件表达式的值为真(True)时，就重复执行某个动作。直到行情信息改变以致条件为假(False)时，循环才结束。

语法如下：

```
While (Condition)
{
    TradeBlazer公式语句;
}
```

Condition是一个逻辑表达式，当Condition为True的时候，TradeBlazer公式语句将会被循环执行，Condition可以是多个条件表达式的逻辑组合，Condition必须用()括起来。

TradeBlazer公式语句是一些语句的组合，如果TradeBlazer公式语句是单条，您可以省略{}，二条或者二条以上的语句必须使用{}。

例如，以下的公式用来计算要产生大于100000成交量需要最近Bar的个数：

```
Vars
    Numeric      SumVolume(0);
    Numeric      Counter (0);
Begin
    While (SumVolume < 100000)
    {
        SumVolume = SumVolume + Vol[Counter]
        Counter = Counter + 1;
    }
End
```

7. 止损

- ❖ 任何一笔交易都不能出现2%以上的风险。
因为价格波动1N表示1%的帐户净值，容许风险为2%的最大止损就是价格波动2N。
- ❖ 为了保证全部头寸的风险最小，如果另外增加单位，前面单位的止损就提高1/2N。
- ❖ 在后面单位因市场波动太快造成“打滑（skid）”或者因开盘跳空而以较大的间隔设置的情况下，止损就有所不同。

8.过滤

- ❖ 过滤条件:

若前一次价格的实际突破是“赢利性突破”，则不入场；无论前次“赢利性突破”是否有入场；

若前一次价格的实际突破是“亏损性突破”，则入场。

- ❖ 过滤后的入场条件:

若前一次突破被过滤，则以系统2长期系统的入场条件入场。

谢谢！