

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2023 年 10 月 26 日 (26.10.2023)



(10) 国际公布号
WO 2023/202149 A1

(51) 国际专利分类号:

H03M 7/42 (2006.01)

(21) 国际申请号:

PCT/CN2022/142929

(22) 国际申请日:

2022 年 12 月 28 日 (28.12.2022)

(25) 申请语言:

中文

(26) 公布语言:

中文

(30) 优先权:

202210413401.4 2022年4月20日 (20.04.2022) CN

(71) 申请人: 苏州浪潮智能科技有限公司

(**INSUR SUZHOU INTELLIGENT TECHNOLOGY CO., LTD.**) [CN/CN]; 中国江苏省苏州市吴中区吴中经济开发区郭巷街道官浦路1号9幢, Jiangsu 215000 (CN)。

(72) 发明人: 张永兴 (**ZHANG, Yongxing**); 中国江苏省

苏州市吴中区吴中经济开发区郭巷街道官浦路1号9幢, Jiangsu 215000 (CN)。陈静静 (**CHEN, Jingjing**); 中国江苏省苏州市吴中区吴中经济开发区郭巷街道官浦路1号9幢, Jiangsu 215000 (CN)。

吴睿振 (**WU, Ruizhen**); 中国江苏省苏州市吴中区吴中经济开发区郭巷街道官浦路1号9幢, Jiangsu 215000 (CN)。孙华锦 (**SUN, Huajin**); 中国江苏省苏州市吴中区吴中经济开发区郭巷街道官浦路1号9幢, Jiangsu 215000 (CN)。王凇 (**WANG, Lin**); 中国江苏省苏州市吴中区吴中经济开发区郭巷街道官浦路1号9幢, Jiangsu 215000 (CN)。

(74) 代理人: 北京市万慧达律师事务所 (**BEIJING**

WANHUIDA LAW FIRM); 中国北京市海淀区中关村南大街1号友谊宾馆颐园写字楼2楼, Beijing 100873 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家

保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE,

(54) **Title:** STATE SELECTION METHOD AND SYSTEM FOR FINITE STATE ENTROPY ENCODING, AND STORAGE MEDIUM AND DEVICE

(54) 发明名称: 有限状态熵编码的状态选择方法、系统、存储介质及设备

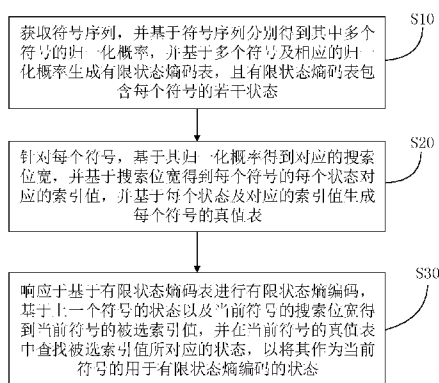


图 1

- S10 Acquire a symbol sequence, obtain, on the basis of the symbol sequence, normalization probabilities of a plurality of symbols in the symbol sequence, and generate a finite state entropy code table on the basis of the plurality of symbols and the corresponding normalization probabilities, wherein the finite state entropy code table includes several states of each symbol
- S20 For each symbol, obtain a corresponding search bit width on the basis of the normalization probability thereof, obtain, on the basis of the search bit width, an index value corresponding to each state of each symbol, and generate a truth table of each symbol on the basis of each state and the corresponding index value
- S30 In response to performing finite state entropy encoding on the basis of the finite state entropy code table, obtain a selected index value of the current symbol on the basis of the state of the previous symbol and the search bit width of the current symbol, and search a truth table of the current symbol for a state corresponding to the selected index value to take same as a state of the current symbol that is used for finite state entropy encoding

(57) **Abstract:** Provided in the present application are a state selection method and system for finite state entropy encoding, and a storage medium and a device. The method comprises: acquiring a symbol sequence, obtaining normalization probabilities of a plurality of symbols in the symbol sequence, and generating a finite state entropy code table on the basis of the plurality of symbols and the corresponding normalization probabilities, wherein the finite state entropy code table includes several states of each symbol; for each symbol, obtaining a corresponding search bit width on the basis of the normalization probability thereof, obtaining, on the basis of the search bit width, an index value corresponding to each state of each symbol, and generating a truth table of each symbol on the basis of each state and the corresponding index value; and if finite state entropy encoding is performed on the basis of the finite state entropy code table, obtaining a selected index value of the current symbol on the basis of the state of the previous symbol and the search bit width of the current symbol, and searching a truth table of the current symbol for a state corresponding to the selected index value to take same as a state of the current symbol that is used for finite state entropy encoding. By means of the present application, the state selection efficiency is greatly improved.

SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA,
UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚
(AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE,
BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR,
HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO,
PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF,
CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN,
TD, TG)。

本国际公布:
— 包括国际检索报告(条约第21条(3))。

(57) 摘要: 本申请提供了一种有限状态熵编码的状态选择方法、系统、存储介质及设备, 方法包括: 获取符号序列, 并得到其中多个符号的归一化概率, 并基于多个符号及相应的归一化概率生成有限状态熵码表, 且有限状态熵码表包含每个符号的若干状态; 针对每个符号, 基于其归一化概率得到对应的搜索位宽, 并基于搜索位宽得到每个符号的每个状态对应的索引值, 并基于每个状态及对应的索引值生成每个符号的真值表; 若基于有限状态熵码表进行有限状态熵编码, 基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值, 并在当前符号的真值表中查找被选索引值所对应的状态, 以将其作为当前符号的用于有限状态熵编码的状态。本申请大大提高了状态选择效率。

有限状态熵编码的状态选择方法、系统、存储介质及设备

相关申请的交叉引用

本申请要求于 2022 年 4 月 20 日提交中国专利局，申请号为 202210413401.4，申请名称为“有限状态熵编码的状态选择方法、系统、存储介质及设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及数据编解码技术领域，尤其涉及一种有限状态熵编码的状态选择方法、系统、非易失性计算机可读存储介质及计算机设备。

背景技术

随着大数据等前沿科学技术的快速发展，催生了数据爆发式的增长，海量数据对现有的存储设备带来巨大的压力。面对持续增长的海量数据，数据压缩成为减轻服务器存储负担、降低存储成本的有效方法。

数据压缩在不丢失有用信息的前提下，缩减数据量以减少存储空间，提高传输、存储和处理效率。无损数据压缩一般通过两种方法来实现：一种是通过字典的方式实现压缩的算法，包括 LZ 系列算法，这类算法能实现重复数据的搜索功能；另一种是基于统计模型的压缩算法，如 Huffman 码（哈夫曼编码，一种常见的变长编码）、算术编码等，这类算法的核心思想是依照符号出现频率分配码长，符号频率越大，码长就越短。

哈夫曼编码简单高效，也能为单个数据符号生成最佳的码字，然而对于给定的符号集来说，其并非总是生成最有效的码字。算术编码可以解决 Huffman 编码的这个痛点，但是算术编码算法实现复杂，且编码效率很低（大致是 Huffman 编码的 1/10），在数据压缩领域无法运用。正因如此，近些年，好多学者都在寻找一个“压缩率接近算术编码，且编码效率接近 Huffman 编码”的新型算法。

FSE（Finite State Entropy，有限状态熵）编码算法是一种结合 Huffman 编码及算术编码的优点的熵编码方案。基于 FSE 还推出一种新型数据压缩算法 Zstandard(zstd)，它是一种由 LZ77 编码、Huffman 编码和 FSE 组成的混合压缩算法，具有更好的压缩性能。

FSE 编码本质上是利用状态编码符号，即将符号的编码转换为状态编码。FSE 在进行符号序列编码时，需要根据前一个符号的状态（state）选择下一个符号的状态，选择的依据为：前一个符号的状态必须是在后一个符号的状态的覆盖范围（State_range）之内。一个符号出现的概率较大，该符号对应的状态会有多个，FSE 编码该符号时需要从它对应的所有状态中选择一个。

在 Zstandard 参考代码中，给出了一种遍历搜索的状态选择方案。假设前一个符号选择的

state_0, 下一个符号为 s, s 对应的所有状态为 State[]。该算法方案需要依次分析符号所对应的每个状态的覆盖范围, 直到找到一个 state_0 所在范围的状态。状态的搜索次数, 与前一个符号的状态有关, 如用硬件实现, 会有如下弊端: 对于特定的符号 s, 符号序列中会有多个 s 符号, 位置不同, 搜索次数会有不同。特定符号 s 的最大搜索次数由其归一化频率 (norm[s]) 决定, 但是对于符号全集, 每个符号的最大搜索次数也会不同。

搜索次数直接对应硬件电路的时钟周期, 硬件电路往往要求统一的时钟周期, 所以上述“搜到即止”的方案无法用于硬件实现。因此, 需要一种对硬件友好的状态搜索方法。

发明内容

本申请提供了一种有限状态熵编码的状态选择方法, 包括以下步骤:

获取符号序列, 并基于符号序列分别得到其中多个符号的归一化概率, 并基于多个符号及相应的归一化概率生成有限状态熵码表, 且有限状态熵码表包含每个符号的若干状态;

针对每个符号, 基于其归一化概率得到对应的搜索位宽, 并基于搜索位宽得到每个符号的每个状态对应的索引值, 并基于每个状态及对应的索引值生成每个符号的真值表;

响应于基于有限状态熵码表进行有限状态熵编码, 基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值, 并在当前符号的真值表中查找被选索引值所对应的状态, 以将其作为当前符号的用于有限状态熵编码的状态。以及

在一些实施例中, 基于搜索位宽得到每个符号的每个状态对应的索引值包括:

基于多个符号的归一化概率得到有限状态熵码表的码表长度, 并基于码表长度得到有限状态熵码表的码表位宽; 以及

针对每个符号的每个状态, 基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值。

在一些实施例中, 针对每个符号的每个状态, 基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值步骤之前, 还包括:

获取每个符号的每个状态的状态取值范围; 以及

根据状态取值范围设置对应的状态区域。

在一些实施例中, 针对每个符号的每个状态, 基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值步骤之前, 还包括:

依次根据多个符号对应的归一化概率, 确定每个符号的状态区域的长度取值。

在一些实施例中, 依次根据多个符号对应的归一化概率, 确定每个符号的状态区域的长度取值步骤, 包括:

依次确定每个符号对应的归一化概率是否为预设值;

响应于归一化概率为预设值, 将对应的符号的状态区域的长度取值设置为唯一取值; 或

响应于归一化概率不为预设值, 将对应的符号的状态区域的长度取值设置为两个取值。

在一些实施例中，针对每个符号的每个状态，基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值包括：

计算码表位宽与每个符号的搜索位宽的差值，得到第一移动位数；以及

基于第一移动位数对每个符号的每个状态的基线状态值进行右移运算，得到每个状态对应的索引值。

在一些实施例中，基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值包括：

计算码表位宽与当前符号的搜索位宽的差值，得到第二移动位数；以及

基于第二移动位数对上一个符号的状态的值进行右移运算，得到当前符号的被选索引值。

在一些实施例中，针对每个符号，基于其归一化概率得到对应的搜索位宽包括：

基于每个符号的归一化概率以及底数 2 构造对数函数，对对数函数求解，得到每个符号对应的搜索位宽。

在一些实施例中，基于多个符号及相应的归一化概率生成有限状态熵码表，且有限状态熵码表包含每个符号的若干状态包括：

基于多个符号的归一化概率得到有限状态熵码表的码表长度；

基于码表长度得到有限状态熵码表中的所有状态；

将每个符号的归一化概率的值作为相应的状态数目；

基于所有状态和每个符号的状态数目得到每个符号的若干状态；以及

基于每个符号的若干状态生成有限状态熵码表。

在一些实施例中，基于多个符号的归一化概率得到有限状态熵码表的码表长度步骤，包括：

获取多个符号的归一化概率的总和；以及

根据总和获取有限状态熵码表的码表长度。

在一些实施例中，基于符号序列分别得到其中多个符号的归一化概率包括：

基于符号序列分别得到其中多个符号的出现频率；以及

基于每个符号的出现频率得到每个符号的归一化概率。

在一些实施例中，基于每个符号的出现频率得到每个符号的归一化概率步骤之后，还包括：

确定多个符号中是否存在相同的归一化概率；以及

响应于存在相同的归一化概率，采用不同的标识对相同的归一化概率进行标识。

在一些实施例中，获取符号序列，并基于符号序列分别得到其中多个符号的归一化概率，并基于多个符号及相应的归一化概率生成有限状态熵码表，且有限状态熵码表包含每个符号的若干状态步骤之前，还包括：

确定符号序列是否为预设符号；以及

响应于符号序列为预设符号，则基于符号序列分别得到其中多个符号的归一化概率。

在一些实施例中，预设符号包括：原文节段长度 `literal_length`，重复长度 `match_length`，和偏

移距离 offset。

在一些实施例中，获取符号序列，并基于符号序列分别得到其中多个符号的归一化概率，并基于多个符号及相应的归一化概率生成有限状态熵码表，且有限状态熵码表包含每个符号的若干状态步骤，包括：

根据有限状态熵码表的码表长度和归一化概率，依次对多个符号进行状态分配。

在一些实施例中，根据有限状态熵码表的码表长度和归一化概率，依次对多个符号进行状态分配步骤，包括：

根据有限状态熵码表的码表长度和归一化概率，采用状态倒序对多个符号进行状态分配。

本申请的另一个或多个方面，还提供了一种有限状态熵编码的状态选择系统，包括：

有限状态熵码表生成模块，配置用于获取符号序列，并基于符号序列分别得到其中多个符号的归一化概率，并基于多个符号及相应的归一化概率生成有限状态熵码表，且有限状态熵码表包含每个符号的若干状态；

真值表生成模块，配置用于针对每个符号，基于其归一化概率得到对应的搜索位宽，并基于搜索位宽得到每个符号的每个状态对应的索引值，并基于每个状态及对应的索引值生成每个符号的真值表；以及

状态选择模块，配置用于响应于基于有限状态熵码表进行有限状态熵编码，基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值，并在当前符号的真值表中查找被选索引值所对应的状态，以将其作为当前符号的用于有限状态熵编码的状态。

在一些实施例中，真值表生成模块包括：

长度和位宽获取子模块，用于基于多个符号的归一化概率得到有限状态熵码表的码表长度，并基于码表长度得到有限状态熵码表的码表位宽；以及

索引值获取子模块，用于针对每个符号的每个状态，基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值。

本申请的又一个或多个方面，还提供了一种非易失性计算机可读存储介质，非易失性计算机可读存储介质存储有计算机可读指令，计算机可读指令被至少一个处理器执行时，使得至少一个处理器执行如上任意方法的步骤。

本申请的再一个或多个方面，还提供了一种计算机设备，包括

至少一个处理器；以及

至少一个存储器，至少一个存储器存储有计算机可读指令，计算机可读指令被至少一个处理器执行时，使得至少一个处理器执行时执行如上任意的的方法。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施

例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的实施例。

图 1 为根据本申请实施例提供的有限状态熵编码的状态选择方法的示意图；

图 2 为根据本申请实施例提供的有限状态熵编码的状态选择系统的示意图；

图 3 为根据本申请实施例提供的实现有限状态熵编码的状态选择方法的非易失性计算机可读存储介质的示意图；

图 4 为根据本申请实施例提供的执行有限状态熵编码的状态选择方法的计算机设备的硬件结构示意图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚明白，以下结合具体实施例，并参照附图，对本申请实施例进一步详细说明。

需要说明的是，本申请实施例中所有使用“第一”和“第二”的表述均是为了区分两个相同名称的非相同的实体或者非相同的参量，可见“第一”“第二”仅为了表述的方便，不应理解为对本申请实施例的限定。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备固有的其他步骤或单元。

基于上述目的，本申请实施例的第一个或多个方面，提出了一种有限状态熵编码的状态选择方法的实施例。图 1 示出的是本申请提供的有限状态熵编码的状态选择方法的实施例的示意图。如图 1 所示，本申请实施例包括如下步骤：

步骤 S10、获取符号序列，并基于符号序列分别得到其中多个符号的归一化概率，并基于多个符号及相应的归一化概率生成有限状态熵码表，且有限状态熵码表包含每个符号的若干状态；

步骤 S20、针对每个符号，基于其归一化概率得到对应的搜索位宽，并基于搜索位宽得到每个符号的每个状态对应的索引值，并基于每个状态及对应的索引值生成每个符号的真值表；

步骤 S30、响应于基于有限状态熵码表进行有限状态熵编码，基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值，并在当前符号的真值表中查找被选索引值所对应的状态，以将其作为当前符号的用于有限状态熵编码的状态。

本申请实施例通过构建有限状态熵码表，并针对有限状态熵码表中的每个符号，基于其归一化概率得到对应的搜索位宽，并基于搜索位宽得到每个符号的每个状态对应的索引值，并基于每个状态及对应的索引值生成每个符号的真值表，以辅助有限状态熵编码时对每个符号从若干状态中选择一个状态来编码；在选择状态时，通过基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值，并在当前符号的真值表中查找被选索引值所对应的状态，以将其作为当前符号的用于有限状态熵编码的状态，从而形成与硬件电路的时钟周期相适配的状态搜索方式，并且相比于现有的依次搜索的方式，大大提高了状态搜索、选择的效率。

本申请方案适用于硬件实现，可以提高有限状态熵编码数据时的状态搜索流程，进而整个加

速有限状态熵编码的速率。

在一些实施例中，基于搜索位宽得到每个符号的每个状态对应的索引值包括：基于多个符号的归一化概率得到有限状态熵码表的码表长度，并基于码表长度得到有限状态熵码表的码表位宽；针对每个符号的每个状态，基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值。

在一些实施例中，针对每个符号的每个状态，基于其状态区域的基线位置的基线状态值、搜索位宽以及码表位宽得到每个状态对应的索引值包括：计算码表位宽与每个符号的搜索位宽的差值，得到第一移动位数；基于第一移动位数对每个符号的每个状态的基线状态值进行右移运算，得到每个状态对应的索引值。

在一些实施例中，基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值包括：计算码表位宽与当前符号的搜索位宽的差值，得到第二移动位数；基于第二移动位数对上一个符号的状态的值进行右移运算，得到当前符号的被选索引值。

在一些实施例中，针对每个符号，基于其归一化概率得到对应的搜索位宽包括：基于每个符号的归一化概率以及底数 2 构造对数函数，对对数函数求解，得到每个符号对应的搜索位宽。

在一些实施例中，基于多个符号及相应的归一化概率生成有限状态熵码表，且有限状态熵码表包含每个符号的若干状态包括：基于多个符号的归一化概率得到有限状态熵码表的码表长度；基于码表长度得到有限状态熵码表中的所有状态；将每个符号的归一化概率的值作为相应的状态数目；基于所有状态和每个符号的状态数目得到每个符号的若干状态；基于每个符号的若干状态生成有限状态熵码表。

在一些实施例中，基于符号序列分别得到其中多个符号的归一化概率包括：基于符号序列分别得到其中多个符号的出现频率；基于每个符号的出现频率得到每个符号的归一化概率。

以下为本申请的有限状态熵编码的状态选择的具体实施例：

Zstandard（快速实时压缩算法）协议的符号（Symbols）是指源数据经过 LZ77 算法搜索查重的输出符号，总计有 4 类符号：literals（原文字母），literal_length（原文节段长度），match_length（重复长度），offset（偏移距离，重复数据之间的距离）。其中 literals 采用 Huffman（哈夫曼）编码，其余三类符号采用 FSE（Finite State Entropy，有限状态熵）编码。所以 Zstandard 压缩算法中需要用到 3 个 FSE 码表（有限状态熵码表），分别适用于 literal-length、match-length、offset 这三类符号。对这 3 类符号独立统计，得到三个概率分布表。

FSE 码表的码表长度为 2 的整数次幂，每个状态对应一个符号。FSE 码表中每个符号对应状态数与该符号的归一化概率相关，因此需要对符号的频率进行归一化（Normalize）生成概率表。归一化计算公式大致如下：

$$norm(x) = round \left(\frac{count(x)}{total_count} * table_size \right) \quad (1)$$

某符号的归一化概率与其出现频次占总数的比例相关。

归一化算法需要满足以下两个原则：(1)归一化概率总和必须等于 Table_size（码表长度）；
(2)某个符号只要出现，归一化后的概率至少为 1。

假设源符号的总数为 1024，FSE 码表的码表长度为 32，某符号 s1 的出现次数为 32，符号 s2 的出现次数为 2；两个符号的归一化概率可以由公式（1）计算得到：

$$\begin{aligned} \text{norm}(s_1) &= \text{round} \left(\frac{32}{1024} * 32 \right) = 1 \\ \text{norm}(s_2) &= \text{round} \left(\frac{2}{1024} * 32 \right) = 0 \end{aligned}$$

由于归一化原则(2)的要求，需要将 norm(s2)的值设置为 1。Zstandard 协议中为了区别是 s1、s2 代表的两种归一化概率等于 1 的情形，将 s2 情形的归一化概率的用“-1”标识，s1 的情形用“1”标识。

以下为构建 FSE 码表的方法：

FSE 码表是 FSE 编码及解码用的工具，它是 FSE 编码的基石。FSE 码表具有三个特点：FSE 码表的长度为 2 的整数次幂；FSE 码表中每个状态会对应一个符号；每个符号对应的状态数目等于归一化概率。

首先为归一化概率为“-1”的符号分配状态，从状态（Table_size-1）倒序分配。伪代码如下：

```
end_state = Table_size - 1;
for(symbol = 0; symbol < max; symbol++)
{
    if(-1 == norm(symbol))
        FSE[end_state--].symbol = symbol;
}
```

然后为其他符号分配状态，伪代码如下：

```
step = (Table_size >> 1) + (Table_size >> 2) +
state = 0;
for(symbol = 0; symbol < max; symbol++){
    if((-1 == norm(symbol)) || 0 == norm(symbol))
        continue;
    for(i = 0; i < norm(symbol); i++){
        FSE[state].symbol = symbol;
        while(state >= end_state)
            state = (state + step) % Table_size;
    }
}
```

FSE 码表中每个状态除对应一个符号外，还对应一个状态区域（State-range），State-range 的描述方式为(Baseline, number-bits)，在 FSE 编解码流程，每个状态的 State-range 用来描述下一个

符号的状态取值范围[Baseline, Baseline+(1<<number-bits)]。其中, <<表示右移运算, 右移运算是将一个二进制位的操作数按指定移动的位数向右移动, 移出位被丢弃, 左边移出的空位或者一律补0, 或者补符号位, 这由不同的机器而定。

对于一个符号, FSE 码表中对应若干个状态, 这些状态对应的 State-range 没有重叠, 而且所有有状态的 State-range 的长度相加, 等于 FSE 码表的总状态数目(Table-size)。

State-range 的长度值必须为 2 的整数次幂(整数为 number-bits)。同一符号的所有状态的 State-range 的长度可能取值 1-2 个。

响应于符号的概率正好为 2 的整数次幂($\text{norm}=2^n$), 那么 State-range 的长度取值惟一:

$$\text{number_bits}=\text{table_log}-n$$

$$\text{table_log}=\log_2^{\text{table_size}}$$

其他情形, State_range 的长度有两个取值:

$$\text{number_bits}_1=\text{table_log}-\text{floor}(\log_2^{\text{norm}})$$

$$\text{number_bits}_2=\text{table_log}-\text{ceil}(\log_2^{\text{norm}})$$

对于同一个符号, 首先需要对所有的状态进行排序。较小状态对应State-range的长度取值为 num_bits_1 , 较大状态取值为: num_bits_2 。

假设 $N=2^{\text{ceil}(\log(\text{norm}))}$, 那么 num_bits_1 对应的状态数为: $N-\text{norm}$; num_bits_2 对应的状态数目为: $2*\text{norm}-N$ 。

State-range的基线 (Baseline),从较大状态的开始分配。

假设FSE码表的长度为64, 某符号s的归一化概率为5, 如果从状态0开始分配, 那么如下状态的对应符号都为s:

[0, 43, 22, 1, 44].排序后的结果为[0, 1, 22, 43, 44]。

$$\text{Table_log}=\log_2^{64}=6;$$

$$\text{number_bits}_1=\text{Table_log}-\text{floor}(\log(5))=6-2=4;$$

$$\text{number_bits}_2=\text{Table_log}-\text{ceil}(\log(5))=3;$$

$N=2^3=8$, 所以 Number-bits 取值为 4 的状态数目为 $N-5=3$, 即[0, 1, 22]。Number-bits 取值为 3 的状态数数目为 2, 即[43, 44]。

State(43)的基线 (Baseline) 为0, State(44)的基线为8 (0+8), State(0)的基线 (Baseline) 为16 (8+8), State(1)的基线为32 (16+16), State(22)的基线为48 (32+16)。

以上计算经整理得到下表 1:

表 1

state_order	0	1	2	3	4
State	0	1	22	43	44
Number_bit	4	4	4	3	3
Base_line	16	32	48	0	8
State_range	16-31	32-47	48-63	0-7	8-15

以上详述的是 FSE 码表构建流程，接下来为 FSE 编码数据的子流程：

从宏观角度看，FSE是对整个符号序列（Sequences）统一编码，可以认为整个符号序列最终编码生成一个码字。

从微观角度看，FSE编码通过编码状态(State)的形式实现符号的熵编码，并且通过FSE码表的状态对符号序列相邻的符号构建一个链接关系。

假设前一个符号s1选择状态为State[39],并且后一个符号s2,且s2可以采用表1编码，FSE编码时，s2需要选择State[39]所在State-range的对应状态,本例中显然选择State[1]；s2状态确定后,就可以编码s1的状态，State[39]编码的位宽为State[1]的number_bits:4,编码值为7（39-32,32为State[1]的baseline）。

假设s3与s2恰好相同，那么s3的状态选择State[1]所在State-range（0-7）对应的state[43],s2编码位宽为state[43]的number-bits:3，编码值为1（1-0,0为State[43]的baseline）。

以上举例描述FSE编码的算法流程，简要归纳为：

第1个符号的状态决定第2个符号的状态，第1个符号的编码码字由第2个状态的range决定；

第2个符号的状态决定第3个符号的状态，第2个符号的编码码字由第3个状态的range决定；

.....

倒数第2个符号的状态决定最后一个符号的状态，倒数第2个符号的编码码字由最后一个的range决定；

前一个符号的状态决定后一个符号的状态，前一个符号的状态编码内容（码长，码字）却由后一个符号决定。FSE编码流程中，整个符号序列每个符号的状态依赖于前一个符号状态，整个编码流程，环环相扣，一步一步串行流下去。从流程上讲，难以并行实现符号序列的FSE编码。

FSE在进行符号序列编码时，需要根据前一个符号的状态（state）选择下一个符号的状态（state），选择的依据为：前一个的状态必须在后一个状态的覆盖范围（State_range）之内。

由前述可知，一个符号出现的概率较大，该符号对应的状态会有多个(等于归一化频率)。FSE编码该符号时需要从它对应的所有状态中选择一个，状态的选择标准前面已经说明。以下需要寻求状态选择方式。

在Zstandard参考代码中，给出了一种遍历搜索的状态选择方案，伪代码形式如下：

```

for(i = 0; i < norm[s]; i++){
    if ( state_0 >= State[i].base  &&
        state_0 < State[i].base + (1 << State[i].number_bits) ) {
        state_1 = State[i];
        break;
    }
}

```

假设前一个符号选择的state_0,下一个符号为s, s对应的所有状态为State[]。由以上伪代码可知,该算法方案需要依次分析符号s所对应的每个状态的覆盖范围,直到找到一个state_0所在范围的状态。状态的搜索次数与前一个符号的状态有关,如用硬件实现,会有如下弊端:对于特定的符号s,符号序列中会有多个s符号,位置不同,搜索次数会有不同。

特定符号s的最大搜索次数由其归一化频率(norm[s])决定,但是对于符号全集,每个符号的最大搜索次数也会不同。

搜索次数直接对应硬件电路的时钟周期,硬件电路往往要求统一的时钟周期。因此上述“搜到即止”的方案无法用于硬件实现,所以需要找到一种对硬件友好的状态选择方案,具体如下:

为每个符号生成一个可用于查找的真值表,每个表包含两组元素:搜索位宽search_bits,假设符号s的归一化概率为norm[s],那么search_bits=ceil(log (norm[s]));以及真值表Search_table[],表的长度等于2的search_bits次方,查找表的每个单元格对应一个状态(state)。

真值表使用方法:假设前一个符号的状态为state0,当前符号为s,状态搜索流程如下:截取state0高位,截取位宽为search_bits,得到一个数字num=state0>>(table_lengths-search_bits),利用真值表查找状态, state1=Search_table[num]。

以下详述 FSE 状态查找的真值表的构建流程,本实施例中,FSE 状态查找的真值表与 FSE 码表的构建流程同步进行:

第一步,FSE 码表构件:

假设符号序列为:

fdgecaedcfcegeafceadedgfegcafgegedgegfcecbfecgegcgefbcdgaefdebfgfdefagfecfgedeaegfedgfbdcdf
 cfefadgcecgfgdcgcbfcdgcegfacedgdcdf

对上述符号序列进行频率统计,得到下表 2:

表 2

a	b	c	d	e	f	g
9	5	16	17	32	24	24

归一化结果如下表 3:

表 3

a	b	c	d	e	f	g
1	-1	2	2	4	3	3

本例中，FSE 码表的码表位宽为：table_log=4；

码表长度为：table_size=1<<table_log=16；

步长 step=(5/8)*table_size+3=13；

首先，状态 15 的对应符号为 b；

其他的状态分配顺序为：0,13,10,7,4,1,14,11,8,5,2,12,9,6,3；

即：

状态 0 对应的符号为 a；

状态 13,10 对应的符号为 c；

状态 7,4 对应的符号为 d；

状态 1,14,11,8 对应的符号为 e；

状态 5,2,12 对应的符号为 f；

状态 9,6,3 对应的符号为 g；

假设 FSE 码表的长度为 16，则构建码表，如下表 4：

表 4

state	symbol	baselin	bit	range
0	a	0	4	0 - 15
1	e	0	2	0 - 3
2	f	8	3	8 - 15
3	g	8	3	8 - 15
4	d	0	3	0 - 7
5	f	0	2	0 - 3
6	g	0	2	0 - 3
7	d	8	3	8 - 15
8	e	4	2	4 - 7
9	g	4	2	4 - 7
10	c	0	3	0 - 7
11	e	8	2	8 - 11
12	f	4	2	4 - 7
13	c	8	3	8 - 15
14	e	12	2	12 - 15
15	b	0	4	0 - 15

第二步，用于查找状态的真值表的构建：

根据归一化结果计算符号的搜索位宽，计算公式如下：

$$\text{search_bits} = \text{ceil}(\log_2^{\text{norm}[s]})$$

or

$$\text{search_bits} = \log_2^{\text{norm}[s]}$$

其中，ceil 表示向上取整，例如，ceil(3.5)=4。

各个符号的搜索位宽如下表 5：

表 5

符号	a	b	c	d	e	f	g
归一化频率(norm)	1	-1	2	2	4	3	3
搜索位宽(search_bits)	0	0	1	1	2	2	2

给各个符号构建真值表：

符号 a：搜索位宽为 0；a 只存在一个状态 0。符号 b：搜索位宽为 0；b 只存在一个状态 15。

符号 c：搜索位宽为 1；b 对应两个状态 10，13；真值表长度为 2；状态 state[10]对应的索引值计算：

$$\text{num}_{10} = \text{state}[10].\text{baseline} \gg (\text{码表位宽} - \text{搜索位宽}) = 0 \gg (4-1) = 0$$

即真值表中索引值“0”对应的值为 10；

状态 state[13]对应的索引值计算：

$$\text{num}_{13} = \text{state}[13].\text{baseline} \gg (\text{码表位宽} - \text{搜索位宽}) = 8 \gg (4-1) = 1$$

即真值表中索引值“1”对应的值为 13；

符号 c 的真值表如下表 6：

表 6

索引	0	1
映射状态	10	13

符号 d：搜索位宽为 1；b 对应两个状态 4，7；真值表长度为 2；

状态 state[4]对应索引计算：num₄=state[4].baseline>>(4-1)=0>>3=0；

状态 state[7]对应索引计算：num₇=state[7].baseline>>(4-1)=8>>3=1；

符号 d 的真值表如下表 7：

表 7

索引	0	1
映射状态	4	7

符号 e：搜索位宽为 2；b 对应 4 个状态 1,8,11,14；真值表长度为 4；状态 state[1]对应索引计

算: $\text{num}_1 = \text{state}[1].\text{baseline} \gg (4-2) = 0 \gg 2 = 0$;

状态 $\text{state}[8]$ 对应索引计算: $\text{num}_8 = \text{state}[8].\text{baseline} \gg (4-2) = 4 \gg 2 = 1$;

状态 $\text{state}[11]$ 对应索引计算: $\text{num}_{11} = \text{state}[11].\text{baseline} \gg (4-2) = 8 \gg 2 = 2$;

状态 $\text{state}[14]$ 对应索引计算: $\text{num}_{14} = \text{state}[14].\text{baseline} \gg (4-2) = 12 \gg 2 = 3$;

符号 e 的真值表如下表 8:

表 8

索引	0	1	2	3
映射状态	1	8	11	14

符号 f: 搜索位宽为 2; b 对应 3 个状态 2,5,12; 真值表长度为 4;

状态 $\text{state}[2]$ 对应索引计算: $\text{num}_1 = \text{state}[2].\text{baseline} \gg (4-2) = 8 \gg 2 = 2$;

此时, 索引 3 对应的状态也为 2, $\text{num}_2 = \text{num}_1 + 1 = 3$;

状态 $\text{state}[5]$ 对应索引计算: $\text{num}_5 = \text{state}[2].\text{baseline} \gg (4-2) = 0 \gg 2 = 0$;

状态 $\text{state}[12]$ 对应索引计算: $\text{num}_{12} = \text{state}[2].\text{baseline} \gg (4-2) = 4 \gg 2 = 1$;

符号 f 的真值表如下表 9:

表 9

索引	0	1	2	3
映射状态	5	12	2	2

符号 g: 搜索位宽为 2; b 对应 3 个状态 3,6,9; 真值表长度为 4;

状态 $\text{state}[3]$ 对应索引计算: $\text{num}_1 = \text{state}[3].\text{baseline} \gg (4-2) = 8 \gg 2 = 2$;

此时, 索引 3 对应的状态也为 3, $\text{num}_2 = \text{num}_1 + 1 = 3$;

状态 $\text{state}[6]$ 对应索引计算: $\text{num}_6 = \text{state}[3].\text{baseline} \gg (4-2) = 0 \gg 2 = 0$;

状态 $\text{state}[9]$ 对应索引计算: $\text{num}_9 = \text{state}[3].\text{baseline} \gg (4-2) = 4 \gg 2 = 1$;

符号 g 的真值表如下表 10:

表 10

索引	0	1	2	3
映射状态	6	9	3	3

第三步, 状态选择 (查找) 示例:

FSE 编码数据时, 采用状态 (state) 来编码符号, 即每个符号用一个状态标示。由于一个符号对应的状态数目并不唯一, 因此编码某个符号时需要选在符号的所对应的所有状态中选择一个, 要选择哪个状态需要利用上一个状态 (前一个符号选择的状态) 来确定。

假设上一个符号选择状态为 $\text{State}_{\text{prev}}$, 后一个符号 $\text{symbol}_{\text{next}}$ 状态 $\text{State}_{\text{next}}$, 那么 $\text{State}_{\text{prev}}$ 必须在

$State_{next}$ 对应的状态区域之内:

$$State_{next} \cdot Baseline \leq State_{prev} < (State_{next} \cdot Baseline + State_{next} \cdot length)$$

常规的方法需要遍历后一个 $symbol_{next}$ 对应的所有状态, 选择一个状态满足上式。

本实施例中, 利用构建的真值表来快速查找状态, 使用方法如下:

1. 获取上一个符号的状态 $State_{prev}$;
2. 根据下一个符号 $symbol_{next}$, 选择其对应的真值表, 得到其搜索位宽 $search_bits$;
3. 计算搜索值 $Search_num = State_{prev} \gg (table_log - search_bits)$;
4. 查表得到下一个符号的状态 $State_{next} = Table[search_num]$ 。

以 $afgegedgegfcecb$ 为例, 介绍每个符号的状态选择流程:

1. 本例中 a 只对应一个状态 0 , 即第一个符号的编码状态为 0 ;
2. 确定 f 的状态, 搜索值 $search_num = 0 \gg (码表位宽 - f \text{ 的搜索位宽}) = 0 \gg (4 - 2) = 0$; 查表得到 f 的状态为 5 ;
3. 确定 g 的状态, 搜索值 $search_num = 5 \gg (码表位宽 - g \text{ 的搜索位宽}) = 5 \gg (4 - 2) = 1$; 查表得到 g 的状态为 9 ;
4. 确定 e 的状态, 搜索值 $search_num = 9 \gg (4 - 2) = 2$; 查表得到 e 的状态为 11 ;
5. 确定 g 的状态, 搜索值 $search_num = 11 \gg (4 - 2) = 2$; 查表得到 g 的状态为 3 ;
6. 确定 e 的状态, 搜索值 $search_num = 3 \gg (4 - 2) = 0$; 查表得到 e 的状态为 1 ;
7. 确定 d 的状态, 搜索值 $search_num = 1 \gg (4 - 2) = 0$; 查表得到 d 的状态为 4 ;

.....

最终会得到整个序列中的状态表, 如下表 11:

表 11

符号	a	f	g	e	g	e	d	g	e	g	f	e	c	e	b
状态	0	5	9	11	3	1	4	6	8	3	5	8	13	14	15

由此, 则选出了 FSE 码表中每个符号的用于编码的状态。

本申请实施例的一个或多个方面, 还提供了一种有限状态熵编码的状态选择系统。图 2 示出的是本申请提供的有限状态熵编码的状态选择系统的实施例的示意图。如图 2 所示, 一种有限状态熵编码的状态选择系统包括: 有限状态熵码表生成模块 10, 配置用于获取符号序列, 并基于符号序列分别得到其中多个符号的归一化概率, 并基于多个符号及相应的归一化概率生成有限状态熵码表, 且有限状态熵码表包含每个符号的若干状态; 真值表生成模块 20, 配置用于针对每个符号, 基于其归一化概率得到对应的搜索位宽, 并基于搜索位宽得到每个符号的每个状态对应的索引值, 并基于每个状态及对应的索引值生成每个符号的真值表; 以及状态选择模块 30, 配置用于响应于基于有限状态熵码表进行有限状态熵编码, 基于上一个符号的状态以及当前符号的搜索位宽得到当前符号的被选索引值, 并在当前符号的真值表中查找被选索引值所对应的状态, 以将其

作为当前符号的用于有限状态熵编码的状态。

本申请实施例的一个或多个方面，还提供了一种非易失性计算机可读存储介质，图 3 示出了根据本申请实施例提供的实现有限状态熵编码的状态选择方法的非易失性计算机可读存储介质的示意图。如图 3 所示，非易失性计算机可读存储介质 3 存储有计算机可读指令 31。该计算机可读指令 31 被处理器执行时实现上述任意一项实施例的方法。

应当理解，在相互不冲突的情况下，以上针对根据本申请的有限状态熵编码的状态选择方法阐述的所有实施方式、特征和优势同样地适用于根据本申请的有限状态熵编码的状态选择系统和非易失性计算机可读存储介质。

本申请实施例的一个或多个方面，还提供了一种计算机设备，包括如图 4 所示的存储器 402 和处理器 401，该存储器 402 中存储有计算机可读指令，该计算机可读指令被该处理器 401 执行时实现上述任意一项实施例的方法。

如图 4 所示，为本申请提供的执行有限状态熵编码的状态选择方法的计算机设备的一个或多个实施例的硬件结构示意图。以如图 4 所示的计算机设备为例，在该计算机设备中包括一个处理器 401 以及一个存储器 402，并还可以包括：输入装置 403 和输出装置 404。处理器 401、存储器 402、输入装置 403 和输出装置 404 可以通过总线或者其他方式连接，图 4 中以通过总线连接为例。输入装置 403 可接收输入的数字或字符信息，以及产生与有限状态熵编码的状态选择系统的用户设置以及功能控制有关的键信号输入。输出装置 404 可包括显示屏等显示设备。

存储器 402 作为一种非易失性计算机可读存储介质，可用于存储非易失性软件程序、非易失性计算机可执行程序以及模块，如本申请实施例中的有限状态熵编码的状态选择方法对应的程序指令/模块。存储器 402 可以包括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需的应用程序；存储数据区可存储有限状态熵编码的状态选择方法的使用所创建的数据等。此外，存储器 402 可以包括高速随机存取存储器，还可以包括非易失性存储器，例如至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。在一些实施例中，存储器 402 可选包括相对于处理器 401 远程设置的存储器，这些远程存储器可以通过网络连接至本地模块。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

处理器 401 通过运行存储在存储器 402 中的非易失性软件程序、指令以及模块，从而执行服务器的各种功能应用以及数据处理，即实现上述方法实施例的有限状态熵编码的状态选择方法。

最后需要说明的是，本文的非易失性计算机可读存储介质（例如，存储器）可以是易失性存储器或非易失性存储器，或者可以包括易失性存储器和非易失性存储器两者。作为例子而非限制性的，非易失性存储器可以包括只读存储器（Read-Only Memory，ROM）、可编程 ROM（Programmable Read Only Memory，PROM）、电可编程 ROM（Electrically Erasable Programmable Read Only Memory，EPROM）、电可擦写可编程 ROM（EEPROM）或快闪存储器。易失性存储器可以包括随机存取存储器（Random Access Memory，RAM），该 RAM 可以充当外部高速缓存存储器。作为例子而非限制性的，RAM 可以以多种形式获得，比如同步 RAM

(DRAM)、动态 RAM (Dynamic Random Access Memory, DRAM)、同步 DRAM (Synchronous Dynamic Random-access Memory, SDRAM)、双数据速率 SDRAM (Double Data Rate SDRAM, DDR SDRAM)、增强 SDRAM (Enhanced Synchronous DRAM, ESDRAM)、同步链路 DRAM (Sync Link DRAM, SLDRAM)、以及直接 Rambus RAM (Direct Rambus RAM, DRRAM)。所公开的方面的存储设备意在包括但不限于这些和其它合适类型的存储器。

本领域技术人员还将明白的是，结合这里的公开所描述的各种示例性逻辑块、模块、电路和算法步骤可以被实现为电子硬件、计算机软件或两者的组合。为了清楚地说明硬件和软件的这种可互换性，已经就各种示意性组件、方块、模块、电路和步骤的功能对其进行了一般性的描述。这种功能是被实现为软件还是被实现为硬件取决于具体应用以及施加给整个系统的设计约束。本领域技术人员可以针对每种具体应用以各种方式来实现的功能，但是这种实现决定不应被解释为导致脱离本申请实施例公开的范围。

以上是本申请公开的示例性实施例，但是应当注意，在不背离权利要求限定的本申请实施例公开的范围的前提下，可以进行多种改变和修改。根据这里描述的公开实施例的方法权利要求的功能、步骤和/或动作不需以任何特定顺序执行。此外，尽管本申请实施例公开的元素可以以个体形式描述或要求，但除非明确限制为单数，也可以理解为多个。

应当理解的是，在本文中使用的，除非上下文清楚地支持例外情况，单数形式“一个”旨在也包括复数形式。还应当理解的是，在本文中使用的“和/或”是指包括一个或者一个以上相关联地列出的项目的任意和所有可能组合。上述本申请实施例公开实施例序号仅仅为了描述，不代表实施例的优劣。

所属领域的普通技术人员应当理解：以上任何实施例的讨论仅为示例性的，并非旨在暗示本申请实施例公开的范围（包括权利要求）被限于这些例子；在本申请实施例的思路下，以上实施例或者不同实施例中的技术特征之间也可以进行组合，并存在如上的本申请实施例的不同方面的许多其它变化，为了简明它们没有在细节中提供。因此，凡在本申请实施例的精神和原则之内，所做的任何省略、修改、等同替换、改进等，均应包含在本申请实施例的保护范围之内。

权利要求书

1.一种有限状态熵编码的状态选择方法，其特征在于，包括以下步骤：

获取符号序列，并基于所述符号序列分别得到其中多个符号的归一化概率，并基于所述多个符号及相应的归一化概率生成有限状态熵码表，且所述有限状态熵码表包含每个符号的若干状态；

针对所述每个符号，基于其归一化概率得到对应的搜索位宽，并基于所述搜索位宽得到所述每个符号的每个状态对应的索引值，并基于所述每个状态及对应的索引值生成所述每个符号的真值表；以及

响应于基于所述有限状态熵码表进行有限状态熵编码，基于上一个符号的状态以及当前符号的搜索位宽得到所述当前符号的被选索引值，并在所述当前符号的真值表中查找所述被选索引值所对应的状态，以将其作为所述当前符号的用于有限状态熵编码的状态。

2.根据权利要求 1 所述的方法，其特征在于，基于所述搜索位宽得到所述每个符号的每个状态对应的索引值包括：

基于所述多个符号的归一化概率得到所述有限状态熵码表的码表长度，并基于所述码表长度得到所述有限状态熵码表的码表位宽；以及

针对所述每个符号的每个状态，基于其状态区域的基线位置的基线状态值、所述搜索位宽以及所述码表位宽得到所述每个状态对应的索引值。

3.根据权利要求 2 所述的方法，其特征在于，所述针对所述每个符号的每个状态，基于其状态区域的基线位置的基线状态值、所述搜索位宽以及所述码表位宽得到所述每个状态对应的索引值步骤之前，还包括：

获取所述每个符号的每个状态的状态取值范围；以及

根据所述状态取值范围设置对应的状态区域。

4.根据权利要求 2 所述的方法，其特征在于，所述针对所述每个符号的每个状态，基于其状态区域的基线位置的基线状态值、所述搜索位宽以及所述码表位宽得到所述每个状态对应的索引值步骤之前，还包括：

依次根据所述多个符号对应的归一化概率，确定所述每个符号的状态区域的长度取值。

5.根据权利要求 4 所述的方法，其特征在于，所述依次根据所述多个符号对应的归一化概率，确定所述每个符号的状态区域的长度取值步骤，包括：

依次确定所述每个符号对应的归一化概率是否为预设值；

响应于所述归一化概率为预设值，将对应的符号的状态区域的长度取值设置为唯一取值；或

响应于所述归一化概率不为预设值，将对应的符号的状态区域的长度取值设置为两个取值。

6.根据权利要求 2 所述的方法，其特征在于，针对所述每个符号的每个状态，基于其状态区域的基线位置的基线状态值、所述搜索位宽以及所述码表位宽得到所述每个状态对应的索引值包

括：

计算所述码表位宽与所述每个符号的所述搜索位宽的差值，得到第一移动位数；以及

基于所述第一移动位数对所述每个符号的每个状态的基线状态值进行右移运算，得到所述每个状态对应的索引值。

7.根据权利要求 2 所述的方法，其特征在于，基于上一个符号的状态以及当前符号的搜索位宽得到所述当前符号的被选索引值包括：

计算所述码表位宽与所述当前符号的搜索位宽的差值，得到第二移动位数；以及

基于所述第二移动位数对所述上一个符号的状态的值进行右移运算，得到所述当前符号的被选索引值。

8.根据权利要求 1 或 2 任意一项所述的方法，其特征在于，针对所述每个符号，基于其归一化概率得到对应的搜索位宽包括：

基于所述每个符号的归一化概率以及底数 2 构造对数函数，对所述对数函数求解，得到所述每个符号对应的搜索位宽。

9.根据权利要求 1 或 2 任意一项所述的方法，其特征在于，基于所述多个符号及相应的归一化概率生成有限状态熵码表，且所述有限状态熵码表包含每个符号的若干状态包括：

基于所述多个符号的归一化概率得到所述有限状态熵码表的码表长度；

基于所述码表长度得到所述有限状态熵码表中的所有状态；

将所述每个符号的归一化概率的值作为相应的状态数目；

基于所述所有状态和所述每个符号的状态数目得到所述每个符号的若干状态；以及

基于所述每个符号的若干状态生成所述有限状态熵码表。

10.根据权利要求 9 所述的方法，其特征在于，所述基于所述多个符号的归一化概率得到所述有限状态熵码表的码表长度步骤，包括：

获取所述多个符号的归一化概率的总和；以及

根据所述总和获取所述有限状态熵码表的码表长度。

11.根据权利要求 1 或 2 任意一项所述的方法，其特征在于，基于所述符号序列分别得到其中多个符号的归一化概率包括：

基于所述符号序列分别得到其中多个符号的出现频率；以及

基于每个符号的出现频率得到所述每个符号的归一化概率。

12.根据权利要求 11 所述的方法，其特征在于，所述基于每个符号的出现频率得到所述每个符号的归一化概率步骤之后，还包括：

确定所述多个符号中是否存在相同的归一化概率；以及

响应于存在相同的归一化概率，采用不同的标识对相同的归一化概率进行标识。

13.根据权利要求 1 或 2 任意一项所述的方法，其特征在于，所述获取符号序列，并基于所述符号序列分别得到其中多个符号的归一化概率，并基于所述多个符号及相应的归一化概率生成有

限状态熵码表，且所述有限状态熵码表包含每个符号的若干状态步骤之前，还包括：

确定所述符号序列是否为预设符号；以及

响应于所述符号序列为预设符号，则基于所述符号序列分别得到其中多个符号的归一化概率。

14. 根据权利要求 13 所述的方法，其特征在于，所述预设符号包括：原文节段长度 `literal_length`，重复长度 `match_length`，和偏移距离 `offset`。

15. 根据权利要求 1 或 2 任意一项所述的方法，其特征在于，所述获取符号序列，并基于所述符号序列分别得到其中多个符号的归一化概率，并基于所述多个符号及相应的归一化概率生成有限状态熵码表，且所述有限状态熵码表包含每个符号的若干状态步骤，包括：

根据所述有限状态熵码表的码表长度和所述归一化概率，依次对所述多个符号进行状态分配。

16. 根据权利要求 15 所述的方法，其特征在于，所述根据所述有限状态熵码表的码表长度和所述归一化概率，依次对所述多个符号进行状态分配步骤，包括：

根据所述有限状态熵码表的码表长度和所述归一化概率，采用状态倒序对所述多个符号进行状态分配。

17. 一种有限状态熵编码的状态选择系统，其特征在于，包括：

有限状态熵码表生成模块，配置用于获取符号序列，并基于所述符号序列分别得到其中多个符号的归一化概率，并基于所述多个符号及相应的归一化概率生成有限状态熵码表，且所述有限状态熵码表包含每个符号的若干状态；

真值表生成模块，配置用于针对所述每个符号，基于其归一化概率得到对应的搜索位宽，并基于所述搜索位宽得到所述每个符号的每个状态对应的索引值，并基于所述每个状态及对应的索引值生成所述每个符号的真值表；以及

状态选择模块，配置用于响应于基于所述有限状态熵码表进行有限状态熵编码，基于上一个符号的状态以及当前符号的搜索位宽得到所述当前符号的被选索引值，并在所述当前符号的真值表中查找所述被选索引值所对应的状态，以将其作为所述当前符号的用于有限状态熵编码的状态。

18. 根据权利要求 17 所述的系统，其特征在于，所述真值表生成模块包括：

长度和位宽获取子模块，用于基于所述多个符号的归一化概率得到所述有限状态熵码表的码表长度，并基于所述码表长度得到所述有限状态熵码表的码表位宽；以及

索引值获取子模块，用于针对所述每个符号的每个状态，基于其状态区域的基线位置的基线状态值、所述搜索位宽以及所述码表位宽得到所述每个状态对应的索引值。

19. 一种非易失性计算机可读存储介质，其特征在于，所述非易失性计算机可读存储介质存储有计算机可读指令，所述计算机可读指令被至少一个处理器执行时，使得所述至少一个处理器执行如权利要求 1-16 任意一项所述方法的步骤。

20.一种计算机设备，包括存储器和处理器，其特征在于，包括：

至少一个处理器；以及

至少一个存储器，所述至少一个存储器存储有计算机可读指令，所述计算机可读指令被所述至少一个处理器执行时，使得所述至少一个处理器执行时执行如权利要求 1-16 任意一项所述的方法。

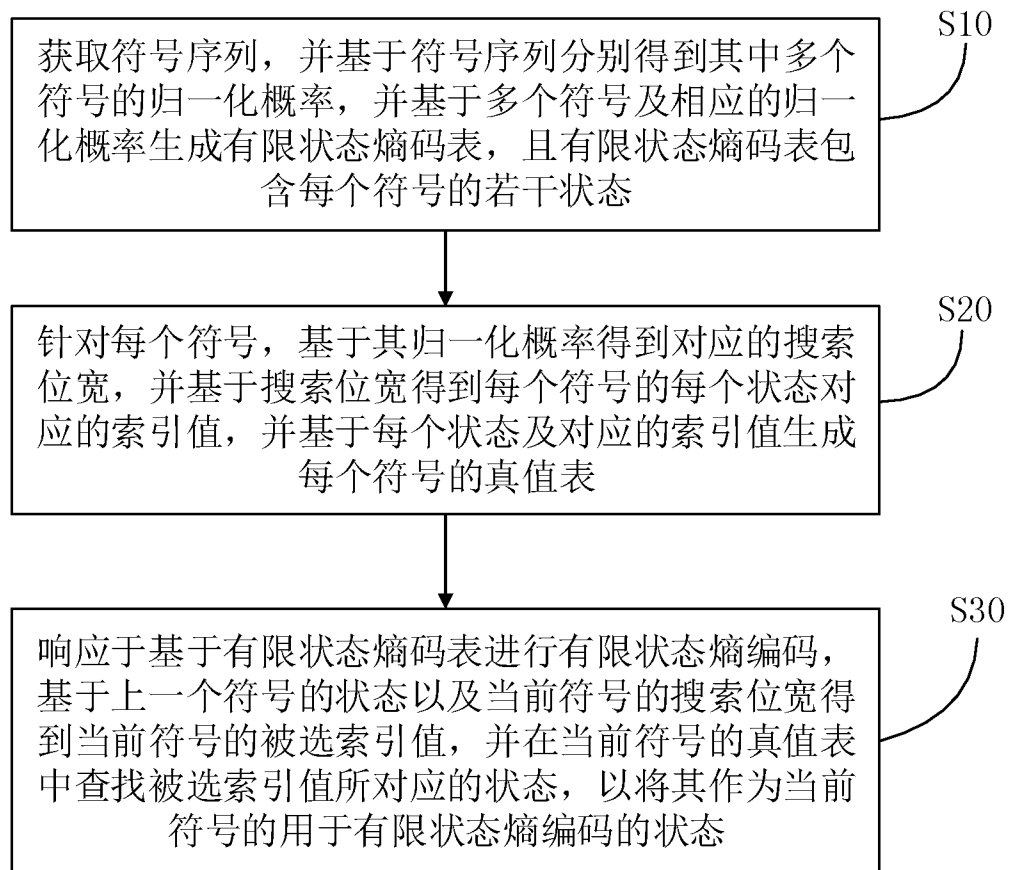


图 1

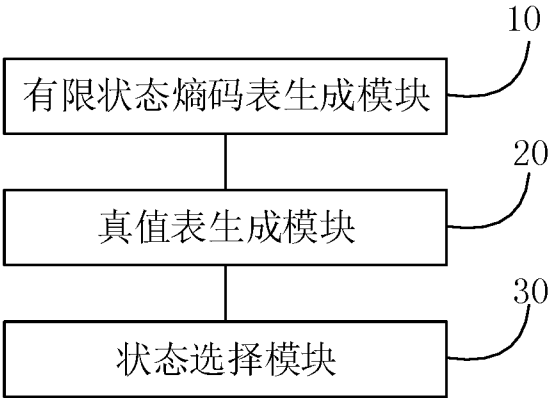


图 2

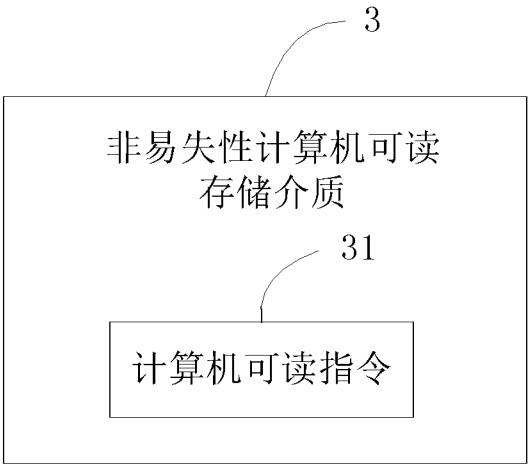


图 3

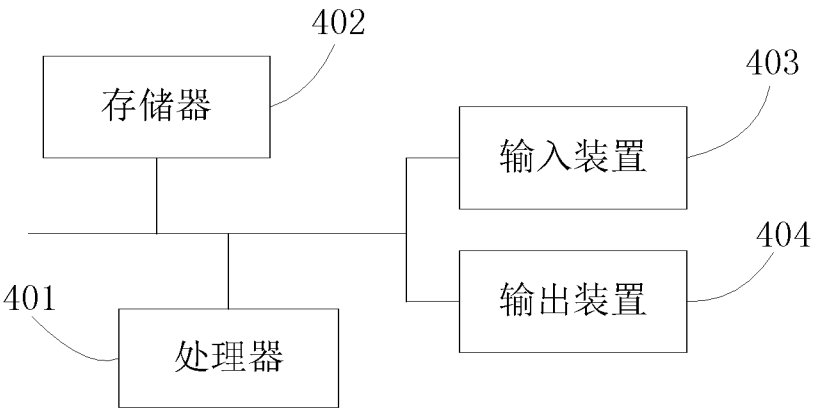


图 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/142929

A. CLASSIFICATION OF SUBJECT MATTER

H03M7/42(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; CNTXT; DWPI; VEN; USTXT; WOTXT; EPTXT; CNKI; IEEE: 熵, 编码, 状态, 概率, 真值, 索引, entropy, encoding, state, probability, true value, index

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 114513210 A (SUZHOU INSPUR INTELLIGENT TECHNOLOGY CO., LTD.) 17 May 2022 (2022-05-17) description, paragraphs 2-74, and figures 1-4	1-20
X	CN 114301468 A (SHANDONG YUNHAI GUOCHUANG CLOUD COMPUTING EQUIPMENT INDUSTRY INNOVATION CENTER CO., LTD.) 08 April 2022 (2022-04-08) description, paragraphs 42-94, and figures 1-7	1-20
X	CN 114039607 A (SHANDONG YUNHAI GUOCHUANG CLOUD COMPUTING EQUIPMENT INDUSTRY INNOVATION CENTER CO., LTD.) 11 February 2022 (2022-02-11) description, paragraphs 2-76, and figures 1-8	1-20
A	CN 113261285 A (GUANGDONG OPPO MOBILE COMMUNICATIONS CO., LTD.) 13 August 2021 (2021-08-13) entire document	1-20
A	US 2002037059 A1 (TEXAS INSTRUMENTS INC.) 28 March 2002 (2002-03-28) entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

09 May 2023

Date of mailing of the international search report

16 May 2023

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
China No. 6, Xitucheng Road, Jimenqiao, Haidian District,
Beijing 100088

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/142929

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	114513210	A	17 May 2022	CN	114513210	B	02 August 2022
CN	114301468	A	08 April 2022	None			
CN	114039607	A	11 February 2022	None			
CN	113261285	A	13 August 2021	WO	2021056214	A1	01 April 2021
US	2002037059	A1	28 March 2002	US	6961392	B2	01 November 2005

国际检索报告

国际申请号

PCT/CN2022/142929

A. 主题的分类

H03M7/42 (2006.01) i

按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

IPC:H03M

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))

CNABS; CNTXT; DWPI; VEN; USTXT; WOTXT; EPTXT; CNKI; IEEE: 熵, 编码, 状态, 概率, 真值, 索引, entropy, encoding, state, probability, true value, index

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 114513210 A (苏州浪潮智能科技有限公司) 2022年5月17日 (2022 - 05 - 17) 说明书第2-74段, 图1-4	1-20
X	CN 114301468 A (山东云海国创云计算装备产业创新中心有限公司) 2022年4月8日 (2022 - 04 - 08) 说明书第42-94段, 图1-7	1-20
X	CN 114039607 A (山东云海国创云计算装备产业创新中心有限公司) 2022年2月11日 (2022 - 02 - 11) 说明书第2-76段, 图1-8	1-20
A	CN 113261285 A (OPPO广东移动通信有限公司) 2021年8月13日 (2021 - 08 - 13) 全文	1-20
A	US 2002037059 A1 (TEXAS INSTRUMENTS INC) 2002年3月28日 (2002 - 03 - 28) 全文	1-20

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2023年5月9日

国际检索报告邮寄日期

2023年5月16日

ISA/CN的名称和邮寄地址

中国国家知识产权局

中国北京市海淀区蓟门桥西土城路6号 100088

授权官员

李文婷

电话号码 (+86) 020-28958148

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2022/142929

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	114513210	A	2022年5月17日	CN 114513210 B	2022年8月2日
CN	114301468	A	2022年4月8日	无	
CN	114039607	A	2022年2月11日	无	
CN	113261285	A	2021年8月13日	WO 2021056214 A1	2021年4月1日
US	2002037059	A1	2002年3月28日	US 6961392 B2	2005年11月1日