数据的表示

作者：Maeteno

原文链接：https://ld246.com/article/1676206061111

来源网站：链滴

进制的转换

在计算机中数据的表示都是二进制的，只有 0 和 1。当一个十进制的数需要保存到计算机中需要先转为二进制。十进制转二进制的规则有两条

1. 整数部分除 2 取余
2. 小数部分乘 2 取整

例如 10 转换为 2 进制的步骤：

```
10 / 2 = 5 ... 0
5  / 2 = 2 ... 1
2  / 2 = 1 ... 0
1  / 2 = 0 ... 1
```

最后 2 进制的值为: 1010

我们再计算一下 0.59 的 2 进制表示

```
0.59 * 2 = 1.18 -> 1
0.18 * 2 = 0.36 -> 0
0.36 * 2 = 0.72 -> 0
0.72 * 2 = 1.44 -> 1
0.44 * 2 = 0.88 -> 0
0.88 * 2 = 1.76 -> 1
// .... 直到小数位等于0
```

码制

码制是指数据在计算机中的编码方式

原码

原码可以理解为就是十进制到二进制的直接转换映射。

数值 X 的原码记为 [X]_原，如果机器字长为 n，则原码的定义如下：

若 X 是纯整数，则 [X]_{原}=\begin{cases} X &\text 0 \leq X \leq 2^{2-1}-1 \\2^{n-1} + |X| &\tex
-(2^{n-1}-1) \leq X \leq 0\end{cases}

若 X 是纯小数，则 [X]_{原}=\begin{cases} X &\text 0 \leq X < 1\\2^0 + |X| &\text -1 <X \leq 0\
nd{cases}

[+1]_原 = 0 \; {0000001}

[-1]_原=1\;0000001

反码

反码的范围，其中 n 表示机器字长。

若 X 是纯整数，则 [X]_{原}=\begin{cases} X &\text 0 \leq X \leq 2^{n-1}-1 \\2^{n-1} + |X| &\tex
-(2^{n-1}-1) \leq X \leq 0\end{cases}

该方程式的语义为：

当 0 \leq X \leq 2^{2-1} -1 时（正数），X_{[原]} 的值等于 X。

当 -(2^{n-1}-1) \leq X \leq 0 时（负数），X_{[原]} 的值等于 $2^{n-1}+|X|$

同时也可以看出来 X 的取值范围为：-(2^{n-1}-1) 到 $2^{n-1} -1$

若 X 是纯小数，则 [X]_{原} =\begin{cases} X &\text 0 \leq X < 1 \\2-2^{-(n-1)}+X &\text -1 <X \leq 0\end{cases}

反码表示法中，最高位是符号位，0 表示正数，1 表示负数。反码的正数表示和原码一致。负数的表是原码的基础上取反。例如 -5 的表示

[-5]_原 = 0 \; 0000101 数据位安位取反 [-5]_反 = 0 \; 1111010

补码

补码的正数表示和原码反码一致。其负数表示法在反码的基础上末尾加 1。例如 -5 的表示：

[-5]_反 = 0 \; 1111010 末尾 +1 [-5]_补 = 0\;1111011

移码

移码表示法是在数 X 上增加一个偏移量来定义的，常用于表示浮点数中的阶码。如果机器字长为 n，定偏移量为 2^{n-1}，则移码的定义如下：

若 X 是纯整数，则 [X]_移 = 2^{n-1} + X \quad (-2^{n-1} \leq X < 2^{n-1}) 其中的 2^{n-1} 是上面说的偏移量

若 X 是纯小数，则 [X]_移 = 1 + X \qquad (-1 \leq X < 1) 在定点小数中，1 就相当于加到最高位上了

实际上，在偏移 2^{n-1} 的情况下，只要将补码的符号位取反便可获得相应的移码表示。

移码的计算过程

计算 +5 的移码

\begin{aligned}
[+5]_移 &= 2^{8-1} + 5 \\
&= 2^7 + 5 \\
&= 128 + 5 \\
&= 133 \\
\text{二进制}&= 1000\,0101
\end{aligned}

计算 -5 的移码

\begin{aligned}
[-5]_移 &= 2^{8-1} + (-5) \\

```
&= 2^7 - 5 \\
&= 128 - 5 \\
&= 123 \\
\text{二进制}&= 0111\,1011
\end{aligned}
```

原文链接：数据的表示