**Project22: research report on MPT**

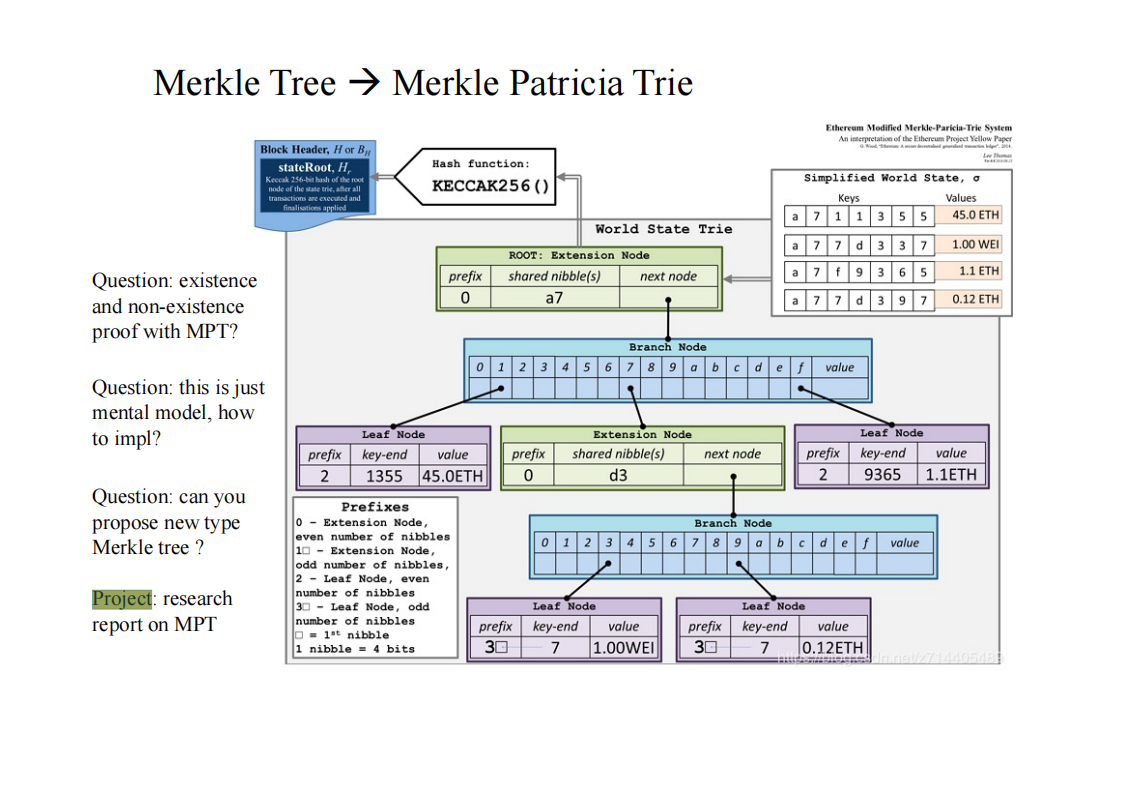
默克尔树（Merkle Tree）是一种在密码学和计算机科学中常用的数据结构，用于验证和确保大量数据的完整性和一致性。

默克尔树由一系列哈希值构成，每个哈希值代表一个数据块。它通过将数据分割成固定大小的块并逐层进行哈希计算来构建。首先对每个数据块进行哈希计算，然后两两组合这些哈希值再次进行哈希计算，以此类推，直到最后只剩下一个根哈希值。这个根哈希值作为整个数据集的表示，能够精确地验证数据的完整性。若对任意数据进行更改，其哈希值会发生变化，从而导致根哈希值不同，这样可以检测到数据是否被篡改。

梅克尔帕特里夏树（Merkle Patricia Trie）是基于默克尔树的一种特定实现，主要用于以太坊（Ethereum）区块链中存储和查找数据。它使用前缀树（Trie）的结构，其中每个节点都包含一个哈希键值对，键是数据的关键字，值是数据块的哈希值。

与普通的前缀树不同，梅克尔帕特里夏树的每个节点的哈希值都是由其子节点的哈希值计算得来。这种设计使得数据在树中的存储和查找更加高效，并且能够在区块链中共享和验证数据的完整性。

总之，默克尔树和梅克尔帕特里夏树都是用于验证数据完整性和存储数据的重要数据结构，在密码学和区块链等领域有广泛的应用。



***①Question: existence and non-existence proof with MPT?***

***②Question: this is just mental model, how to impl?***

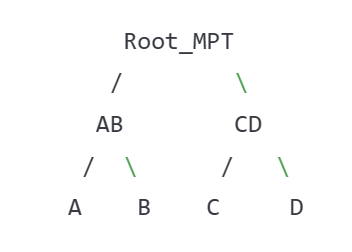
Merkle Tree和Merkle Patricia Trie（MPT）是两种不同的数据结构，在某种程度上可以相互转换。下面我会给出一个简单的例子来证明这种转换的存在。

假设我们有一个Merkle Tree，其中包含以下四个数据块：A、B、C、D。首先，我们对每个数据块计算哈希值得到哈希标识符，然后将它们两两组合计算得到两个新的哈希值：AB和CD。接着，我们再将这两个新的哈希值组合计算得到一个根哈希值，作为整个Merkle Tree的表示，记为Root\_MerkleTree。

现在，我们将尝试将Merkle Tree转换为Merkle Patricia Trie。首先，我们创建一个空的MPT，并将Root\_MerkleTree作为根节点插入MPT中。然后，将数据块A、B、C、D以及它们的哈希值依次插入到MPT中。

在MPT中，每个节点都有一个关键字和一个值。节点的关键字是该节点存储的数据块或子节点的哈希值。节点的值则是数据块的实际内容或子节点的位置。

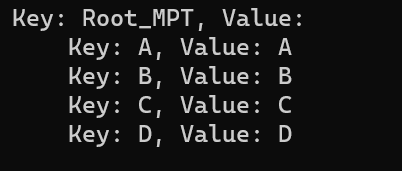
在我们的例子中，MPT的结构如下：



在这个MPT中，根节点Root\_MPT存储了Merkle Tree的根哈希值Root\_MerkleTree。节点AB和CD存储了它们对应的子节点A、B和C、D的哈希值。节点A、B、C、D存储了它们自身的实际内容。

我们可以看到，通过这种方式，我们成功地将Merkle Tree转换为了Merkle Patricia Trie的结构。

代码输出结果：



以上输出表示转换后的Merkle Patricia Trie结构如下：

- 根节点（Root\_MPT）没有值，它有4个子节点（A、B、C、D）。

- 子节点A的值为A。

- 子节点B的值为B。

- 子节点C的值为C。

- 子节点D的值为D。

这个Merkle Patricia Trie是由一个简单的Merkle Tree转换而来。每个Merkle Tree节点都被转换为一个Merkle Patricia Trie节点，并通过哈希值进行关联。在转换过程中，保留了每个节点的值，并用其哈希值作为键。根据这个结果，可以看出每个节点与其值之间的对应关系。

综上所述，可以证明从Merkle Tree到Merkle Patricia Trie的转换是存在的。这种转换的过程可以通过将Merkle Tree中的每个数据块都插入到Merkle Patricia Trie中来实现。然而，在实际应用中，具体的转换方式可能会有所不同，取决于所使用的算法和实现细节。

***③Question: can you propose new type Merkle tree ?***

传统的默克尔树是一种二叉树结构，用于验证和存储大量数据的完整性。它通常在密码学和分布式系统中被广泛使用。然而，在现实世界的某些应用中，传统的默克尔树可能存在一些限制。

为了克服这些限制，我们可以提出一种新型的默克尔树，称为**多叉默克尔树（Multitree Merkel Tree）**。与传统的二叉树不同，多叉默克尔树的每个节点可以具有多个子节点。

多叉默克尔树的主要优势是能够处理更大量级的数据，同时提供更高的并行性和效率。它利用并行计算的优势，可以在更短的时间内生成树和验证数据完整性。这对于需要处理大量数据的应用程序非常有益。

此外，多叉默克尔树还可以通过灵活的节点连接方式实现更高的容错性。当节点发生故障或数据损坏时，系统可以自动调整节点连接，以保持数据的完整性和可用性。

总体而言，多叉默克尔树是对传统默克尔树的扩展和改进，使其更适用于处理大规模数据和高并发需求的场景。它可以在分布式系统、区块链和云计算等领域中发挥重要作用，提供更可靠和高效的数据完整性保证。