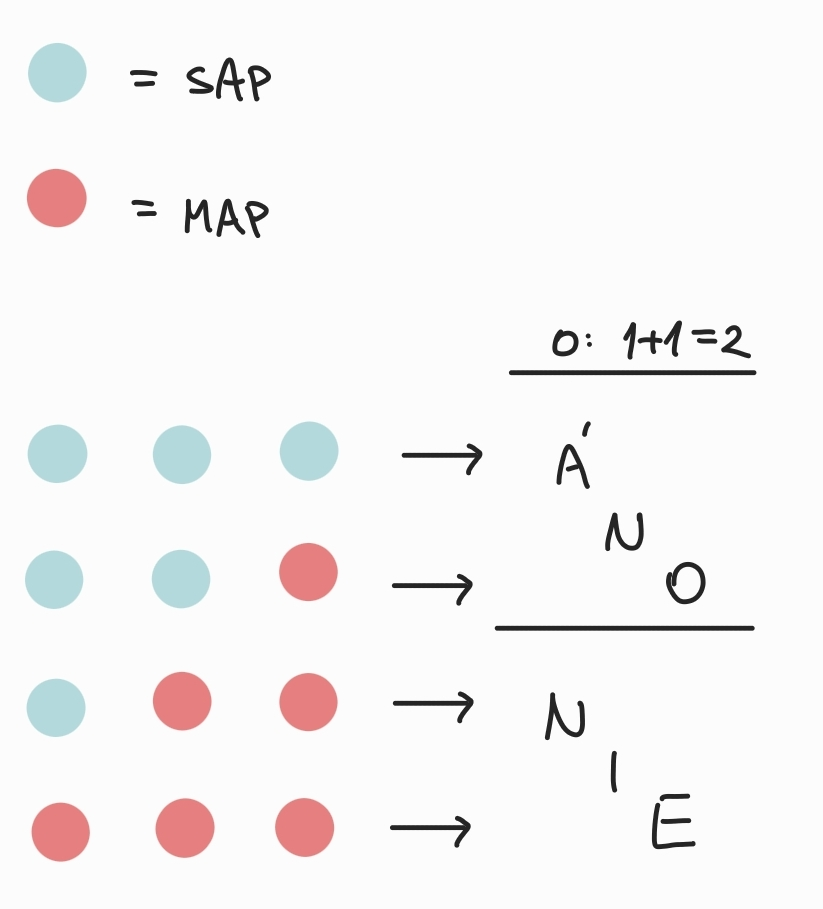
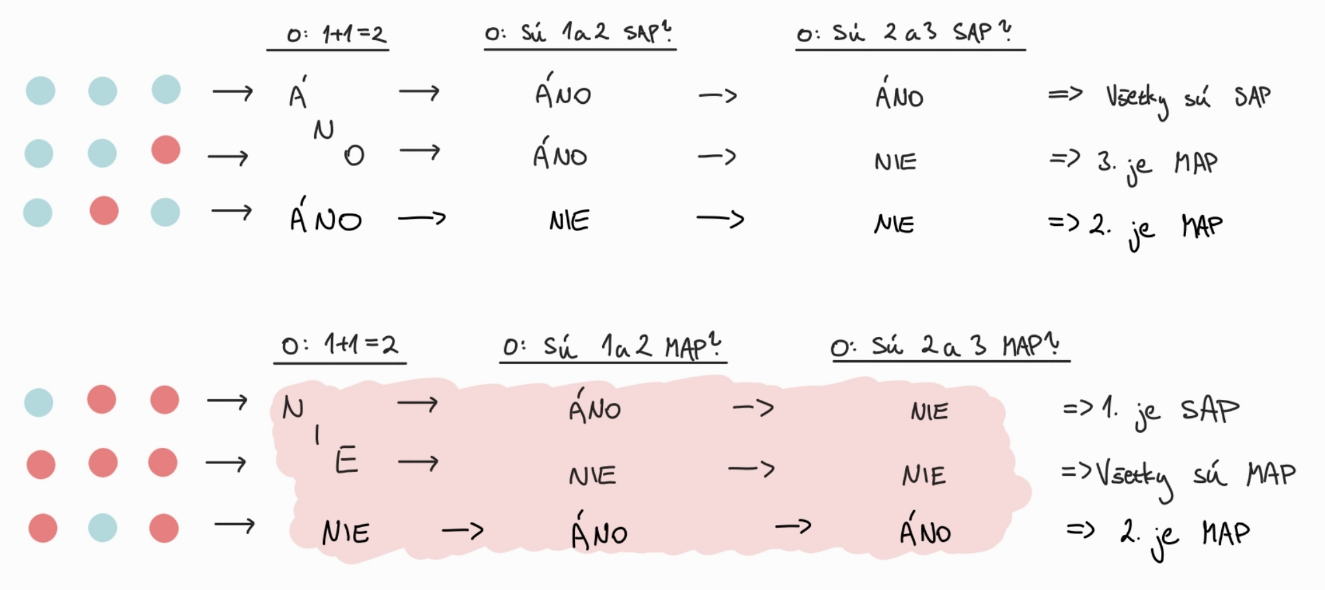
**Meziuzelná autorizace**

1. Keďže jednotlivé uzly od seba vieme rozlíšiť, môžeme si ich očíslovať od 1 po *n*, kde *n* je počet uzlov.
2. Vyberieme prvé tri uzly a spýtame sa ich otázku na ktorú dopredu poznáme odpoveď, čím určíme, či je väčšina uzlov v danej trojici SAP alebo MAP. Keďže SAP hovoria vždy pravdu a MAP vždy klamú, ak dostaneme pravdivú odpoveď, v trojici sa musia nachádzať 2 alebo 3 SAP uzly. (Pri klamlivej odpovedi 2 alebo 3 MAP uzly). Spýtame sa teda: **„Je pravda, že 1+1=2?“**



1. Teraz musíme určiť presný počet uzlov, ako aj ich poradie. Toto dokážeme spraviť v každom prípade dvoma otázkami. Využijeme pri tom vlastnosť, že uzly navzájom vedia aký protokol používa každý uzol.
   1. Ak sme na otázku 1+1=2? dostali odpoveď **ÁNO**, spýtame sa nasledujúce otázky:
      1. **„Sú uzly 1 a 2 SAP?“**
      2. **„Sú uzly 2 a 3 SAP?“**
   2. Ak sme na otázku 1+1=2? dostali odpoveď **NIE**, spýtame sa nasledujúce otázky:
      1. **„Sú uzly 1 a 2 MAP?“**
      2. **„Sú uzly 2 a 3 MAP?“**

Takýmto spôsobom sa bude vždy dať presne určiť, ktorý uzol v trojici používa aký protokol, pretože elementy v otázkach **i.** sa prekrývajú s elementami v otázkach **ii.** . Ak by sme teda mali kombináciu **SAP, SAP, MAP**, na otázku **i.** dostaneme odpoveď ÁNO, ale na otázku **ii.** dostaneme NIE, logicky z toho môžeme vyvodiť, že ak (1 a 2) sú SAP ale (2 a 3) nie sú, potom jedine 3 môže byť MAP aby platili dané odpovede.

Pri otázkach v prípade **b.**, teda ak prevažujú MAP uzly, berieme pri vyvodzovaní logických záverov samozrejme opačné pravdivostné hodnoty odpovedí.

1 2 3

1 2 3

(\*v tomto diagrame nie sú uvedené kombinácie **MAP, SAP, SAP** a **MAP, MAP, SAP**, no ich riešenie je analogické.)

1. Pre každú trojicu teda vieme zistiť jednotlivé protokoly tromi otázkami. V prípade, že máme *n* uzlov, bude riešenie ***n***otázok. Bude tomu tak aj v prípade, že počet uzlov nie je násobok trojky. V takom prípade sa protokol zvyšného jedného alebo dvoch uzlov ktoré už nemajú trojicu dozvieme tak, že sa spýtame predošlej trojice, aké protokoly používajú tie zvyšné uzly. To znova dáva 1 otázku na uzol, čiže výsledný počet bude stále ***n***. Taktiež sa po zistení prvých troch uzlov môžeme stále pýtať ich na protokoly ďalších uzlov, čo nám opäť dá ***n*** riešení.
2. Špecifickým prípadom, kedy by sa počet otázok mohol skrátiť by bolo, keby zostával jeden neurčený uzol, a vo všetkých predošlých sme našli len 2 SAP, potom na základe podmienky, že sa tam musia nachádzať minimálne 3 SAP, by tento uzol musel byť SAP, a ušetríme jednu otázku, teda riešenie bude ***n-1***. Taktiež, ak by ostávali na určenie dva uzly, a vo všetkých predošlých sme našli len 1 SAP, tieto dva musia byť tiež SAP, riešením teda bude ***n-2.*** (Platí analogicky aj pre MAP uzly).

**Rozšírenie:**

1. Ak budú niektoré uzly shy, musíme k problému pristúpiť trochu inak. Znovu sa prvej trojice spýtame **„Je pravda, že 1+1=2?“.** Použili sme zatiaľ teda **1** otázku.
2. Ak dostaneme odpoveď ÁNO, vieme že trojica bude dávať pravdivé odpovede, ak dostaneme NIE, vieme, že pravda bude opačná k tomu, čo odpovedia.
3. Týchto uzlov sa spýtame aký protokol používa každý ďalší uzol, kde použijeme ďalších ***n-3*** otázok.
4. Zostanú nám na určenie pôvodné tri uzly, ktorých protokol zistíme tak, že sa spýtame ľubovoľnej trojice SAP uzlov z už určených zvyšných uzlov. To sú ďalšie **3** otázky.
5. Dokopy na určenie teda použijeme ***n+1*** otázok [ *1*+(*n-3*)+*3* ]