In the library there is no statement of the M-N theorem, just paraphrases of the three clause of the cycle. I have introduced mn1(Alph,L), mn2(Alph,L), mn3(Alph,L) and stated the cycle strictly.

```
THM ∀Alph:Type.
    Fin(Alph) ⇒
    (∀L:(Alph*→Prop).
     (mn1(Alph,L) ⇒ mn2(Alph,L)) &
     (mn2(Alph,L) ⇒ mn3(Alph,L)) &
     (mn3(Alph,L) ⇒ mn1(Alph,L)))
```

```
mn1(Alph,L) == ∃St:Type, Auto:Automata(Alph;St). L = LangOf(Auto) & Fin(St)
```

```
mn2(Alph,L)
== ∃R:(Alph*→Alph*→Prop).
   (EquivRel on Alph*, R) c∧ (Fin(Alph*/R) &
                             (∃g:((Alph*/R)→𝔹). ∀l:Alph*. L(l) ⟺ ∪g(l)) &
                             (ExtInvariant R))
```

```
mn3(Alph,L) == Fin(Alph*/L-induced Equiv) & (∀l:Alph*. Dec(L(l)))
```

```
ExtInvariant R == ∀x,y,z:Alph*. R(x,y) ⇒ R(z @ x,z @ y)
```

```
∪g == g
```

This union operator was introduced by me to improve the match with informal concepts of theorem. Here are the relevant theorems.

```
1. Thm* (EquivRel on A, R) ⇒ (∀g:((A/R)→𝔹). ∪g ∈ A→𝔹)
2. Thm* (EquivRel on A, R) ⇒
        (∀g:((A/R)→𝔹), x:A. ∪g(x) ⟺ (∃i:A/R. g(i) & x∈i∈A/R))
3. Thm* (EquivRel on A, R) ⇒ (∀g:((A/R)→𝔹), x:A. ∪g(x) ⟺ g([x]))
4. Thm* (EquivRel on Alph*, R) ⇒
        (ExtInvariant R) ⇒
        (∀g:((Alph*/R)→𝔹).
         (∀l:Alph*. L(l) ⟺ ∪g(l)) ⇒ Alph*/L-induced Equiv = Alph*/Rg)
```