Irreflexive Ring vs. Value Comparison (≠) constraint

Are they equivalent?
Looks the same except for the ring and value comp constraints on the predicate. Much clearer as value constraints. Note the formatting problem when doing Copy and Paste on the verbalization.

Guest1 is an entity type.
Reference Scheme: Guest1 has Guest1_id.
Reference Mode: .id.
Data Type: Numeric: Auto Counter.

Fact Types:
Guest1 has Guest1_id.
Guest1 introduces Guest1 to Guest1.
Guest1 introduces Guest1 to Guest1.

It is possible that for some Guest1, and Guest1, that Guest1 introduces that Guest1, to more than one Guest1,
and that for some Guest1, and Guest1, that Guest1, introduces more than one Guest1, to that Guest1,
and that for some Guest1, and Guest1, more than one Guest1, introduces that Guest1, to that Guest1.

In each population of Guest1 introduces Guest1 to Guest1, each Guest1, Guest1, Guest1 combination occurs at most once.

This association with Guest1, Guest1, Guest1 provides the preferred identification scheme for
Guest1IntroducesGuest1ToGuest1.

No Guest1 introduces the same Guest1 to some Guest1.
No Guest1 introduces some Guest1 to the same Guest1.

It is impossible that some Guest1 introduces some Guest1 to the same Guest1.

No Guest1 introduces the same Guest1 to some Guest1.
No Guest1 introduces some Guest1 to the same Guest1.

It is impossible that some Guest1 introduces some Guest1 to the same Guest1.

Guest is an entity type.
Reference Scheme: Guest has Guest_id.
Reference Mode: .id.
Data Type: Numeric: Auto Counter.

Fact Types:
Guest has Guest_id.
Guest introduces Guest to Guest.

It is possible that for some Guest, and Guest, that Guest, introduces that Guest, to more than one Guest,
and that for some Guest, and Guest, that Guest, introduces more than one Guest, to that Guest,
and that for some Guest, and Guest, more than one Guest, introduces that Guest, to that Guest.

In each population of Guest introduces Guest to Guest, each Guest, Guest, Guest combination occurs at most once.

This association with Guest, Guest, Guest provides the preferred identification scheme for
GuestIntroducesGuestToGuest.

If Guest, introduces some Guest, to some Guest,
then Guest, is not equal to Guest.
If Guest, introduces some Guest, to some Guest,
then Guest, is not equal to Guest.
If some Guest, introduces Guest, to some Guest,
then Guest, is not equal to Guest.