Nakanishi, Y. and Suzuki, S. Osaka J. Math. 24 (1987), 217–225

ON FOX'S CONGRUENCE CLASSES OF KNOTS

Dedicated to Professor Shin'ichi Kinoshita for his 60th birthday

YASUTAKA NAKANISHI AND SHIN'ICHI SUZUKI

(Received December 3, 1985)

R.H. Fox introduced the notion of congruence classes of knots in [3], and he gave a necessary condition for congruence in terms of Alexander matrices and polynomials. In this note we will improve his condition and discuss some of its consequences.

1. Congruence classes of knots

In this note we only consider 1-dimensional tame oriented knots k in an oriented 3-sphere S^3 . Two knots k and k' are said to be *equivalent*, iff there is an orientation preserving homeomorphism from (S^3, k) onto (S^3, k') , and each equivalence class of knots is called a *knot type*. A knot k is called *trivial* (or *unknotted*) iff there exists a disk D in S^3 with $\partial D = k$.

DEFINITION (Fox [3]). Let *n* and *q* be non-negative integers. The knot types κ and λ are said to be *congruent modulo n,q*, written $\kappa \equiv \lambda \pmod{n,q}$, iff there are knots $k_0, k_1, k_2, \dots, k_l$, integers c_1, c_2, \dots, c_l , and trivial knots m_1, m_2, \dots, m_l such that

(1) k_{i-1} and m_i are disjoint,

(2) k_i is obtained from k_{i-1} by $1/c_i n$ -surgery along m_i (see [9, 10] for a/b-surgery),

(3) the linking number $lk(k_{i-1}, m_i) \equiv 0 \pmod{q}$, and

(4) k_0 represents κ , and k_1 represents λ .

DEFINITION. Two knot types κ and λ are said to be *q*-congruent modulo n, written $\kappa \equiv_{q} \lambda \pmod{n}$, iff they satisfy the conditions (1), (2), (4) in the above and the following condition (3'):

 $(3') \quad lk(k_{i-1}, m_i) = q.$

We note that these relations are equivalence relations.

Fox [3] pointed out that congruence modulo 0, q is just the knot equivalence, and that any two knot types are congruent modulo 1, q, because if a knot is obtained from another by changing an overpass to underpass, they belong to the same congruence class modulo 1, q (see Fig. 1).