Prop: Let $b \in B_n$. Then b has a canonical expression of the form $\Delta_n^k b_n^+$ for some $k \in \mathbb{Z}$, $b_n^+ \in B_n^+$.

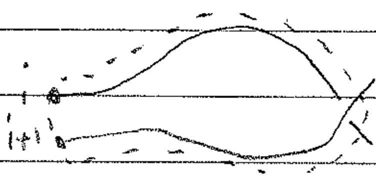
Def: A positive braid word b is *simple* if no two strands cross each other twice.

Lemma: Any two simple positive braid words b, b' which give equivalent braids are related by a sequence of R-III moves.

Pf: Suppose σ_i is the leftmost generator in b. Since b is simple, strands i and $i+1$ cross once in b.

The signed number of crossings between two strands is invariant under braid relations. Since $b' \sim b$ and b' is positive, strands i and $i+1$ cross once in b'.

Thus we have the following local picture in b':



Other strands may appear but no two strands may cross twice. Each crossing in the interior of the region cut off by strands $i, i+1$ either cuts of an upper quadrant which is an empty triangle, or contains another crossing which

cuts off an empty upper triangle. In either case, an R-III move will reduce the number of interior crossings in the picture by 1. After all such crossings are removed, the crossing of strands $i, i+1$ may be slid left via R-III moves.
$\boxtimes$

Corollary: Simple braids are classified by the list of pairs of strands which cross.

<u>Construction for proposition:</u>

First, replace each σ_i^{-1} which appears with $\Delta_n^{-1}\Delta_n\sigma_i^{-1}$, where $\Delta_n\sigma_i^{-1}$ can be expressed as a positive braid by sliding a crossing to the right and cancelling with σ_i. Use the commutativity of Δ_n to slide all copies of Δ_n^{-1} to the left. This gives us $\Delta_n^{-k}b$ for some $k \in \mathbb{N}$, $b \in B_n^+$.

Then perform the algorithm described below, which transforms b via a series of R-III moves into a sequence $S = (s_1, \ldots, s_l)$ of simple positive braid words which are <u>maximal</u> in the sense that for each $j>1$, every crossing which may be slid to the leftmost position in s_j crosses a pair of strands which are crossed in s_{j-1}.

<u>Note</u>: Maximality implies that if $s_j \sim \Delta_n$, then $s_i \sim \Delta_n \, \forall i < j$.

<u>The algorithm:</u>

Set $S_0 = ()$.

Let $b = w_1 \ldots w_j$ be the positive braid word.

Let $S_k = (s_{k,1}, s_{k,2}, \ldots, s_{k,i})$ be the sequence of maximal simple braid words constructed at step k. Construct S_{k+1} as follows:

Call subroutine $Add(S_k, i, w_{k+1})$. If this succeeds, set S_{k+1} to the result. If it fails, set $S_{k+1} = (s_{k,1}, \ldots, s_{k,i}, w_{k+1})$.

$Add(S, j, w)$:

If the word $s_j w$ is not simple, FAIL.

If w may be slid to the left in $s_j w$ and $j > 1$, call $Add(S, j-1, w)$. If this succeeds, return the result.

In any other case, return

$(s_1, \ldots, s_{j-1}, s_j w, s_{j+1}, \ldots, s_i)$,

where $(s_1, \ldots, s_i) = S$.

One then obtains the word $\Delta_n^{-l} \Delta_n^m S'$ where $S = \Delta_n^m S'$, with S' a sequence of maximal simple positive braids not equivalent to Δ_n, and $l, m \in N$.

Then $\Delta^{m-\ell} S'$ is the Greedy Normal Form (GNF) for b.

Proof of proposition

It is not difficult to verify that at each stage t one has

1) $w_1 \ldots w_t \sim S_t$ via a sequence of R-III moves.
2) S_t is a sequence of maximal simple positive braid words.

We wish to show that the GNF is invariant under braid relations.

a) $\sigma_i^{-1}\sigma_i = \emptyset$ and $\sigma_i\sigma_i^{-1} = \emptyset$:

After replacing σ_i^{-1} one finds that $x\sigma_i\sigma_i^{-1}y$ becomes $x\Delta_n\Delta_n^{-1}y$ which gets sent to $\Delta_n^{-1}\bar{x}\Delta_n y$, where $\bar{x}$ is a $180°$ rotation of x about the axis along which the braid runs. This is equivalent by R-III moves to $\Delta_n^{-1}\Delta_n x y$. For any word z, it is easy to check that the algorithm on positive braid words gives $GNF(\Delta_n z) = \Delta_n GNF(z)$. Thus it suffices to check invariance under R-III moves.

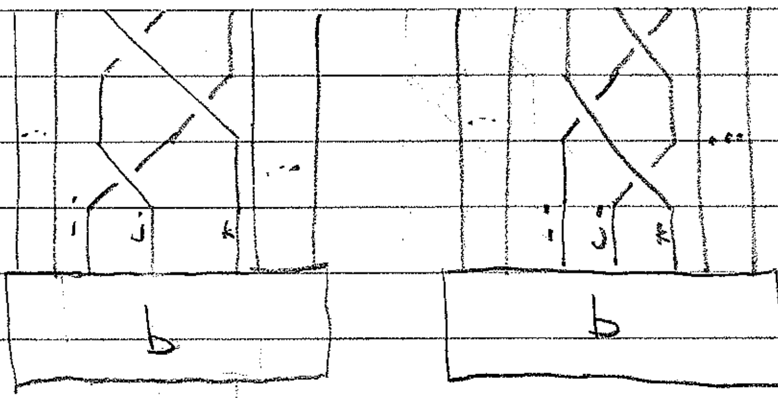
b) R-III moves.

The two sequences $\sigma_i\sigma_{i+1}\sigma_i$ and $\sigma_{i+1}\sigma_i\sigma_{i+1}$

cross the same three distinct pairs of strands.

In the above algorithm a generator σ_i can slide through a simple braid b iff the strands crossed in σ_i are not crossed in b and are adjacent on the far side of b.

consider:



On the right, ${}_j\times_k$ passes through b if at the bottom we have strands $\cdots jk \cdots$. On the left, ${}_j\times_k$ passes through b if we have $\cdots jk \cdots$ for any compatible position of i.
Similarly, ${}_i\times_j$ and ${}_i\times_k$ pass through b for the same choices of strand orderings on the right and left pictures.

Furthermore ${}_i\times_j$ is absorbed into b under the same conditions on the bottom strands for both pictures, namely that i is to the left of and not adjacent to j.

Crossings j,k and i,k work similarly.
Thus the algorithm places crossings (i,j), (j,k) and (i,k) in the same simple braids s_ℓ, s_m, s_n respectively, for either side of the braid relation.
Therefore, it produces the same GNF.