Lindsey Wingate
Formal Language and Autonoma
November 9, 2016

<u>Proof</u>
Let abDiff(w) for string w be the number of a's in the string mius the number of b's.

Let the set of strings abEqual be defined as $\{w \in \{a,b\}^* \mid abDiff(w) = 0\}$.

Consider the grammar $G=(NT, \Sigma, R, S)$ where $NT = \{S,T,U\}$, $\Sigma = \{a, b\}$ and the rules in R are:

S->T|U|SS
T->aSb|*e*
U->bSa|*e*

1. If s produces a string w of terminal and or nonterminal symbols, then the difference of a's and b's in w, or abDiff, will be 0.
    The only symbols that allow the addition of an a or b when constructing a string are aSb and bSa.
    If a=1 and b=-1, each time one of these pairs is added they will cancel each other out.
    Let n equal any integer equal or greater than 0.
    an + bn, in other words, 1n + -1n or n-n = 0.
    Any combination of an+bn will be even.
    Therefore, any combinations of the terminals in the grammar will result in the abDiff equaling 0.
2. If a string w is in abEqual, then S can produce w.
    Base Case 1:
        Let the length of w equal n. Let n equal 0.
        S branches to T which branches to *e*. Also, S branches to U which branches to *e*.
        Therefore, S can produce the empty string, or w at length 0.
    Base Case 2:
        Let the length of w equal n. Let n equal 2.
        S branches to T which branches to aSb, producing an equal number of a's and b's (with an empty string in S).
        S also branches to U, which branches to bSa, also producing an equal number of a's and b's (with an empty string in S).
        Any grammar rule adding a's and b's to a constructed string adds them equally so abDiff(w) is always 0.
        Therefore, S can produce a non-empty string where abDiff(w)=0 and any w has an even length.

    Induction:

Suppose (2) is true for n=k. Let n=k+2. The language the grammar G produces any string of length k+2 or less.

abEqual only produces strings of an even length (see 1 for proof).

Case 1:

Let w equal aabb.

Case 2:

Let w equal abab.

Case 3:

Let w equal bbaa.

Case 4:

Let w equal baba.

Therefore, If a string w is in abEqual, then S can produce w