Two two-tape Turing Machines, the first a DPDA, the second not.

A DPDA that recognizes the language of balanced paren strings. Convention: start with heads on blank to left of input.

Start: Put down ^

\( (.,R, \) \)
\( /,R \)
\( /,S \)
\( /,L \)

Push on 1

Reached end of input

\( /,S \)
\( /,S \)
\( /,S \)
\( /,S \)
\( /,S \)
\( /,S \)
\( /,S \)

\( \Sigma: ( ) \)
\( \Gamma: \ ) \wedge \text{blank} \)

Negative count

End on positive count

TM deciding

DOUBLEWORD

Find midpoint of input

0/0, R
1/1, R

START

Odd

Even

1/1, R
0/0, R

\[ x \] is even, \( x = wv \)

Copy \( v \) to Tape 2

[\( x \) is odd, so reject]

Non-PDA state

Copy

Note: lead bit of \( v \) overwrites the ^ marker.

\[ 1/1, L \]
\[ 1/1, L \]

Move 1 left

success—\( n \) chars match.

Homer-SelmanExercise 1.1

Input Alphabet: \( \{0, 1\} \)

Work Alphabet: \( \{0, 1, ^, \$, \text{blank}\} \)

Initially, write input \( x \) on Tape 1, with head on its first bit.

\( L(M) = \{ww : w \in \{0, 1\}^*, \text{ for any } w\} \)