

Report for DB/src/Dictionary.java -- Constructor

1.0 - Source Code Based Testing:

1.1 - Control Flow Based Testing:

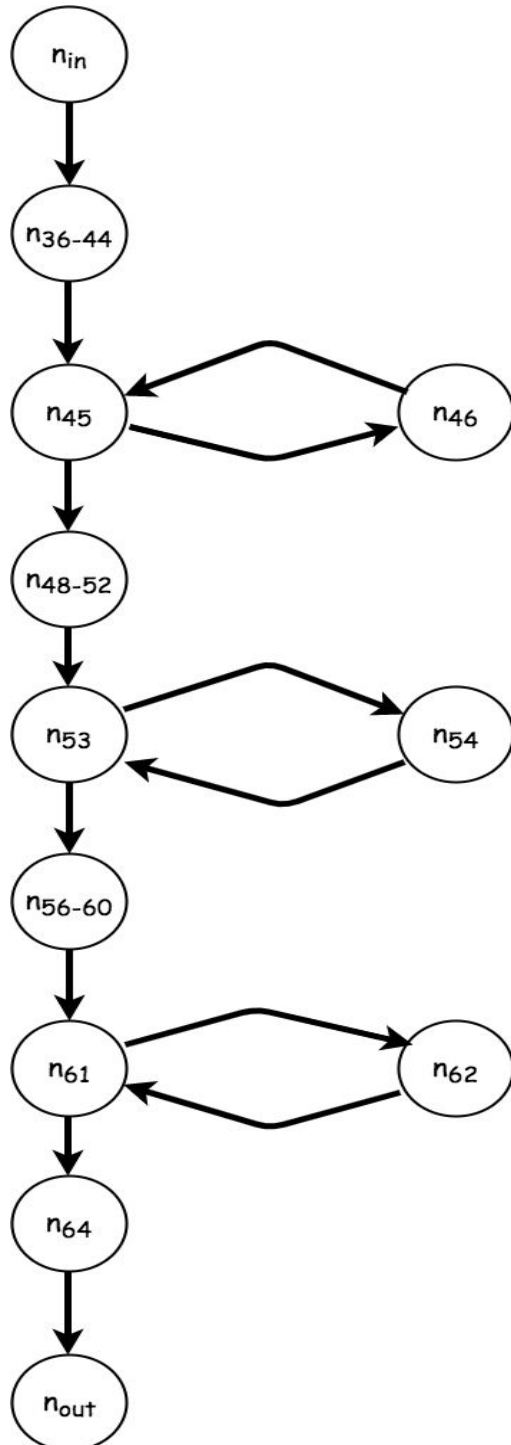


Exhibit 1: Control Flow Diagram

Since there are no conditionals, C_{3b} cannot be tested, Therefore, I will only perform Boundary Interior Testing, which in this specific case subsumes C_2 which always subsumes C_1 criteria.

1.1.1 Boundary Interior Analysis

LOOP COVERAGE

ID	PARAMETERS			Coverage	
	eng	spec	bad	path	%
T_0_0	"/path/to/empty.txt"	"/path/to/empty.txt"	"/path/to/empty.txt"	In, 36-44, 45, 48-52, 53, 56-60, 61, 64, out	$\frac{1}{3} = 33.3\%$
T_0_1	"/path/to/single_entry.txt"	"/path/to/single_entry.txt"	"/path/to/single_entry.txt"	In, 36-44, 45, 46, 48-52, 53, 54, 56-60, 61, 62, 64, out	$\frac{2}{3} = 66.6\%$
T_0_2	"/path/to/two_entries.txt"	"/path/to/two_entries.txt"	"/path/to/two_entries.txt"	In, 36-44, $(45, 46)^2$, 48-52, $(53, 54)^2$, 56-60, $(61, 62)^2$, 64, out	$\frac{3}{3} = 100\%$

1.2 - Data Flow Based Testing

SEE DATA FLOW DIAGRAM IN THE NEXT PAGE

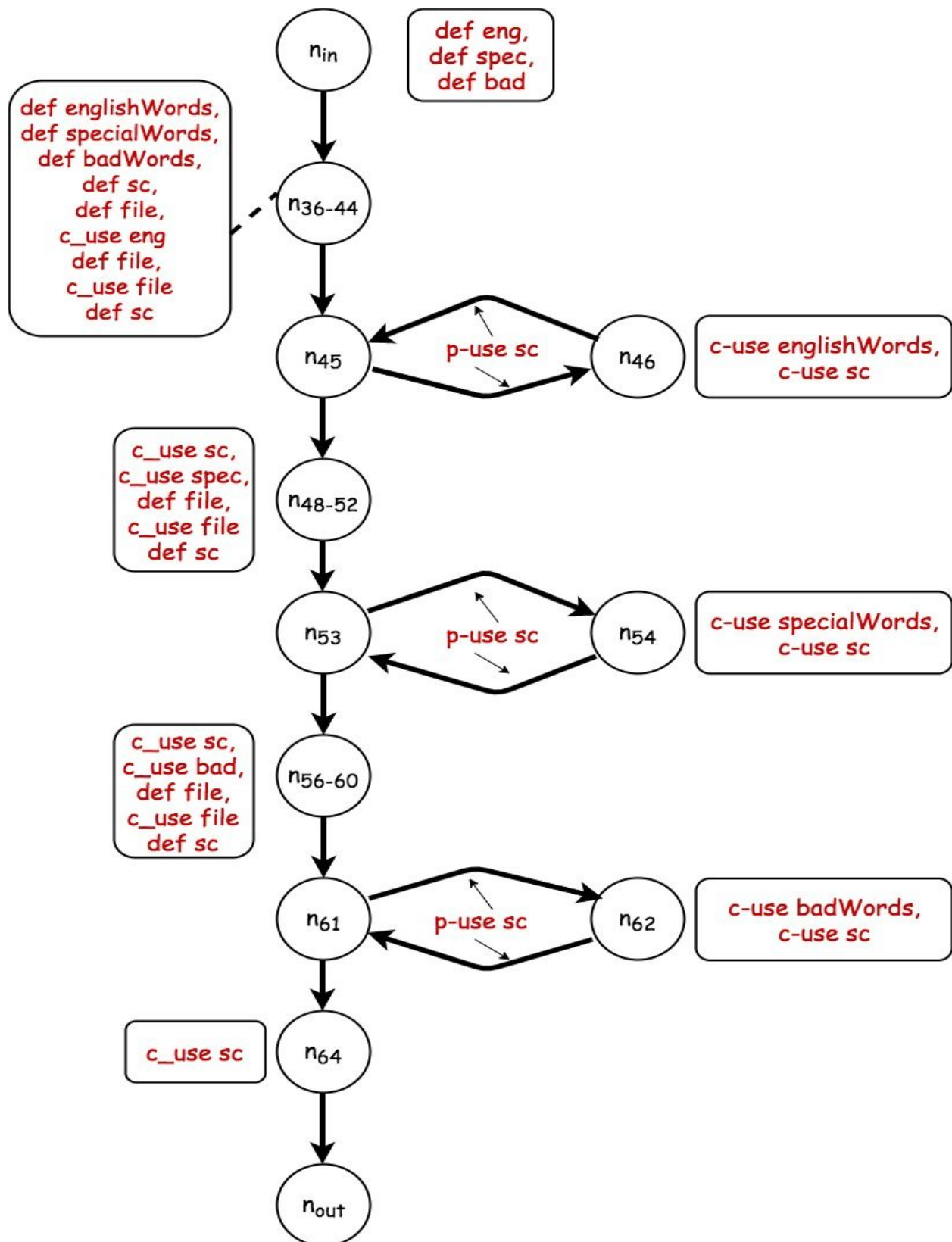


Exhibit 2: Data Flow Diagram

DEF/C-USE SET(S)

Node n	DEF(N)	C-USE(N)
in	{ eng, spec, bad }	∅
36-44	{ englishWords, specialWords, badWords, sc, file }	{ eng, file }
45	∅	∅
46	∅	{ englishWords, sc }
48-52	{ file, sc }	{ sc, spec, file }
53	∅	∅
54	∅	{ specialWords, sc }
56-60	{ file, sc }	{ sc, bad, file }
61	∅	∅
62	∅	{ badWords, sc }
64	∅	{ sc }
out	∅	∅

P-USE SET(S)

EDGE (n, m)	P-USE(N)
45-46	{ sc }
53-54	{ sc }
61-62	{ sc }

DCU/DPU SETS

Node n	Variable x	DCU(x, n)	DPU(x, n)
in	eng	{ 36-44 }	Ø
in	spec	{ 48-52 }	Ø
in	bad	{ 56-60 }	Ø
36-44	englishWords	{ 46 }	Ø
36-44	specialWords	{ 54 }	Ø
36-44	badWords	{ 62 }	Ø
36-44	file ₁	{ 36-44 }	Ø
48-52	file ₂	{ 48-52 }	Ø
56-60	file ₃	{ 56-60 }	Ø
36-44	sc ₁	{ 46, 48-52 }	{ 45-46 }
48-52	sc ₂	{ 54, 56-60 }	{ 53-54 }
56-60	sc ₃	{ 62, 64 }	{ 61-62 }

All-Use criteria already met by boundary analysis tests... all DCUs and DPUs executed

2.0 - Specification Based Testing

3.0 - Object Oriented Testing