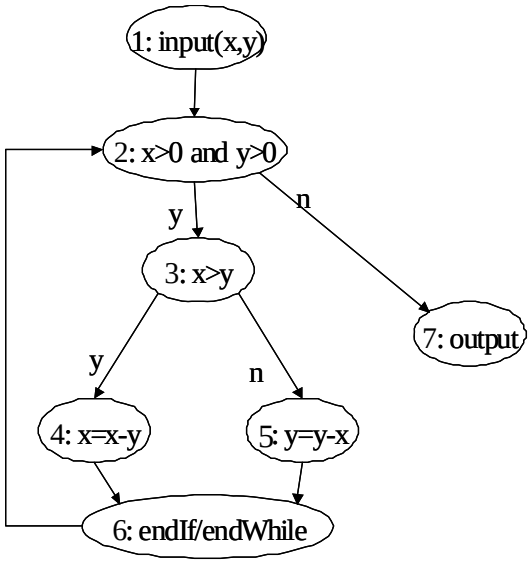
INF 1050
UKEOPPGAVER 12: TESTING (WHITE-BOX TESTING)
LØSNINGSFORSLAG

---

1) Control Flow Graph



2) Test sets (in form of paths) and test inputs

a) Statement coverage

T={(12346235627)} Input(X,Y)=(2,1)

b) Edge coverage

T={(12346235627)} Input(X,Y)=(2,1)

c) Condition coverage: (two clauses: x>0, y>0)

T={(12346235627)} Input (X,Y)=(2,1)

This covers the cases where both clauses are true and when the second one becomes false.

Covering the two other truth value combinations requires to input values of the form (0,*) and (0,0), respectively. Those, however, should result into exceptions being raised or the output will be meaningless. In that case, only two truth value combinations already covered by statement/edge coverage can be exercised without triggering error conditions. This is a peculiar case. For the sake of sake of simplicity, the current program just outputs the invalid result. In practice one may want to test these conditions to test the robustness of the software.

d) Path coverage

Heuristic: Loop skipped (Error condition), taken once, 3 times

A=2346, B=2356

T={	(127),	skipped, Input(X,Y)=(0,1)
	(1A27)	Impossible (requires x>y and x-y<=0)
	(1B27)	Once, Input(X,Y)=(2,2) (requires x=y)
	(1AAA27)	Impossible
	(1ABA27)	Impossible
	(1AAB27)	3 times, Input(X,Y)=(3,1) (requires x = 3y)
	(1ABB27)	3 times, Input(X,Y)=(3,2) (requires 2x = 3 y)
	(1BBB27)	3 times, Input(X,Y)=(1,3) (requires 3x = y)
	(1BAA27)	Impossible
	(1BBA27)	Impossible
	(1BAB27)	3 times, Input(X,Y)=(2,3) (requires 2y=3x)
	}	

Detailed Examples: solving inequalities

Summary of procedure:
xi, yi denote values of x, y at iteration i
x1, y1 are the original inputs
For each iteration we compute valid inequalities in terms of x1 and y1
If we find contradictions, this is an impossible path

(1ABA27): Impossible

x1 > y 1
x2 = x1 – y1
x2 <= y1 ⇔ x1 <= 2 y1
y2 = y1-x2
x2 > y2 ⇔ x1 – y1 > y1 – x1 + y1 ⇔ 2x1 > 3 y1
x3 = x2 – y2
x3 <= 0 ⇔ x2 – y2 <= 0 ⇔ x1 – y1 – y1 +x1 – y1 <= 0 ⇔ 2x1 <= 3y1

(1BAB27): Possible (2 y1 = 3 x1)

x1 <= y1
y2 = y1 – x1
x1 > y2 ⇔ 2x1 > y1
x2 = x1 – y2
x2 <= y2 ⇔ x1 <= 2y1 – 2x1 ⇔ 3x1 <= 2y1
y3 = y2 – x2
y3 <= 0 ⇔ y1 – 2x1 + y1 – x1 <= 0 ⇔ 2y1 <= 3 x1

***