DATA FLOW COVERAGE MEASURES (I)

(1) def(x)

set of all nodes in CFG with def access to x;

(2) **p-use(x**)

set of all nodes in CFG with p-use access to x;

(3) c-use(x)

set of all nodes in CFG with c-use access to x;

(4) **def-clear(x)**

= { (s_0, s_n) ,

$s_0 \in def(x)$, $s_i \notin def(x)$, $\forall i = 1, \ldots, n$,

$s_i \neq s_j$, $\forall i, j = 1, \ldots, n, i \neq j$ }

$\forall s \in def(x)$ we define

(5) **dpu(s,x)**

= { s' ,

$(s, s') \in def-clear(x)$ and

$s' \in p-use(x)$ }

(6) **dcu(s,x)**

= { s' ,

$(s, s') \in def-clear(x)$ and

$s' \in c-use(x)$ }

(7) **du(s,x)**

= $dpu(s, x) \cup dcu(s, x)$

DATA FLOW COVERAGE MEASURES (II)

For all program variables *x* and for all statements *s* in *def(x)*,
the test suite contains the following paths $(s, s') \in def-clear(x)$:

(a) **ALL-DEFS**:

one path (s, s'), $s' \in dpu(s, x) \cup dcu(s, x)$

-> each variable definition is used at least once

(b) **ALL-P-USES**: *-> covers branch testing*

one path $(s, s') \forall s' \in dpu(s, x)$

-> each variable definition is tested in all its predicate uses

(c) **ALL-C-USES**:

one path $(s, s') \forall s' \in dcu(s, x)$

-> each variable definition is tested in all its computational uses

(d) **ALL-P-USES/SOME-C-USES**: *-> covers (a), (b)*

all-p-uses $\cup$

if $dpu(s, x) = \emptyset$ then one path (s, s') for one $s' \in dcu(s, x)$ fi

(e) **ALL-C-USES/SOME-P-USES**: *-> covers (a), (c)*

all-c-uses $\cup$

if $dcu(s, x) = \emptyset$ then one path (s, s') for one $s' \in dpu(s, x)$ fi

(f) **ALL-USES**: *-> covers (e), (d)*

one path $(s, s') \forall s' \in du(s, x)$

(g) **ALL-DU-PATHS** *-> covers (f)*

all paths $(s, s') \forall s' \in du(s, x)$