Addendum to the Article "Types vs. PDGs in Information Flow Analysis" – Proofs and Operational Semantics

Heiko Mantel and Henning Sudbrock

Computer Science Department, TU Darmstadt, Germany
{mantel,sudbrock}@mais.informatik.tu-darmstadt.de

This document contains proofs for theorems from the article "*Types vs. PDGs in Information Flow Analysis*" [MS13] (in Sections 1 and 2). Moreover as an addendum to the article it contains the operational semantics for the considered programming language (in Section 3).

1 Proof of Lemma 1

Before proving Lemma 1 from [MS13] we prove several propositions that relate paths in the graph $PDG(CFG_c^{I,O})$ where c is of the form $\mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}$, $\mathsf{while}\ (e)\ \mathsf{do}\ c_1\ \mathsf{od}$, or $c_1; c_2$ to paths in the graphs $PDG(CFG_{c_1}^{I,O})$ and (if applicable) $PDG(CFG_{c_2}^{I,O})$. In the proofs, we write $p + k$ for the path that is obtained from p by adding k to each node on p that is a natural number, and leaving *start* and *stop* unchanged. Moreover, we write $p - k$ for the path that is obtained from p by subtracting k from each node on p that is a natural number, and leaving *start* and *stop* unchanged.

Proposition 1. *Let* $c = \mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}$ *and* $x, y \in Var$. *Then the following hold:*

1. *There is a path from in to out in* $PDG(CFG_c^{\{x\},\{y\}})$ *that contains more than* 2 *nodes if and only if one of the following conditions are satisfied:*
   (a) *there is a path from in to out in* $PDG(CFG_{c_1}^{\{x\},\{y\}})$,
   (b) *there is a path from in to out in* $PDG(CFG_{c_2}^{\{x\},\{y\}})$,
   (c) $x \in fv(e)$ *and there is a path from start to out in* $PDG(CFG_{c_1}^{\{x\},\{y\}})$ *that contains more than* 2 *nodes, or*
   (d) $x \in fv(e)$ *and there is a path from start to out in* $PDG(CFG_{c_2}^{\{x\},\{y\}})$ *that contains more than* 2 *nodes.*
2. *There is a path from start to out in* $PDG(CFG_c^{\{x\},\{y\}})$ *that contains more than* 2 *nodes if and only there is such a path in* $PDG(CFG_{c_1}^{\{x\},\{y\}})$ *or in* $PDG(CFG_{c_2}^{\{x\},\{y\}})$.

Proof. We firstly prove Statement (1) of the proposition.

1. By the construction of the CFG for commands the following holds:
   (a) Let $n, n' \notin \{1, start, stop\}$ be nodes of CFG_c. Then n' is data (control) dependent on n for CFG_c if and only if $n' \ominus 1$ is data (control) dependent on

$n \ominus 1$ for CFG_{c_1} or $n' \ominus 1 \ominus |c_1|$ is data (control) dependent on $n \ominus 1 \ominus |c_1|$ for CFG_{c_2}.

Moreover, n is not data dependent on 1 and 1 is not data dependent on n. Moreover, n is control dependent on 1 if and only if $n \ominus 1$ is control dependent on *start* for CFG_{c_1} or $n \ominus 1 \ominus |c_1|$ is control dependent on *start* for CFG_{c_2}.

2. By the construction of the PDG and the construction of the CFG for commands we obtain the following:
   (a) Let $n \notin \{1, start, stop\}$ be a node of CFG_c. Then there is an edge (in, n) in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there is an edge $(in, n \ominus 1)$ in the graph $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or an edge $(in, n \ominus 1 \ominus |c_1|)$ in $PDG(CFG_{c_2}^{\{x\},\{y\}})$.
   Moreover, there is an edge (n, out) in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there is an edge $(n \ominus 1, out)$ in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or an edge $(n \ominus 1 \ominus |c_1|, out)$ in $PDG(CFG_{c_2}^{\{x\},\{y\}})$.
   (b) There is an edge $(in, 1)$ in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if $x \in fv(e)$. Moreover, there is no edge $(1, out)$ in $PDG(CFG_c^{\{x\},\{y\}})$.
   (c) There is an edge (in, out) in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there is an edge (in, out) in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or in $PDG(CFG_{c_2}^{\{x\},\{y\}})$.
3. Assume that there is a path p from *in* to *out* in $PDG(CFG_c^{\{x\},\{y\}})$.
   (a) If $p = \langle in, out\rangle$, then by (2c) p is also a path in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or in $PDG(CFG_{c_2}^{\{x\},\{y\}})$.
   (b) If $p = \langle in, 1\rangle.p'$ for some p', then, by (2b), $x \in fv(e)$ and $p' \neq \langle out\rangle$. Moreover, by (1a) the first node in p' is control dependent on *start* in CFG_c, and, since due to (1a) all edges in p' derive from edges in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or $PDG(CFG_{c_1}^{\{x\},\{y\}})$, $\langle start\rangle.(p' \ominus 1)$ is a path from *start* to *out* that contains more than 2 nodes in the graph $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or in the graph $PDG(CFG_{c_1}^{\{x\},\{y\}})$.
   (c) If $p \neq \langle in, out\rangle$ and $p \neq \langle in, 1\rangle.p'$ for some p', then, using (1a) and (2a), $p - 1$ is a path in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ or in $PDG(CFG_{c_1}^{\{x\},\{y\}})$.
4. Assume that there is a path p from *in* to *out* in $PDG(CFG_{c_1}^{\{x\},\{y\}})$. Then, by (1a), (2a), and (2c), $p + 1$ is a path in $PDG(CFG_c^{\{x\},\{y\}})$.
5. Assume that there is a path from *in* to *out* in $PDG(CFG_{c_2}^{\{x\},\{y\}})$. Then, by (1a), (2a), and (2c), $p + (1 + |c_1|)$ is a path in $PDG(CFG_{c_2}^{\{x\},\{y\}})$.
6. Assume that $x \in fv(e)$ and that there is a path from *start* to *out* in the graph $PDG(CFG_{c_1}^{\{x\},\{y\}})$, i.e., of the form $\langle start\rangle.p'$. Then, by (1a), (2a), and (2b), the sequence $\langle in, 1\rangle.(p' + 1)$ is a path in $PDG(CFG_c^{\{x\},\{y\}})$.
7. Assume finally that $x \in fv(e)$ and that there is a path from *start* to *out* in $PDG(CFG_{c_2}^{\{x\},\{y\}})$ that contains more than 2 nodes, i.e., of the form $\langle start\rangle.p'$. Then, by (1a), (2a), and (2b), the sequence $\langle in, 1\rangle.(p' + (1 + |c_1|))$ is a path in $PDG(CFG_c^{\{x\},\{y\}})$.

Statement (2) of the proposition is seen as follows:. Using (1) and (2) of the proof of the first statement of the proposition, paths from Node *start* to Node *out* in

$PDG(CFG_c^{\{x\},\{y\}})$ that contains more than 2 nodes are exactly the paths obtained from paths from *start* to *out* in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ and $PDG(CFG_{c_2}^{\{x\},\{y\}})$, respectively, into which the node representing the guard of the conditional (i.e., Node 1) is inserted directly after node *start*, and 1 respectively $1+|c_1|$ is added to each node.

Proposition 2. *Let $c = c_1; c_2$ and $x, y \in Var$. Then the following hold:*

1. *There is a path from in to out in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there exists $z \in Var$ such that there is a path from in to out in $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and a path from in to out in $PDG(CFG_{c_2}^{\{z\},\{y\}})$.*
2. *There is a path from start to out in $PDG(CFG_c^{\{x\},\{y\}})$ that contains more than 2 nodes if and only if one of the following conditions is satisfied:*
   (a) *there exists $z \in Var$ such that there is a path from start to out that contains more than 2 nodes in the graph $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and a path from in to out in $PDG(CFG_{c_2}^{\{z\},\{y\}})$, or*
   (b) *there is a path from start to out in $PDG(CFG_{c_2}^{\{x\},\{y\}})$ that contains more than 2 nodes.*

Proof. We firstly prove Statement (1).

1. By the construction of the CFG for commands, the following hold:
   (a) Let $n, n' \notin \{start, stop\}$ be nodes of CFG_c. Then n' is control dependent on n for CFG_c, if and only if n' is control dependent on n for CFG_{c_1}, or if $n \ominus |c_1|$ is control dependent on $n \ominus |c_1|$ for CFG_{c_2}.
   (b) Let $n, n' \notin \{start, stop\}$ be nodes of CFG_c. Then n' is data dependent on n for CFG_c if and only if one of the following conditions is satisfied:
      i. n' is data dependent on n for CFG_{c_1}
      ii. $n' \ominus |c_1|$ is data dependent on $n \ominus |c_1|$ for CFG_{c_2}
      iii. There exists $z \in Var$ such that $z \in def_{c_1}(n)$, $z \in use_{c_2}(n' \ominus |c_1|)$, a definition of z at n reaches *stop* in CFG_{c_1}, and a definition of z at *start* reaches $n' \ominus |c_1|$ in CFG_{c_2}.

      Note that condition (iii) is equivalent to the existence of $z \in Var$ such that there is an edge from n to *out* in $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and an edge from *in* to $(n' \ominus |c_1|)$ in $PDG(CFG_{c_2}^{\{z\},\{y\}})$.
2. By the definition of PDGs and the definition of CFGs of commands there is an edge (in, out) in $PDG(CFG_c^{\{z\},\{y\}})$ if and only if $x = y$ and (in, out) is an edge both in $PDG(CFG_{c_1}^{\{z\},\{y\}})$ and in $PDG(CFG_{c_2}^{\{z\},\{y\}})$.
3. Assume that there is a path p from *in* to *out* in $PDG(CFG_c^{\{x\},\{y\}})$.
   (a) If $p = \langle in, out \rangle$, then, by (2), we conclude (setting $z = x = y$).
   (b) If p contains, besides *in* and *out*, only nodes in the set $\{1, \ldots, |c_1|\}$, then a definition of y at the one but last node of p reaches *stop* in CFG_c, and, hence, a definition of y at *start* reaches *stop* in CFG_{c_2}. Hence, p is a path in $PDG(CFG_{c_1}^{\{x\},\{y\}})$ and $\langle in, out \rangle$ is a path in $PDG(CFG_{c_2}^{\{y\},\{y\}})$. Thus, we conclude setting $z = y$.

(c) If p contains, besides *in* and *out*, only nodes in $\{|c_1|+1,\dots,|c_1;c_2|\}$, we argue as in the previous case with roles of c_1 and c_2 switched, concluding by setting $z = x$.

(d) If p contains nodes in both $\{1,\dots,|c_1|\}$ and $\{|c_1|+1,\dots,|c_1;c_2|\}$, then, by the definition of the CFG, $p = p_1 . p_2$ where p_1 contains only nodes in $\{1,\dots,|c_1|\}$ and p_2 contains only nodes in $\{|c_1|+1,\dots,|c_1;c_2|\}$. With (1b.iii) it follows that there is z such that $p_1 . \langle out\rangle$ is a path in the graph $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and $\langle in\rangle . p_2$ is a path in $PDG(CFG_{c_2}^{\{z\},\{y\}})$.

4. Now assume that there exists $z \in Var$ and paths p_1 and p_2 from *in* to *out* in $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and in $PDG(CFG_{c_2}^{\{z\},\{y\}})$, respectively. With (1b.iii) it follows that there exists a path from *in* to *out* in $PDG(CFG_{c}^{\{x\},\{z\}})$ (by joining these two paths).

We now prove Statement (2).

1. By the construction of the CFG for $c_1;c_2$ and the definition of postdominance, a node n is control dependent on *start* in CFG_c if and only if n is control dependent on *start* in CFG_{c_1} or $n \ominus |c_1|$ is control dependent on *start* in CFG_{c_2}.
2. Assume that p is a path from *start* to *out* in $PDG(CFG_{c}^{\{x\},\{y\}})$ that contains more than 2 nodes.
   (a) If p contains, besides *start* and *out*, only nodes in $\{|c_1|+1,\dots,|c_1;c_2|\}$, then $p - |c_1|$ is a path from *start* to *out* in $PDG(CFG_{c_2}^{\{x\},\{y\}})$ that contains more than 2 nodes (by (1), (1a), (1b), and the proof of the first statement of the proposition).
   (b) Otherwise, using (1) and arguing analogously to the proof of the first part of the proposition, there exists z such that there are paths from *start* to *out* in the graph $PDG(CFG_{c_1}^{\{x\},\{z\}})$ (containing more than 2 nodes) and from *in* to *out* in the graph $PDG(CFG_{c_1}^{\{z\},\{y\}})$.
3. The backwards direction (assuming paths in the graphs $PDG(CFG_{c_1}^{\{x\},\{y\}})$ and $PDG(CFG_{c_2}^{\{x\},\{y\}})$) is analogous to the previous cases.

Proposition 3. *Let $c = \mathsf{while}\,(e)\,\mathsf{do}\; c_1\;\mathsf{od}$. Then the following hold:*

1. There is a path from in to out in $PDG(CFG_{c}^{\{x\},\{y\}})$ if and only if there exist $z_1,\dots,z_k$ (for some $k > 1$) with $z_1 = x$ and $z_k = y$ such that for each $i \in \{1,\dots,k-1\}$ one of the following conditions is satisfied:
(a) there is a path from in to out in $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$, or
(b) $z_i \in fv(e)$ and there is a path from start to out in $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$ that contains more than 2 nodes.

2. There is a path from start to out in $PDG(CFG_{c}^{\{x\},\{y\}})$ that contains more than 2 nodes if and only if there exist $z_1,\dots,z_k$ (for some $k > 0$) such that $z_k = y$, there is a path from start to out in $PDG(CFG_{c_1}^{\{x\},\{z_1\}})$ that contains more than 2 nodes, and for each $i \in \{1,\dots,k-1\}$ one of the following conditions is satisfied:
(a) there is a path from in to out in $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$, or

(b) $z_i \in fv(e)$ *and there is a path from start to out in* $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$ *that contains more than 2 nodes.*

Proof. We start proving Statement (1) of the proposition.

1. By the construction of the CFG for commands the following hold:
   (a) Let $n, n' \notin \{1, start, stop\}$ be nodes of CFG_c. Then n' is data dependent on n for CFG_c if and only if one of the following holds:
      i. $n' \ominus 1$ is data dependent on $n \ominus 1$ for CFG_{c_1} or
      ii. there exists $z \in Var$ such that $(n \ominus 1, out)$ is an edge in the graph $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and $(in, n' \ominus 1)$ is an edge in $PDG(CFG_{c_1}^{\{z\},\{y\}})$.

   Moreover, Node 1 is data dependent on n if and only if there exists $z \in fv(e)$ such that (n, out) is an edge in $PDG(CFG_{c_1}^{\{x\},\{z\}})$.

   (b) Let $n, n' \notin \{1, start, stop\}$ be nodes of CFG_c. Then n' is control dependent on n for CFG_c if and only if $n' \ominus 1$ is control dependent on $n \ominus 1$ for CFG_{c_1}.

   Moreover, n is control dependent on 1 if and only if $n \ominus 1$ is control dependent on $start$ for CFG_{c_1}.
2. By the construction of the PDG and the construction of the CFG for commands the following hold:
   (a) Let $n \notin \{1, start, stop\}$ be a node of CFG_c. Then there is an edge (in, n) in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there is an edge $(in, n \ominus 1)$ in the graph $PDG(CFG_{c_1}^{\{x\},\{y\}})$.

   Moreover, there is an edge (n, out) in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there is an edge $(n \ominus 1, out)$ in $PDG(CFG_{c_1}^{\{x\},\{y\}})$.

   (b) There is an edge $(in, 1)$ in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if $x \in fv(e)$. Moreover, there is no edge $(1, out)$ in $PDG(CFG_c^{\{x\},\{y\}})$.

   (c) There is an edge (in, out) in $PDG(CFG_c^{\{x\},\{y\}})$ if and only if there is an edge (in, out) in $PDG(CFG_{c_1}^{\{x\},\{y\}})$.
3. Assume that there is a path p from in to out in $PDG(CFG_c^{\{x\},\{y\}})$.
   (a) If $p = \langle in, out \rangle$, then by (2c) it is also a path in $PDG(CFG_{c_1}^{\{x\},\{y\}})$.

   (b) If p consists of more than 2 nodes, then by (1) and (2) p consists of segments that correspond to paths in $PDG(CFG_{c_1}^{\{x\},\{y\}})$, potentially separated by Node 1 (representing the guard of the loop), where the edges (n, n') between these segments correspond to edges (n, out) in $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$ and (in, n') respectively $(start, n')$ in $PDG(CFG_{c_1}^{\{z_{i+1}\},\{z_{i+2}\}})$ for a sequence of z_i.

   (c) Hence, there exist $z_1, \ldots, z_k$ and paths from in to out respectively from $start$ to out (containing more than 2 nodes when starting an $start$) in the PDGs $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$, where $z_1 = x$ and $z_k = y$.
4. Assume now that there exist $z_1, \ldots, z_k$ with $z_1 = x$ and $z_k = y$ and paths from in to out respectively from $start$ to out (containing more than 2 nodes) in the PDGs $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$.
   (a) Using (1) and (2), these paths can be concatenated using the same arguments as in Propositions 1 and 2, where Node $start$ is replaced by Node 1.

Statement (2) of the proposition is proven analogously to Statement (1), with the only difference being that the first segment of the path now begins at Node *start*, not at Node *in*.

We prove a generalization of Lemma 1 from [MS13] that permits arbitrary security domains for the program counter in the typing judgment. The generalization permits an inductive proof over the construction of the command c.

Proposition 4. *Let $c \in Com$, $\Gamma : Var \to \mathcal{D}$, $pc \in \mathcal{D}$, and $y \in Var$. Let Γ' be the environment with $pc \vdash \Gamma\,\{c\}\,\Gamma'$. Let $X \subseteq Var$ be such that $x \in X$ if and only if there is a path $\langle in, \ldots, out\rangle$ in $PDG(CFG_c^{\{x\},\{y\}})$. Then one of the following two conditions is satisfied:*
1. *$\Gamma'(y) = pc \sqcup \left(\bigsqcup_{x \in X} \Gamma(x)\right)$, and there is a path from start to out in the graph $PDG(CFG_c^{\{x\},\{y\}})$ for some $x \in Var$ that contains more than 2 nodes.*
2. *$\Gamma'(y) = \bigsqcup_{x \in X} \Gamma(x)$, and there is no such path in $PDG(CFG_c^{\{x\},\{y\}})$.*

Proof (Proof of Proposition 4). The proof is by induction on the structure of c.

Assume that $c = \mathsf{skip}$:
1. By the typing rule [skip], $\Gamma = \Gamma'$. Hence, $\Gamma(y) = \Gamma'(y)$.
2. The node *start* is the only node in CFG_{skip} with two outgoing edges, and, hence, all control dependency edges in $PDG(CFG_{\mathsf{skip}}^{\{x\},\{y\}})$ originate at *start*.
3. Node 1 (representing the skip-statement) has empty def and use sets. Hence, $PDG(CFG_{\mathsf{skip}}^{\{x\},\{y\}})$ contains no data dependency edges from or to Node 1, and it contains an edge (in, out) if and only if $x = y$.
4. By (2) and (3), there is a path from *in* to *out* in $PDG(CFG_{\mathsf{skip}}^{\{x\},\{y\}})$ if and only if $x = y$, i.e., $X = \{y\}$.
5. Moreover, by (2) and (3) there is no path from *start* to *out* in the graph $PDG(CFG_{\mathsf{skip}}^{\{x\},\{y\}})$ that contains more than 2 nodes.
6. Hence, by (4) and (5), it suffices to show that $\Gamma'(y) = \Gamma(y)$ to conclude this case, and $\Gamma'(y) = \Gamma(y)$ holds by (1).

Assume that $c = z{:=}e$:
1. By the same argument as in the case for skip, all control dependency edges in $PDG(CFG_{z:=e}^{\{x\},\{y\}})$ originate at *start*.
2. Be the definition of def and use sets, $def_c(1) = \{z\}$ and $use_c(1) = fv(e)$ (where 1 is the node representing the assignment). Hence, by the definition of data dependence there is a data dependency edge $(in, 1)$ in $PDG(CFG_{z:=e}^{\{x\},\{y\}})$ if and only if $x \in fv(e)$ and there is a data dependency edge $(1, out)$ if and only if $y = z$. Moreover, there is an edge (in, out) if and only if $x = y$ and $x \neq z$.
3. Assume that $y = z$.
   (a) By (1) and (2), there is a path from *in* to *out* if and only if $x \in fv(e)$ (the path $\langle in, 1, out\rangle$). Hence, $X = fv(e)$.
   (b) Since node 1 is control dependent on *start* there is a path from *start* to *out* that contains more than 2 nodes (the path $\langle start, 1, out\rangle$).

(c) Hence, we must show that $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{x \in fv(e)} \Gamma(x)\right)$. This equality holds due to the typing rules [assign] and [exp].
4. Assume now that $y \neq z$.
   (a) By (1) and (2), there is a path from *in* to *out* if and only if $x = y$ and $x \neq z$ (the path $\langle in, out \rangle$). Hence, $X = \{y\}$.
   (b) Moreover, by (1) and (2) there is no path from *start* to *out* that contains more than 2 nodes.
   (c) Hence, we must show that $\Gamma'(y) = \Gamma(y)$. This holds due to the typing rule [assign].

Assume that $c = c_1; c_2$:

1. By typing rule [seq] there is a domain environment Γ'' such that $pc \vdash \Gamma\{c_1\}\Gamma''$ and $pc \vdash \Gamma''\ \{c_2\}\ \Gamma'$.
2. Let X_2 be the set of variables such that $x \in X_2$ if and only if there is a path from *in* to *out* in $PDG(CFG_{c_2}^{\{x\},\{y\}})$ By the induction hypothesis for k and c_2 (instantiating the environment with Γ'') one of the following conditions is satisfied:
   (a) $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{x \in X_2} \Gamma''(x)\right)$ and there is a path from *start* to *out* in the graph $PDG(CFG_{c_2}^{\{x\},\{y\}})$ that contains more than 2 nodes, or
   (b) $\Gamma'(y) = \bigsqcup_{x \in X_2} \Gamma''(x)$, and there is no such path in $PDG(CFG_{c_2}^{\{x\},\{y\}})$.
3. For each $z \in X_2$, let X_z be the set of variables such that $x \in X_z$ if and only if there is a path from *in* to *out* in $PDG(CFG_{c_1}^{\{x\},\{z\}})$. For each $z \in X_2$, by the induction hypothesis for c_1 (instantiating the environment with Γ and the variable with z) one of the following conditions is satisfied:
   (a) $\Gamma''(z) = pc \sqcup \left(\bigsqcup_{x \in X_z} \Gamma(x)\right)$ and there is a path from *start* to *out* in the graph $PDG(CFG_{c_1}^{\{x\},\{z\}})$ that contains more than 2 nodes, or
   (b) $\Gamma''(z) = \bigsqcup_{x \in X_z} \Gamma(x)$, and there is no such path in $PDG(CFG_{c_1}^{\{x\},\{z\}})$.
4. By Proposition 2, there is a path from *in* to *out* in $PDG(CFG_{c_1;c_2}^{\{x\},\{y\}})$ if and only if there exists $z \in Var$ such that there is a path from *in* to *out* in $PDG(CFG_{c_1}^{\{x\},\{z\}})$ and a path from *in* to *out* in $PDG(CFG_{c_2}^{\{z\},\{y\}})$. Hence, it follows from the definitions of X_2 and X_z that $X = \bigcup_{z \in X_2} X_z$.
5. We distinguish the cases (2a) and (2b). Assume firstly that (2a) holds.
   (a) Due to (2a) and Proposition 2, for all $z \in Var$ there is a path from *start* to *out* in $PDG(CFG_{c_1;c_2}^{\{z\},\{y\}})$ that contains more than 2 nodes. Hence, we must show that $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{x \in X} \Gamma(x)\right)$.
   (b) Due to (2a), $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{z \in X_2} \Gamma''(z)\right)$.
   (c) Due to (3), for each $z \in X_2$ either $\Gamma''(z) = pc \sqcup \left(\bigsqcup_{x \in X_z} \Gamma(x)\right)$ or $\Gamma''(z) = \left(\bigsqcup_{x \in X_z} \Gamma(x)\right)$ holds.
   (d) It follows from (b) and (c) that $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{z \in X_2} \bigsqcup_{x \in X_z} \Gamma(x)\right)$. Hence, by (4), $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{x \in X} \Gamma(x)\right)$.
6. Assume now that (2b) holds.
   (a) Due to (2b), $\Gamma'(y) = \bigsqcup_{z \in X_2} \Gamma''(z)$.
   (b) For each $z \in X_2$, either (3a) or (3b) holds. We distinguish the cases that (3b) holds for all $z \in X_2$ and that (3b) does not hold for some $z \in X_2$.

(c) Assume firstly that (3b) holds for all $z \in X_2$.

i. By Proposition 2, $PDG(CFG^{\{x\},\{y\}}_{c_1;c_2})$ does not contain a path from *start* to *out* that contains more than 2 nodes.

ii. For all $z \in X_2$, it follows from (3b) that $\Gamma''(z) = \big(\bigsqcup_{x \in X_z} \Gamma(x)\big)$.

iii. From (a) and (ii) it follows that $\Gamma'(y) = \bigsqcup_{z \in X_2} \bigsqcup_{x \in X_z} \Gamma(x)$. Hence, by (4), $\Gamma'(y) = \bigsqcup_{x \in X} \Gamma(x)$.

(d) Assume now that there exists $z \in X_2$ such that (3a) holds for z.

i. Hence, $\Gamma''(z) = pc \sqcup \big(\bigsqcup_{x \in X_z} \Gamma(x)\big)$.

ii. Moreover, $PDG(CFG^{\{x\},\{z\}}_{c_1})$ contains a path from *start* to *out* that contains more than 2 nodes.

iii. Since $z \in X_2$ there is a path from *in* to *out* in $PDG(CFG^{\{z\},\{y\}}_{c_2})$.

iv. By Proposition 2, (ii), and (iii), $PDG(CFG^{\{x\},\{y\}}_{c_1;c_2})$ contains a path from *start* to *out* that contains more than 2 nodes.

v. Due to (3), for each $z \in X_2$ either $\Gamma''(z) = pc \sqcup \big(\bigsqcup_{x \in X_z} \Gamma(x)\big)$ or $\Gamma''(z) = \big(\bigsqcup_{x \in X_z} \Gamma(x)\big)$ holds.

vi. From (a), (i), and (v), it follows that $\Gamma'(y) = pc \sqcup \big(\bigsqcup_{z \in X_2} \bigsqcup_{x \in X_z} \Gamma(x)\big)$. Hence, by (4), $\Gamma'(y) = pc \sqcup \big(\bigsqcup_{x \in X} \Gamma(x)\big)$.

Assume that $c = \mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}$:

1. Let $\{z_1, \ldots, z_l\} = fv(e)$, and $t = dom(z_1) \sqcup \ldots \sqcup dom(z_l)$.
2. By the typing rule [if], there are environments Γ'_1 and Γ'_2 such that $\Gamma' = \Gamma'_1 \sqcup \Gamma'_2$, $pc \sqcup t \vdash \Gamma\ \{c_1\}\ \Gamma'_1$, and $pc \sqcup t \vdash \Gamma\ \{c_2\}\ \Gamma'_2$.
3. For $i \in \{1,2\}$, let X_i be the set of variables such that $x \in X_i$ if and only if there is a path from *in* to *out* in $PDG(CFG^{\{x\},\{y\}}_{c_i})$.
4. Applying the induction hypothesis for both k and c_1 and k and c_2 (instantiating the program counter security level with $pc \sqcup t$), it follows that for $i \in \{1,2\}$ one of the following two conditions is satisfied:
   (a) $\Gamma'_i(y) = pc \sqcup t \sqcup \big(\bigsqcup_{x \in X_i} \Gamma(x)\big)$ and there is a path from *start* to *out* in the graph $PDG(CFG^{\{x\},\{y\}}_{c_i})$ that contains more than 2 nodes, or
   (b) $\Gamma'_i(y) = \bigsqcup_{x \in X_i} \Gamma(x)$ and there is no such path in $PDG(CFG^{\{x\},\{y\}}_{c_i})$.
5. It follows from Proposition 1 that there exists a path from *in* to *out* in the graph $PDG(CFG^{\{x\},\{y\}}_{\mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}})$ if and only if there is such a path in the graph $PDG(CFG^{\{x\},\{y\}}_{c_1})$ or in $PDG(CFG^{\{x\},\{y\}}_{c_2})$ (i.e., $x \in X_1 \cup X_2$), or if $x \in fv(e)$ and there is a path from *start* to *out* in $PDG(CFG^{\{x\},\{y\}}_{c_1})$ or in $PDG(CFG^{\{x\},\{y\}}_{c_2})$ that contains more than 2 nodes.
6. We do a case distinction on whether there exists a path from *start* to *out* in the graph $PDG(CFG^{\{x\},\{y\}}_{\mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}})$ that contains more than 2 nodes. Assume firstly that there is such a path.
   (a) Hence, by Proposition 1, there is such a path in $PDG(CFG^{\{x\},\{y\}}_{c_1})$ or in $PDG(CFG^{\{x\},\{y\}}_{c_2})$. Assume without loss of generality that there is such a path in $PDG(CFG^{\{x\},\{y\}}_{c_1})$.
   (b) Then, by (5), $X = X_1 \cup X_2 \cup fv(e)$.
   (c) Moreover, by (4), $\Gamma'_1(y) = pc \sqcup t \sqcup \big(\bigsqcup_{x \in X_1} \Gamma(x)\big)$.

(d) Moreover, by (4), either $\Gamma_2'(y) = pc \sqcup t \sqcup \left(\bigsqcup_{x\in X_2} \Gamma(x)\right)$ or $\Gamma_2'(y) = \bigsqcup_{x\in X_2} \Gamma(x)$.

(e) Hence, since $\Gamma' = \Gamma_1' \sqcup \Gamma_2'$, it follows from (1), (c), (d), and (e) that $\Gamma'(y) = pc \sqcup \left(\bigsqcup_{x\in X} \Gamma(x)\right)$.

7. Assume that there is no path $\langle \mathit{start}, \ldots, \mathit{out}\rangle$ in $\mathit{PDG}(\mathit{CFG}^{\{x\},\{y\}}_{\mathsf{if}\,(e)\,\mathsf{then}\,c_1\,\mathsf{else}\,c_2\,\mathsf{fi}})$ that contains more than 2 nodes.
   (a) Hence, by Proposition 1, there is no such path in $\mathit{PDG}(\mathit{CFG}^{\{x\},\{y\}}_{c_1})$ or in $\mathit{PDG}(\mathit{CFG}^{\{x\},\{y\}}_{c_2})$.
   (b) In consequence, by (5), $X = X_1 \cup X_2$.
   (c) Moreover, by (4), $\Gamma_i'(y) = \bigsqcup_{x\in X_i} \Gamma(x)$ for $i \in \{1, 2\}$.
   (d) Hence, since $\Gamma' = \Gamma_1' \sqcup \Gamma_2'$, it follows from (b) and (c) that $\Gamma'(y) = \bigsqcup_{x\in X} \Gamma(x)$.

Assume that $c = \mathsf{while}\,(e)\,\mathsf{do}\,c_1\,\mathsf{od}$

1. Since $pc \vdash \Gamma\ \{\mathsf{while}\,(e)\,\mathsf{do}\,c_1\,\mathsf{od}\}\ \Gamma'$ is derivable it follows from typing rule [while] that there exist $k \in \mathbb{N}$ and sequences $\Gamma_0', \ldots, \Gamma_{k+1}'$, $\Gamma_0'', \ldots, \Gamma_k''$, and $t_0, \ldots, t_k$ such that the following hold (for $0 \le i \le k$):
   (a) $\Gamma_0' = \Gamma$,
   (b) $\Gamma_{k+1}' = \Gamma_k' = \Gamma'$,
   (c) $\Gamma_i' \vdash_e t_i\ \{\}$,
   (d) $pc \sqcup t_i \vdash \Gamma_i'\ \{c_1\}\ \Gamma_i''$, and
   (e) $\Gamma_{i+1}' = \Gamma_i'' \sqcup \Gamma$.
2. We say that a loop run of the loop $\mathsf{while}\,(e)\,\mathsf{do}\,c_1\,\mathsf{od}$ induces a dependency of z' on z if one of the following two conditions is satisfied:
   (a) there is a path from *in* to *out* in $\mathit{PDG}(\mathit{CFG}^{\{z\},\{z'\}}_{c_1})$ or
   (b) $z \in \mathit{fv}(e)$ and there is a path from *start* to *out* in $\mathit{PDG}(\mathit{CFG}^{\{z\},\{z'\}}_{c_1})$ that contains more than 2 nodes.
3. Hence, by Proposition 3, $x \in X$ if and only if there is a sequence of distinct variables $z_1, \ldots, z_l$ such that $x = z_1$, $z_l = y$, and the loop c induces a dependency of z_{i+1} on z_i for all $i \in \{1, \ldots, l-1\}$.
4. By the induction hypothesis and (1d) it follows that, for all $z' \in \mathit{Var}$ and all $i \in \mathbb{N}$, one of the following holds, where Z is the set of all z such that there is a path from *in* to *out* in $\mathit{PDG}(\mathit{CFG}^{\{z\},\{z'\}}_{c_1})$:
   (a) $\Gamma_i''(z') = \left(\bigsqcup_{z\in Z} \Gamma_i'(z)\right) \sqcup pc \sqcup t_i$ if there is a path from *start* to *out* in the graph $\mathit{PDG}(\mathit{CFG}^{\{z\},\{z'\}}_{c_1})$ that contains more than 2 nodes, and
   (b) $\Gamma_i''(z') = \bigsqcup_{z\in Z} \Gamma_i'(z)$ if there is no such path.
5. From (1c) and typing rule [exp] it follows that $t_i = \bigsqcup_{x\in \mathit{fv}(e)} \Gamma_i'(x)$.
6. From the definition in (2) and from (4) and (5) it follows that for all $z' \in \mathit{Var}$ one of the following holds, where Z' is the set of all z such that the loop c induces a dependency of z' on z:
   (a) $\Gamma_i''(z') = \left(\bigsqcup_{z\in Z'} \Gamma_i'(z)\right) \sqcup pc$ if there is a path from *start* to *out* in the graph $\mathit{PDG}(\mathit{CFG}^{\{x\},\{z'\}}_{c_1})$ that contains more than 2 nodes, and
   (b) $\Gamma_i''(z') = \bigsqcup_{z\in Z'} \Gamma_i'(z)$ if there is no such path.
7. Hence, by (1e), it follows that for all $z \in \mathit{Var}$ one of the following holds:

(a) $\Gamma'_{i+1}(z') = \left(\bigsqcup_{z \in Z'} \Gamma'_i(z)\right) \sqcup pc \sqcup \Gamma(z')$ if there is a path from *start* to *out* in $PDG(CFG_{c_1}^{\{z\},\{z'\}})$ that contains more than 2 nodes, and
(b) $\Gamma'_{i+1}(z') = \left(\bigsqcup_{z \in Z'} \Gamma'_i(z)\right) \sqcup \Gamma(z')$ if there is no such path.

8. Using (7) and starting with $\Gamma'(y) = \Gamma'_k(y)$, we unfold the equation further and further, thereby collecting the term $\Gamma(x)$ for all $x \in X$ on the right hand side of the equation. As a result, the following holds:
   (a) $\Gamma'(y) = \bigsqcup_{x \in X} \Gamma(x) \sqcup pc$, if, for any $x \in Var$, there exists $z_1, \ldots, z_k \in Var$ and a path from *start* to *out* in $PDG(CFG_{c_1}^{\{x\},\{z_1\}})$ that contains more than 2 nodes and paths from *in* to *out* in the graphs $PDG(CFG_{c_1}^{\{z_i\},\{z_{i+1}\}})$, and
   (b) $\Gamma'(y) = \bigsqcup_{x \in X} \Gamma(x)$ if such paths do not exist for any $x \in Var$.
9. But then, with Statement (2) of Proposition 3, it follows that
   (a) $\Gamma'(y) = \bigsqcup_{x \in X} \Gamma(x) \sqcup pc$, if, there exists a path from *start* to *out* in the graph $PDG(CFG_{c}^{\{z_x\},\{y\}})$ that contains more than 2 nodes, and
   (b) $\Gamma'(y) = \bigsqcup_{x \in X} \Gamma(x)$ if such a path does not exist.

   This concludes the proof.

Proof (Proof of Lemma 1 from [MS13]). Lemma 1 from [MS13] follows from Proposition 4 for $pc = l$.

2 Proof of Theorem 4 from [MS13]

To prove Theorem 4 from [MS13], we generalize the definition of $PDG^{\|}(CFG_c^{H,L})$ to the form $PDG^{\|}(CFG_c^{I,O}, mds)$ where I, O are arbitrary sets of variables and $mds : Mod \to \mathcal{P}(Var)$ is a function that specifies modes before the execution of c.

Definition 1. *For $c \in Com$ and $mds : Mod \to \mathcal{P}(Var)$ we define the function $modes_{c,mds} : (N_c \times Mod) \to \mathcal{P}(Var)$ by $x \in modes_{c,mds}(n, m)$ if and only if for all paths $\langle start, \ldots, n\rangle$ in CFG_c one of the following two conditions is satisfied:*
- *there exists n' on the path such that $c[n']$ acquires m for x, and for all nodes n'' following n' on the path $c[n'']$ does not release m for x, or*
- *$x \in mds(m)$ and for all nodes n' on the path $c[n']$ does not release m for x.*

Definition 2. *Let $c \in Com$. Then $PDG^{\|}(CFG_c^{I,O}, mds) = (N, E \cup E')$ where $(N, E) = PDG(CFG_c^{I,O})$ and $(n, n') \in E'$ if and only if one of the following holds:*
1. *$n = in$ and there exist a variable $x \in I \cap use_c(n')$, a node $n'' \in N$ with $x \notin modes_{c,mds}(n'', asm\text{-}nowrite)$, and a path p from n'' to n' with $x \notin def_c(n''')$ for every node n''' on p with $n''' \neq n''$ and $n''' \neq n'$,*
2. *$n' = out$ and there exist a variable $x \in O \cap def_c(n')$, a node $n'' \in N$ with $x \notin modes_{c,mds}(n'', asm\text{-}noread)$, and a path p from n to n'' such that $x \notin def_c(n''')$ for every node n''' on p with $n''' \neq n$ and $n''' \neq n''$, or*
3. *$n \in \{1, \ldots, |c|\}$, $c[n] \in Exp$, and $n' = out$.*

To prove Theorem 4 from [MS13], we establish the following more general proposition.

Proposition 5. *Let mds be a mode state, Λ be a partial environment that is consistent with mds, and $c \in Com$. Assume that no partial environment Λ' exists such that $\vdash \Lambda\,\{c\}\,\Lambda'$ is derivable in the type system from [MSS11]. Then there exist $x, y \in Var$ with $\Lambda\langle x\rangle = h$ and $dom(y) = l$ and a path p of the form $\langle in, \ldots, n, out\rangle$ in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$ where $n \in \{1, \ldots, |c|\}$ and one of the following conditions is satisfied:*

1. *$y \in def_c(n)$, there is a node n' with $y \notin modes_{c,mds}(n', asm\text{-}noread)$, and a definition of y at n reaches n' in CFG_c, or*
2. *$c[n] \in Exp$.*

Proof. The proof is by induction on the structure of the command c.

Assume that $c = \mathsf{skip}$**.** Then $\vdash \Lambda\,\{c\}\,\Lambda$ is derivable, which contradicts the assumptions of the lemma.

Assume that $c = x{:=}e$**.** If $x \in dom(\Lambda)$ then $\vdash \Lambda\,\{c\}\,\Lambda'$ is derivable for some Λ' with rule [assign_2]. In consequence, $x \notin dom(\Lambda)$. Moreover, if $dom(x) = h$ then $\vdash \Lambda\,\{c\}\,\Lambda$ is derivable with rule [assign_1]. In consequence, $dom(x) = l$. Moreover, if $\Lambda\langle y\rangle = l$ for all $y \in fv(e)$ then $\vdash \Lambda\,\{c\}\,\Lambda$ would be derivable with rule [assign_1]. In consequence, there exists $y \in fv(e)$ with $\Lambda\langle y\rangle = h$.

Since $y \in fv(e)$ it follows that $y \in use_c(1)$, and, hence, the pair $(in, 1)$ is an edge in $PDG(CFG_c^{\{y\},\{x\}})$. In consequence, the pair is an edge in the graph $PDG^{\|}(CFG_c^{\{y\},\{x\}}, mds)$.

Since $x \notin dom(\Lambda)$, Λ is consistent with mds, and $dom(x) = l$, it follows that $x \notin mds(asm\text{-}noread)$. In consequence, $x \notin modes_{c,mds}(stop, asm\text{-}noread)$. Moreover, the definition of x at Node 1 reaches $stop$ in CFG_c. Hence, $(1, out)$ is an edge in $PDG^{\|}(CFG_c^{\{y\},\{x\}}, mds)$, and Condition (1) is satisfied for this edge.

Hence, $\langle in, 1, out\rangle$ is a path in $PDG^{\|}(CFG_c^{\{y\},\{x\}}, mds)$ that satisfies all required conditions.

Assume that $c = c_1; c_2$**.** If there exist Λ'' and Λ' such that $\vdash \Lambda\,\{c_1\}\,\Lambda''$ and $\vdash \Lambda''\,\{c_2\}\,\Lambda'$ are derivable then $\vdash \Lambda\,\{c\}\,\Lambda'$ is derivable with rule [seq]. In consequence, there does not exist Λ'' and Λ' such that $\vdash \Lambda\,\{c_1\}\,\Lambda''$ and $\vdash \Lambda''\,\{c_2\}\,\Lambda'$ are derivable. We distinguish the two cases (1) that there is no Λ'' such that $\vdash \Lambda\,\{c_1\}\,\Lambda''$ is derivable, and (2) that if $\vdash \Lambda\,\{c_1\}\,\Lambda''$ is derivable for some Λ'' then there is no Λ' such that $\vdash \Lambda''\,\{c_2\}\,\Lambda'$ is derivable.

1. Assume that there is no Λ'' such that $\vdash \Lambda\,\{c_1\}\,\Lambda''$ is derivable.
   (a) By the induction hypothesis there are $x, y \in Var$ with $\Lambda\langle x\rangle = h$ and $dom(y) = l$ and a path p from in to out in $PDG^{\|}(CFG_{c_1}^{\{x\},\{y\}}, mds)$ such that Condition (1) or Condition (2) is satisfied for the one-but-last node of p. To determine a path from in to out in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$ we distinguish between whether Condition (1) or Condition (2) is satisfied for that node.

i. Assume that $y \in \mathit{def}_{c_1}(n)$, that there exists a node n' with $y \notin \mathit{modes}_{c_1,mds}(n', \mathit{asm\text{-}noread})$, and that a definition of y at n reaches n' in CFG_{c_1}.
It follows from $y \in \mathit{def}_{c_1}(n)$ that $y \in \mathit{def}_{c_1;c_2}(n)$. It follows from $y \notin \mathit{modes}_{c_1,\Lambda}(n, \mathit{asm\text{-}noread})$ that $y \notin \mathit{modes}_{c_1;c_2,\Lambda}(n, \mathit{asm\text{-}noread})$. Moreover, since a definition of y at n reaches n' in CFG_{c_1}, it also reaches n' in CFG_c.
Hence, the last edge (n, out) of p is an edge in $\mathit{PDG}^{\|}(\mathit{CFG}_c^{\{x\},\{y\}}, \mathit{mds})$. Thus p is a path in $\mathit{PDG}^{\|}(\mathit{CFG}_c^{\{x\},\{y\}}, \mathit{mds})$.

ii. Assume that $n \in \{1, \ldots, |c_1|\}$ and $c_1[n] \in \mathit{Exp}$. Then the last edge (n, out) of p is an edge in $\mathit{PDG}^{\|}(\mathit{CFG}_c^{\{x\},\{y\}}, \mathit{mds})$. Thus p is a path in $\mathit{PDG}^{\|}(\mathit{CFG}_c^{\{x\},\{y\}}, \mathit{mds})$.

2. Let Λ'' such that $\vdash \Lambda\,\{c_1\}\,\Lambda''$ is derivable, and such that there is no Λ' for which $\vdash \Lambda''\,\{c_2\}\,\Lambda'$ is derivable.
(a) Define the function $\mathit{mds}'' : \mathit{Mod} \to \mathcal{P}(\mathit{Var})$ by $\mathit{mds}''(m) = \mathit{modes}_{c_1,mds}(\mathit{stop}, m)$ It follows from the definition of $\mathit{modes}_{c,mds}$ and the typing rules that Λ'' is compatible with mds''.
(b) By the induction hypothesis there exist $z, y \in \mathit{Var}$ with $\Lambda''\langle z\rangle = h$ and $\mathit{dom}(y) = l$ such that there is a path p_2 in $\mathit{PDG}^{\|}(\mathit{CFG}_{c_2}^{\{z\},\{y\}}, \mathit{mds}'')$ that is of the form $\langle \mathit{in}, \ldots, n, \mathit{out}\rangle$ with $n \in \{1, \ldots, |c_2|\}$ and where Condition (1) or Condition (2) is satisfied for the edge (n, out).
(c) Let Γ be the unique environment such that $l \vdash \Lambda\langle\cdot\rangle\,\{c_1\}\,\Gamma$ is derivable in the type system from [HS06]. Then, by Lemma 1 from [MS13], $\Gamma(z) = \bigsqcup_{x\in X} \Lambda\langle x\rangle$ where X contains all $x \in \mathit{Var}$ such that there exists a path from in to out in $\mathit{PDG}(\mathit{CFG}_{c_1}^{\{x\},\{z\}})$. We denote this path with p_x for $x \in X$.
Like in the proof of Lemma 1 in [MS13], it follows from the existence of the paths p_x and p_2 that there exists a path p from in to out in the graph $\mathit{PDG}^{\|}(\mathit{CFG}_{c_1;c_2}^{\{x\},\{y\}}, \mathit{mds})$. That Condition (1) or Condition (2) is satisfied for the last edge in this path p follows from the construction of p and from (a) and (b). Hence, we can conclude this case if there exists $x \in X$ with $\Lambda\langle x\rangle = h$.
Assume now that $\Lambda\langle x\rangle = l$ for all $x \in X$. Then $\Gamma(z) = l$. It follows from $\Gamma(z) = l$, $\Lambda''\langle z\rangle = h$, and the derivability of $l \vdash \Lambda\langle\cdot\rangle\,\{c_1\}\,\Gamma$ and $\vdash \Lambda\,\{c_1\}\,\Lambda''$ that $\mathit{dom}(z) = h$ and $z \notin \mathit{dom}(\Lambda'')$. Then, since Λ'' is consistent with mds'', $z \notin \mathit{mds}''(\mathit{asm\text{-}nowrite})$. Let n' be a node such that $z \in \mathit{use}_{c_1}(n')$ and such that a definition of z at n' reaches stop in CFG_{c_1}, or let $n' = \mathit{out}$ if no such node exists. Then, by Definition 2 the pair (in, n') is an edge in $\mathit{PDG}^{\|}(\mathit{CFG}_{c_1}^{\{x\},\{z\}}, \mathit{mds})$ for any x, because the pair (n', out) is an edge in $\mathit{PDG}^{\|}(\mathit{CFG}_{c_1}^{\{x\},\{z\}}, \mathit{mds})$ by construction of n'. Hence, $\langle \mathit{in}, n', \mathit{out}\rangle$ is a path in $\mathit{PDG}^{\|}(\mathit{CFG}_{c_1}^{\{x\},\{z\}}, \mathit{mds})$, and we can conclude as in the above case for $\Lambda\langle x\rangle = h$

Assume that $c = \textsf{if}\ (e)\ \textsf{then}\ c_1\ \textsf{else}\ c_2\ \textsf{fi}$**.** Since there is no partial environment Λ' such that $\vdash \Lambda\,\{c\}\,\Lambda'$ is derivable in the type system, by the typing rule [if] one of the following conditions is satisfied: $\Lambda\langle e\rangle = h$, or there is no Λ' such that $\vdash \Lambda\,\{c_1\}\,\Lambda'$ is derivable, or there is no Λ' such that $\vdash \Lambda\,\{c_2\}\,\Lambda'$ is derivable.

1. Assume that $\Lambda\langle e\rangle = h$. Then there exists $x \in fv(e)$ such that $\Lambda\langle x\rangle = h$. Hence, $(in, 1)$ is an edge in $PDG(CFG_c^{\{x\},\{y\}})$. In consequence, $(in, 1)$ is an edge in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$.
   Let y be a variable with $dom(y) = l$. Then the pair $(1, out)$ is an edge in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$ because $c[1] \in Exp$.
   Hence, $\langle in, 1, out\rangle$ is a path from in to out in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$, and Condition (2) is satisfied for the last edge $(1, out)$.
2. Assume that there is no Λ' such that $\vdash \Lambda\,\{c_1\}\,\Lambda'$ is derivable. Then, by the induction hypothesis for c_1, there are x, y, and a path p_1 in the graph $PDG^{\|}(CFG_{c_1}^{\{x\},\{y\}}, mds)$ with the properties stated by the lemma. But then one obtains a path p in the graph $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$ with the properties required by the lemma by increasing each node $n \in \{1, \ldots, |c_1|\}$ in p_1 by 1.
3. Assume that there is no Λ' such that $\vdash \Lambda\,\{c_2\}\,\Lambda'$ is derivable. The proof is as in the previous case, exploiting the induction hypothesis for c_2.

Assume that $c = \mathsf{while}\,(e)\,\mathsf{do}\,c_1\,\mathsf{od}$**:** Since there is no Λ' such that the judgment $\vdash \Lambda\,\{\mathsf{while}\,(e)\,\mathsf{do}\,c_1\,\mathsf{od}\}\,\Lambda'$ is derivable, by the typing rule [sub] there is no Λ'' with $\Lambda \sqsubseteq \Lambda'' \sqsubseteq \Lambda'$ such that $\vdash \Lambda''\,\{\mathsf{while}\,(e)\,\mathsf{do}\,c_1\,\mathsf{od}\}\,\Lambda''$ is derivable. I.e., by the typing rule [while], for all Λ'' with $\Lambda \sqsubseteq \Lambda'' \sqsubseteq \Lambda'$ one of the following conditions is satisfied: $\Lambda''\langle e\rangle = h$ or $\vdash \Lambda''\,\{c_1\}\,\Lambda''$ is not derivable.

1. Assume that $\Lambda''\langle e\rangle = h$. Then the proof is as in the case for conditionals, defining $p = \langle in, 1, out\rangle$.
2. Assume that $\vdash \Lambda''\,\{c_1\}\,\Lambda''$ is not derivable. Then, by the induction hypothesis for c_1, there are $x, y \in Var$ and a path p_1 in $PDG^{\|}(CFG_{c_1}^{\{x\},\{y\}}, mds)$ with the properties stated by the lemma. But then one obtains a path p in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds)$ with the properties required by the lemma by increasing each node $n \in \{1, \ldots, |c_1|\}$ in p_1 by 1.

Now we prove Theorem 4 from [MS13].

Proof. The proof is by contradiction. Assume that that there is no partial environment Λ' such that $\vdash \Lambda_0\,\{c\}\,\Lambda'$ is derivable. Then, by Proposition 5, there exist $x, y \in Var$ with $\Lambda_0\langle x\rangle = h$ and $dom(y) = l$, such that there is a path from in to out in $PDG^{\|}(CFG_c^{\{x\},\{y\}}, mds_0)$.

By the definition of Λ_0 we obtain that $dom(x) = h$.

Hence, the path in $PDG^{\|}(CFG_{c)}^{\{x\},\{y\}}, mds_0)$ is also a path in the PDG with multi-threaded dependencies for high inputs and low outputs $PDG_{H,L}(CFG_c)$.

In consequence, c is not accepted by the PDG-based analysis for threads.

3 Derivation Rules for the Judgment $\langle c, mem\rangle \Downarrow mem'$

The derivation rules for the judgment $\langle c, mem\rangle \Downarrow mem'$ are defined in Figure 1, where $\langle e, mem\rangle \Downarrow v$ denotes that expression $e \in Exp$ evaluates to value $v \in Val$ in memory $mem \in (Var \to Val)$.

$$\frac{}{\langle \mathsf{skip}, mem \rangle \Downarrow mem} \qquad \frac{\langle e, mem \rangle \Downarrow v}{\langle x{:=}e, mem \rangle \Downarrow mem[x \mapsto v]}$$

$$\frac{\langle c_1, mem \rangle \Downarrow mem' \quad \langle c_2, mem' \rangle \Downarrow mem''}{\langle c_1; c_2, mem \rangle \Downarrow mem''}$$

$$\frac{\langle e, mem \rangle \Downarrow \mathsf{True} \quad \langle c_1, mem \rangle \Downarrow mem'}{\langle \mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}, mem \rangle \Downarrow mem'} \qquad \frac{\langle e, mem \rangle \Downarrow \mathsf{False} \quad \langle c_2, mem \rangle \Downarrow mem'}{\langle \mathsf{if}\ (e)\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\ \mathsf{fi}, mem \rangle \Downarrow mem'}$$

$$\frac{\langle e, mem \rangle \Downarrow \mathsf{False}}{\langle \mathsf{while}\ (e)\ \mathsf{do}\ c\ \mathsf{od}, mem \rangle \Downarrow mem} \qquad \frac{\begin{array}{c}\langle e, mem \rangle \Downarrow \mathsf{True} \quad \langle c, mem \rangle \Downarrow mem' \\ \langle \mathsf{while}\ (e)\ \mathsf{do}\ c\ \mathsf{od}, mem' \rangle \Downarrow mem''\end{array}}{\langle \mathsf{while}\ (e)\ \mathsf{do}\ c\ \mathsf{od}, mem \rangle \Downarrow mem''}$$

Figure 1. Derivation rules for the judgment $\langle c, mem \rangle \Downarrow mem'$

References

[HS06] S. Hunt and D. Sands. On Flow-Sensitive Security Types. In *ACM Symposium on Principles of Programming Languages*, pages 79–90, 2006.

[MS13] H. Mantel and H. Sudbrock. Types vs. PDGs in Information Flow Analysis. In *Proceedings of the 22nd International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR)*, volume 7844 of *LNCS*, pages 106–121, Leuven, Belgium, 2013. Springer. To appear.

[MSS11] H. Mantel, D. Sands, and H. Sudbrock. Assumptions and Guarantees for Compositional Noninterference. In *IEEE Computer Security Foundations Symposium*, pages 218–232, 2011.