経路依存フローグラフを用いた プログラム・スライスの形式的定義とその妥当性

直井 邦彰　　高橋 直久

NTT ソフトウェア研究所
〒180 東京都武蔵野市緑町 3-9-11
NTT 武蔵野研究開発センタ

あらまし　本稿では，先に提案した経路依存フローグラフ (PDFG) と呼ぶ有向グラフを用いて，各種スライスを統一的枠組の上で形式的に表現する手法を提案する．プログラム・スライシングとは，手続き型言語のプログラムから，ある着目する性質を持つ文集合 (スライスと呼ぶ) を抽出する技術である．これまでに，様々なスライス表現法が与えられているが，各種スライスを統一的に扱える表現形式は提案されていなかった．提案手法では，まず，各種スライスの一般化した表現形式を与える．次に，PDFG 上で各種依存関係を定義し，これら関係の組合せとしてスライスを定義する．これにより，スライスの性質が，スライス作成法とは独立に議論可能となる．本稿では，更に，作成法が既に与えられているスライスについて，それらスライスと本手法で表現するスライスとが同じ文集合となることを証明することにより，提案手法の妥当性を示す．

Formal definition of program slice using a path dependence flow graph and its adequacy

Kuniaki NAOI　　Naohisa TAKAHASHI

NTT Software Laboratories
Midori-cho, 3-9-11, Musashino-shi, Tokyo, 180, JAPAN

Abstract　A unified framework for formally representing various kinds of program slices are presented. Eight typical slices are defined as combinations of three orthogonal primitive attributes, first introduced by G.A. Venkatesh. Fundamental relationships are formalized on a path dependence flow graph proposed earlier, and all slices are represented using these relationships. As a result, the proposed framework makes it possible to precisely discuss the properties of program slices independently of the slicing procedures. The proposed method is shown to be adequate by proving that the slice provided by this method is the same as that provided by previous methods.

1. まえがき

手続き型言語のプログラムから，ある着目する性質を満たす文集合を抽出する技術はプログラム・スライシングと呼ばれ，プログラムの簡単化，並列化，デバッグや保守などのために広く用いられている[1)-12)]．ここで，抽出した文集合はスライスと呼ばれる[1)]．これまでに，様々な種類のスライスが提案され，それぞれ作成法が与えられている[1)-9)]．しかし，従来の作成法は，各種スライスを作成するためのアルゴリズムが繁雑であったり，同じ枠組の上で各種スライスを求めることができないという問題があった[13)]．更に，従来，各種スライスは，それぞれ与えられた作成法により求まる文集合として定義されるため，各スライス間での包含関係など，スライスの性質を厳密に議論することができなかった．本稿では，上記の問題を解決するために，先に提案した経路依存フローグラフ (PDFG : Path Dependence Flow Graph)[14)-17),13),18)] と呼ぶ有向グラフを用いて，各種スライスを統一的な枠組の上で表現する方法を提案する．ここで，PDFG とは，プログラムにおける，データ，経路，制御の依存関係を表現できる有向グラフであり，依存関係に基づくプログラム解析[19)] に適したグラフ表現形式である．本稿では，まず，スライスを静的 / 動的，逆方向 / 順方向，実行可 / クロージャからなる互いに直交した概念である 3 つの基本的な属性で表現し，これら属性の組合せとして 8 種類のスライスが定義できることを示す．次に，これらのスライスに対して，PDFG を用いた表現法を与える．更に，8 種類のスライスのうち計算法が既に提案されているスライスについては，それらスライスが，本手法により与えられるスライスと同じ文集合を表現していることを証明する．

本稿の構成は以下の通り．2. では，スライスを分類・整理する．3. から 7. では，文献 (13) に基づき，PDFG とその上でのポート間関係，ポート集合，依存関係をまとめる．8. では，文献 (13) に基づき，PDFG を用いたスライス表現法を示す．9. では，提案手法による表現と，従来手法により作成されるスライスとを比較することにより，提案手法の妥当性を示す．10. では，9. で示した定理の詳細な証明を示す．

2. プログラム・スライシング

手続き型言語のプログラムにおいて，ソースコードから，注目する性質を持つ文集合を抽出する技術をプログラム・スライシングと呼び，抽出した文集合をスライスと呼ぶ[1)]．求めたい性質に応じて様々なスライシング技法[1)-8)] が提案されており，これらスライスは，文献 (13) の通り，入力 f_i，方向 f_d，実行可能性 f_e と呼ぶ 3 つの属性[12)] に従い分類される．ここで，$f_i=st$ は静的，$f_i=dy$ は動的，$f_d=bk$ は逆方向，$f_d=fw$ は順方向，$f_e=ex$ は実行可，$f_e=cl$ はクロージャのスライスを表す．そして，(f_i,f_d,f_e) をスライス属性と呼び，スライスの性質を表すために用いる．例えば，(st,bk,ex) のスライスは，静的逆方向実行可のスライスを表す．

また，スライスを求める対象となるプログラムを Q，Q への入力値を I，着目する文を S，着目する変数の集合を $\mathcal{V}$ としたとき，これらの値とスライス属性の 7 つ組 $(Q,I,S,\mathcal{V},f_i,f_d,f_e)$ によりスライスを表す[13)]．

3. 経路依存フローグラフ

本章では，5. 以降の議論の準備のため，文献 (13) に従い，PDFG の概要を述べる．

3.1 有向グラフにおける基本用語

$\mathcal{N}$ をノード集合，$\mathcal{A}$ をアーク集合，$\mathcal{G}=\{\mathcal{N},\mathcal{A}\}$ を有向グラフとする．このとき，$n\in\mathcal{N}$ は任意個の入力ポートと出力ポートを持ち，n の入力ポート集合を $\mathcal{I}(n)$，出力ポート集合を $\mathcal{O}(n)$ と表す．また，$(x\in\mathcal{O}(n))\wedge(n\in\mathcal{N})$ なる x から $(y\in\mathcal{I}(m))\wedge(m\in\mathcal{N})$ なる y への有向枝をアークといい，(x,y) と表す．更に，$a=(x,y)$ のとき，x を a の始点，y を a の終点，a を n の出力アーク，m の入力アークという．また，入力アークを持たないノードを開始ノード，出力アークを持たないノードを終了ノードという．更に，x を $\mathcal{G}$ の任意のポートとしたとき，x を入力ポートまたは出力ポートとするノードを $\mathcal{N}_d(x)$ と表す．

3.2 PDFG のノード属性

start, *end*, *switch-macro*, *merge-macro*, *operational*, *load*, *store* を PDFG のノード属性という．最初の 5 つの属性は，それぞれ，データフロー計算モデル[20)] における開始，終了，*switch*, *merge*, 演算のノードに対応し，*load*, *store* は変数を読み書きするノードを示す．上記ノード属性からなる集合を PDFG ノード属性集合といい，$\mathcal{N}_0$ と表す．ただし，*switch-macro* 属性のノードは，同じ文に対応する複数の *switch* ノードを 1 つのノードで表現したものであり[15)]，以下の通り定義される．

[定義 1] *switch-macro* ノード．

switch-macro 属性のノード n は，2 組の入力 $< x_1, x_2,...,x_m >$，y と，2 組の出力 $< z_{t_1},z_{t_2},...,z_{t_m} >$，$< z_{f_1},z_{f_2},...,z_{f_m} >$ を持つ *switch* 命令であり，y から受信したトークンの値が真ならば z_{t_i} に，偽ならば z_{f_i} に，$x_i(1\leq i\leq m)$ からの入力トークンを送信する．ここで，x_i を分岐対象入力ポート，y を真偽決定入力ポート，z_{t_i} を真方向出力ポート，z_{f_i} を偽方向出力ポートという．n の真方向出力ポートにはラベル t を，偽方向出力ポートにはラベル f を，それぞれ付与する．そして，n の出力ポートからラベルを返す関数を $\mathcal{L}_p$ とする． □

同様に，*merge-macro* 属性のノードは，同じ文に対応する複数の *merge* ノードを 1 つのノードで表現したものであり，以下の通り定義する．

[定義 2] *merge-macro* ノード．

merge-macro 属性のノードは，2 組の入力 $< x_1, x_2,...,x_m >$，$< y_1,y_2,...,y_m >$ と，1 組の出力 $< z_1,z_2,...,z_m >$ を持つ *merge* 命令であり，x_i および $y_i(1\leq i\leq m)$ からトークンを受信したとき，それぞれ z_i にトークンを送信する． □

3.3 PDFG のアーク属性

switch-macro, *merge-macro*, *start*, *end* のいずれかの属性を持つノード間の順序関係を表すアークは *path* 属性を持つという．同様に，ある変数に対する *load*, あるいは *store* 属性のノード間の順序関係を表すアー

クは *impe* 属性，同じ文に対応する複数のノード間でのデータ授受の関係を表すアークは *func* 属性を持つという．そして，*store* 属性，*switch-macro* 属性，*merge-macro* 属性のノードをそれぞれ終点とするアークは，*st* 属性，*wt* 属性，*mt* 属性を持つという．また，その他の属性のノードを終点とするアークは *ot* 属性を持つという．

[定義 3] PDFG アーク属性集合.

$\mathcal{A}_k=\{path, impe, func\}$，$\mathcal{A}_t=\{st, wt, mt, ot\}$ としたとき，直積 $\mathcal{A}_k\times\mathcal{A}_t$ を PDFG アーク属性集合といい，$\mathcal{A}_0^2$ と表す．また，$\mathcal{A}_{if}=\{impe, func\}$，$\mathcal{A}_{ns}=\{wt, mt, ot\}$ としたとき，集合 $\mathcal{A}_{if}\times\{st,mt,ot\}$ を $\mathcal{A}^2_{nw}$，集合 $\mathcal{A}_{if}\times\{wt\}$ を $\mathcal{A}^2_w$，$\mathcal{A}^2_{nw}\cup\mathcal{A}^2_w$ を $\mathcal{A}^2_{np}$，$(\mathcal{A}_{if}\times\mathcal{A}_{ns})\cup\{(func, st)\}$ を $\mathcal{A}^2_d$ と表す． □

3.4 PDFG の定義

[定義 4] 経路依存フローグラフ (PDFG).

プログラム Q の経路依存フローグラフは，ノード集合 $\mathcal{N}$，アーク集合 $\mathcal{A}$，開始ノード s，終了ノード e の 4 つ組 $(\mathcal{N},\mathcal{A},s,e)$ で表される有向グラフである．ここで，$n\in\mathcal{N}$ のノード属性を a_n，$a\in\mathcal{A}$ のアーク属性を a_a とすると，$a_n\in\mathcal{N}_0$，$a_a\in\mathcal{A}_0^2$ である． □

PDFG は手続き型プログラムに対する表現形式であり，プログラムにおける，データ，経路，制御の 3 つの依存関係が表現できる．図 1(a) に示すプログラムの PDFG の例を図 1(b) に示す．

4. 逐次 PDFG

本章では，本稿で対象とする言語と PDFG のクラスについて述べる．

4.1 対象プログラム

本稿では，図 2 の BNF 記法に示すように，C 言語に対して制御構造と変数の型を簡単化したプログラミング言語を対象とする．図で，*Prog* はプログラム，*var* は変数，*const* は定数，*op* は演算子である．また，[ ] 内は省略可能とする．この言語は，図に示すように，代入文，if 文，while 文を持ち，goto 文を持たない．また，基本データ型の変数を持つが，配列など構造データおよびポインタ変数は持たない．

4.2 対象とする PDFG のクラス

本節では，PDFG において，グラフのノードと分岐文との順序関係をアークとして保持する，グラフ表現形式について述べる．

Q をあるプログラムとしたとき，Q の文をノードとし，文の間の制御移行の関係をノード間のアークとする有向グラフは，Q の制御フローグラフ (CFG : Control Flow Graph)[21] と呼ばれている．以下，$\mathcal{G}_p$ を Q の PDFG，$\mathcal{N}_p$ を $\mathcal{G}_p$ のノード集合，$\mathcal{G}_c$ を Q の CFG，$\mathcal{N}_c$ を $\mathcal{G}_c$ のノード集合，$\mathcal{N}_c^{sw}$ を $\mathcal{G}_c$ の *switch* ノード集合として，議論を進める．また，$d\in\mathcal{A}^2_{np}$．

$\mathcal{G}_p$ において，$n_p\in\mathcal{N}_p$ に対応する $\mathcal{G}_c$ のノード $n_c\in\mathcal{N}_c$ と，$\mathcal{G}_c$ の任意の *switch* ノード $m_c\in\mathcal{N}_c^{sw}$ との $\mathcal{G}_c$ における順序関係が，属性 d のアークにより表された有向グラフを逐次 PDFG と呼び，以下の通り定義する．

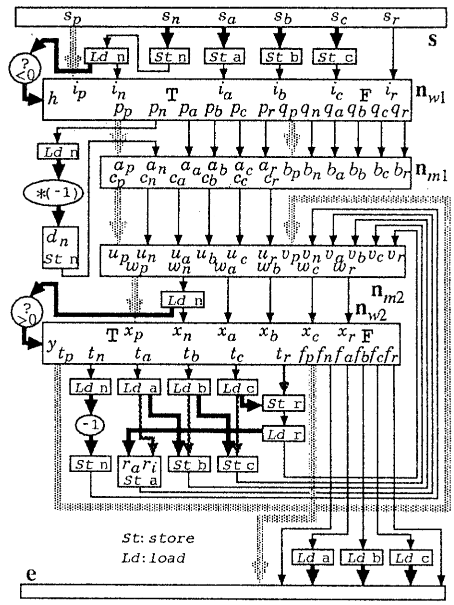
[定義 5] 逐次 PDFG.

$\mathcal{G}_p$ において，$n_p\in\mathcal{N}_p$ から $m_p\in\mathcal{N}_p$ まで属性 d のアークを辿って到達できるとき，n_p に対応する $\mathcal{G}_c$ の

```
main(int a,int b,int c,int n){
  int r;
  if(n < 0) n=-n;
  while(n > 0){
    r=c; c=b; b=a; a=r; --n;
  }
  exit(a, b, c);
}
```

path
func
(impe,wt)
(impe,mt)
(impe,ot)
(impe,st)

(a) ローテートを計算するプログラム
(a) A rotation program.



(b) (a)のプログラムのPDFGの例
(b) An example of a PDFG corresponding to program (a).

図 1 サンプルプログラム
Fig. 1 Sample program.

ノード $n_c\in\mathcal{N}_c$ から，m_p に対応する $\mathcal{G}_c$ のノード $m_c\in\mathcal{N}_c$ まで，$\mathcal{G}_c$ において，属性 *exec* のアークを辿って到達する場合，以下の [性質 a] から [性質 c] を保持する PDFG を逐次 PDFG と呼ぶ．

[性質 a] $\mathcal{G}_c$ において，$b_c\in\mathcal{N}_c^{sw}$ を経由して到達できるならば，$\mathcal{G}_p$ において，n_p から m_p まで辿る際，b_c に対応した $\mathcal{G}_p$ のノードを経由する．

[性質 b] $\mathcal{G}_c$ において，$b_c\in\mathcal{N}_c^{sw}$ を必ず経由するならば，$\mathcal{G}_p$ において，n_p から m_p まで辿る際，b_c に対応した $\mathcal{G}_p$ のノードを必ず経由する．

[性質 c] $\mathcal{G}_c$ において，$b_c\in\mathcal{N}_c^{sw}$ を経由しなくても到達できるならば，$\mathcal{G}_p$ において，n_p から m_p まで辿る際，b_c に対応した $\mathcal{G}_p$ のノードを経由しなくても到達できる． □

本稿では，以下，逐次 PDFG を対象として議論を進める．そして，単に PDFG という場合には，逐次 PDFG を指す．

5. PDFG におけるポート間関係

本章では，文献 (13) での議論をもとに，PDFG 上のポート間関係をまとめる．本章では，$\mathcal{G}_p=(\mathcal{N},\mathcal{A},$

```
        < Prog >  ::=  main ( [ < Init-var-list > ] ) < Body >
        < Body >  ::=  { [ < Locl-var-list > ] < Stmt-list > [ exit ( < Retr-var-list > ) ; ] }
< Init-var-list >  ::=  < Decl-list >
< Retr-var-list >  ::=  < Var-list >
< Locl-var-list >  ::=  < Decl-list >
    < Decl-list >  ::=  < Decl > [ < Decl-list > ]
        < Decl >  ::=  < type > < var > ;
     < Var-list >  ::=  < var > | < var > , < Var-list >
    < Stmt-list >  ::=  < Stmt > [ < Stmt-list > ]
        < Stmt >  ::=  { < Stmt-list > } | < var > = < Expr > ; | while ( < Expr > ) < Stmt >
                     | if ( < Expr > ) < Stmt > | if ( < Expr > ) < Stmt > else < Stmt > | ;
        < Expr >  ::=  < var > | < const > | < Expr > < op > < Expr > | < op > < Expr >
```

図 2 言語の構文
Fig. 2 Syntax of the programming language.

$s,e)$ をプログラム Q の PDFG, $\mathcal{N}_b$ を $\mathcal{G}_p$ の *switch-macro* ノードの集合として議論を進める.

5.1 ノード内依存関係

[定義 6] 関係 N_D と関係 N_C.

$(x\in\mathcal{I}(n))\wedge(y\in\mathcal{O}(n))\wedge(n\in\mathcal{N})$ なる x, y を考える. このとき, y から送信するトークンの値が x から受信したトークンの値に依存する場合, または, $\mathcal{N}_d(x)$ が *store* ノードの場合, x から y に関係 N_D があるという. 同様に, y からトークンを送信するかどうかが x から受信したトークンの値に依存する場合, x から y に関係 N_C があるという. ここで, N_D はノード内データ依存関係, N_C はノード内制御依存関係を表す. □

5.2 ポート間関係

[定義 7] 関係集合 $\mathcal{R}$.

$\mathcal{R}=\{a,N_D,N_C|a\in\mathcal{A}_0^2\}$ □

[定義 8] ポート間関係.

$\mathcal{N}_d(x)\in\mathcal{N}$ なる x から $\mathcal{N}_d(y)\in\mathcal{N}$ なる y へ関係 $f\in\mathcal{R}$ が成り立つとき, x から y へポート間関係 f があるといい, $x\xrightarrow{f}y$ と表す. このとき, ポート間関係に対して定義 9で与えられる演算 $\cdot$, $|$, $+$, $\circ$, $*$ を許す. すなわち, f, g をポート間関係としたとき, 定義 9の右辺が成立するならば, 左辺にある $h\in\{f\cdot g, f|g, f^+, f^\circ, f^*\}$ はポート間関係であるといい, x から y へポート間関係 h があるという. □

[定義 9] ポート間関係の演算.

$$x\xrightarrow{f\cdot g}y \triangleq (x\xrightarrow{f}z) \wedge (z\xrightarrow{g}y)$$
$$x\xrightarrow{f|g}y \triangleq (x\xrightarrow{f}y) \vee (x\xrightarrow{g}y)$$
$$x\xrightarrow{f^+}y \triangleq x^{f|(\underrightarrow{f\cdot f^+})}y$$
$$x\xrightarrow{f^\circ}y \triangleq (x\xrightarrow{f}y) \vee (x=y)$$
$$x\xrightarrow{f^*}y \triangleq x^{(\underrightarrow{f^\circ})^+}y$$

ここで, x, y, z は $\mathcal{G}_p$ の任意ポート, f, g は $\mathcal{G}_p$ における任意のポート間関係. □

そして, $\mathcal{G}_p$ の任意のポート間における, 共通入力関係 N_I, 分岐間関係 P_W, 分岐終了間関係 P_E を, 以下に示す.

[定義 10] 関係 N_I, 関係 P_W, 関係 P_E.

$$x\xrightarrow{N_I}x_o \triangleq (y\xrightarrow{N_D}x)\wedge(y\xrightarrow{N_D}x_o)\wedge(x\neq x_o)\wedge(\mathcal{N}_d(x)\in\mathcal{N}_b)$$
$$x\xrightarrow{P_W}y \triangleq x^{((path,mt)\cdot \underrightarrow{N_D})^*\cdot(path,wt)}y$$
$$x\xrightarrow{P_E}y \triangleq x^{((path,mt)\cdot \underrightarrow{N_D})^*\cdot(path,ot)}y$$

ただし, x, x_o, y は $\mathcal{G}_p$ の任意のポート. □

6. PDFG におけるポート集合

本章では, 文献 (13)での議論をもとに, PDFG におけるポート集合を整理する. 本章では, $\mathcal{G}_p=(\mathcal{N},\mathcal{A},s,e)$ をプログラム Q の PDFG, $\mathcal{N}_b$ を $\mathcal{G}_p$ の *switch-macro* ノード集合, $\mathcal{N}_{sb}$ を $\mathcal{N}_b\cup\{s\}$ として議論を進める.

6.1 分岐支配ポート集合

[定義 11] *path* 始点分岐ポート集合 $\mathcal{P}_{sb}$.

$$\mathcal{P}_{sb} \triangleq \{p|(\mathcal{N}_d(p)\in\mathcal{N}_{sb})\wedge \exists x(\mathcal{N}_d(x)\in\mathcal{N})\exists c(c\in\mathcal{A}_t)[p\xrightarrow{(path,c)}x]\}$$ □

[定義 12] 同一分岐方向ポート集合 $\mathcal{F}$.

$$\mathcal{F}(p)\triangleq\{x|\Big(\mathcal{N}_d(x)=\mathcal{N}_d(p)\Big)\wedge\Big(x\in\mathcal{O}(\mathcal{N}_d(x))\Big)\wedge \bigg(\Big((\mathcal{N}_d(p)\in\mathcal{N}_b)\wedge(\mathcal{L}_p(x)=\mathcal{L}_p(p))\Big)\vee(\mathcal{N}_d(p)=s)\bigg)\}$$

ただし, $p\in\mathcal{P}_{sb}$. □

[定義 13] 分岐支配ポート集合 $\mathcal{D}_p$.

$$\mathcal{D}_p(p) \triangleq \{x|(x\notin\mathcal{P}_{sb})\wedge(y^{(\underrightarrow{a|N_D})^*\cdot b^\circ}x)\wedge(y\in\mathcal{F}(p))\}$$

ただし, $(p\in\mathcal{P}_{sb})\wedge(a\in\mathcal{A}_{nw}^2)\wedge(b\in\mathcal{A}_w^2)$. □

6.2 到達依存ポート集合

[定義 14] 非通過到達関係 *pskip*.

$$pskip(x,y,\mathcal{Z})\triangleq\exists\langle x_1,x_2,\ldots,x_i\rangle\Big((x_1=x)\wedge(x_i=y)\wedge (i>1)\wedge\forall j(1\leq j<i)[x_j\xrightarrow{d|N_D}x_{j+1}]\Big) \Big[\forall j(1<j\leq i)\forall z(z\in\mathcal{Z})[\mathcal{N}_d(x_j)\neq\mathcal{N}_d(z)]\Big]$$

ただし, x, y は $\mathcal{G}_p$ の任意のポート, $\mathcal{Z}$ は $\mathcal{G}_p$ の任意のポート集合. $d\in\mathcal{A}_{np}^2$. □

[定義 15] 同一ノードポート集合 $\mathcal{N}_{ps}$.

$$\mathcal{N}_{ps}(x) \triangleq \{p|\mathcal{N}_d(x)=\mathcal{N}_d(p)\}$$

ただし, x は $\mathcal{G}_p$ の任意のポート. □

[定義 16] 到達ポート集合 $\mathcal{E}$.

$$\mathcal{E}(x) \triangleq \{y|pskip(y,x,\mathcal{N}_{ps}(y))\wedge\exists y_o[y_o\xrightarrow{N_I}y]\}$$

ただし, x は $\mathcal{G}_p$ の任意のポート. □

[定義 17] ノード内被後支配関係 N_P.

$$x\xrightarrow{N_P}y \triangleq (x\xrightarrow{N_I}y)\wedge\neg\exists z\Big(\mathcal{N}_d(z)=e\Big)[x\in\mathcal{E}(z)]$$

ただし, x, y は $\mathcal{G}_p$ の任意のポート. □

[定義 18] 片方向到達境界集合 $\mathcal{B}$.

$$\mathcal{B}(x) \triangleq \{y|\exists x_o(x_o\xrightarrow{N_I}x)\Big[\Big((x_o\xrightarrow{N_P}x)\wedge(y=x)\Big)\vee \Big(\neg(x_o\xrightarrow{N_P}x)\wedge pskip(x,z,\mathcal{B}(x))\wedge(z\xrightarrow{d}y)\wedge$$

$\exists x'_o\Big((x'_o\in\mathcal{O}(\mathcal{N}_d(x_o)))\wedge(\mathcal{L}_p(x'_o)=\mathcal{L}_p(x_o))\Big)$
$[(x'_o\in\mathcal{E}(r))\wedge(\mathcal{N}_d(y)=\mathcal{N}_d(r))]\Big)\Big]\}$

ただし，x は $\mathcal{G}_p$ の任意のポート．$d\in\mathcal{A}^2_{np}$．□

[定義 19] 到達依存ポート集合 $\mathcal{B}_{Rd}$.

$\mathcal{B}_{Rd}(y)\triangleq\{x|(x_o\xrightarrow{N_I}x)\wedge\neg(x_o\xrightarrow{N_F}x)\wedge pskip(x,y,\mathcal{B}(x))\}$

ただし，y は $\mathcal{G}_p$ の任意のポート．□

6.3 履歴付きポート集合

本節では，PDFG における経路の表現法と，PDFG の履歴付きポート集合について示す．

6.3.1 経路

[定義 20] 経路.

以下の条件 (a)，(b) を満たす $j(\geq 1)$ 個のポートの並びを経路といい，$\langle p_1,p_2,...,p_j\rangle$ と表す．ここで，p_j を最終ポートといい，経路から経路の最終ポートを返す関数を $\mathcal{P}_t$ とする．(a) $\forall i(1\leq i\leq j)[p_i\in\mathcal{P}_{sb}]$．(b) $j>1$ ならば，$\forall i(1\leq i<j)[p_i\xrightarrow{P_W\cdot N_D}p_{i+1}]$．□

[定義 21] 経路間の隣接関係 $\mathcal{P}_{nbr}$.

$\mathcal{P}_{nbr}(P_x,P_y,q)\triangleq$
$\forall i(1\leq i\leq j)[p_{x_i}=p_{y_i}]\wedge(j+1=k)\wedge(q=p_{y_k})$

ただし，$P_x=\langle p_{x_1},p_{x_2},...,p_{x_j}\rangle$，$P_y=\langle p_{y_1},p_{y_2},...,p_{y_k}\rangle$ は $\mathcal{G}_p$ の任意の経路，$q\in\mathcal{P}_{sb}$．□

[定義 22] 分岐通過経路集合 $\mathcal{P}_b$.

$\mathcal{P}_b(P,p)\triangleq\{\langle p_1,p_2,...,p_k\rangle|(p=p_k)\wedge(1\leq k\leq i)\}$

ただし，$P=\langle p_1,p_2,...,p_i\rangle$ は $\mathcal{G}_p$ の経路，$p\in\mathcal{P}_{sb}$．□

6.3.2 履歴付きポート集合の定義

以下，P を Q の経路として議論を進める．

[定義 23] 履歴付きポート.

直積集合の和集合 $\bigcup_{p\in\mathcal{P}_{sb}}\mathcal{D}_p(p)\times\mathcal{P}_b(P,p)$ を P に対する履歴付きポート集合といい，$\mathcal{H}_{ps}(P)$ と表す．また，$u=(x,h)\in\mathcal{H}_{ps}(P)$ を履歴付きポート，x を u のポート，h を u の履歴という．□

[定義 24] 履歴付きポート間の関係.

$u_i=u_j\triangleq(x_i=x_j)\wedge(h_i=h_j)$
$u_i\xrightarrow{r}u_j\triangleq(x_i\xrightarrow{r}x_j)\wedge(n\in\mathcal{N}_b)\wedge$
$\Bigg(\Big((x_i\notin\mathcal{I}(n))\wedge(h_i=h_j)\Big)\vee$
$\Big((x_i\in\mathcal{I}(n))\wedge(q=\mathcal{P}_t(h_j))\wedge(\mathcal{P}_{nbr}(h_i,h_j,q))\Big)\Bigg)$

ただし，$r\in\mathcal{R}$，$u_i=(x_i,h_i)\in\mathcal{H}_{ps}(P)$，$u_j=(x_j,h_j)\in\mathcal{H}_{ps}(P)$．□

また，定義9における演算は，履歴付きポートに対しても適用可能である．

7. PDFG における依存関係

本章では，文献 (13) の議論をもとに，静的依存関係と動的依存関係を整理する．本章では，P をプログラム Q の経路，$\mathcal{G}_p=\{\mathcal{N},\mathcal{A},s,e\}$ を Q の PDFG，$\mathcal{N}_{so}$ を $\mathcal{G}_p$ の *store* または *operational* ノードの集合として議論を進める．

7.1 PDFG 上の静的依存関係

$\mathcal{G}_p$ の任意のポート間における，静的データ連鎖関係 S_D，静的到達依存関係 S_R，静的制御連鎖関係 S_C を，以下に示す．

[定義 25] PDFG 上の静的依存関係.

$x\xrightarrow{S_D}y\triangleq(x^{(a|\underrightarrow{N_D})^+}y)\wedge(a\in\mathcal{A}^2_d)$
$x\xrightarrow{S_R}y\triangleq(x\xrightarrow{N_C}z)\wedge\Big(z\in\mathcal{B}_{Rd}(y)\Big)$
$x\xrightarrow{S_C}y\triangleq(x\xrightarrow{S_R}y)\wedge\Big(\mathcal{N}_d(y)\in\mathcal{N}_{so}\Big)$

ただし，$\Big(\mathcal{N}_d(x)\in\mathcal{N}\Big)\wedge\Big(\mathcal{N}_d(y)\in\mathcal{N}\Big)$．□

7.2 PDFG 上の動的依存関係

$\mathcal{G}_p$ の P に対する任意の履歴付きポート間における，動的データ連鎖関係 D_D，動的制御連鎖関係 D_C，経路多重データ連鎖関係 M_D，経路多重制御連鎖関係 M_C を，以下に示す．

[定義 26] PDFG 上の動的依存関係.

$u\xrightarrow{D_D}v\triangleq(u^{(a|\underrightarrow{N_D})^+}v)\wedge(a\in\mathcal{A}^2_d)$
$u\xrightarrow{D_R}v\triangleq(u\xrightarrow{N_C}v)\vee\Big((u\xrightarrow{D_R}w)\wedge(w^{d|\underrightarrow{N_D}}v)\wedge$
$(u=(x,h))\wedge(v=(y,i))\wedge(d\in\mathcal{A}^2_{np})\wedge(x\xrightarrow{S_R}y)\Big)$
$u\xrightarrow{D_C}v\triangleq(u\xrightarrow{D_R}v)\wedge(v=(y,i))\wedge\Big(\mathcal{N}_d(y)\in\mathcal{N}_{so}\Big)$
$u\xrightarrow{M_D}v\triangleq(v=(y,i))\wedge$
$\Big(w=(y,j)\in\mathcal{H}_{ps}(P)\Big)\wedge(u\xrightarrow{D_D}w)$
$u\xrightarrow{M_C}v\triangleq(v=(y,i))\wedge$
$\Big(w=(y,j)\in\mathcal{H}_{ps}(P)\Big)\wedge(u\xrightarrow{D_C}w)$

ただし，$\Big(u\in\mathcal{H}_{ps}(P)\Big)\wedge\Big(v\in\mathcal{H}_{ps}(P)\Big)$．□

8. PDFG を用いたスライスの表現

本章では，文献 (13) での議論をもとに，スライス ($Q,I,S,\mathcal{V},f_i,f_d,f_e$) の表現法を示す．本章では，$\mathcal{G}_p=(\mathcal{N},\mathcal{A},s,e)$ を Q の PDFG とする．また，$f_i=dy$ のとき，Q に I を与えて実行した際の経路を P と表す．

まず，$\mathcal{G}_p$ において，依存関係の始点となるポート集合を定義する．

[定義 27] 着目ポート集合.

S と $\mathcal{V}$ に対して，次の条件 (a)，(b) を満たす $\mathcal{G}_p$ のポート y の集合を着目ポート集合と呼び，$\mathcal{N}_A(S,\mathcal{V})$ と表す．(a) $\mathcal{N}_d(y)$ は S に対応したノード．(b) $\mathcal{N}_d(y)$ は，$a\in\mathcal{V}$ に対する *load* ノードまたは *store* ノード．□

更に，$f_i=dy$ の場合，S の位置に対応した履歴を h としたとき，履歴付きポート (x,h) を u と表す．このとき，Q の静的スライスのノード集合 $\mathcal{T}^{f_e}_{f_d}$，Q の P に対する動的スライスのノード集合 $\mathcal{Y}^{f_e}_{f_d}$ は以下の通り表される．ただし，f_d は方向属性，f_e は実行可能性属性である．また，$x\in\mathcal{N}_A(S,\mathcal{V})$.

$\mathcal{T}^{cl}_{bk}(x)\triangleq\{\mathcal{N}_d(y)|y^{(S_D|\underrightarrow{S_C})^+}x\}$
$\mathcal{T}^{cl}_{fw}(x)\triangleq\{\mathcal{N}_d(y)|x^{(S_D|\underrightarrow{S_C})^+}y\}$
$\mathcal{T}^{ex}_{bk}(x)\triangleq\{\mathcal{N}_d(y)|y^{(S_D|\underrightarrow{S_C})^+}x\}$
$\mathcal{T}^{ex}_{fw}(x)\triangleq\{\mathcal{N}_d(y)|(x^{(S_D|\underrightarrow{S_C})^+}z)\wedge(y^{(S_D|\underrightarrow{S_C})^*}z)\}$
$\mathcal{Y}^{cl}_{bk}(u)\triangleq\{\mathcal{N}_d(y)|(v=(y,i)\in\mathcal{H}_{ps}(P))\wedge(v^{(D_D|\underrightarrow{D_C})^+}u)\}$
$\mathcal{Y}^{cl}_{fw}(u)\triangleq\{\mathcal{N}_d(y)|(v=(y,i)\in\mathcal{H}_{ps}(P))\wedge(u^{(D_D|\underrightarrow{D_C})^+}v)\}$
$\mathcal{Y}^{ex}_{bk}(u)\triangleq\{\mathcal{N}_d(y)|(v=(y,i)\in\mathcal{H}_{ps}(P))\wedge(v^{(M_D|\underrightarrow{M_C})^+}u)\}$
$\mathcal{Y}^{ex}_{fw}(u)\triangleq\{\mathcal{N}_d(y)|(v=(y,i)\in\mathcal{H}_{ps}(P))\wedge$
$(u^{(D_D|\underrightarrow{D_C})^+}w)\wedge(v^{(M_D|\underrightarrow{M_C})^*}w)\}$

ここで，ノード集合に対応する文集合が，求めるスライスである．

9. PDFGを用いたスライスの妥当性

本章では，8. で示したスライス表現と，従来のスライスを比較考察する．

9.1 CFG上のポート間関係

まず，従来のスライス定義の際に用いられているCFGの定義を以下に示す．

[定義 28] 制御フローグラフ．

プログラム Q のCFGは，ノード集合 $\mathcal{N}$，アーク集合 $\mathcal{A}$，開始ノード s，終了ノード e の4つ組 $(\mathcal{N},\mathcal{A},s,e)$ で表される有向グラフである．ここで，ノード $n\in\mathcal{N}$ は Q の実行文を，s は開始文を，e は終了文を表す．ここで，ノード $n,m\in\mathcal{N}$ 間で制御移行の関係があるとき，n から m へのアークは属性 $exec$ を持つという．また，各ノードは高々2本の出力アークを持ち，出力アークを2本持つノードを $switch$ ノードという．$switch$ ノードの出力ポートには，$switch$-$macro$ ノードと同様に，ラベル t または f を付与する．そして，$switch$ ノードの出力ポートからラベルを返す関数を $\mathcal{L}_p^c$ とする．

次に，CFGのノード n において，$u\in\mathcal{I}(n)$，$v\in\mathcal{O}(n)$ なる u，v を考える．このとき，u から v に関係 ${N_D}^c$ があるという．また，${N_D}^c$ はCFGノード内依存関係を表す．

アーク属性 $exec$ はPDFGのアーク属性 $d\in\mathcal{A}_0^2$ に対応し，関係 ${N_D}^c$ はPDFGにおける関係 N_D と関係 N_C に対応する．ここで，集合 $\{exec,{N_D}^c\}$ を $\mathcal{R}$ に対応づけると，PDFGと同様に，CFGにおいてもポート間関係が表現できる．更に，CFGにおけるポート間関係に対して定義9の演算を許す．このとき，CFGの任意の2つのポート u，v 間における，被後支配 (Post-dominated) 関係[22] C_P，制御依存関係[22] C_C は，以下のように表される．

$$u\xrightarrow{C_P}v \triangleq \forall\langle u_1,u_2,\ldots,u_i\rangle\Big((u_1=u)\wedge(u_i\in\mathcal{I}(e))\wedge$$
$$(i>1)\wedge\forall j(1\leq j<i)[u_j\xrightarrow{exec|{N_D}^c}u_{j+1}]\Big)$$
$$\Big[\exists j(1<j\leq i)[u_j=v]\Big]$$
$$u\xrightarrow{C_C}v \triangleq \exists w\Big[(w\xrightarrow{{N_D}^c}u)\wedge\neg(w\xrightarrow{C_P}v)\wedge$$
$$\exists\langle u_1,u_2,\ldots,u_i\rangle\Big((u_1=u)\wedge(u_i=v)\wedge(i>1)\wedge$$
$$\forall j(1\leq j<i)[u_j\xrightarrow{exec|{N_D}^c}u_{j+1}]\Big)$$
$$\Big[\forall j(1\leq j<i)[u_j\xrightarrow{C_P}v]\Big]\Big]$$

□

9.2 CFGとPDFGとの対応関係

次に，関係 S_C と関係 C_C の対応について考察する．以下では，$\mathcal{G}_p=(\mathcal{N}_p,\mathcal{A}_p,s_p,e_p)$ をプログラム Q のPDFG，$\mathcal{G}_c=(\mathcal{N}_c,\mathcal{A}_c,s_c,e_c)$ を Q のCFGとして議論を進める．

一般に，Q の文は $\mathcal{G}_p$ では複数のノードにより表される．$n\in\mathcal{N}_c$ に対応するPDFGのノード集合を $\mathcal{N}_{cp}(n)$ と表す．更に，$n\in\mathcal{N}_p$ から $m\in\mathcal{N}_c$ を返す関数 $\mathcal{N}_f$，$\mathcal{N}_d(x)\in\mathcal{N}_p$ なる x から $m\in\mathcal{N}_c$ を返す関数 $\mathcal{N}_{cd}$ を以下の通り定義する．

[定義 29] PDFGからCFGへのノード変換関数．

$$\mathcal{N}_f(n) \triangleq m \quad \text{s.t.} \quad n\in\mathcal{N}_{cp}(m)$$
$$\mathcal{N}_{cd}(x) \triangleq \mathcal{N}_f(\mathcal{N}_d(x))$$

ただし，$(n\in\mathcal{N}_p)\wedge(\mathcal{N}_d(x)\in\mathcal{N}_p)$． □

このとき，制御依存関係 C_C と静的制御連鎖関係 S_C とに関して次の定理が成立する．

[定理 1] $\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)$

$$\Big[\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[u\xrightarrow{(C_C)^+}v]\Leftrightarrow$$
$$\exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[x\xrightarrow{S_C}y]\Big]$$

(証明) **10.** に示す． □

9.3 静的スライスの比較

定理1は，$\mathcal{G}_p$ における関係 S_C と，$\mathcal{G}_c$ における関係 C_C の連鎖により与えられる $\mathcal{G}_p$ のノード集合とが等しいことを意味する．一方，$\mathcal{G}_p$ における関係 S_D は，Paduaらによるデータ依存関係[23),21)] の連鎖と同様の関係を表している．ところで，静的スライスは，データ依存関係の連鎖と制御依存関係の連鎖により表される[24)]．従って，8. で示した静的スライスは，従来の静的スライスと同じ文集合を表すといえる．

9.4 動的スライスの比較

本節では，8. で示した動的スライスと，従来の動的スライスとの対応について考察する．ここでは，8. と同様に，P を，プログラム Q に入力 I を与えて実行させた際の経路とする．以下では，$\mathcal{G}_p=(\mathcal{N},\mathcal{A},s,e)$ をプログラム Q のPDFGとして議論を進める．

9.4.1 従来の動的依存関係

まず，$\mathcal{G}_h$ の任意の2つの履歴付きポート間における，データデータ (Data-Data)[5)] ポート間関係 K_{DD}，テスト制御 (Test-Control)[5)] ポート間関係 K_{TC}，文同一 (Identity)[5)] ポート間関係 K_I は，以下の通り表される．

[定義 30] 関係 K_{DD}，関係 K_{TC}，関係 K_I．

$$u\xrightarrow{K_{DD}}v\triangleq(u\xrightarrow{(b|N_D)^+}v)\wedge(\mathcal{N}_d(x)\in\mathcal{N}_s)\wedge(\mathcal{N}_d(y)\in\mathcal{N}_l)$$
$$u\xrightarrow{K_{TC}}v \triangleq u\xrightarrow{D_C}v$$
$$u\xrightarrow{K_I}v \triangleq x=y$$

ただし，$b\in\{impe\}\times\mathcal{A}_{ns}$．また，$u=(x,h)\in\mathcal{H}_{ps}(P)$，$v=(y,i)\in\mathcal{H}_{ps}(P)$．更に，$\mathcal{N}_s$ は $store$ ノードまたは $start$ ノードの集合で，$\mathcal{N}_l$ は $load$ ノードまたは end ノードの集合． □

Korelらのデータデータ関係は，計算経路に沿ってデータ依存関係が成り立つことを表す．また，Korelらは，if-then-else文とwhile文に対する影響範囲 (scope of influence) を定義し，計算経路に沿って，影響範囲の中の文を経由して到達できる文にテスト制御関係があると定義した[5)]．更に，文同一関係は，同じ文が実行履歴上で異なる位置に現れることを表す．

ところで，定義30の K_{DD} の定義において，関係 $(b|N_D)^+$ は，データ依存関係を表す．また，K_{TC} の定義において，D_C は，定義25より関係 S_C を用いて定義される．ここで，S_C は，定義26より，関係 S_R を用いて定義されるが，S_R は，影響関係を一般的に表した関係である．そして，K_{DD}，K_{TC}，K_I は，いずれも履歴付きポート間関係であり，履歴付きポートは，定義23より，経路を表すといえる．

従って，$(u\xrightarrow{K_{DD}}v)\wedge\Big(u=(x,h)\in\mathcal{H}_{ps}(P)\Big)\wedge\Big(v=(y,i)\in\mathcal{H}_{ps}(P)\Big)$ のとき，そしてそのときに限り，$\mathcal{N}_{cd}(x)$ から $\mathcal{N}_{cd}(y)$ へのデータデータ関係[5)] があるといえ

る． K_{TC}， K_I についても同様である．

9.4.2 従来の動的スライス

次に，従来の逆方向動的クロージャ・スライスと実行可スライスの各ノード集合 $\mathcal{Z}^{cl}$， $\mathcal{Z}^{ex}$ は，以下の通り表現できる．

$$\mathcal{Z}^{cl}(u) \triangleq \{\mathcal{N}_d(y)|(v=(y,i)\in\mathcal{H}_{ps}(P))\wedge (v \xrightarrow{(K_{DD}|K_{TC}|(func,c)|N_D)^+} u)\}$$

$$\mathcal{Z}^{ex}(u) \triangleq \{\mathcal{N}_d(y)|(v=(y,i)\in\mathcal{H}_{ps}(P))\wedge (v \xrightarrow{(K_{DD}|K_{TC}|K_I|(func,c)|N_D)^+} u)\}$$

ただし， u の意味は 8. と同じであり， $c\in\mathcal{A}_t$．

9.4.3 動的スライスの対応関係

以下の定理から， 8. で示した逆方向動的スライスは，従来から提案されている逆方向動的スライス5)-7)を正しく表しているといえる．

[定理 2] $\mathcal{Z}^{cl}(u)=\mathcal{Y}^{cl}_{bk}(u)$

(証明) $v \xrightarrow{(K_{DD}|K_{TC}|(func,c)|N_D)^+} u$

$\Leftrightarrow v \xrightarrow{(N_D|b|D_C|(func,c))^+} u$ ∵定義30より．

$\Leftrightarrow v \xrightarrow{(D_D|D_C)^+} u$ $\because \{(func,c)\}\cup\{b\}=\mathcal{A}^2_d$.

ただし， $b\in\{impe\}\times\mathcal{A}_{ns}$． □

[定理 3] $\mathcal{Z}^{ex}(u)=\mathcal{Y}^{ex}_{bk}(u)$

(証明) $v \xrightarrow{(K_{DD}|K_{TC}|K_I|(func,c)|N_D)^+} u$

$\Leftrightarrow v \xrightarrow{((D_D|D_C)|K_I)^+} u$ ∵定理2より．

$\Leftrightarrow v \xrightarrow{(M_D|M_C)^+} u$ ∵定義26より． □

10. 定理1の証明

本章では，定理1の証明を示す．本章では， 9.3 と同様に， $\mathcal{G}_p=(\mathcal{N}_p,\mathcal{A}_p,s_p,e_p)$ をプログラム Q の PDFG, $\mathcal{G}_c=(\mathcal{N}_c,\mathcal{A}_c,s_c,e_c)$ を Q の CFG として，議論を進める．また， $\mathcal{G}_c$ の *switch* ノードの集合を $\mathcal{N}^{sw}_c$ と表す．

10.1 CFG 上での諸定義

PDFG の場合と同様に， $\mathcal{G}_c$ における CFG 共通入力関係 ${N_I}^c$， CFG 非通過到達関係 $pskip^c$， CFG 同一ノードポート集合 $\mathcal{N}^c_{ps}$， CFG 到達ポート集合 $\mathcal{E}^c$， CFG ノード内被後支配関係 ${N_P}^c$， CFG 片方向到達境界集合 $\mathcal{B}^c$， CFG 到達依存ポート集合 $\mathcal{B}^c_{Rd}$， CFG 静的到達依存関係 ${S_R}^c$ を，以下の通り定義する．

[定義31] ${N_I}^c, pskip^c, \mathcal{N}^c_{ps}, \mathcal{E}^c, {N_P}^c, \mathcal{B}^c, \mathcal{B}^c_{Rd}, {S_R}^c$.

$$u \xrightarrow{{N_I}^c} u_o \triangleq (v \xrightarrow{{N_D}^c} u)\wedge(v \xrightarrow{{N_D}^c} u_o)\wedge(u\neq u_o)$$

$$pskip^c(u,v,\mathcal{W}) \triangleq \exists\langle u_1,u_2,\ldots,u_i\rangle \Big((u_1=u)\wedge (u_i=v)\wedge(i>1)\wedge\forall j(1\leq j<i)[u_j \xrightarrow{exec|{N_D}^c} u_{j+1}]\Big) \Big[\forall j(1<j\leq i)\forall w(w\in\mathcal{W})[\mathcal{N}_d(u_j)\neq\mathcal{N}_d(w)]\Big]$$

$$\mathcal{N}^c_{ps}(u) \triangleq \{q|\mathcal{N}_d(u)=\mathcal{N}_d(q)\}$$

$$\mathcal{E}^c(u) \triangleq \{v|pskip^c(v,u,\mathcal{N}^c_{ps}(v))\wedge\exists v_o[v_o \xrightarrow{{N_I}^c} v]\}$$

$$u \xrightarrow{{N_P}^c} v \triangleq (u \xrightarrow{{N_I}^c} v)\wedge\neg\exists w\Big(\mathcal{N}_d(w)=e_c\Big)[u\in\mathcal{E}^c(w)]$$

$$\mathcal{B}^c(u) \triangleq \{v|\exists u_o(u_o \xrightarrow{{N_I}^c} u)[\Big((u_o \xrightarrow{{N_P}^c} u)\wedge(v=u)\Big)\vee \Big(\neg(u_o \xrightarrow{{N_P}^c} u)\wedge pskip^c(u,w,\mathcal{B}^c(u))\wedge(w \xrightarrow{exec} v)\wedge (u_o\in\mathcal{E}^c(r))\wedge(\mathcal{N}_d(v)=\mathcal{N}_d(r))\Big)]\}$$

$$\mathcal{B}^c_{Rd}(v) \triangleq \{u|(u_o \xrightarrow{{N_I}^c} u)\wedge\neg(u_o \xrightarrow{{N_P}^c} u)\wedge pskip^c(u,v,\mathcal{B}^c(u))\}$$

$$u \xrightarrow{{S_R}^c} v \triangleq u\in\mathcal{B}^c_{Rd}(v)$$

ただし， u， u_o， v は $\mathcal{N}_c$ の任意のポート， $\mathcal{W}$ は $\mathcal{G}_c$ の任意のポート集合． □

上記 ${N_I}^c$ について，定義28より次の性質が導かれる．

[性質 1] $(\mathcal{N}_d(u)\in\mathcal{N}_c)\wedge(\mathcal{N}_d(v)\in\mathcal{N}_c)\wedge(u \xrightarrow{{N_I}^c} v) \Leftrightarrow (u\in\mathcal{O}(n))\wedge(v\in\mathcal{O}(n))\wedge(u\neq v)\wedge(n\in\mathcal{N}^{sw}_c)$ □

10.2 CFG と PDFG との対応関係

本節では， $\mathcal{G}_c$ と $\mathcal{G}_p$ とのグラフ構造に対する対応関係を示す．

[補題 1] $d\in\mathcal{A}^2_{np}$ のとき，

$$(\mathcal{N}_d(x)\in\mathcal{N}_p)\wedge(\mathcal{N}_d(y)\in\mathcal{N}_p)\wedge(x \xrightarrow{d|N_D} y) \Rightarrow \exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big) [u \xrightarrow{(exec|{N_D}^c)^*} v].$$

(証明) $\mathcal{A}^2_{np}=(impe,func)\times\mathcal{A}_t$ なので， $x \xrightarrow{(impe,t)} y$ を $x \xrightarrow{A} y$， $x \xrightarrow{(func,t)|N_D} y$ を $x \xrightarrow{B} y$ と置けば， $x \xrightarrow{d|N_D} y=(x \xrightarrow{A} y)\vee(x \xrightarrow{B} y)$．ただし， $t\in\mathcal{A}_t$． *impe* アークは， Q の文と文との間の順序関係を表すため， $x \xrightarrow{A} y$ ならば $\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)[u \xrightarrow{(exec|{N_D}^c)^+} v]$．一方， *func* アークと関係 N_D は， Q の文の中でのポート間関係を表すため， $x \xrightarrow{B} y$ ならば $\exists u[\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)=\mathcal{N}_{cd}(y)]$． □

[補題 2] $\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)$

$$\Big[\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[u \xrightarrow{{N_I}^c} v] \Leftrightarrow \exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[x \xrightarrow{N_I} y]\Big]$$

(証明) [⇒ の証明] $\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[u \xrightarrow{{N_I}^c} v]$ のとき，性質1よりより $n(=m)$ は *switch* ノード．そして， $l\in\mathcal{N}_{cp}(n)$ なる *switch-macro* ノード l が必ず存在するため，定義10より， $\exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[x \xrightarrow{N_I} y]$.

[⇐ の証明] ⇒ の場合と同様に証明できる． □

[補題 3] $d\in\mathcal{A}^2_{np}$ のとき，

$$\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c) \Big[\exists u\Big((n=\mathcal{N}_d(u))\wedge\exists u_o[u_o \xrightarrow{{N_I}^c} u]\Big) \exists v(m=\mathcal{N}_d(v))[u \xrightarrow{(exec|{N_D}^c)^+} v] \Leftrightarrow \exists x\Big((n=\mathcal{N}_{cd}(x))\wedge\exists x_o[x_o \xrightarrow{N_I} x]\Big) \exists y(m=\mathcal{N}_{cd}(y))[x \xrightarrow{(d|N_D)^+} y]\Big]$$

(証明) [⇒ の証明] まず， $\mathcal{G}_c$ の任意のノード $l\in\mathcal{N}_c$ では，変数の読み出しまたは書き込みが必ず行われる．そのため， l に対して， $\mathcal{G}_p$ において， $l_p\in\mathcal{N}_{cp}(l)$ なる *load* ノードまたは *store* ノードが必ず存在する．ところで，定義5の性質 (a) より， $\mathcal{G}_p$ では，各ノード $m'_p\in\mathcal{N}_p$ に対して， $\mathcal{G}_c$ における $m'=\mathcal{N}_f(m'_p)$ と任意の $n'\in\mathcal{N}^{sw}_c$ との順序関係が，属性 d のアークにより必ず表される．従って， $\mathcal{N}_c$ において， $n\in\mathcal{N}^{sw}_c$ の出力ポート $u\in\mathcal{O}(n)$ から， $m\in\mathcal{N}_c$ のポート v $(\mathcal{N}_d(v)\in\mathcal{N}_c)$ まで， *exec* 属性のアークを辿って到達できるとき， $\mathcal{G}_p$ の *switch-macro* ノード $n_p=\mathcal{N}_{cp}(n)$ の出力ポート $x\in\mathcal{O}(n_p)$ から， $(\mathcal{N}_d(y)=m_p)\wedge(m_p\in\mathcal{N}_{cp}(m))$ なるポート y まで，属性 d のアークを辿って到達できる．

[⇐ の証明] 補題1より明らか． □

[補題 4] $d\in\mathcal{A}^2_{np}$ のとき，

$\exists n(n\in\mathcal{N}_c)\exists m(m\in\mathcal{N}_c)\exists b(b\in\mathcal{N}_c)$

$$\Big[\exists x'(n{=}\mathcal{N}_{cd}(x'))\exists y'(m{=}\mathcal{N}_{cd}(y'))[x'^{(d|\underline{N_D})^+}\!\!\to y']\wedge$$
$$\forall u(n{=}\mathcal{N}_d(u))\forall v(m{=}\mathcal{N}_d(v))$$
$$[\forall\langle u_1,u_2,...,u_i\rangle\Big((u_1{=}u)\wedge(u_i{=}v)\wedge(i{>}1)\wedge$$
$$\forall j(1{\le}j{<}i)[u_j\overset{exec|\underline{N_D}^c}{\to}u_{j+1}]\Big)$$
$$[\exists j(1{<}j{\le}i)\exists w\Big((b{\in}\mathcal{N}_d(w))\wedge\exists w_o[w_o\overset{N_I^c}{\to}w]\Big)$$
$$[\mathcal{N}_d(u_j){=}\mathcal{N}_d(w)]]]\Big]\Rightarrow$$
$$\forall x(n{=}\mathcal{N}_{cd}(x))\forall y(m{=}\mathcal{N}_{cd}(y))$$
$$\Big[\forall\langle x_1,x_2,...,x_i\rangle\Big((x_1{=}x)\wedge(x_i{=}y)\wedge(i{>}1)\wedge$$
$$\forall j(1{\le}j{<}i)[x_j\overset{d|\underline{N_D}}{\to}x_{j+1}]\Big)$$
$$[\exists j(1{<}j{\le}i)\exists z\Big((b{\in}\mathcal{N}_{cd}(z))\wedge\exists z_o[z_o\overset{N_I}{\to}z]\Big)$$
$$[\mathcal{N}_d(x_j){=}\mathcal{N}_d(z)]]]\Big]$$

(証明) まず，$\exists x'(n{=}\mathcal{N}_{cd}(x'))\exists y'(m{=}\mathcal{N}_{cd}(y'))[x'^{(d|\underline{N_D})^+}\!\!\to y']$ は，$\mathcal{G}_p$ において，$\mathcal{N}_d(x')$ から $\mathcal{N}_d(y')$ まで，d 属性のアークを辿って到達できることを意味する．そして，$\forall u(n{=}\mathcal{N}_d(u))\forall v(m{=}\mathcal{N}_d(v))[\forall\langle u_1,u_2,...,u_i\rangle\Big((u_1{=}u)\wedge(u_i{=}v)\wedge(i{>}1)\wedge\forall j(1{\le}j{<}i)[u_j\overset{exec|\underline{N_D}^c}{\to}u_{j+1}]\Big)[\exists j(1{<}j{\le}i)\exists w\Big((b{\in}\mathcal{N}_d(w))\wedge\exists w_o[w_o\overset{N_I^c}{\to}w]\Big)[\mathcal{N}_d(u_j){=}\mathcal{N}_d(w)]]]$ は，$\mathcal{G}_c$ において，n から m へ，$exec$ 属性のアークを辿る際，$b{\in}\mathcal{N}_c^{sw}$ を必ず経由することを意味する．そのため，定義5の性質(b)より，$\mathcal{G}_p$ において，u から v まで d 属性のアークを辿る際，$(b{=}\mathcal{N}_d(w))\wedge\exists w_o[w_o\overset{N_I}{\to}w]$ なる w を必ず経由する． □

[補題 5]　$d{\in}\mathcal{A}_{np}^2$ のとき，
$$\exists n(n{\in}\mathcal{N}_c)\exists m(m{\in}\mathcal{N}_c)\exists b(b{\in}\mathcal{N}_c)$$
$$\Big[\exists x'(n{=}\mathcal{N}_{cd}(x'))\exists y'(m{=}\mathcal{N}_{cd}(y'))[x'^{(d|\underline{N_D})^+}\!\!\to y']\wedge$$
$$\exists u(n{=}\mathcal{N}_d(u))\exists v(m{=}\mathcal{N}_d(v))$$
$$[\exists\langle u_1,u_2,...,u_i\rangle\Big((u_1{=}u)\wedge(u_i{=}v)\wedge(i{>}1)\wedge$$
$$\forall j(1{\le}j{<}i)[u_j\overset{exec|\underline{N_D}^c}{\to}u_{j+1}]\Big)$$
$$[\forall j(1{<}j{\le}i)\forall w\Big((b{\in}\mathcal{N}_d(w))\wedge\exists w_o[w_o\overset{N_I^c}{\to}w]\Big)$$
$$[\mathcal{N}_d(u_j){\ne}\mathcal{N}_d(w)]]]\Big]\Rightarrow$$
$$\exists x(n{=}\mathcal{N}_{cd}(x))\exists y(m{=}\mathcal{N}_{cd}(y))$$
$$\Big[\exists\langle x_1,x_2,...,x_i\rangle\Big((x_1{=}x)\wedge(x_i{=}y)\wedge(i{>}1)\wedge$$
$$\forall j(1{\le}j{<}i)[x_j\overset{d|\underline{N_D}}{\to}x_{j+1}]\Big)$$
$$[\forall j(1{<}j{\le}i)\forall z\Big((b{\in}\mathcal{N}_{cd}(z))\wedge\exists z_o[z_o\overset{N_I}{\to}z]\Big)$$
$$[\mathcal{N}_d(x_j){\ne}\mathcal{N}_d(z)]]]\Big]$$

(証明) 定義5の性質(c)を用い，補題4と同様に証明できる． □

10.3　静的到達依存関係の比較

本節では，$\mathcal{G}_c$ における関係 $S_R{}^c$ と，$\mathcal{G}_p$ における関係 S_R を比較する．

[補題 6]　$\forall n(n{\in}\mathcal{N}_c)\forall m(m{\in}\mathcal{N}_c)\forall l(l{\in}\{t,f\})$
$$\Big[\exists u(n{=}\mathcal{N}_d(u))\exists v\Big((m{=}\mathcal{N}_d(v))\wedge$$
$$\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)[v{\in}\mathcal{E}^c(u)]\Leftrightarrow$$
$$\exists x(n{=}\mathcal{N}_{cd}(x))\exists y\Big((m{=}\mathcal{N}_{cd}(y))\wedge$$
$$\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)[y{\in}\mathcal{E}(x)]\Big]$$

(証明) $d{\in}\mathcal{A}_{np}^2$ として議論を進める．[⇒ の証明] $\exists u(n{=}\mathcal{N}_d(u))\exists v\Big((m{=}\mathcal{N}_d(v))\wedge\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)[v{\in}\mathcal{E}^c(u)]$ のとき，定義31より，$\exists u(n{=}\mathcal{N}_d(u))\exists v\Big((m{=}\mathcal{N}_d(v))\wedge\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)[pskip^c(v,u,\mathcal{N}_{ps}^c(v))]$．更に，定義31より，$\exists u(n{=}\mathcal{N}_d(u))\exists v\Big((m{=}\mathcal{N}_d(v))\wedge\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)\Big[\exists\langle u_1,u_2,...,u_i\rangle\Big((u_1{=}v)\wedge(u_i{=}u)\wedge(i{>}1)\wedge\forall j(1{\le}j{<}i)[u_j\overset{exec|\underline{N_D}^c}{\to}u_{j+1}]\Big)\Big[\forall j(1{<}j{\le}i)\forall w(w{\in}\mathcal{N}_{ps}^c(v))[\mathcal{N}_d(u_j){\ne}\mathcal{N}_d(w)]\Big]\Big]$．補題3，補題2，補題5より，$\exists x(n{=}\mathcal{N}_{cd}(x))\exists y\Big((m{=}\mathcal{N}_{cd}(y))\wedge\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)\Big[\exists\langle x_1,x_2,...,x_i\rangle\Big((x_1{=}y)\wedge(x_i{=}x)\wedge(i{>}1)\wedge\forall j(1{\le}j{<}i)[x_j\overset{d|\underline{N_D}}{\to}x_{j+1}]\Big)\Big[\forall j(1{<}j{\le}i)\forall z(z{\in}\mathcal{N}_{ps}(y))[\mathcal{N}_d(x_j){\ne}\mathcal{N}_d(z)]\Big]\Big]$．そして，定義14より，$\exists x(n{=}\mathcal{N}_{cd}(x))\exists y\Big((m{=}\mathcal{N}_{cd}(y))\wedge\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)[pskip(y,x,\mathcal{N}_{ps}(y))]$．更に，定義16より，$\exists x(n{=}\mathcal{N}_{cd}(x))\exists y\Big((m{=}\mathcal{N}_{cd}(y))\wedge\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)[y{\in}\mathcal{E}(x)]$．

[⇐ の証明] 対偶により証明する．$\neg\exists u(n{=}\mathcal{N}_d(u))\exists v\Big((m{=}\mathcal{N}_d(v))\wedge\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)[v{\in}\mathcal{E}^c(u)]$ を仮定する．まず，定義31より，$\forall u(n{=}\mathcal{N}_d(u))\forall v\Big((m{=}\mathcal{N}_d(v))\wedge\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)[\neg pskip^c(v,u,\mathcal{N}_{ps}^c(v))]$．更に，定義31より，$\forall u(n{=}\mathcal{N}_d(u))\forall v\Big((m{=}\mathcal{N}_d(v))\wedge\exists v_o[v_o\overset{N_I^c}{\to}v]\wedge(l{=}\mathcal{L}_p^c(v))\Big)\Big[\forall\langle u_1,u_2,...,u_i\rangle\Big((u_1{=}v)\wedge(u_i{=}u)\wedge(i{>}1)\wedge\forall j(1{\le}j{<}i)[u_j\overset{exec|\underline{N_D}^c}{\to}u_{j+1}]\Big)\Big[\exists j(1{<}j{\le}i)\exists w(w{\in}\mathcal{N}_{ps}^c(v))[\mathcal{N}_d(u_j){=}\mathcal{N}_d(w)]\Big]\Big]$．補題3，補題2，補題4より，$\forall x(n{=}\mathcal{N}_{cd}(x))\forall y\Big((m{=}\mathcal{N}_{cd}(y))\wedge\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)\Big[\forall\langle x_1,x_2,...,x_i\rangle\Big((x_1{=}y)\wedge(x_i{=}x)\wedge(i{>}1)\wedge\forall j(1{\le}j{<}i)[x_j\overset{d|\underline{N_D}}{\to}x_{j+1}]\Big)\Big[\exists j(1{<}j{\le}i)\exists z(z{\in}\mathcal{N}_{ps}(y))[\mathcal{N}_d(x_j){=}\mathcal{N}_d(z)]\Big]\Big]$．そして，定義14より，$\forall x(n{=}\mathcal{N}_{cd}(x))\forall y\Big((m{=}\mathcal{N}_{cd}(y))\wedge\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)[\neg skip(y,x,\mathcal{N}_{ps}(y))]$．更に，定義16より，$\neg\exists x(n{=}\mathcal{N}_{cd}(x))\exists y\Big((m{=}\mathcal{N}_{cd}(y))\wedge\exists y_o[y_o\overset{N_I}{\to}y]\wedge(l{=}\mathcal{L}_p(y))\Big)[y{\in}\mathcal{E}(x)]$． □

[補題 7]　$\forall n(n{\in}\mathcal{N}_c)\forall m(m{\in}\mathcal{N}_c)$
$$\Big[\exists u\Big(n{=}\mathcal{N}_d(u)\Big)\exists v\Big(m{=}\mathcal{N}_d(v)\Big)[u\overset{N_P^c}{\to}v]\Leftrightarrow$$
$$\exists x\Big(n{=}\mathcal{N}_{cd}(x)\Big)\exists y\Big(m{=}\mathcal{N}_{cd}(y)\Big)[x\overset{N_P}{\to}y]\Big]$$

(証明)　[⇒ の証明]　$u\overset{N_P^c}{\to}v$ のとき，定義31より，$(u\overset{N_I^c}{\to}v)\wedge\neg\exists w\Big(\mathcal{N}_d(w){=}e_c\Big)[u{\in}\mathcal{E}^c(w)]$．ここで，$e_p{\in}\mathcal{N}_{cp}(e_c)$ であるので，補題2，補題6より，$\exists x\Big($

$\mathcal{N}_{cd}(x)=\mathcal{N}_d(u)\Big)\exists y\Big(\mathcal{N}_{cd}(y)=\mathcal{N}_d(v)\Big)[x\xrightarrow{N_I}y]\wedge\forall x\forall y\Big((\mathcal{N}_{cd}(x)=\mathcal{N}_d(u))\wedge(\mathcal{N}_{cd}(y)=\mathcal{N}_d(v))\wedge(x\xrightarrow{N_I}y)\Big)\Big[\neg\exists z\Big(\mathcal{N}_d(z)=e_p\Big)[x\in\mathcal{E}(z)]\Big]$. 従って，定義 17より，$\exists x\Big(\mathcal{N}_{cd}(x)=\mathcal{N}_d(u)\Big)\exists y\Big(\mathcal{N}_{cd}(y)=\mathcal{N}_d(v)\Big)[x\xrightarrow{N_P}y]$.

[⇐ の証明]　⇒ の場合と同様に証明できる．□

[補題 8]　$\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)$

$$\Big[\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[v\in\mathcal{B}^c(u)]\Leftrightarrow \exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[y\in\mathcal{B}(x)]\Big]$$

(証明)　[⇒ の証明]　対偶により証明する．$\neg(y\in\mathcal{B}(x))$ を仮定する．定義 18より，$\exists x_o(x_o\xrightarrow{N_I}x)\Big[\Big(\Big(\neg(x_o\xrightarrow{N_P}x)\vee(y\neq x)\Big)\wedge\neg\Big(pskip(x,z,\mathcal{B}(x))\wedge(z\xrightarrow{d}y)\wedge\exists x'_o\Big((x'_o\in\mathcal{O}(\mathcal{N}_d(x_o)))\wedge(\mathcal{L}_p(x'_o)=\mathcal{L}_p(x_o))\Big)[(x'_o\in\mathcal{E}(r))\wedge(\mathcal{N}_d(y)=\mathcal{N}_d(r))]\Big)\Big)\vee\Big((x_o\xrightarrow{N_P}x)\wedge\neg(y\neq x)\Big)\Big]$.

ここで，$\exists x_o(x_o\xrightarrow{N_I}x)[\neg(x_o\xrightarrow{N_P}x)\vee(y\neq x)]$ のとき，補題 7より，$\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)\Big[\exists u_o(u_o\xrightarrow{{N_I}^c}u)[\neg(u_o\xrightarrow{{N_P}^c}u)\vee(v\neq u)]\Big]$.

また，$\exists x_o(x_o\xrightarrow{N_I}x)\Big[\Big(\neg pskip(x,z,\mathcal{B}(x))\wedge(z\xrightarrow{d}y)\wedge\exists x'_o\Big((x'_o\in\mathcal{O}(\mathcal{N}_d(x_o)))\wedge(\mathcal{L}_p(x'_o)=\mathcal{L}_p(x_o))\Big)[(x'_o\in\mathcal{E}(r))\wedge(\mathcal{N}_d(y)=\mathcal{N}_d(r))]\Big)\Big]$ のとき，定義 14より，$\exists x_o(x_o\xrightarrow{N_I}x)\Big[\exists r'\Big((x\xrightarrow{(d|N_D)^+}r')\wedge(r'\xrightarrow{d|N_D}y)\Big)\Big[\exists x'_o\Big((x'_o\in\mathcal{O}(\mathcal{N}_d(x_o)))\wedge(\mathcal{L}_p(x'_o)=\mathcal{L}_p(x_o))\Big)[(x'_o\in\mathcal{E}(r'))]\Big]\Big]$. 補題 3より，$\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)\Big[\exists u_o(u_o\xrightarrow{{N_I}^c}u)\Big[\exists w'\Big((u\xrightarrow{(exec|{N_D}^c)^+}w')\wedge(w'\xrightarrow{exec|{N_D}^c}v)\Big)\Big[(u_o\in\mathcal{E}^c(r))\wedge(\mathcal{N}_d(w')=\mathcal{N}_d(r))\Big]\Big]\Big]$. 従って，定義 31より，$\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)\Big[\exists u_o(u_o\xrightarrow{{N_I}^c}u)\Big[\neg pskip^c(u,w,\mathcal{B}_c(u))\wedge(w\xrightarrow{exec^c}v)\wedge(u_o\in\mathcal{E}^c(r))\wedge(\mathcal{N}_d(v)=\mathcal{N}_d(r))\Big]\Big]$.

更に，$\exists x_o(x_o\xrightarrow{N_I}x)[(x_o\xrightarrow{N_P}x)\wedge(y\neq x)]$ のとき，補題 7より，$\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)\Big[\exists u_o(u_o\xrightarrow{{N_I}^c}u)[(u_o\xrightarrow{{N_P}^c}u)\vee(v\neq u)]\Big]$.

以上より，$\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)\Big[\exists u_o(u_o\xrightarrow{{N_I}^c}u)\Big[\neg(u_o\xrightarrow{{N_P}^c}u)\vee(v\neq u)\vee\neg\Big(pskip^c(u,w,\mathcal{B}_c(u))\wedge(w\xrightarrow{exec^c}v)\wedge(u_o\in\mathcal{E}^c(r))\wedge(\mathcal{N}_d(v)=\mathcal{N}_d(r))\Big)\vee(u_o\xrightarrow{{N_P}^c}u)\vee(v\neq u)\Big]\Big]$. 従って，定義 31より，$\exists u\Big(\mathcal{N}_d(u)=\mathcal{N}_{cd}(x)\Big)\exists v\Big(\mathcal{N}_d(v)=\mathcal{N}_{cd}(y)\Big)[\neg(v\in\mathcal{B}^c(u))]$.

[⇐ の証明]　⇒ の場合と同様に証明できる．□

[補題 9]　$\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)$

$$\Big[\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[v\in\mathcal{B}^c_{Rd}(u)]\Leftrightarrow \exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[y\in\mathcal{B}_{Rd}(x)]\Big]$$

(証明)　[⇒ の証明]　対偶により証明する．定義 19，定義 18，補題 2，補題 7，補題 8，定義 31を用いて，補題 8と同様に証明できる．

[⇐ の証明]　⇒ の場合と同様に証明できる．□

[補題 10]　$\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)$

$$\Big[\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[u\xrightarrow{{S_R}^c}v]\Leftrightarrow \exists x_o\Big(n=\mathcal{N}_{cd}(x_o)\Big)\exists y_o\Big(m=\mathcal{N}_{cd}(y_o)\Big)[x_o\xrightarrow{S_R}y_o]\Big]$$

(証明) 定義 25，補題 9，定義 31より証明できる．□

10.4　CFG 上での関係の比較

本節では，$\mathcal{G}_c$ 上での関係を比較する．

まず，$\mathcal{G}_c$ における関連 (relevant) 関係[24] を C_R とすると，以下のように表される．

[定義 32]　関係 C_R.

$$u\xrightarrow{C_R}v\triangleq\exists\langle u_1,u_2,\ldots,u_i\rangle\Big((u_1=u)\wedge(u_i=v)\wedge(i>1)\wedge \forall k(1\leq k<i)[u_k\xrightarrow{exec|{N_D}^c}u_{k+1}]\Big)$$
$$\exists\langle v_1,v_2,\ldots,v_j\rangle\Big((u\xrightarrow{{N_I}^c}v_1)\wedge(v_j\in\mathcal{I}(e_c))\wedge (j>1)\wedge\forall l(1\leq l<j)[v_l\xrightarrow{exec|{N_D}^c}v_{l+1}]\Big)$$
$$\Big[\forall k(1<k\leq i)\forall l(1<l\leq j)[\mathcal{N}_d(u_k)\neq\mathcal{N}_d(v_l)]\Big]$$ □

次に，$\mathcal{G}_c$ での関係 ${S_R}^c$ と関係 C_R，および，関係 C_R と関係 C_C とを比較する．

[補題 11]

$\forall u(\mathcal{N}_d(u)\in\mathcal{N}_c)\forall v(\mathcal{N}_d(v)\in\mathcal{N}_c)[u\xrightarrow{{S_R}^c}v\Leftrightarrow u\xrightarrow{C_R}v]$

(証明)　[⇒ の証明]　$u\xrightarrow{{S_R}^c}v$ ならば，$\exists u_o[(u_o\xrightarrow{{N_I}^c}u)\wedge\neg(u_o\xrightarrow{{N_P}^c}u)]\wedge\exists w[\Big(w\in\mathcal{B}^c(u)\Big)\wedge(w\xrightarrow{(exec|{N_D}^c)^+}q)\wedge\Big(q\in\mathcal{I}(e_c)\Big)]\wedge pskip^c(u,v,\mathcal{B}^c(u))$ なので，$u\xrightarrow{C_R}v$.

[⇐ の証明] 図 2に示した言語の構文と，定義 32，定義 31より明らか．□

[補題 12]　$\forall u(\mathcal{N}_d(u)\in\mathcal{N}_c)\forall v(\mathcal{N}_d(v)\in\mathcal{N}_c)[u\xrightarrow{C_R}v\Leftrightarrow u\xrightarrow{(C_C)^+}v]$

(証明)　Weiss らの文献 (24)参照．□

10.5　PDFG 上での関係の比較

本節では，$\mathcal{G}_p$ 上での関係 S_R と関係 S_C を比較する．

[補題 13]　$\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)$

$$\Big[\exists x_o\Big(n=\mathcal{N}_{cd}(x_o)\Big)\exists y_o\Big(m=\mathcal{N}_{cd}(y_o)\Big)[x_o\xrightarrow{S_R}y_o]\Leftrightarrow \exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[x\xrightarrow{S_C}y]\Big]$$

(証明)　[⇒ の証明] まず，$x_o\xrightarrow{S_R}y_o$ のとき，定義 25，定義 19より $\mathcal{N}_d(x_o)$ は *switch-macro* ノードなので，$\exists x[x\xrightarrow{N_C}x_o]$. また，任意の $m\in\mathcal{N}_c$ に対して，$m=\mathcal{N}_{cd}(y)$ なる *store* ノードまたは *operational* ノード y は必ず存在する．

[⇐ の証明] 定義 25より明らか．□

10.6　定理 1 の証明

(1) 補題 12，補題 11より，$\forall u(\mathcal{N}_d(u)\in\mathcal{N}_c)\forall v(\mathcal{N}_d(v)\in\mathcal{N}_c)[u\xrightarrow{(C_C)^+}v\Leftrightarrow u\xrightarrow{{S_R}^c}v]$.　(2) 補題 10，補題 13より，$\forall n(n\in\mathcal{N}_c)\forall m(m\in\mathcal{N}_c)\Big[\exists u\Big(n=\mathcal{N}_d(u)\Big)\exists v\Big(m=\mathcal{N}_d(v)\Big)[u\xrightarrow{{S_R}^c}v]\Leftrightarrow\exists x\Big(n=\mathcal{N}_{cd}(x)\Big)\exists y\Big(m=\mathcal{N}_{cd}(y)\Big)[x\xrightarrow{S_C}y]\Big]$. 従って，(1)，(2) より，定理 1が示される．□

11. むすび

統一的な枠組の上で様々なスライスを形式的に表現する手法を提案した．本稿では，まず，従来のスライスを3種類の属性に従い8通りに分類した．次に，PDFG上で各種依存関係を定義し，これら依存関係の組合せによりスライスを表現した．更に，提案手法により得られるスライスが，従来のスライスを正しく表現していることを証明することにより，提案手法の妥当性を示した．

現在，Infeasible Path[14)-16),18)]や計算経路などを考慮することによりスライスの精度を向上させる方法[25)]や，表現したスライスを効率的に求める手法[13)]，更に，各種スライスの用途の検討を進めている．また，より複雑な制御構造を持つプログラムなど，対象プログラムを拡大する方法の検討を進める予定である．

謝辞 日ごろ御指導御討論頂く後藤滋樹部長，伊藤正樹リーダはじめ広域コンピューティング研究部の皆様に感謝いたします．

参考文献

1) Weiser M.: "Program slicing", IEEE Trans. Softw.Eng., Vol. SE-10, No. 4, pp.352–357 (1984).
2) Ottenstein K.J. and Ottenstein L.M.: "The program dependence graph in a software development environment", ACM SIGPLAN Notices, Vol. 19, No. 5, pp.177–184 (1984).
3) Horwitz S., Prins J. and Reps T.: "Integrating noninterfering versions of programs", ACM Trans.Prog.Lang.Syst., Vol. 11, No. 3, pp.345–387 (1989).
4) Horwitz S., Reps T. and Binkley D.: "Interprocedural slicing using dependence graphs", ACM Trans.Prog.Lang.Syst., Vol. 12, No. 1, pp.26–60 (1990).
5) Korel B. and Laski J.: "Dynamic program slicing", Inf.Process.Lett., Vol. 29, pp.155–163 (1988).
6) Korel B.: "PELAS—Program Error-Locating Assistant System", IEEE Trans.Softw.Eng., Vol. SE-14, No. 9, pp.1253–1260 (1988).
7) Agrawal H. and Horgan J.R.: "Dynamic program slicing", ACM SIGPLAN Notices, Vol. 25, No. 6, pp.246–256 (1990).
8) 下村隆夫："変数値エラーにおける Critical Slice に基づくバグ究明戦略"，情報処理学会論文誌，Vol. 33, No. 4, pp.501–511 (1992).
9) Korel B. and Laski J.: "Dynamic slicing of computer programs", Journal of Systems and Software, Vol. 13, pp.187–195 (1990).
10) Gallagher K.B. and Lyle J.R.: "Using program slicing in software maintenance", IEEE Trans.Softw.Eng., Vol. 17, No. 8, pp.751–761 (1991).
11) 高橋直久："ソフトウェア・リエンジニアリングにおける情報の構造と変換"，情報処理学会情報システム研究会，No. 42-3, pp.27–36 (1993).
12) Venkatesh G.A.: "The semantic approach to program slicing", ACM SIGPLAN Notices, Vol. 26, No. 6, pp.107–119 (1991).
13) 直井邦彰，高橋直久："経路依存フローグラフを用いたプログラム・スライシング"，電子情報通信学会ソフトウェアサイエンス研究会，No. SS93-25, pp. 41–48 (1993).
14) 直井邦彰，高橋直久："プレスブルガー算術を用いた Infeasible Path 検出法"，電子情報通信学会知能ソフトウェア工学研究会，No. KBSE92-38, pp.57–64 (1992).
15) 直井邦彰，高橋直久："経路依存フローグラフを用いた Infeasible Path 検出法"，電子情報通信学会論文誌，Vol. J76-D-I, No. 8, pp.429–439 (1993).
16) 直井邦彰，高橋直久："経路依存フローグラフを用いたプログラム解析"，NTT R&D, Vol. 42, No. 8, pp.1007–1016 (1993).
17) 直井邦彰，高橋直久："経路依存フローグラフを用いたプログラム・スライシング"，情報処理学会第46回全国大会講演論文集，6E-7 (1993).
18) 直井邦彰，高橋直久："経路依存フローグラフを用いた Infeasible Path 検出における計算量削減法"，情報処理学会第48回全国大会講演論文集，6G-2 (1994).
19) 直井邦彰，高橋直久："経路依存フローグラフを用いたプログラム解析システム"，情報処理学会第47回全国大会講演論文集，5D-9 (1993).
20) 雨宮真人："データフロー・アーキテクチャについて"，コンピュータソフトウェア，Vol. 1, No. 1, pp.42-63 (1984).
21) Aho A.V., Sethi R. and Ullman J.D.: "Compilers: Principles, Techniques, and Tools", Addison-Wesley (1986).
22) Ferrante J., Ottenstein K.J. and Warren J.D.: "The program dependence graph and its use in optimization", ACM Trans.Prog.Lang.Syst., Vol. 9, No. 3, pp.319–349 (1987).
23) Padua D.A. and Wolfe M.J.: "Advanced compiler optimizations for supercomputers", Comm. ACM, Vol. 29, No. 12, pp.1184–1201 (1986).
24) Weiss M.: "The transitive closure of control dependence: The iterated join", ACM Letters on Programming Languages and Systems, Vol. 1, No. 2, pp.178–190 (1992).
25) 直井邦彰，高橋直久："ソフトウェア意味構成管理モデルにおけるスライシング技法を用いたプログラム解析"，情報処理学会第49回全国大会講演論文集，2N-6 (1994).