

プログラミング言語処理系論 2010 レポート課題

5/12講義分

- (課題1) XMLの文法がLL(1)であることを示し、XMLのパースを書け。
 - (課題2) dc.outputを解析し、どのような受理機械が生成されたか述べよ
 - (課題3) linesの定義の1行目が lines expr となっていて、expr lines となっていない理由を受理機械の動作から説明せよ
-

5/26 講義分

(課題4) 変数の扱いについて、以下の点から考察し、コードを改良せよ。

(a) 関数には引数を渡せないのか？

(b) 引数が渡されると、変数名とデータの対応がどのように変更されるか。また、関数からリターンするとき、どのような処理が必要か？(ヒントキーワード: 環境、スコープ)

(課題5) 四則演算と基本的な制御構造が実現できればプログラミング言語の骨格は完成し、関数を実現できれば文句なしである。

(a) 自分でdcc6以上のプログラミング言語を一つ定義し、その規格を書いてみよ。

(b) 規格からparserを生成せよ。Parserの出力は必ずしも実行コードである必要はない。

6/2講義分

(課題6)

JAVA VMの仕様を読み、以下の観点から整理して特徴を記せ。

- (1) VMの命令があつかうデータ
- (2) メモリ(スタックとヒープ)管理
- (3) スレッド管理
- (4) オブジェクト管理

(課題6')

Microsoftが提供する.NETの中でのVM環境であるCLR(共通言語ランタイム)について、Java VMと同じ事を調査せよ。

6/2講義分(続き)

□ (課題7)

手近にあるコンパイラをひとつ対象にし、calling conventionとフレームを解析せよ。この時に

以下のことに注意せよ

(1)コンパイラ・OS・CPUを明記すること

(2)解析の手法を明らかにすること

(3)calling conventionは、呼び出し側と呼び出される側の約束であるが、そのときに呼び出される側から呼び出し側へも約束が存在することに注意せよ

6/9講義分

- (課題8) ループ i, j, k の順序をどのようにしてもよいことを証明せよ。それならば、どのような順序で実行するのがよいのか。現在支配的なメモリアーキテクチャを前提として考えよ。
 - (課題8') ループアンローリングは計算の順序どころか、変数へのアクセスの順番をまったく変更しないことを示せ。それにも係わらず性能が上がるのがよくあるのはなぜか、現在支配的なCPUアーキテクチャを前提として考えよ(ループのオーバヘッドの削減は理由にならない)。
-

6/9 講義分

- (課題9) 以下の命令列に対してvalue numberingを行え。授業中で概説したアルゴリズムを補完し、計算の過程も記すこと。

$$g = x + y$$

$$h = u - v$$

$$i = x + y$$

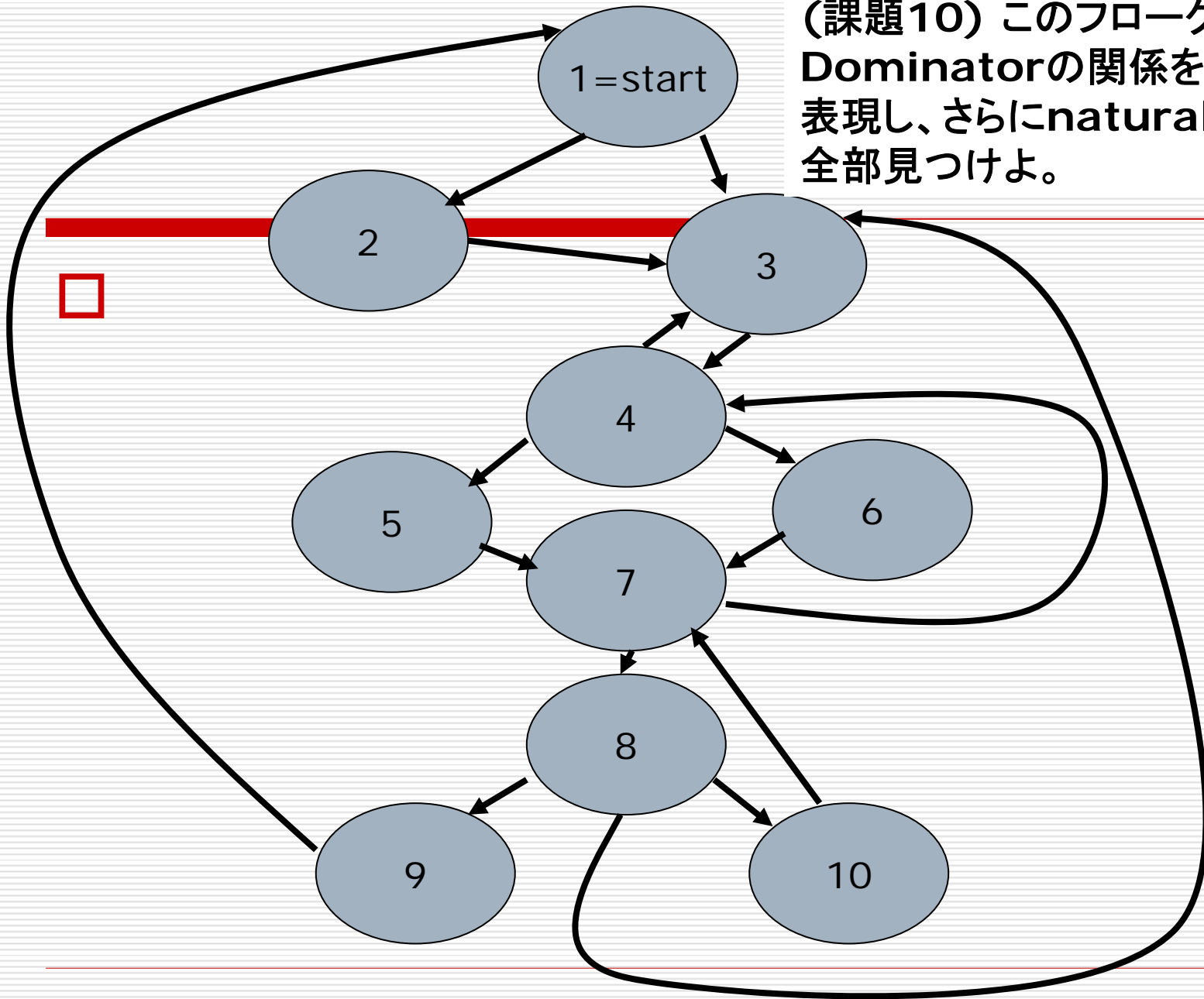
$$x = u - v$$

$$u = g + h$$

$$v = i + x$$

$$w = u + v$$

(課題10) このフローグラフで Dominator の関係を木構造で表現し、さらに natural loop を全部見つけよ。



6/23 講義分

(課題11)この(連立)方程式を解け

$$\text{in}[1] = \text{out}[1] - \{a\}; \quad \text{out}[1] = \text{in}[2]$$

$$\text{in}[2] = \{a\} + (\text{out}[2] - \{b\});$$

$$\text{out}[2] = \text{in}[3] + \text{in}[6]$$

$$\text{in}[3] = \{c, b\} + (\text{out}[3] - \{c\}); \quad \text{out}[3] = \text{in}[4]$$

$$\text{in}[4] = \{b\} + (\text{out}[4] - \{a\}); \quad \text{out}[4] = \text{in}[5]$$

$$\text{in}[5] = \{a\} + \{\text{out}[5]\}; \quad \text{out}[5] = \text{in}[6]$$

$$\text{in}[6] = \text{out}[6]; \quad \text{out}[6] = \text{in}[7] + \text{in}[2]$$

$$\text{in}[7] = \{c\} + \text{out}[7]; \quad \text{out}[7] = \phi$$

6/23講義分

□ 定理:前スライドの繰り返しは単調に増加(和型)、または単調に減少(積型)する。

□ 定理:コンパイラ最適化では、in,outの各集合は有限集合であるので、必ず収束する。

(課題12)各繰り返しにおいてgenとkillが一定であると仮定した上で上の定理2つを証明せよ。この結果として、「出てきた解は最小(または最大)である」ことを併せて証明せよ。

(課題13) KnoopらのLazy Code Motionの論文を読んでサマリーを書け。(Knoop, et.al., Lazy Code Motion, Programming Language Design and Implementation '92, 1992.)

6/30講義分

- (課題14) 自分で適当にloop, if, compoundを持つ簡単なプログラミング言語を想定し、フローグラフのパターンを決め、 ϕ の挿入場所を特定せよ。
-