

## Positive

- Like: init/final/flow definitions at the start.
- Like: step-by-step program slicing examples.
- Like: discussion of if/do handling in program slicing.

## Comments

### Formatting

- Formatting of algorithms, function definitions, and tables in this draft detracts from the readability of those elements. Program examples, however, are generally well laid-out. Do not mix pseudocode, natural language, and programming syntax in one algorithm description. It may be a good idea would be present function definitions in a more formal way like it is done in a book.

### Data Structure

- The “at most three addresses per command” simplification is troublesome. Note that the language explicitly permits more complex expressions; you’ll either have to declare some programs unsupported or write code to translate input to your simplification (and output back to a form familiar to the user).
- The tree/array representation of the program input is overly complex. The tree in the illustration seems to be a mix of a syntax tree (where the nodes are syntax constructions) and a control flow graph (where the nodes are statements). Adding blank nodes for if statement actions seems redundant: you already allow  $y:=y-1$  to be a child of  $x:=x+1$ , so  $x:=x+1$  might as well take the place of the blank node in your tree. It may also be better to have the guarded command be a child of its guard condition, avoiding having two different types of children for the if/do nodes. The performance considerations could be presented better, they are currently somewhat difficult to follow. You need labels to allow *the user* to refer to a point of interest; internally, references to Java objects would probably work for things like keeping track of the program flow.

### Constant Folding

- This section feels less polished than the others. Constant folding is a transformation of program code; live variables analysis is the analysis used. Mentioning “monotone framework”, and specifying the new transfer functions, in combination with the book/slides, does describe how you could get the LV results. The implementation section seems to focus on implementing live variable analysis; but is somewhat confusing as it seems to describe updating a *single* table while examining the *current* statement – additional context would help here. More focus on the actual application of the live variables analysis, as well as input/output examples would be helpful.