CSCI 742 - Compiler Construction

Lecture 34

Available Expressions Analysis

Instructor: Hossein Hojjat

April 18, 2018

Recap: Live Variable Analysis

- Compute which variables are live at each program point
- Live variable information flows backward
- Derive a system of constraints between live variable sets

$$\mathit{in}(S) = \big(\mathit{out}(S) \setminus \mathit{def}(S)\big) \ \cup \ \mathit{use}(S)$$
$$\mathit{out}(S) = \bigcup_{S_i \in \mathit{succ}(S)} \mathit{in}(S_i)$$

- Solve constraints in an iterative algorithm

Available Expressions

Idea: some computation may be a redundant repetition of earlier computation within the same program

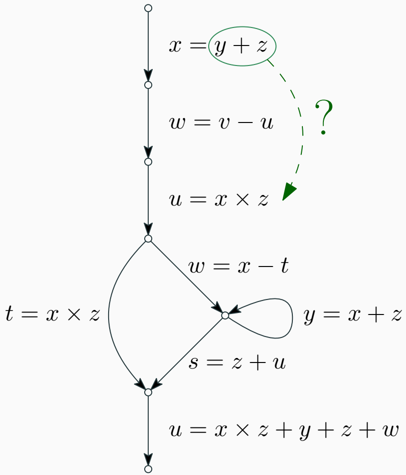
An expression like `x+y` is available at a statement S if

- Every path from the entry node to S compute `x+y` before reaching S
- There are no assignments to `x` or `y` since the last time `x+y` was computed on the paths to S

Optimization: If an expression is available, don't need to recompute it

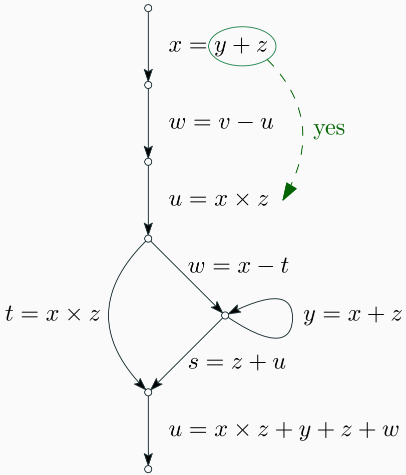
Example: Available Expression

- Is the expression available?



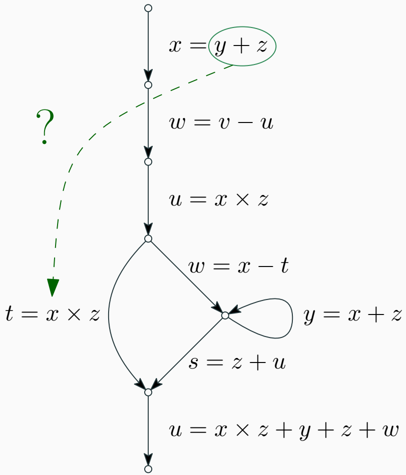
Example: Available Expression

- Is the expression available?



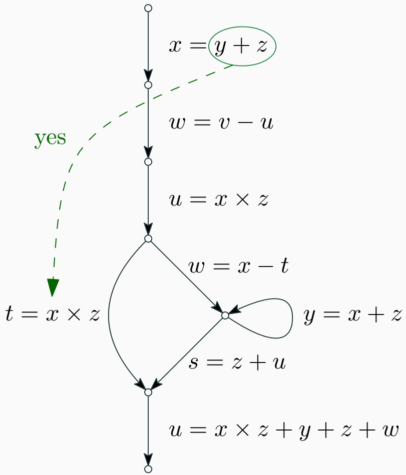
Example: Available Expression

- Is the expression available?



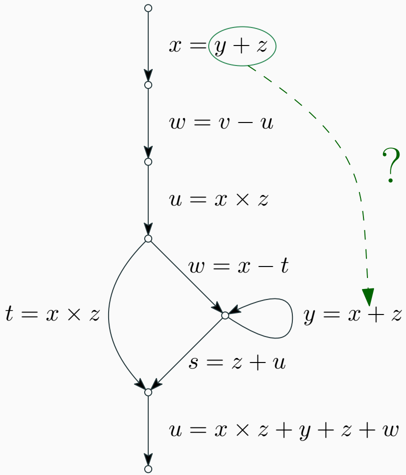
Example: Available Expression

- Is the expression available?



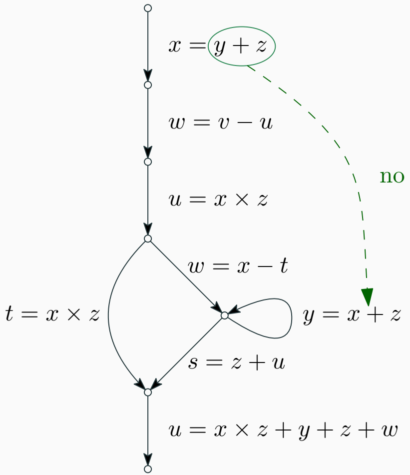
Example: Available Expression

- Is the expression available?



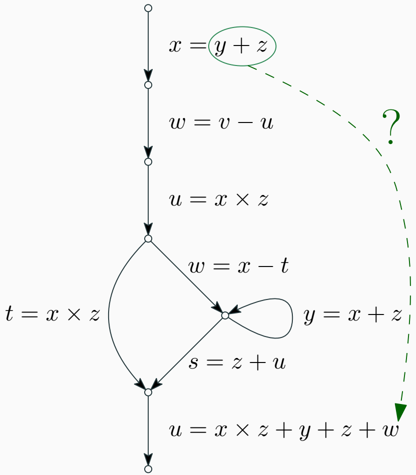
Example: Available Expression

- Is the expression available?



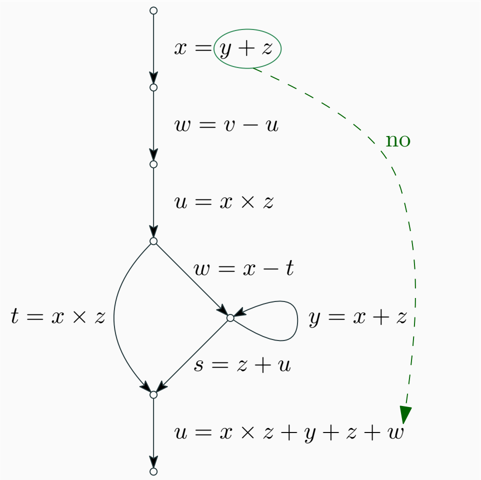
Example: Available Expression

- Is the expression available?



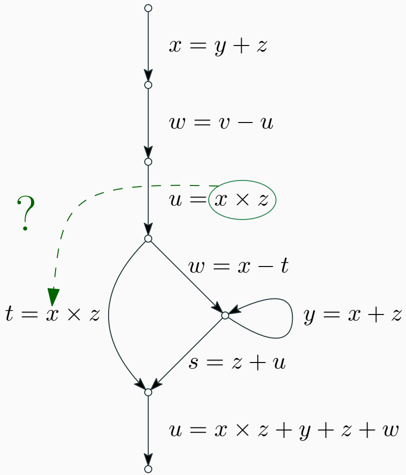
Example: Available Expression

- Is the expression available?



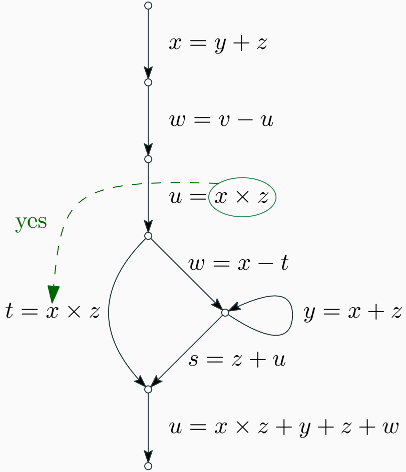
Example: Available Expression

- Is the expression available?



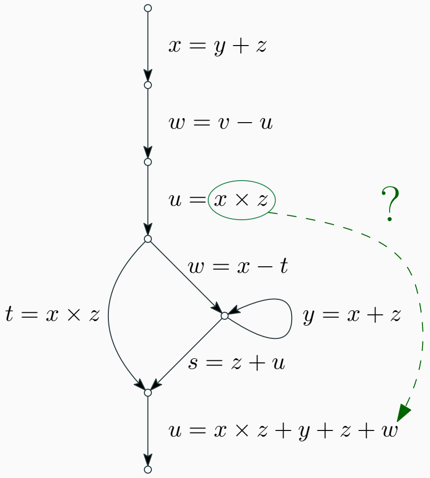
Example: Available Expression

- Is the expression available?



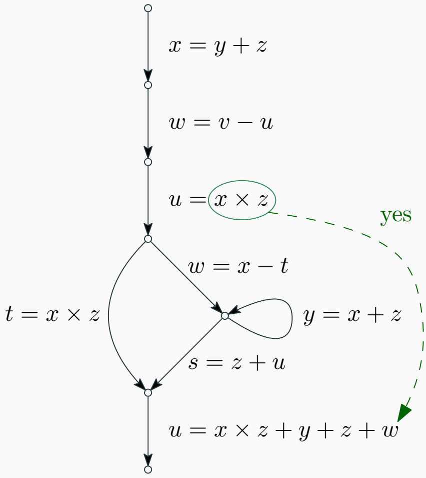
Example: Available Expression

- Is the expression available?



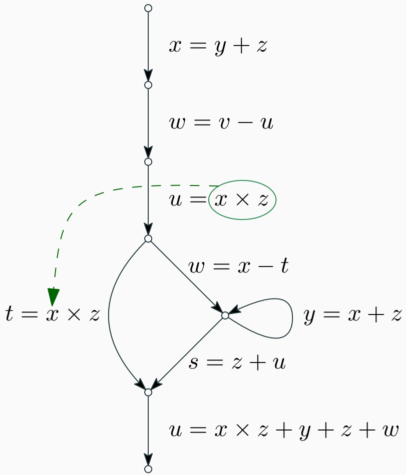
Example: Available Expression

- Is the expression available?



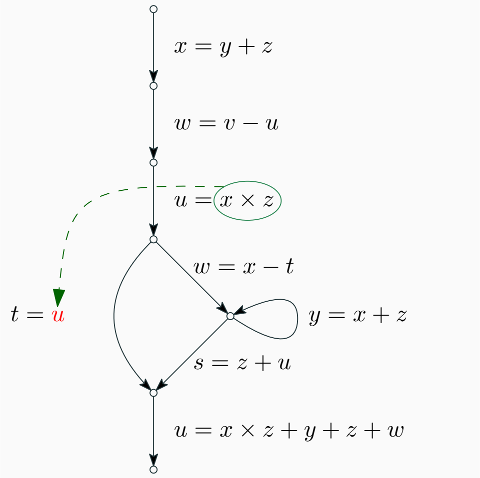
Example: Available Expression

- How can we use available expression?



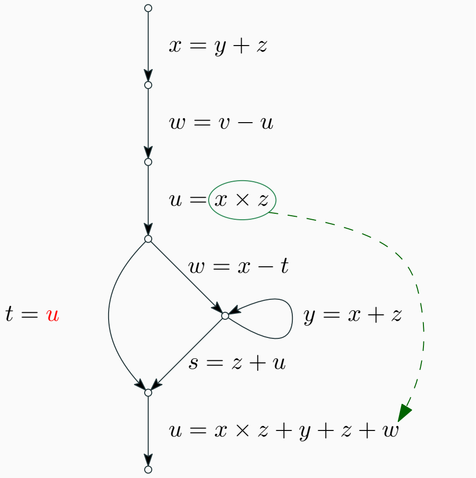
Example: Available Expression

- How can we use available expression?



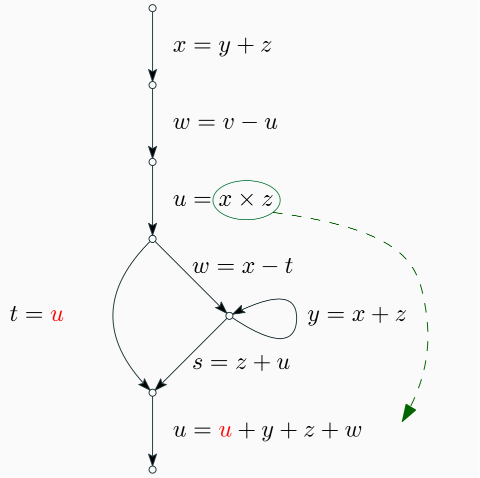
Example: Available Expression

- How can we use available expression?



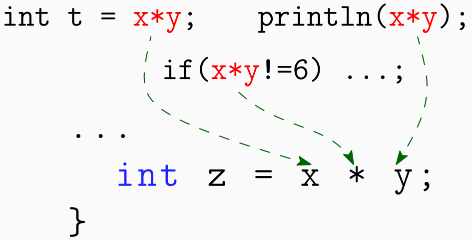
Example: Available Expression

- How can we use available expression?



Available Expression Analysis: Forward

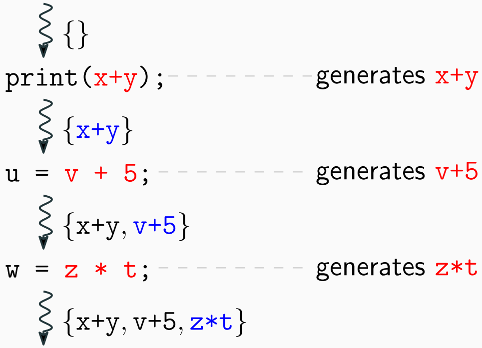
- Available expression analysis is a **forward** data-flow analysis
- Information from past instructions must be propagated forward through the program to discover which expressions are available



- Unlike variable liveness, expression availability flows forward through the program
- Like liveness, each instruction has an effect on the availability information as it flows past

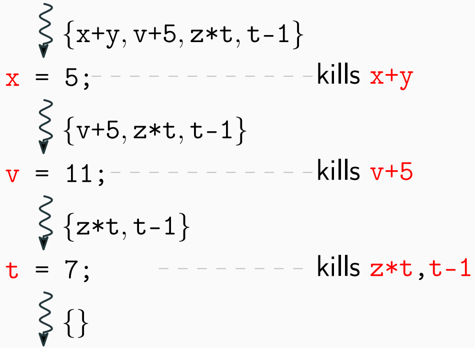
Available Expression Analysis

- A statement makes an expression **available** when it generates (computes) its current value



Available Expression Analysis

- A statement makes an expression **unavailable** when it kills (invalidates) its current value



Available Expression Analysis

- As in Live Variable Analysis, we create functions *gen*(S) and *kill*(S) which give the sets of expressions the statement S generates and kills
- Assignment to a variable $\mathtt{x}$ kills all expressions in the program which contain occurrences of $\mathtt{x}$ ($E_{\mathtt{x}}$)

$\{\mathtt{y-1}\}$

gen(`println(x+5)`) $= \{$ `x+5` $\}$

$\{\mathtt{y-1}, \mathtt{x+5}\}$

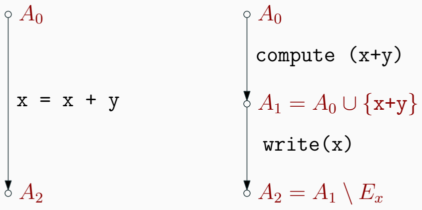
$\{\mathtt{y-1}, \mathtt{x+5}\}$

kill(`x = 3`) $= \{ E_{\mathtt{x}} \}$

$\{\mathtt{y-1}\}$

Available Expression Analysis

- If a statement both generates and kills expressions, remove the killed expressions after adding the generated ones



$\textit{gen}(\texttt{x = x + y}) = \{\texttt{x+y}\}$ $\textit{kill}(\texttt{x = x + y}) = \{\, E_{\texttt{x}} \,\}$

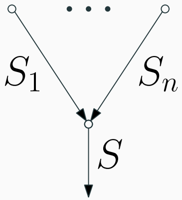
In general:

- $\textit{in}(S)$: set of available expressions immediately before statement S
- $\textit{out}(S)$: set of available expressions immediately after statement S

$$\textit{out}(S) = \big(\textit{in}(S) \cup \textit{gen}(S)\big) \ \setminus \ \textit{kill}(S)$$

Multiple Successors

- An expression is available at beginning of statement S if it is available at the end of all predecessor statements



$$\textit{in}(S) = \bigcap_{S_i \in \textit{pred}(S)} \textit{out}(S_i)$$

Data-flow Equations

- Start with CFG and derive a system of constraints between sets of available expressions

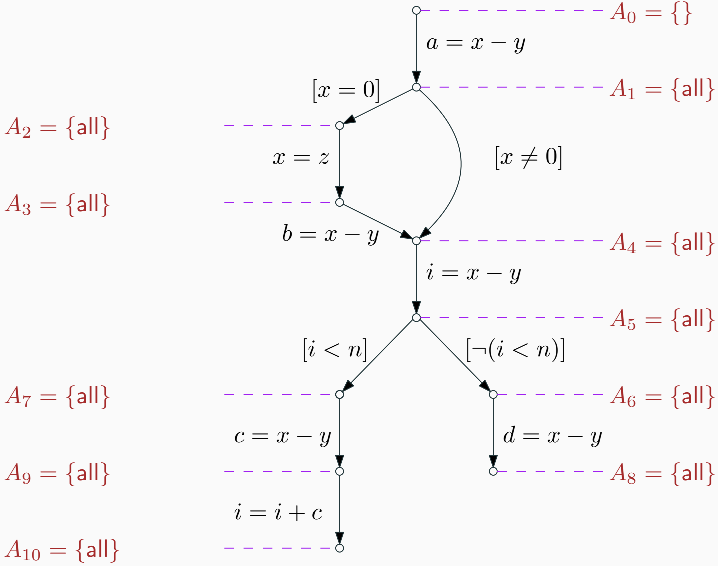
$$\mathit{out}(S) = \big(\mathit{in}(S) \cup \mathit{gen}(S)\big) \ \setminus \ \mathit{kill}(S)$$
$$\mathit{in}(S) = \bigcap_{S_i \in \mathbf{pred}(S)} \mathit{out}(S_i)$$

Solve constraints:

- Start with empty set of available expressions at start and universal set of available expressions everywhere else
- Iteratively apply constraints
- Stop when we reach a fixed point

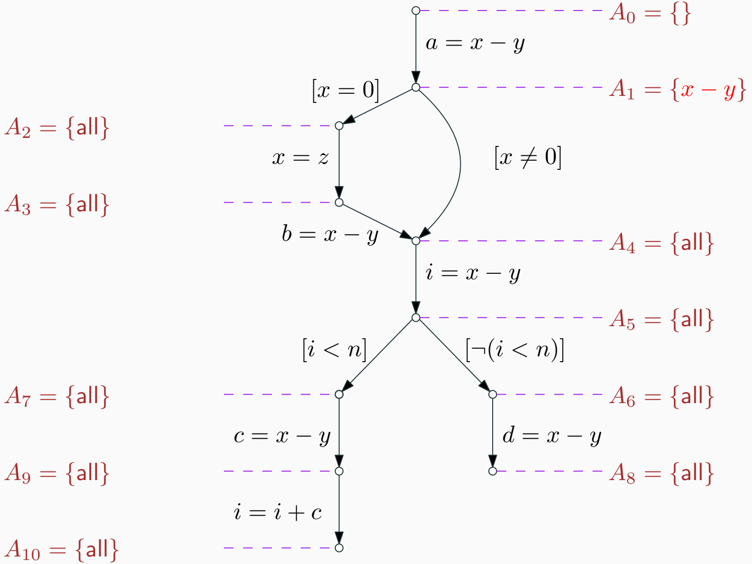
Exercise

Compute the set of available expressions at each point of the program



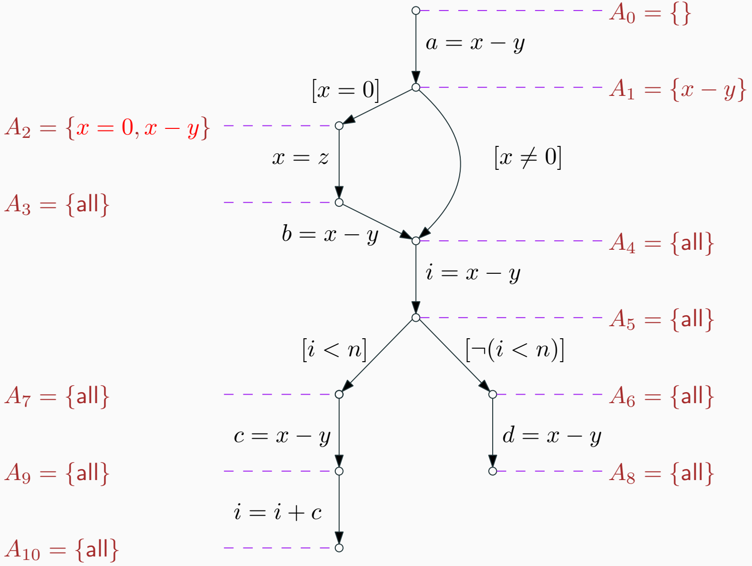
Exercise

Compute the set of available expressions at each point of the program

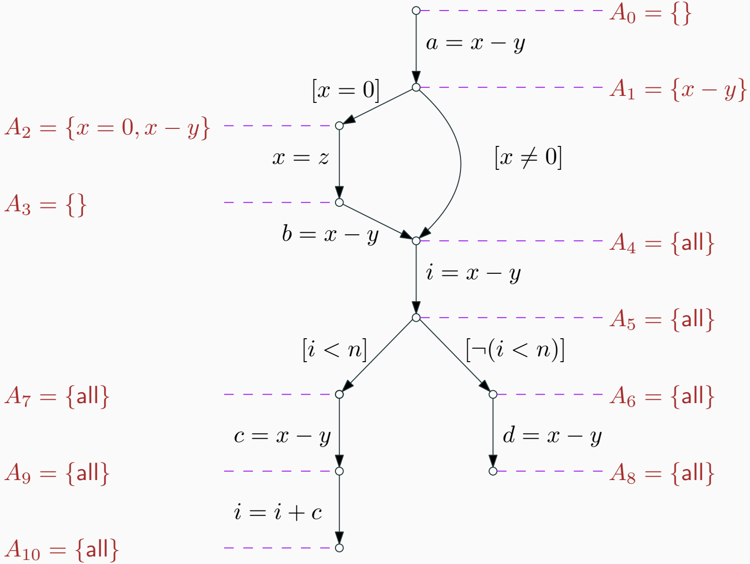


Exercise

Compute the set of available expressions at each point of the program

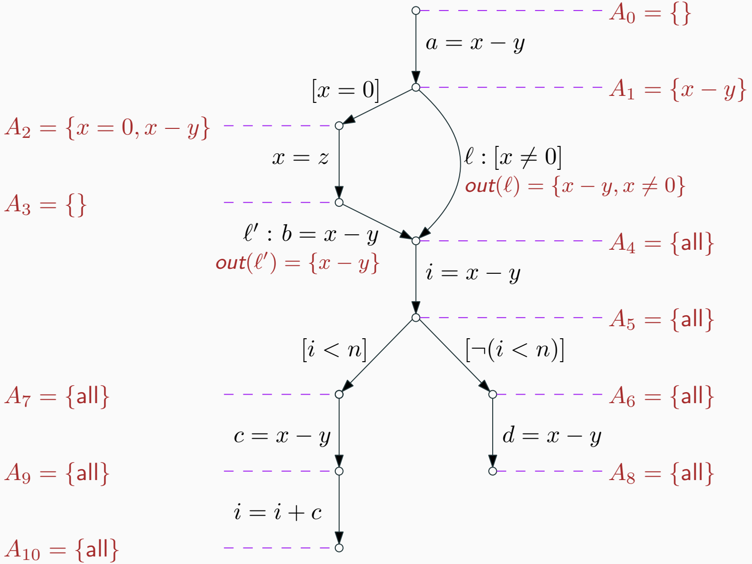


Compute the set of available expressions at each point of the program

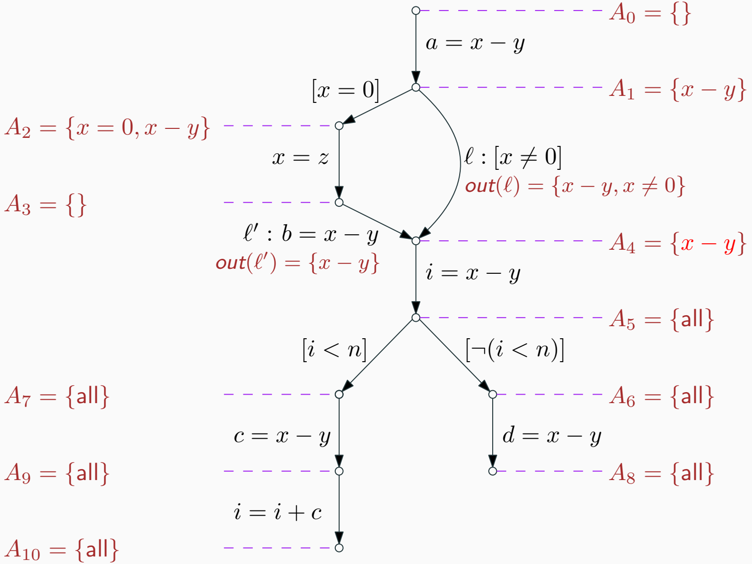


Exercise

Compute the set of available expressions at each point of the program

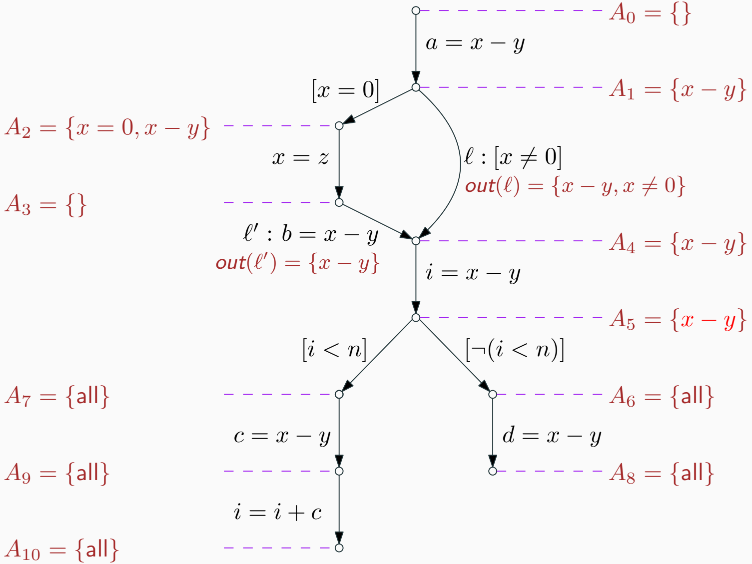


Compute the set of available expressions at each point of the program



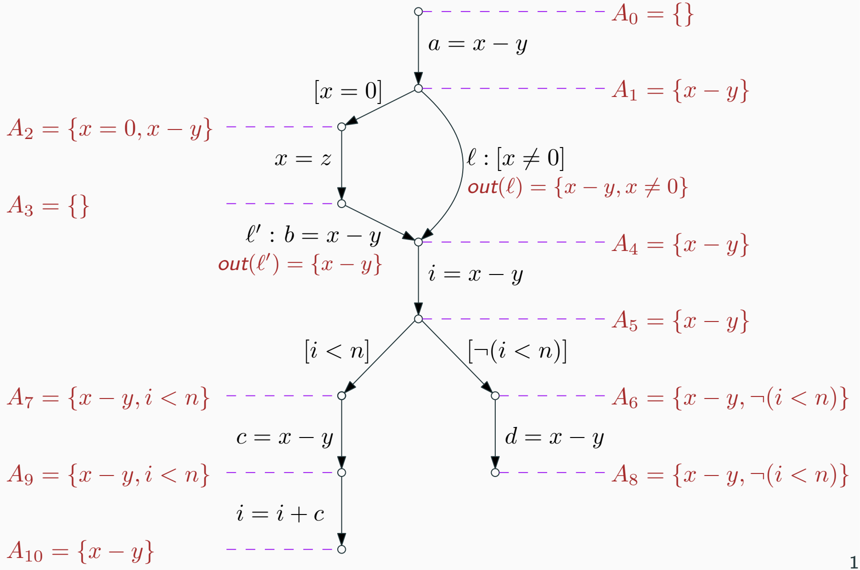
Exercise

Compute the set of available expressions at each point of the program



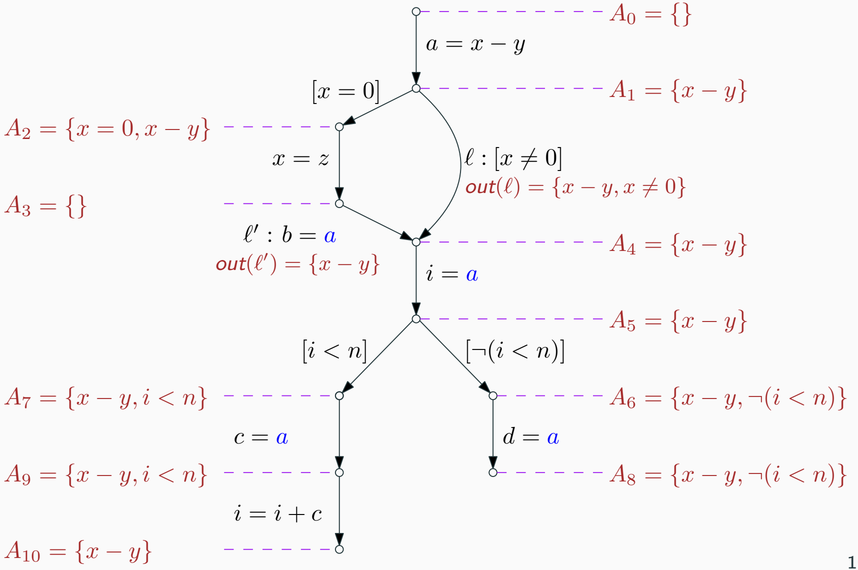
Exercise

Compute the set of available expressions at each point of the program



Exercise

Compute the set of available expressions at each point of the program



Data-flow Equations Comparison

Live Variable Analysis

$$\textit{in-live}(S) = \big(\textit{out-live}(S) \setminus \textit{def}(S)\big) \;\cup\; \textit{use}(S)$$
$$\textit{out-live}(S) = \bigcup_{S_i \in \textit{succ}(S)} \textit{in-live}(S_i)$$

Available Expression Analysis

$$\textit{out-avail}(S) = \big(\textit{in-avail}(S) \cup \textit{gen}(S)\big) \;\setminus\; \textit{kill}(S)$$
$$\textit{in-avail}(S) = \bigcap_{S_i \in \textit{pred}(S)} \textit{out-avail}(S_i)$$