Context Free Languages

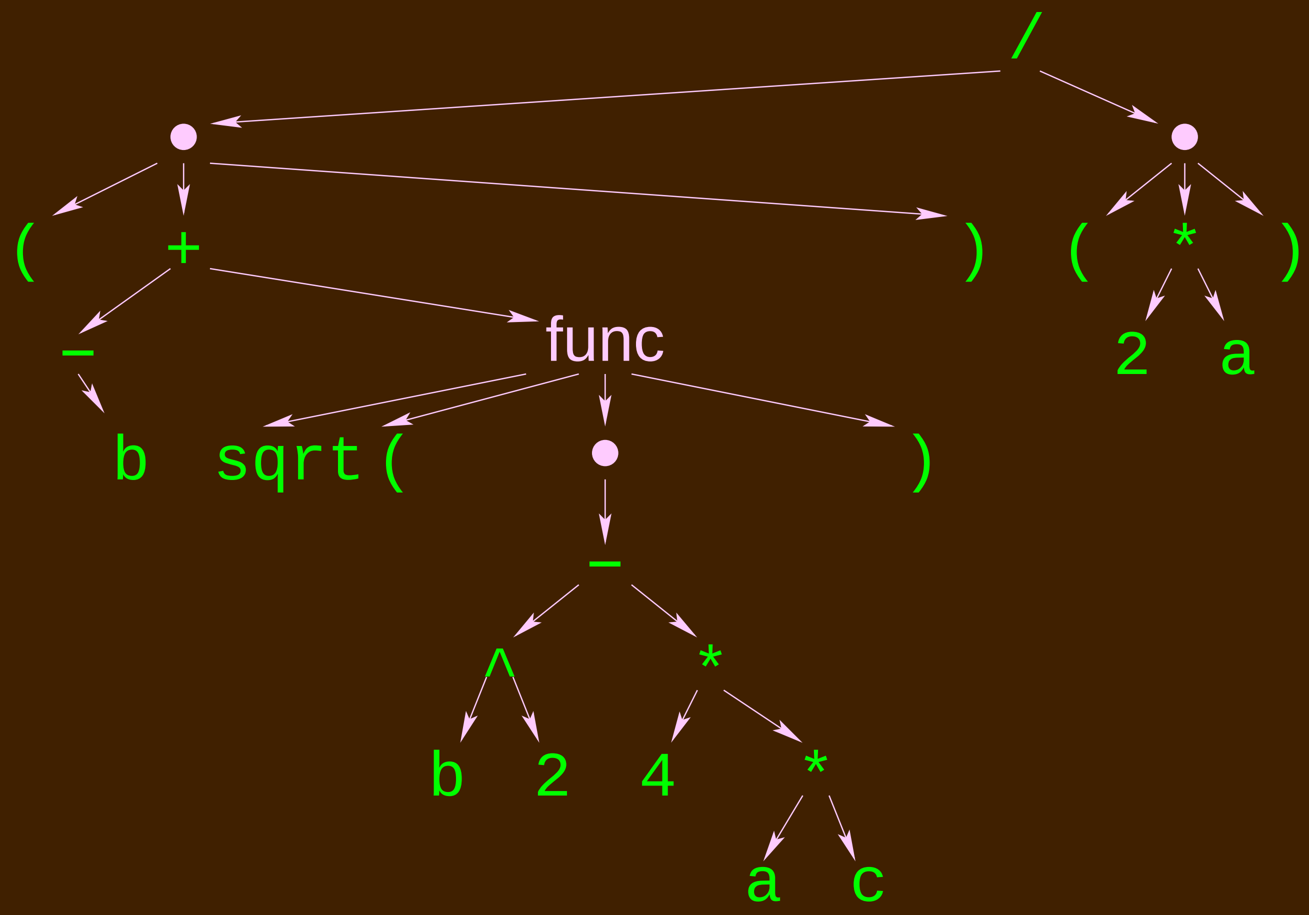
Mark Greenstreet, CpSc 421, Term 1, 2006/07

Lecture Outline

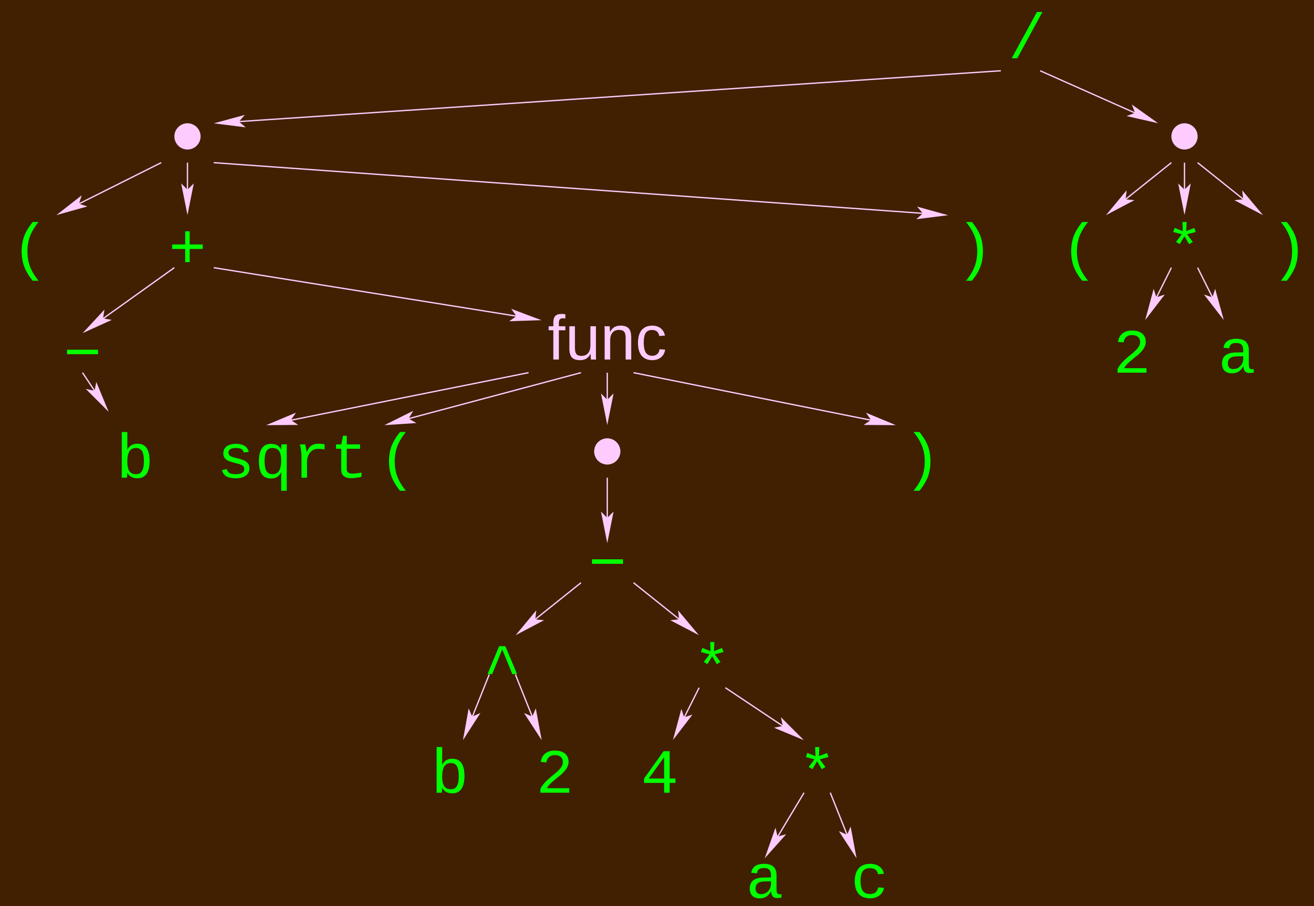
Context Free Languages

- CFG Examples
- Formal Definition
- The Regular Languages are Context Free

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$

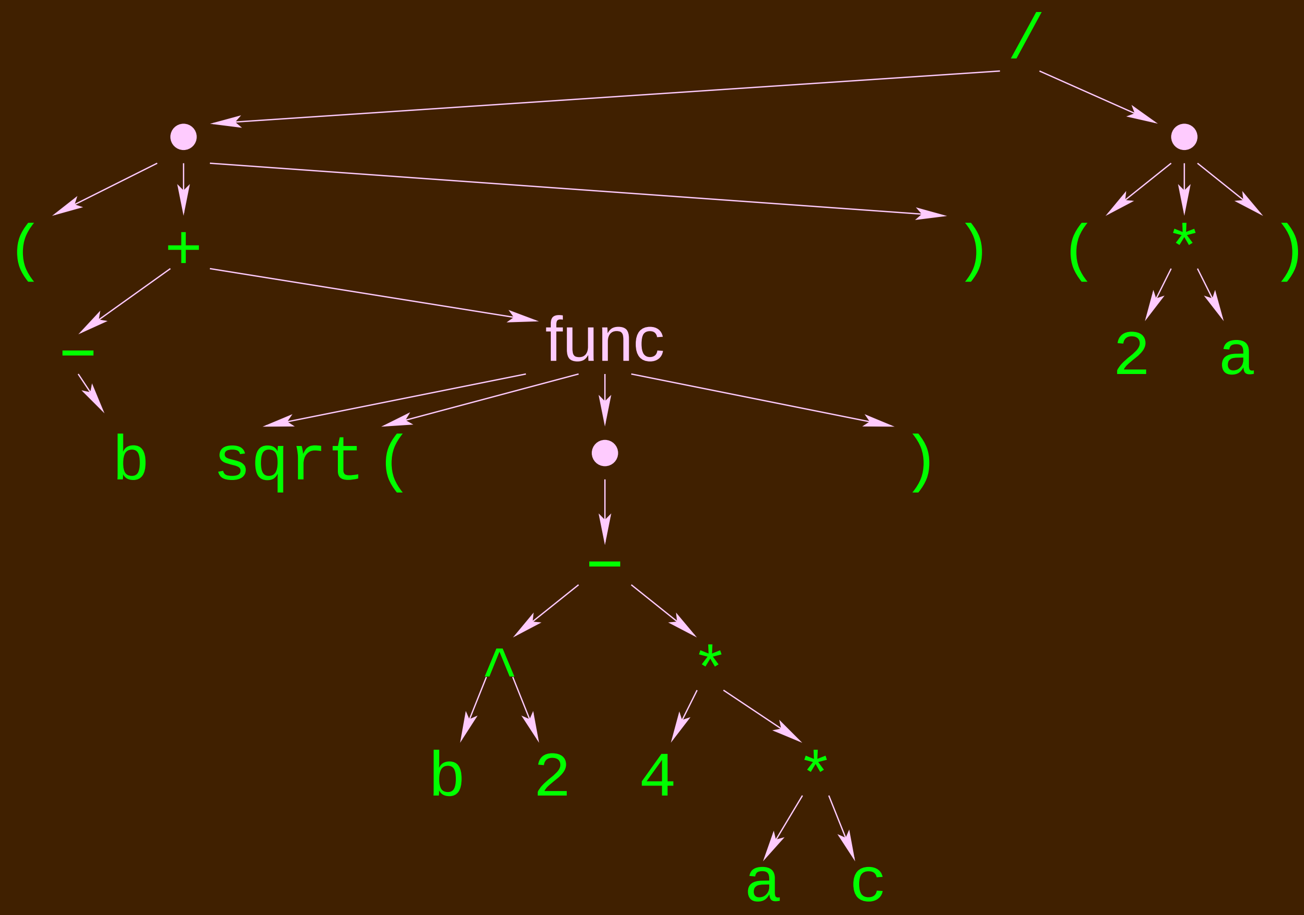


$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



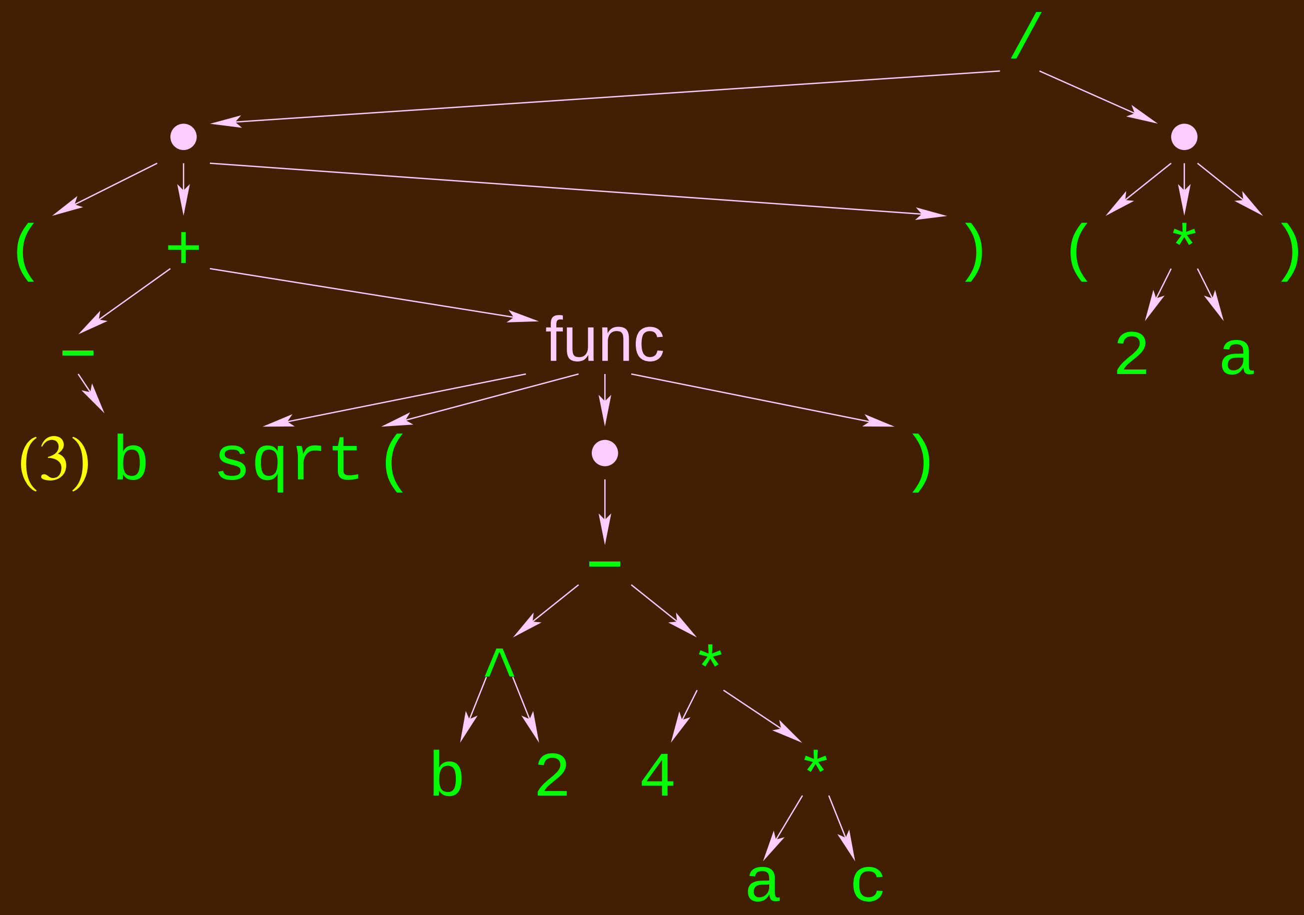
(− b + sqrt (b ^ 2 − 4 * a * c)) / (2 * a)

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



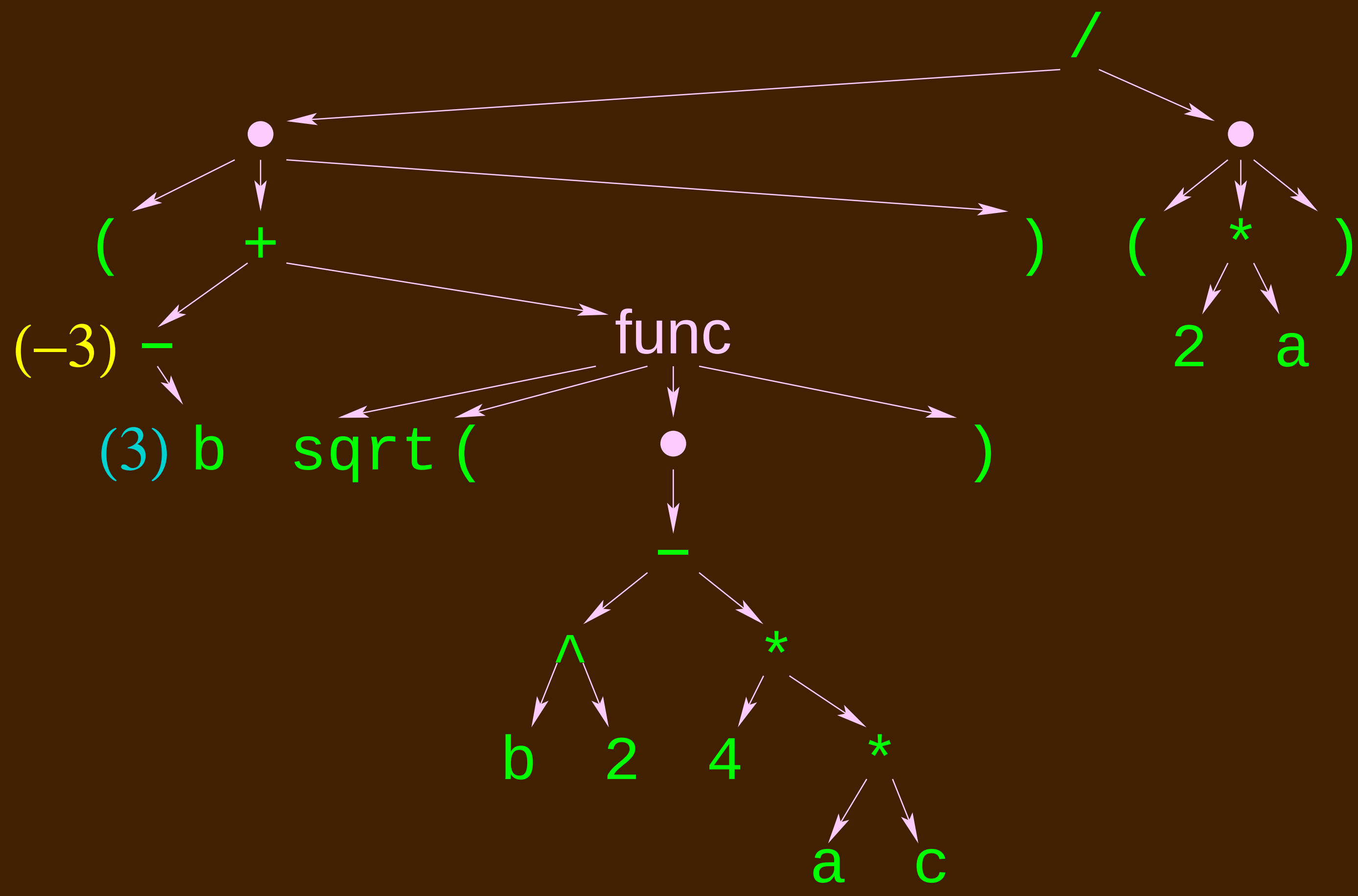
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



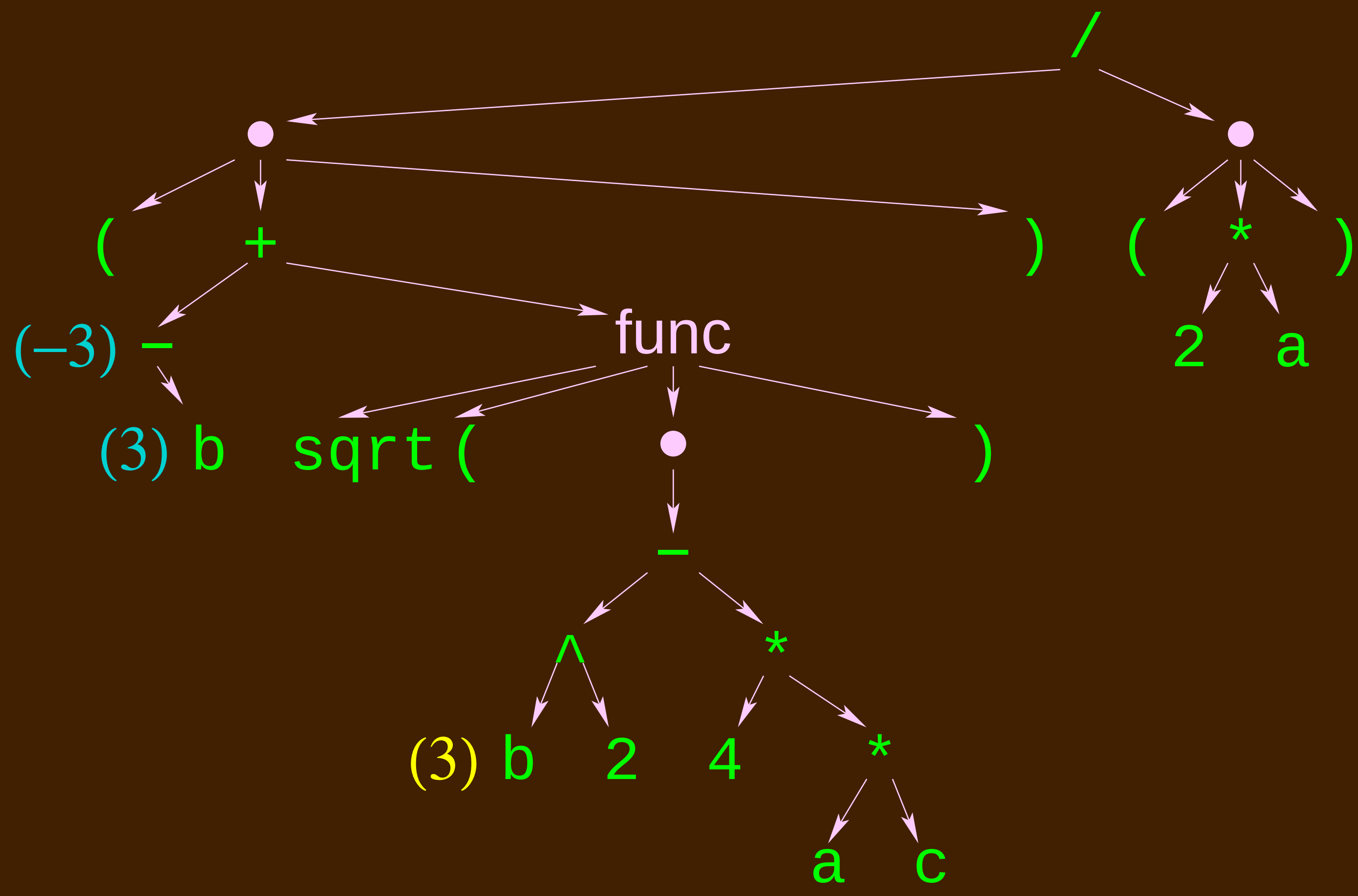
a = 1, b = 3, c = −4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



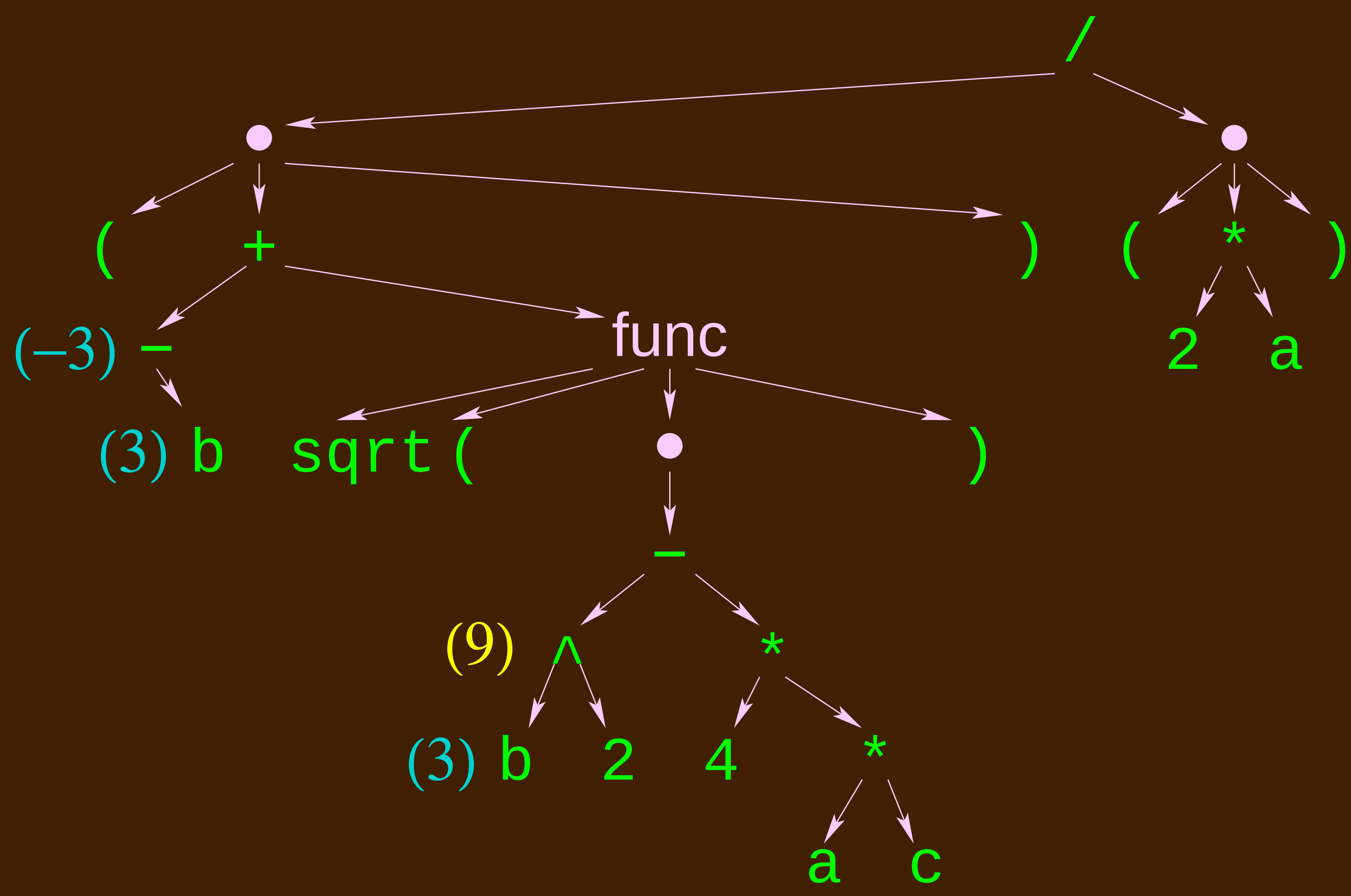
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



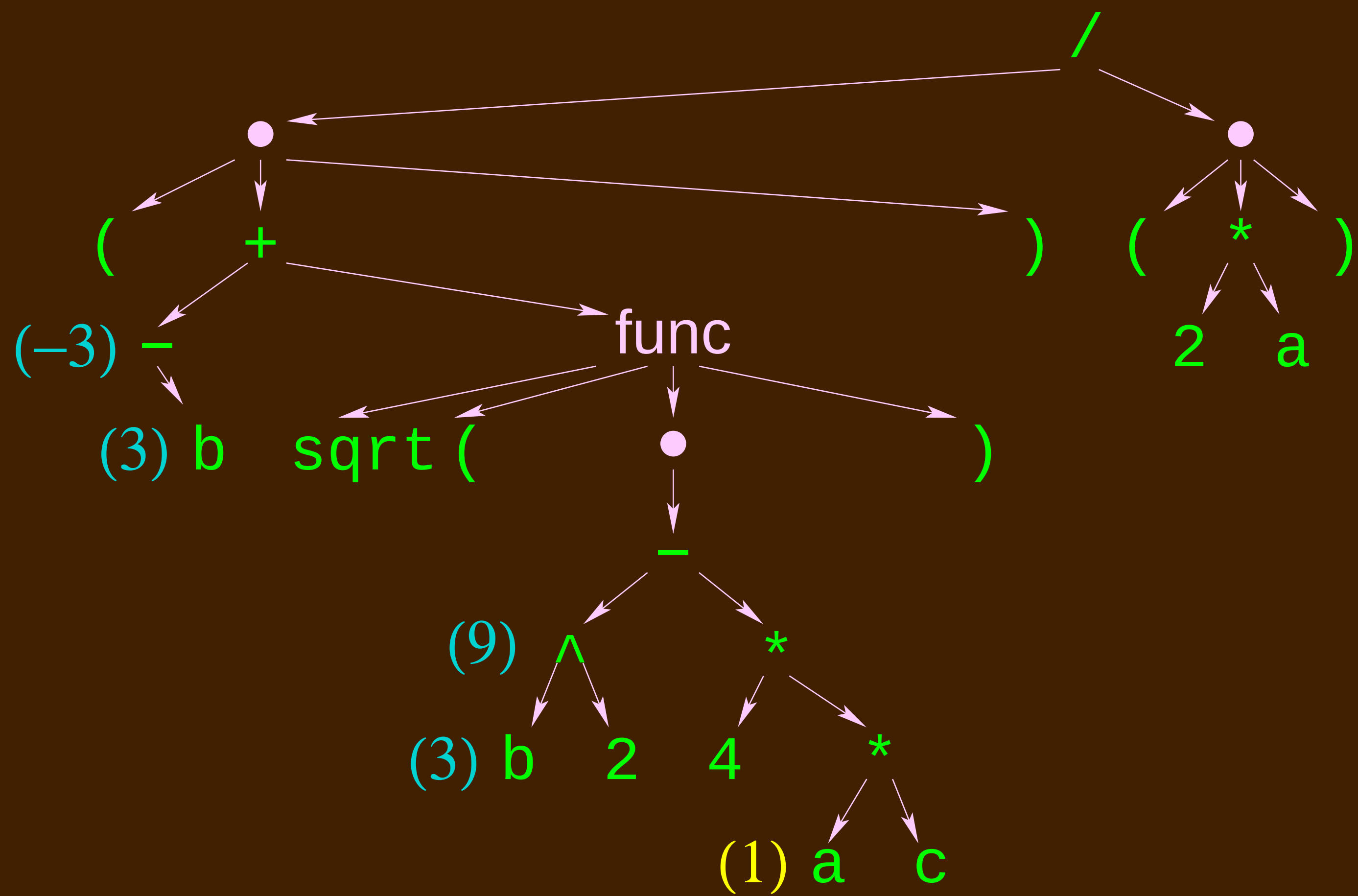
a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



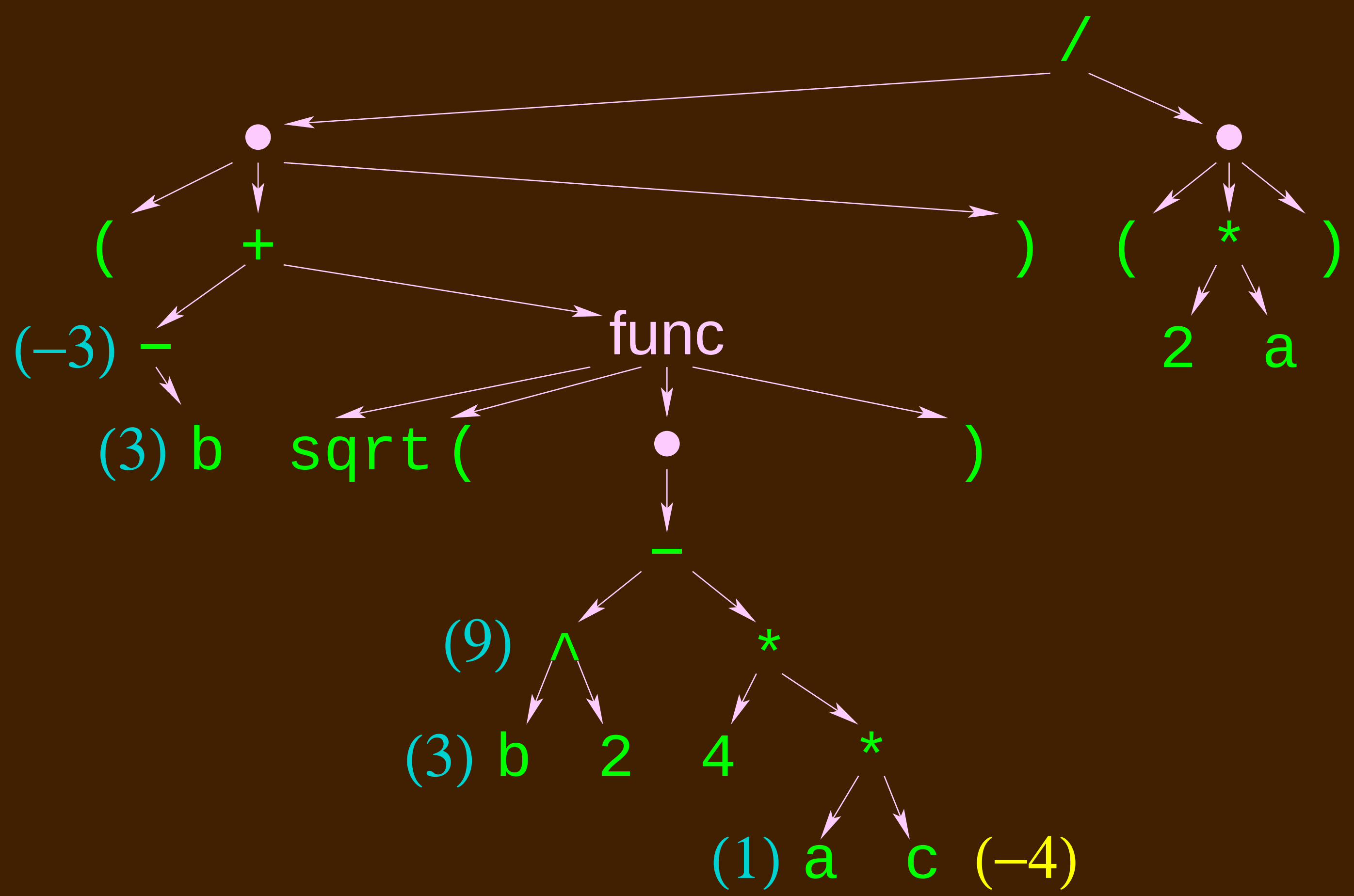
a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



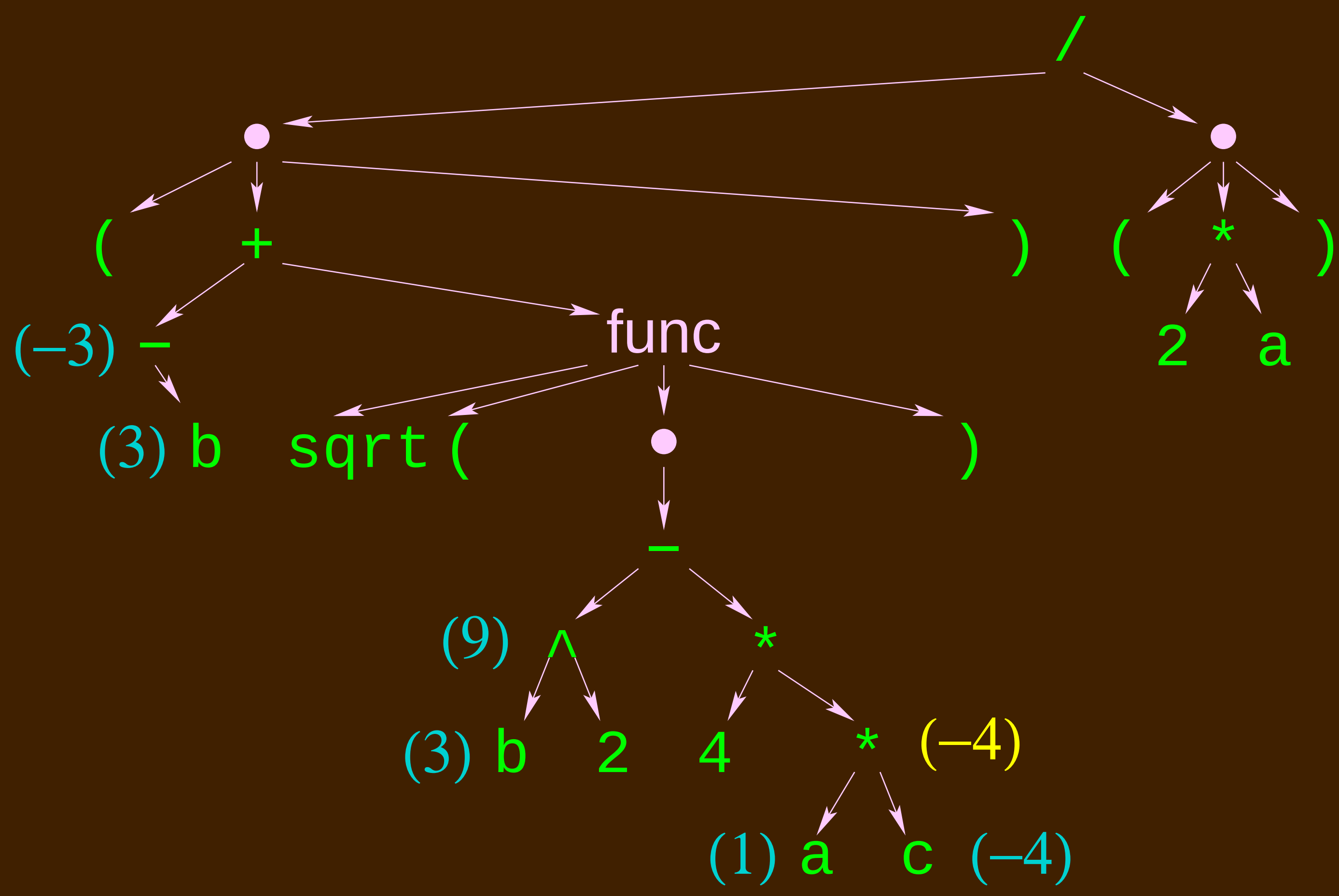
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



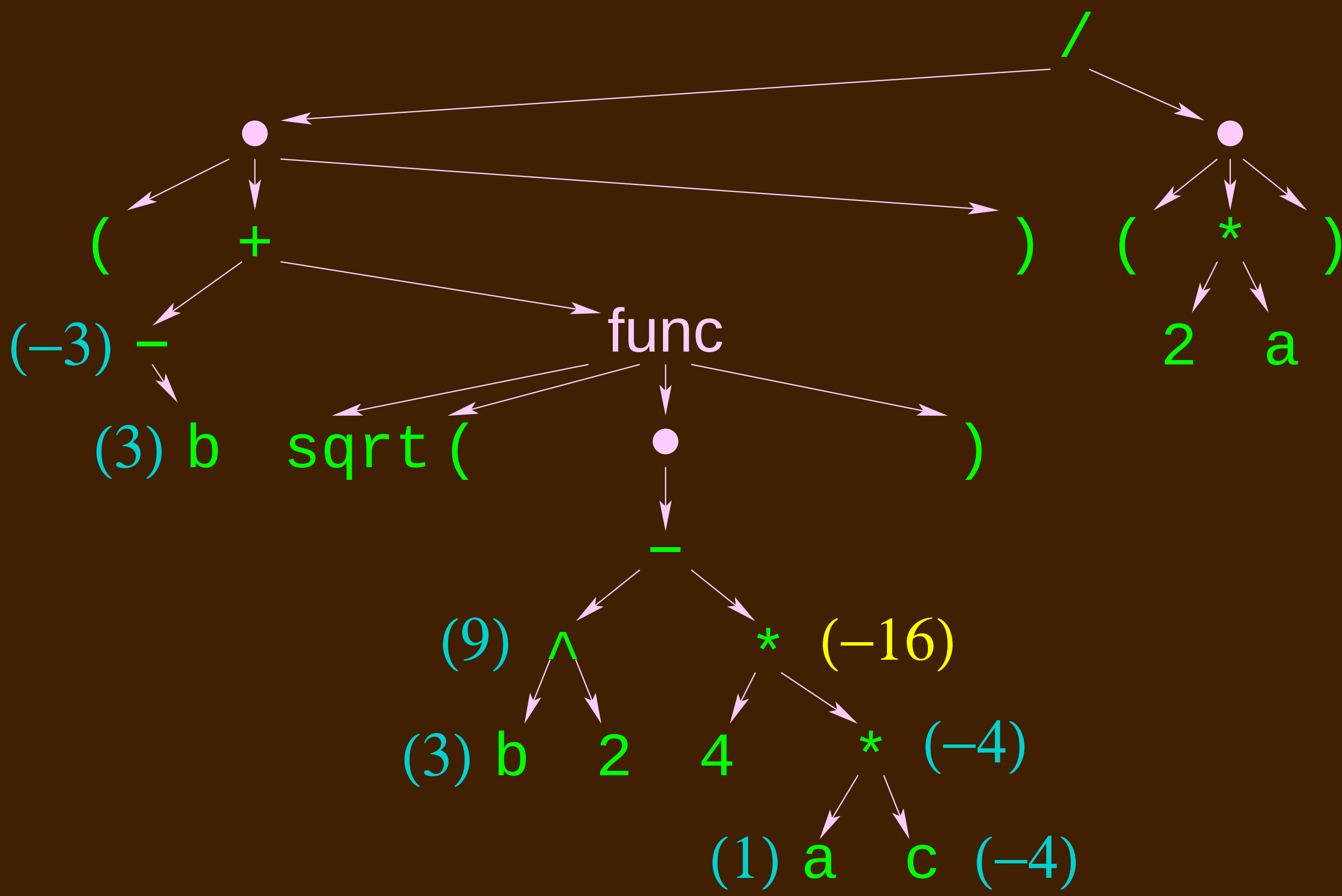
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



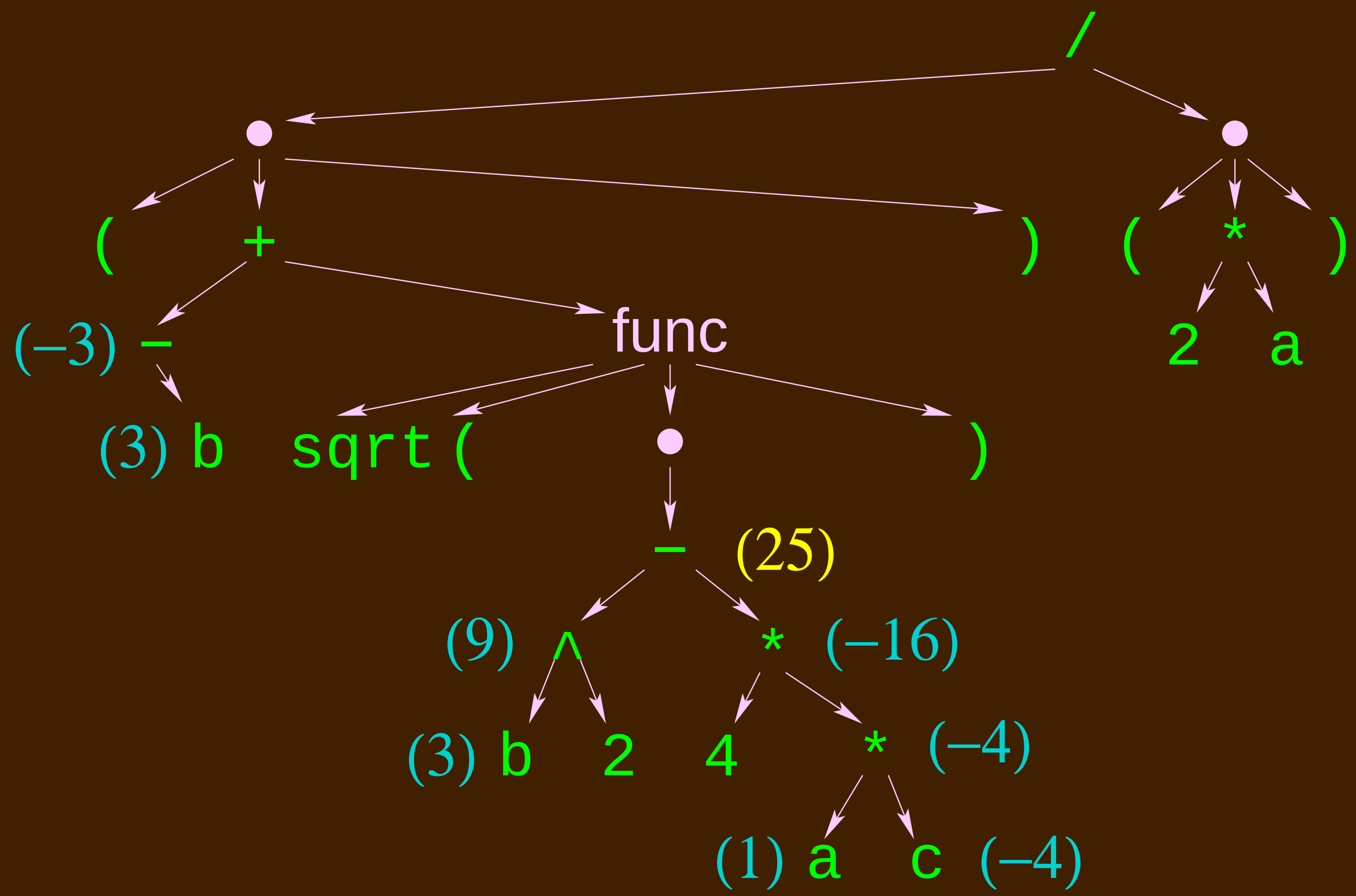
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



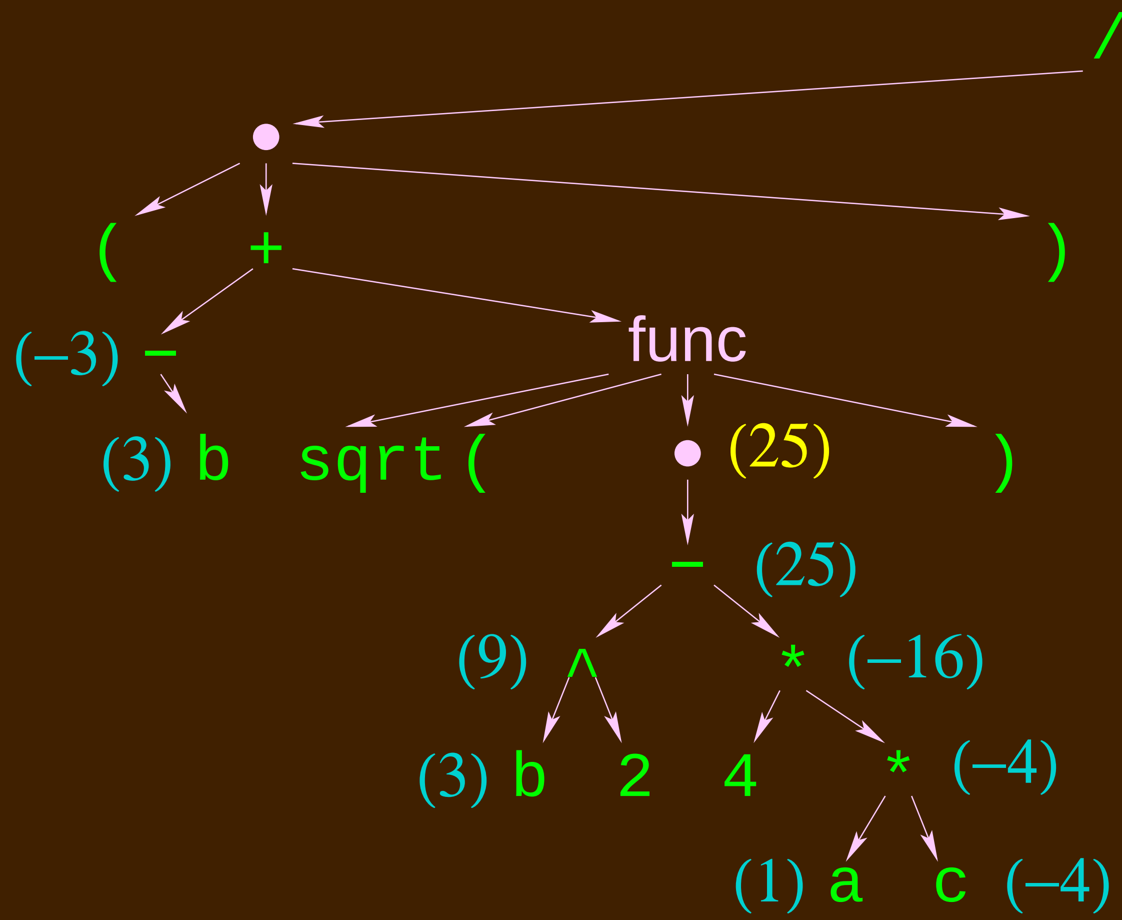
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



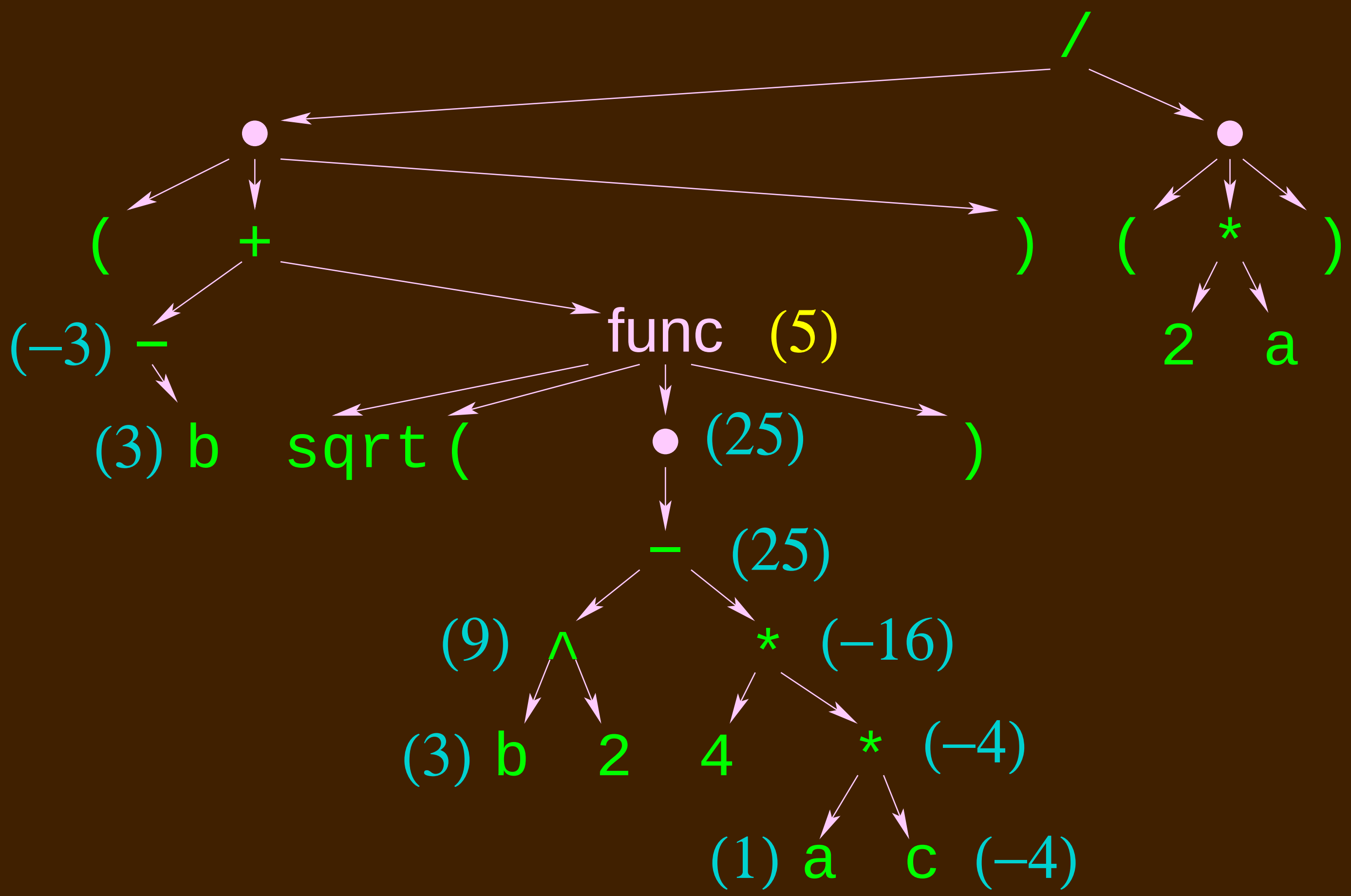
a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



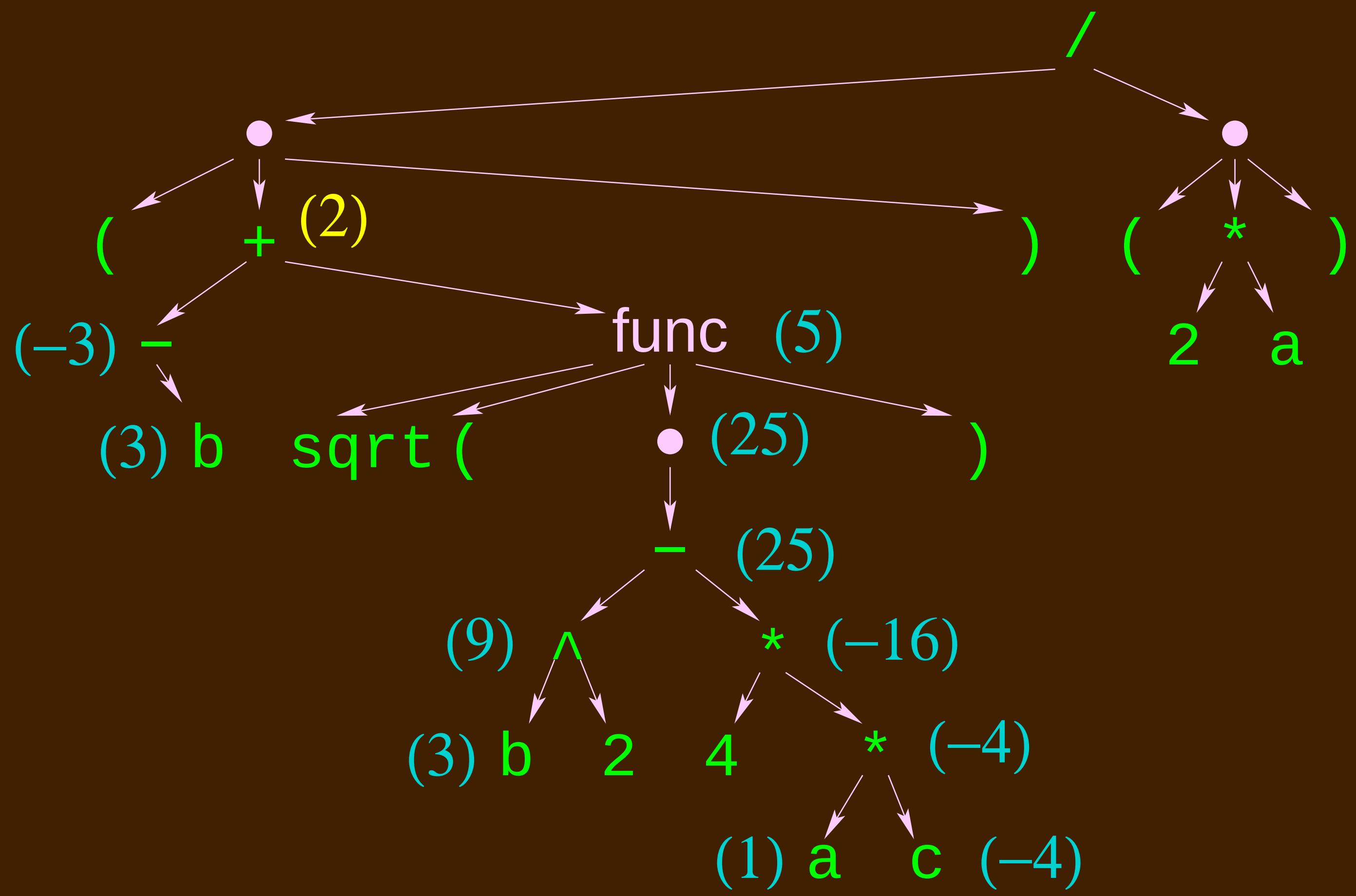
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



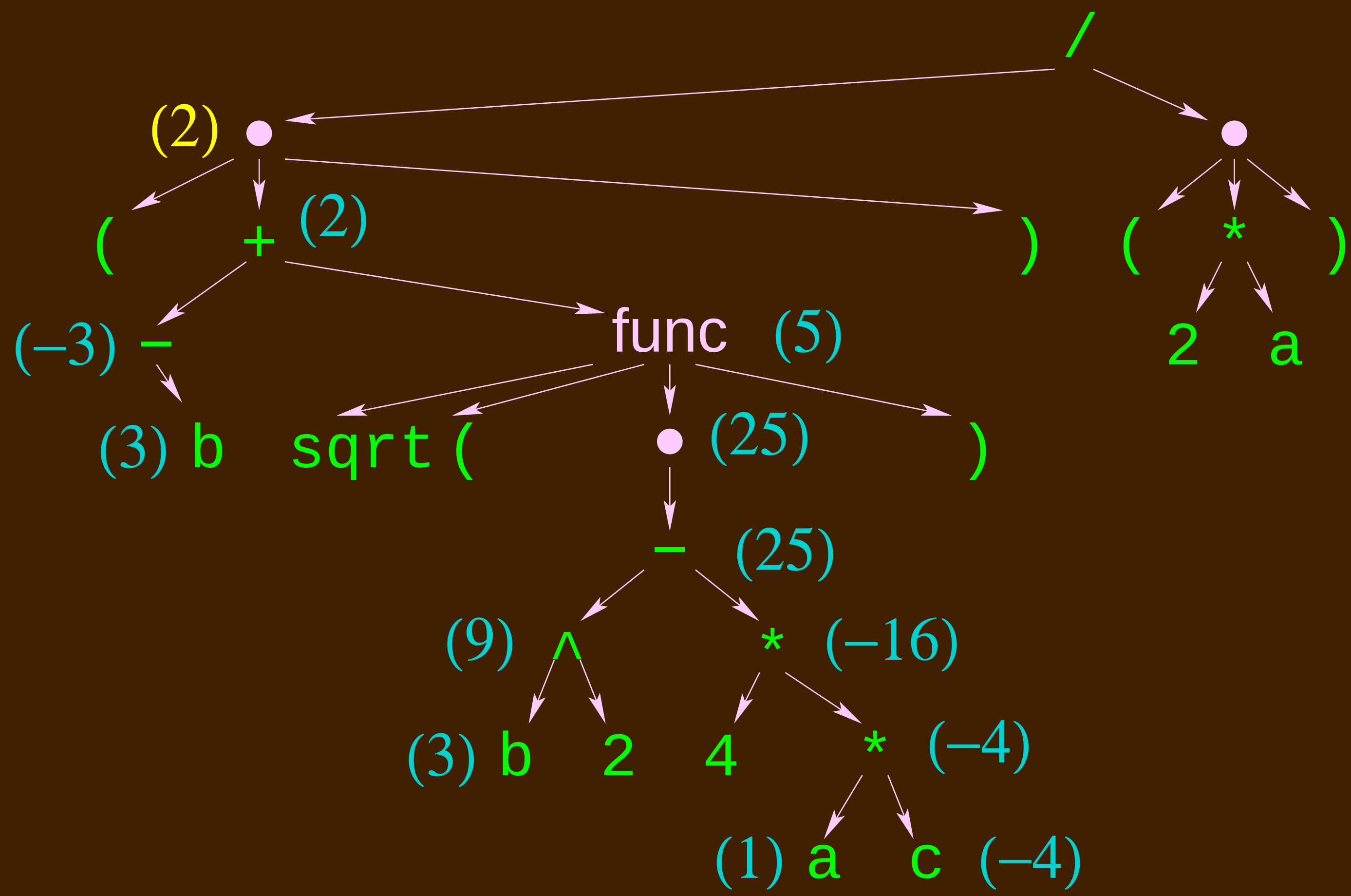
a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



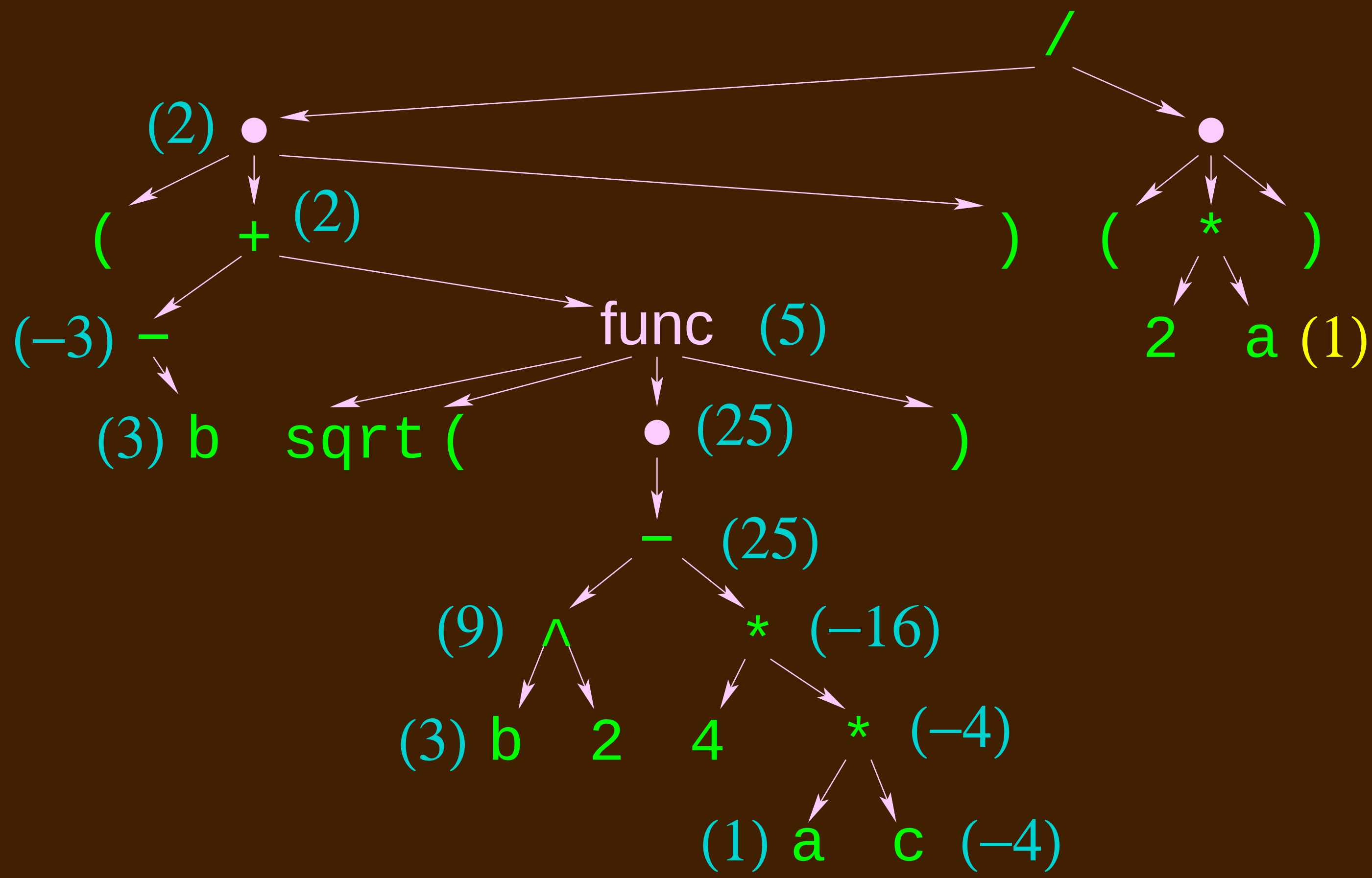
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



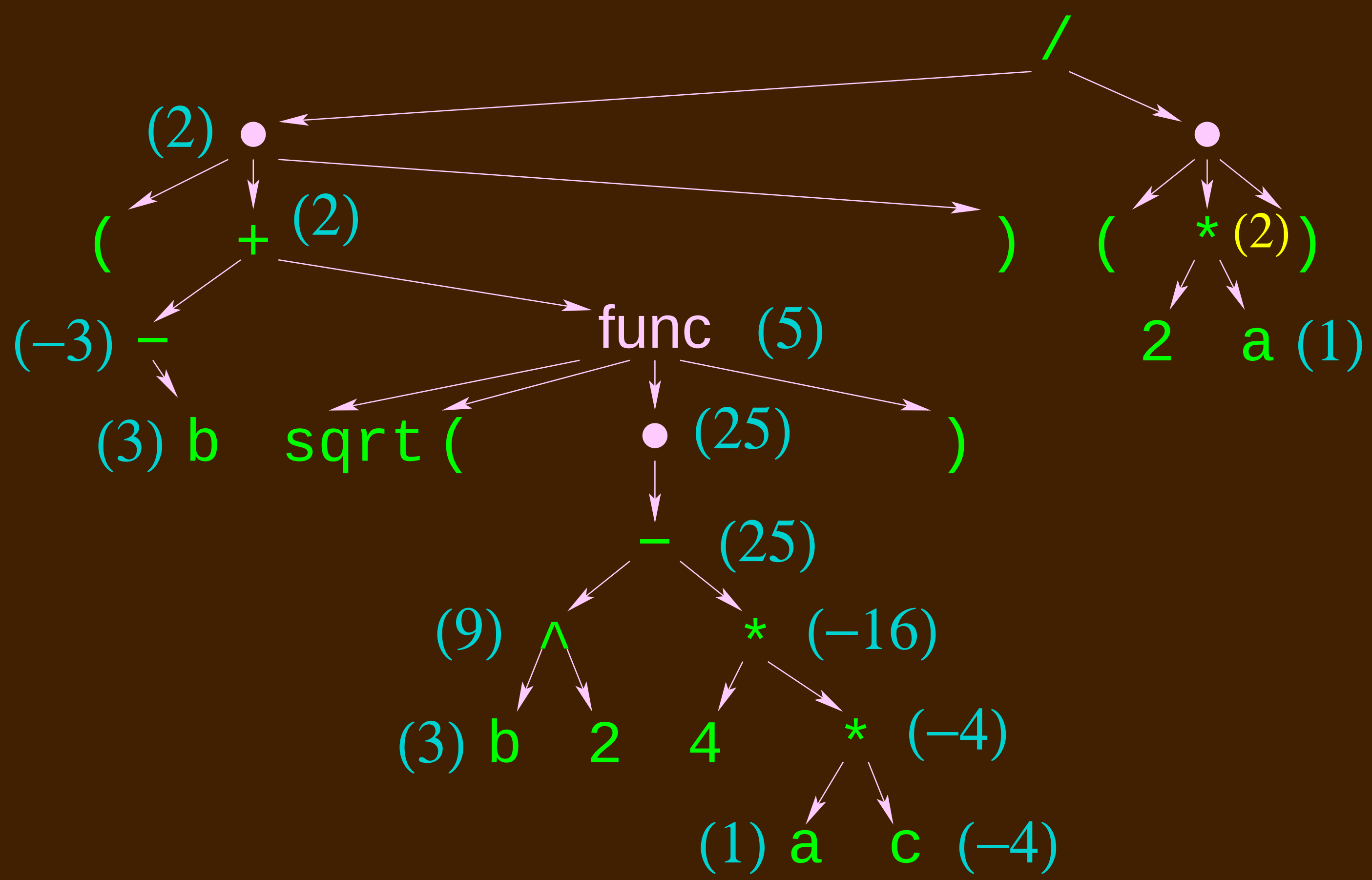
a = 1, b = 3, c = –4.

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



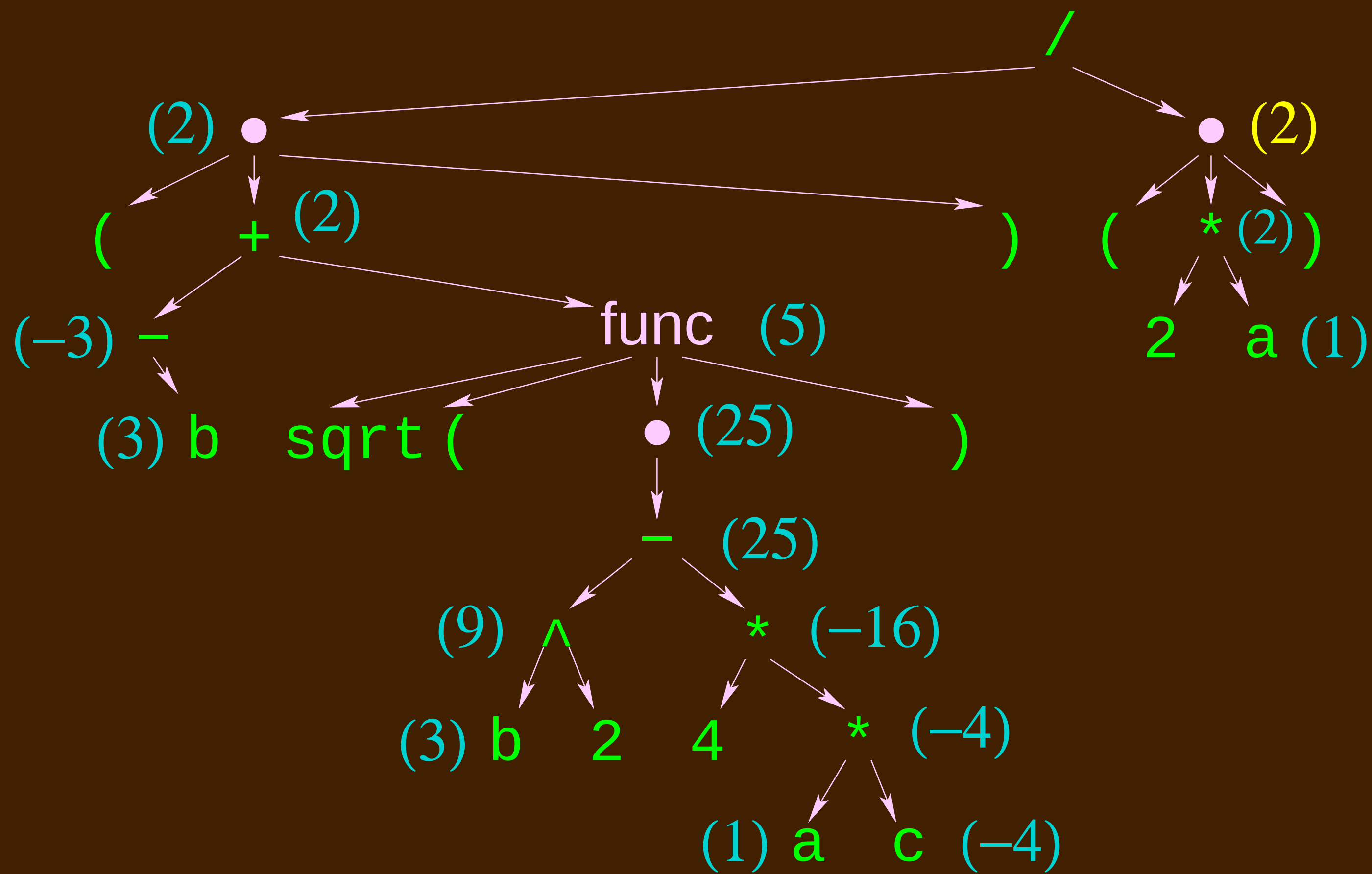
a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



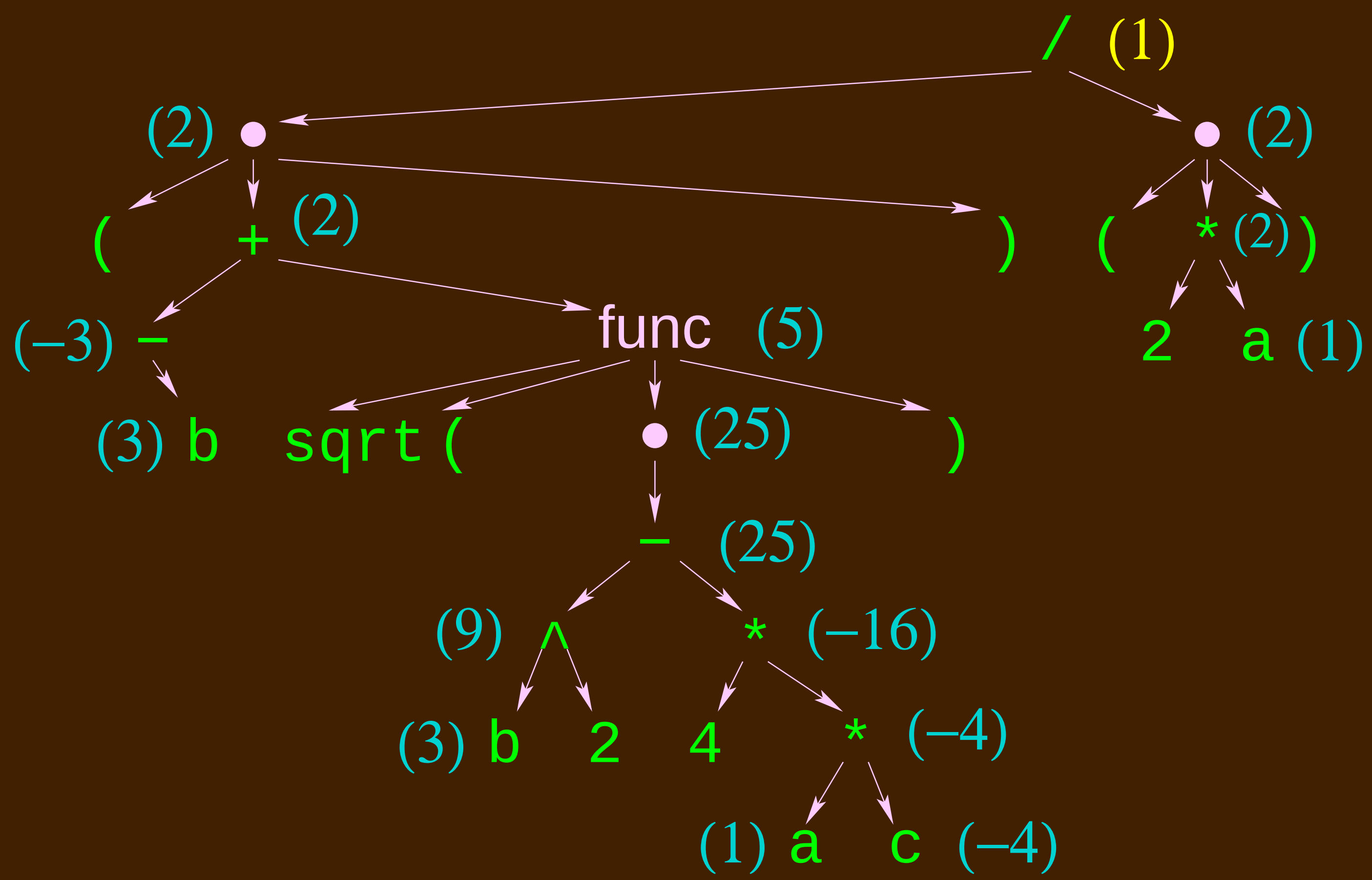
a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



a = 1, b = 3, c = –4.

$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$



a = 1, b = 3, c = −4.

Arithmetic Expressions

$G = (V, \Sigma, R, Expr)$, where

$$
\begin{aligned}
V &= \{Expr, ExprList, NonEmptyExprList\} \\
\Sigma &= \{\texttt{INTEGER}, \texttt{IDENTIFIER}, \texttt{PLUS}, \texttt{MINUS}, \\
&\qquad \texttt{TIMES}, \texttt{DIVIDE}, \texttt{EXP}, \\
&\qquad \texttt{LPAREN}, \texttt{RPAREN}, \texttt{COMMA}\} \\
R &= \text{See next slide}
\end{aligned}
$$

Rules for Arithmetic Expressions

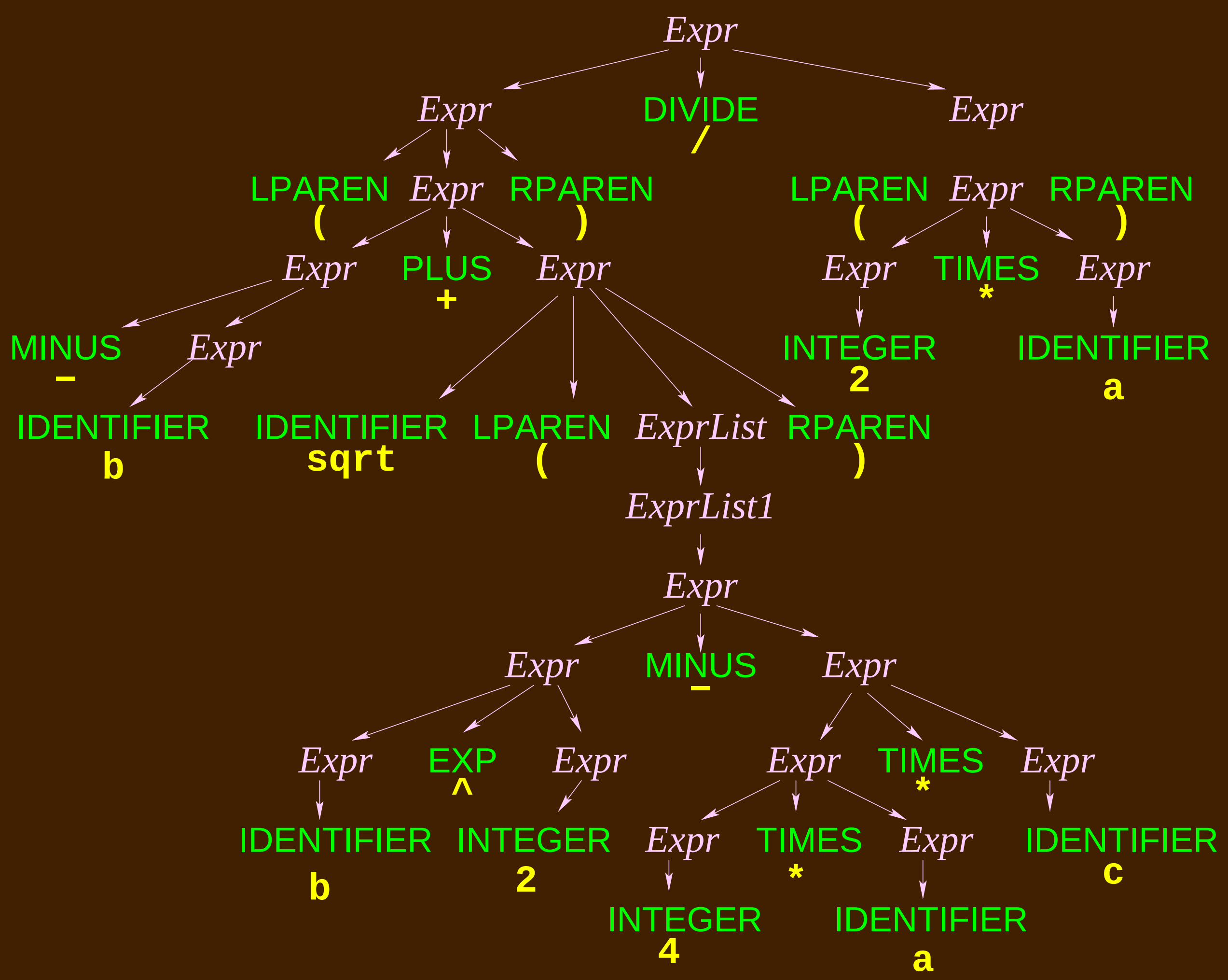
$$
\begin{array}{rcl}
\mathit{Expr} & \rightarrow & \texttt{INTEGER} \quad \mid \quad \texttt{IDENTIFIER} \\
& \mid & \mathit{Expr}\ \texttt{PLUS}\ \mathit{Expr} \quad \mid \quad \mathit{Expr}\ \texttt{MINUS}\ \mathit{Expr} \\
& \mid & \mathit{Expr}\ \texttt{TIMES}\ \mathit{Expr} \quad \mid \quad \mathit{Expr}\ \texttt{DIVIDE}\ \mathit{Expr} \\
& \mid & \mathit{Expr}\ \texttt{EXP}\ \mathit{Expr} \quad \mid \quad \texttt{LPAREN}\ \mathit{Expr}\ \texttt{RPARE} \\
& \mid & \texttt{MINUS}\ \mathit{Expr} \\
& \mid & \texttt{IDENTIFIER}\ \texttt{LPAREN}\ \mathit{ExprList}\ \texttt{RPAREN} \\
\mathit{ExprList} & \rightarrow & \epsilon \quad \mid \quad \mathit{ExprList1} \\
\mathit{ExprList1} & \rightarrow & \mathit{Expr} \\
& \mid & \mathit{ExpList1}\ \texttt{COMMA}\ \mathit{Expr}\ .
\end{array}
$$

Rules for Arithmetic Expressions

$$
\begin{array}{rcll}
\mathit{Expr} & \rightarrow & \mathtt{INTEGER} & \mid\ \mathtt{IDENTIFIER} \\
 & \mid & \mathit{Expr}\ \mathtt{PLUS}\ \mathit{Expr} & \mid\ \mathit{Expr}\ \mathtt{MINUS}\ \mathit{Expr} \\
 & \mid & \mathit{Expr}\ \mathtt{TIMES}\ \mathit{Expr} & \mid\ \mathit{Expr}\ \mathtt{DIVIDE}\ \mathit{Expr} \\
 & \mid & \mathit{Expr}\ \mathtt{EXP}\ \mathit{Expr} & \mid\ \mathtt{LPAREN}\ \mathit{Expr}\ \mathtt{RPARE} \\
 & \mid & \mathtt{MINUS}\ \mathit{Expr} & \\
 & \mid & \mathtt{IDENTIFIER}\ \mathtt{LPAREN}\ \mathit{ExprList}\ \mathtt{RPAREN} & \\
\mathit{ExprList} & \rightarrow & \epsilon & \mid\ \mathit{ExprList1} \\
\mathit{ExprList1} & \rightarrow & \mathit{Expr} & \\
 & \mid & \mathit{ExpList1}\ \mathtt{COMMA}\ \mathit{Expr}\ . &
\end{array}
$$

Generating an Expression

$$(-b + sqrt(b^2 - 4 * a * c))/(2 * a)$$



Context-Free Grammars

- A context-free grammar (CFG) is a 4-tuple, (V, Σ, R, S) where
  - V is a finite set of variables –
    V is sometimes called the “stack alphabet”;
  - Σ is a finite set of symbols –
    Σ is the input alphabet;
  - R is a finite set of rules –
    $R \subseteq V \times (V \cup \Sigma^*)$; in English, a rule maps a variable to a string of variables and/or symbols;
  - S is the start variable.

Derivations (1/2)

- If $\alpha \in V$ and $\alpha \rightarrow w$ for some $w \in (V \cup \Sigma)^*$, then we say that α produces w.
  - Example: $S \rightarrow \texttt{0}S\texttt{1}$.
  - We say that S produces $\texttt{0}S\texttt{1}$.
- If $x = y\alpha z$ with $y, z \in (V \cup \Sigma)^*$ and $\alpha \in V$, and $alpha \rightarrow w$ for some $w \in (V \cup \Sigma)^*$, then we say that x yields w.
  - Example: $S \rightarrow \texttt{0}S\texttt{1}$.
  - We say that $\texttt{00}S\texttt{10}S\texttt{1}$ yields $\texttt{000}S\texttt{110}S\texttt{1}$.
  - Likewise $\texttt{00}S\texttt{10}S\texttt{1}$ yields $\texttt{00}S\texttt{100}S\texttt{11}$. $\texttt{0000}S\texttt{110}S\texttt{1}$.
- We write $x \Rightarrow w$ to denote that x yields w.

Derivations (2/2)

- $x, w \in (V \cup \Sigma)^*$ and w can be obtained from x by applying zero or more rules, then we say that x **derives** w.
- We write $x \stackrel{*}{\Rightarrow} w$ to denote that x derives w.
  - Example: $S \;\to\; \mathtt{0}S\mathtt{1} \mid SS \mid \epsilon$.
  - $S \Rightarrow SS \Rightarrow 0S1S \Rightarrow 0S10S1 \Rightarrow 010S1 \Rightarrow 0100S11 \Rightarrow 010011$.
  - Thus, $x \stackrel{*}{\Rightarrow} w$.
- Formally, $x \stackrel{*}{\Rightarrow} w$ iff
  - $w = x$, or
  - $\exists y.\ (x \stackrel{*}{\Rightarrow} y) \wedge (y \Rightarrow w)$.
  - Note that this is an inductive definition. Therefore, we can use induction to prove properties of derivations.
    - Example: same grammar as above.
    - $S \stackrel{*}{\Rightarrow} S$. Thus, $S \stackrel{*}{\Rightarrow} SS$. Thus, $S \stackrel{*}{\Rightarrow} 0S1 \ldots$
      Thus, $S \stackrel{*}{\Rightarrow} 010011$.

Context-Free Languages

Let $G = (V, \Sigma, R, S)$ be a context-free grammar.

- $L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$.
- A language is context free iff it is the language of some CFG.
- Examples of context-free languages:
  - $0^n 1^n$.
  - The set of strings in $\{0, 1\}^*$ with an equal number of 0's and 1's.
  - Arithmetic expressions as defined on slides 4 and 5.
  - For most programming languages, the set of syntactically correct programs is a context free language.

Regular Languages are Context-Free

Let A be a regular language. Let α be a regular expression that describes A.

case $\alpha = \mathtt{c}$, $\mathtt{c} \in \Sigma$: $G = (\{S\}, \Sigma, \{S \to \mathtt{c}\}, S)$.

case $\alpha = \epsilon$: $G = (\{S\}, \Sigma, \{S \to \epsilon\}, S)$.

case $\alpha = \emptyset$: $G = (\{S\}, \Sigma, \emptyset, S)$.

case $\alpha = \alpha_1 \cup \alpha_2$:

- Let $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ be CFGs such that $L(G_1) = L(\alpha_1)$ and $L(G_2) = L(\alpha_2)$.
- I'll assume $V_1 \cap V_2 = \emptyset$. This can be achieved by renaming variable if neccessary. Likewise, I'll assume that there is a variable S, with $S \notin V_1$ and $S \notin V_2$.
- Let $G = (V, \Sigma, R, S)$ with $V = V_1 \cup V_2 \cup \{S\}$ and $R = R_1 \cup R_2 \cup \{S \ \to \ S_1 \mid S_2\}$.
- $L(G) = L(\alpha_1 \cup \alpha_2)$. The proof is based on the observation that the first step of a derivation in G starts by choosing S_1 or S_2 to obtain eventually a string from $L(\alpha_1)$ or $L(\alpha_2)$ respectively.

The Rest of the Proof

case $\alpha = \alpha_1 \circ \alpha_2$:

- Define G_1 and G_2 as for the previous case.
- Let $G = (V, \Sigma, R, S)$ with $V = V_1 \cup V_2 \cup \{S\}$ and $R = R_1 \cup R_2 \cup \{S \ \rightarrow \ S_1 \ S_2\}$.
- $L(G) = L(\alpha_1 \circ \alpha_2)$. The proof is based on the observation that the first step of a derivation in G produes $S_1 \ S_2$. Thus the complete derivation will produce a string from $L(G_1)$ followed by a string from $L(G_2)$. Conversely, any string in $L(\alpha_1 \circ \alpha_2)$ is in $L(G)$.

case $\alpha = \alpha_1^*$:

- Define G_1 as for the previous cases.
- Let $G = (V, \Sigma, R, S)$ with $V = V_1 \cup \{S\}$ and $R = R_1 \cup \{S \ \rightarrow \ \epsilon \mid S \ S_1\}$.
- $L(G) = L(\alpha_1^*)$. We can show $L(G) \subseteq L(\alpha_1^*)$ by induction on the derivation of a string $w \in L(G)$. Likewise, we show $L(G) \subseteq L(\alpha_1^*)$ by induction on the number of strings in $L(\alpha_1)$ that are concatenated together to produce a string $w \in L(\alpha_1^*)$.

Regular vs. Context-Free

- Every regular language is a context free language.
  Proof just given.
- There are languages that are context-free but not regular.
  Example: 0^n1^n.
- Conclusion: $RL \subset CFL$.

Ambiguity

$2 + 3 * 4$

$$\begin{array}{rcl} Expr & \rightarrow & Expr\ \texttt{PLUS}\ Expr \ \mid\ Expr\ \texttt{TIMES}\ Expr \\ & \mid & \texttt{INTEGER} \end{array}$$

Arithmetic Terminals

Regular Expressions:

$$\mathtt{INTEGER} \equiv \mathtt{DIGITDIGIT}^*$$

$$\mathtt{DIGIT} \equiv \mathtt{0} \cup \mathtt{1} \cup \mathtt{2} \cup \mathtt{3} \cup \mathtt{4} \cup \mathtt{5} \cup \mathtt{6} \cup \mathtt{7} \cup \mathtt{8} \cup \mathtt{9}$$

$$\mathtt{IDENTIFIER} \equiv \mathtt{ISTARTITAIL}^*$$

$$\mathtt{ISTART} \equiv \mathtt{A} \cup \mathtt{B} \cup \ldots \cup \mathtt{Z} \cup \mathtt{a} \cup \mathtt{b} \cup \ldots \cup \mathtt{z}$$

$$\mathtt{ITAIL} \equiv \mathtt{ISTART} \cup \mathtt{DIGIT}$$

PLUS	≡	+	MINUS	≡	-
TIMES	≡	*	DIVIDE	≡	/
EXP	≡	∧	COMMA	≡	,
LPAREN	≡	(	RPAREN	≡	)