О платформенно-независимой спецификации и верификации стандартных математических функций

Н.В. Шилов (Иннополис Университет),
И.С. Ануреев, Е.В. Бодин, А.Д. Кондратьев,
А.В. Промский (ИСИ СО РАН, Новосибирск)

Список литературы вместо предисловия

- Кулямин В.В. *Стандартизация и тестирование реализаций математических функций, работающих с числами с плавающей точкой*. Программирование, 2007, том 33, вып.3, с.1-29.

Список литературы вместо предисловия

- Harrison J. *Formal Verification of Square Root Algorithms*. Formal Methods in System Design. 2003. Vol. 22(2), pp.143-153.
- Abstract: *We discuss the formal verification of some low-level mathematical software for the Intel® Itanium® architecture.* A number of important algorithms have been proven correct using the HOL Light theorem prover. After briefly surveying some of our formal verification work, we discuss in more detail the verification of a square root algorithm, which helps to illustrate why some features of HOL Light, in particular programmability, make it especially suitable for these applications.

Список литературы вместо предисловия

- Grohoski G. *Verifying Oracle's SPARC Processors with ACL2*. Slides of the Invited talk for 14th International Workshop on the ACL2 Theorem Prover and Its Applications, 2017. (Available at http://www.cs.utexas.edu/users/moore/acl2/workshop-2017/slides-accepted/grohoski-ACL2_talk.pdf.)

О проекте

- Цель проекта «*Платформенно-независимый подход к формальной спецификации и верификации стандартных математических функций*» (РФФИ № 17-01-00789) – разработка *платформенно-независимого инкрементального комбинированного* подхода к спецификации и верификации (а в дальнейшем – реализации и сертификации) стандартных математических функций (таких как $sqrt, cos, sin$ и так далее).

Платформенно-независимый подход

- Простая аксиоматизацию машинной арифметики в терминах вещественной арифметики (то есть арифметики поля $\boldsymbol{R}$ вещественных чисел),
- но не фиксируя ни основание системы счисления, ни формата машинного слова, ни другие машинно-зависимые детали представления «вещественных» чисел.

Инкрементальный подход

- Спецификация и верификация начинается с рассмотрения наиболее «простого» случая – элементарной спецификации и верификации простого алгоритма, работающего с вещественными числами,
- а заканчивается – модификацией элементарной спецификации и алгоритма для машинной арифметике и верификацией этого алгоритма, работающего в машинной арифметике.

Комбинированный подход

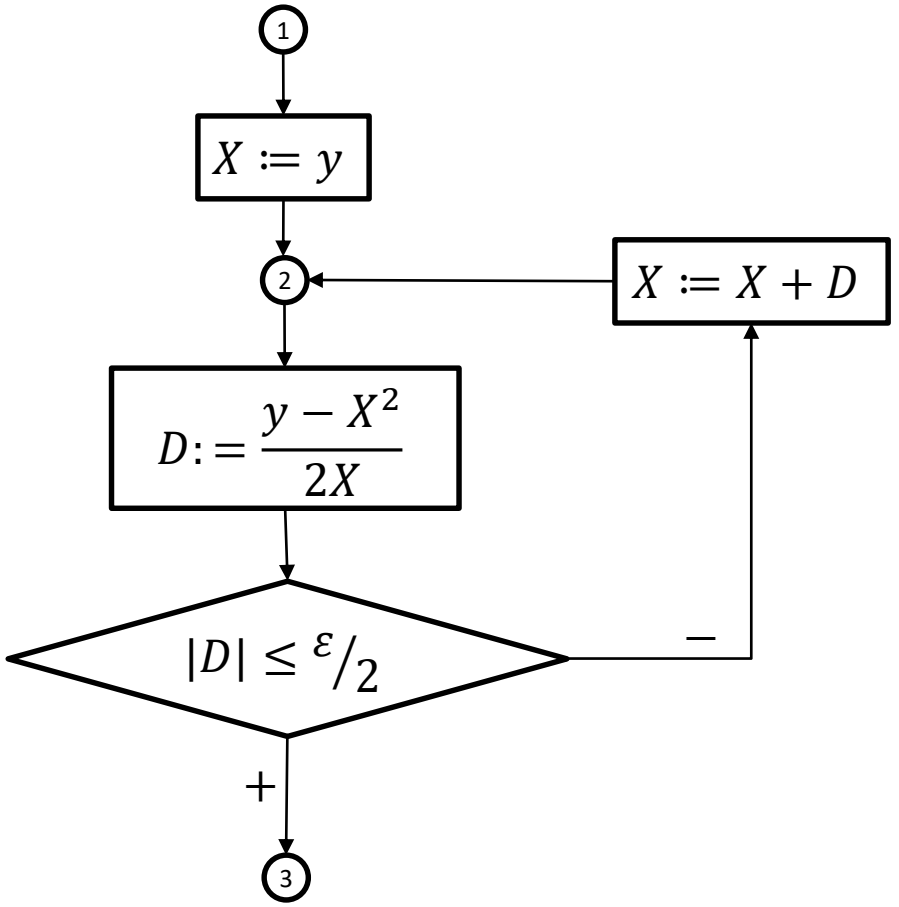
- Для элементарного случая мы проводим «ручную» верификацию (с ручкой и бумагой),
- затем выполняем ручную верификацию алгоритма, работающего в машинной арифметике, используя верификацию для элементарного случая в качестве «конспекта»,
- а заканчиваем – верификацией с использованием автоматизированной системы построения/поиска доказательства для того, что бы исключить апелляцию к «очевидности» в ручной верификации.

О чём доклад?

- Представлен опыт применения подхода для спецификации и верификации стандартной математической функции квадратного корня.
- В частности, для вычислений с фиксированной запятой верифицированная точностью $(\varepsilon + 2\delta)$, где ε – желаемая (определяемая пользователем) точность, а δ – единица последнего разряда (ulp – Unit in the Last Place).

Элементарный уровень – вещественные числа

- Вход: Любой неотрицательный вещественный аргумент $y \geq 0$ и желаемая положительная точность $\varepsilon > 0$.
- Алгоритм: реализация метода Ньютона.
- Выход: Вещественное значение переменной X аппроксимирующее $\sqrt{y}$ с точностью ε.



Спецификация, аннотация и верификация в элементарном случае

- Спецификация:
  - предусловие: $y > 1$(!) и $\varepsilon > 0$;
  - постусловие: $|X - \sqrt{y}| \leq \varepsilon$;
  - утверждение тотальной корректности:

$$[y > 1 \;\&\; \varepsilon > 0]SQRT\left[|X - \sqrt{y}| \leq \varepsilon\right].$$

- Аннотации (индуктивные утверждения):
  1. предусловие;
  2. инвариант: предусловие и $\sqrt{y} < X \leq y$;
  3. постусловие.
- Метод верификации: ручная верификация по Флойду-Хоару.

А.П. Ершов – С.С. Лаврову (март 1983 г.)

Поздравляю с днем рождения, милый друг!
Шестьдесят уже как выпало на круг.
Завтра то ли, как вчера, себя вести,
То ли новую программу завести.

Чтобы грамотно программу составлять,
Надо пред- и постусловия задать,
Надо точный счетчик времени иметь
И циклический инвариант привлечь.

Предусловие — твой праздничный венец,
Постусловие — у всех один конец,
Время катится без помощи чужой,
Для инварианта — будь самим собой!

Аксиоматизация машинной арифметики с «фиксированной запятой» – тип данных T

- Множество значений Val – это некоторое конечное подмножество рациональных (и, следовательно, вещественных) чисел такое, что
  - оно содержит наименьшее inf и наибольшее sup числа,
  - а также все числа из диапазона $[inf, sup]$ с шагом $\delta > 0$ (ulb – Unit of Least Precision или Unit in the Last Place);

  и включает все целые числа Int из этого диапазона.
- Допустимые бинарные отношения – все стандартные равенства и неравенства в рамках диапазона $[inf, sup]$.

Аксиоматизация машинной арифметики с «фиксированной запятой» – тип данных T

- Допустимые операции – это
  - машинное сложение $\oplus$ и вычитание $\ominus$; если результат математического сложения (вычитания) принадлежит диапазону $[inf, sup]$, то результат машинного сложения (соответственно – машинного вычитания) совпадает с результатом математических операций;

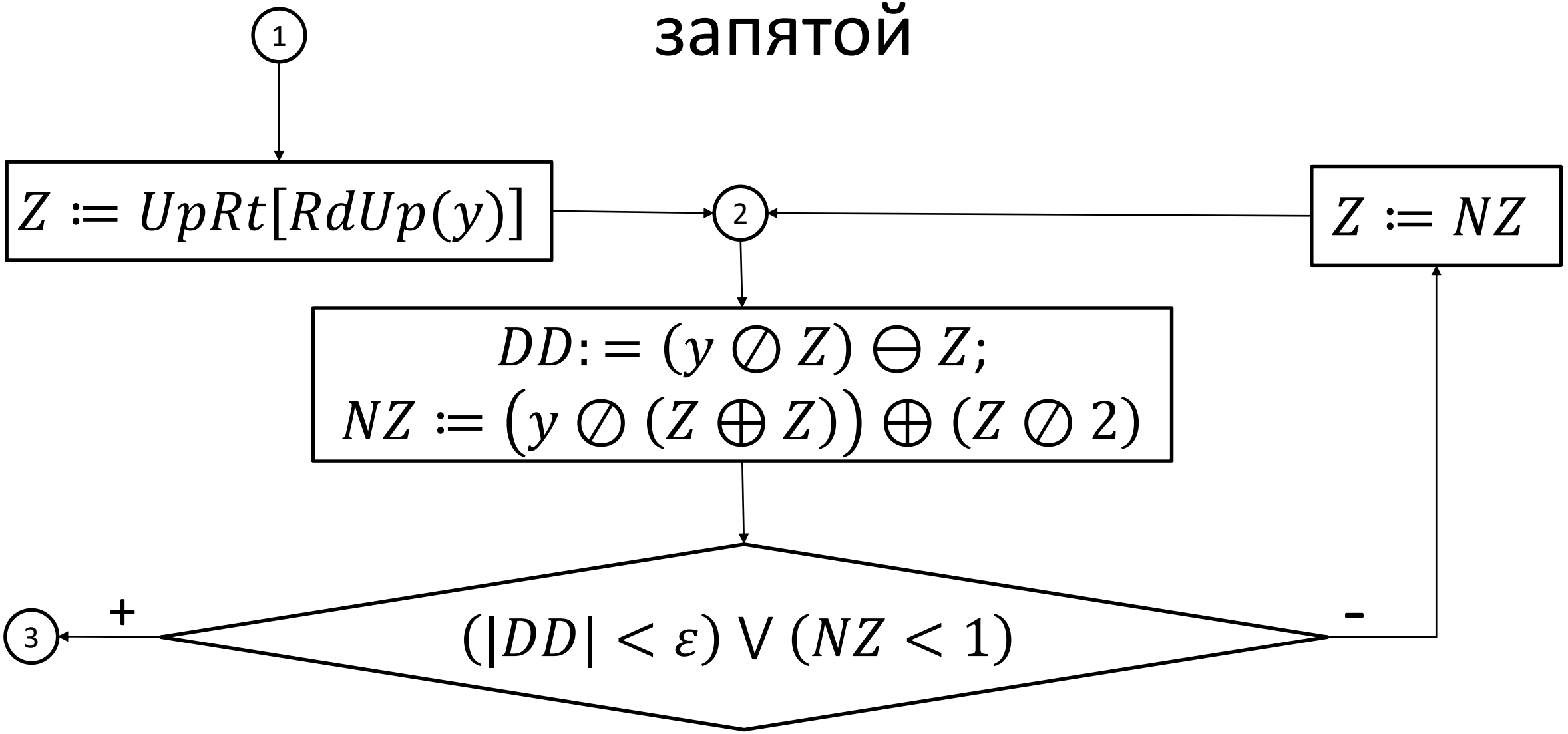
Аксиоматизация машинной арифметики с «фиксированной запятой» – тип данных T

- машинное умножение $\otimes$ и деление $\oslash$; эти операции возвращают округлённое значение соответствующих математических с округлением к ближайшему числу с фиксированной запятой, причём, для любых $x, y \in Val$
  - если $(xy) \in Val$, то $(x \otimes y) = (xy)$, иначе $|(x \otimes y) - (xy)| \leq {}^{\delta}/_{2}$;
  - если ${}^{x}/_{y} \in Val$, то $(x \oslash y) = {}^{x}/_{y}$, иначе $\left|\left(\frac{x}{y}\right) - {}^{x}/_{y}\right| \leq {}^{\delta}/_{2}$.

Существование и реализуемость типа данных T

- Вариант реализации такого тип данных (его виртуальная машина) доступен на BitBucket (https://bitbucket.org/ainoneko/lib_verify/src/f52ec4d8499224ba6bdf0c4d06078d9e45e46436/val_t/Val_T_C.md?at=default&fileviewer=file-view-default).
- Доказательство существования типа данных выполнено с использованием системы ACL2 и доступно на Github (https://github.com/apple2-66/c-light/tree/master/experiments/square-root).

Алгоритм в арифметике с фиксированной запятой



Спецификация и верификация алгоритма в арифметике с фиксированной запятой

- $\left[1 < y \;\&\; 0 < \varepsilon \;\&\; \sqrt{y} \leq UpRt[RdUp(y)] \leq \sqrt{y} + {}^{1}/_{2}\right]$
  $SQRT\ [|\sqrt{y} - NZ| \leq\ (\varepsilon + 2\delta)]$.
- Выполнена ручная верификация методом Флойда-Хоара.
- Выполнимость предусловия (существования массива) доказаны вручную и автоматически с использованием системы ACL2 и доступно на Github (https://github.com/apple2-66/c-light/tree/master/experiments/square-root).

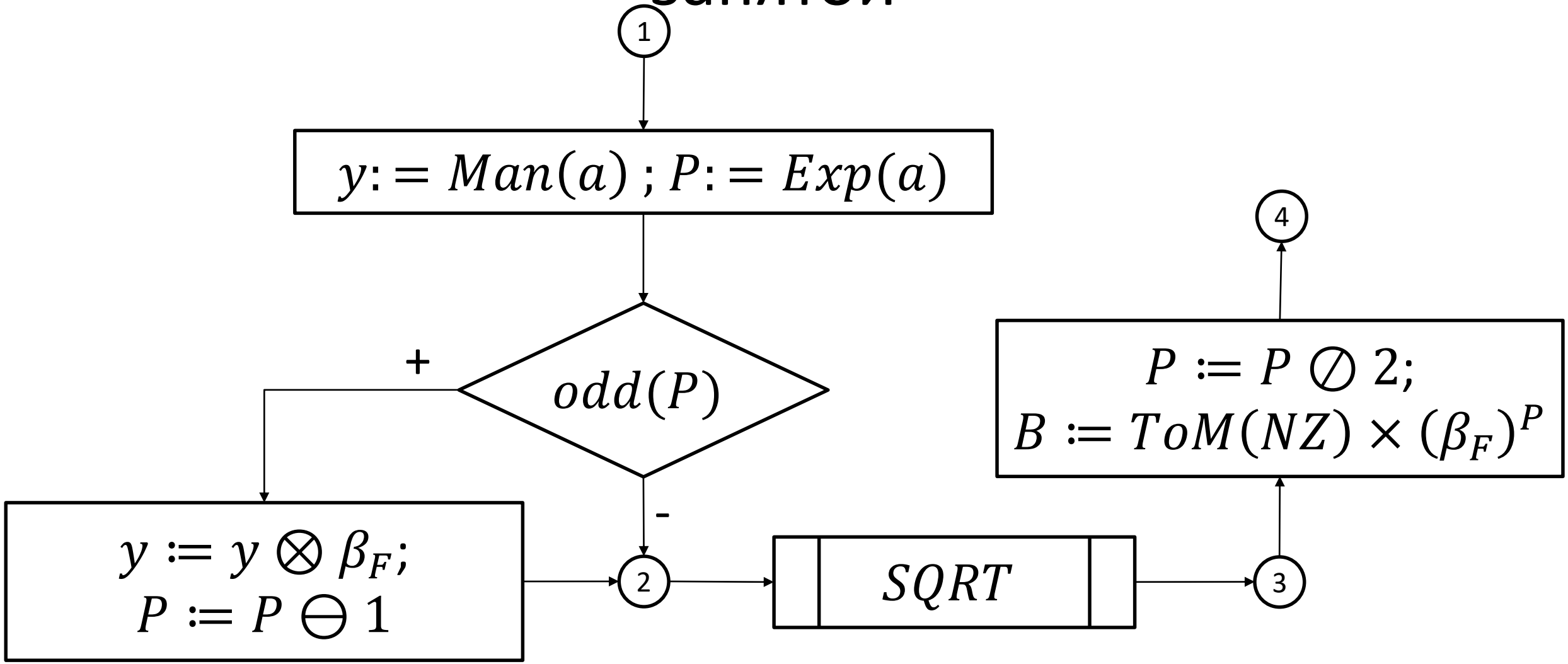
А как быть с «плавающей запятой»?

- Идея алгоритма вычисления квадратного корня в арифметике с плавающей запятой проста:

$$\sqrt{m \times b^e} = \begin{cases} \sqrt{m} \times b^{e/2}, \text{если } e \text{ — чётно;} \\ \sqrt{m \times b} \times b^{\lfloor e/2 \rfloor}, \text{если } e \text{ — нечётно.} \end{cases}$$

- Мы аксиоматизируем только операции вычисления нормализованной мантиссы и экспоненты (возвращающие по числу с плавающей запятой числа с фиксированной запятой).
- Специфицирован и вручную верифицирован алгоритм представленный на следующем слайде.

Алгоритм в арифметике с плавающей запятой



Что делать (дальше)?

- Завершить автоматизированную верификацию алгоритмов для обоих вариантов машинной арифметики.
- «Доказать теорему существования»: посмотреть, для каких реальных платформ и реализаций стандартной функции наша верификация является гарантией корректности (для чего должно быть достаточно проверить, что
  - выполнены наши аксиомы для машинной арифметики этой платформы,
  - а вычисления реализуют алгоритмы со слайдов 17 и 20).