

Digit Expansions

Jonas Bayer, Marco David, Abhik Pal and Benedikt Stock

March 24, 2023

Abstract

We formalize how a natural number a can be expanded as

$$a = \sum_{k=0}^l a_k b^k$$

for some base b and prove properties about functions that operate on such expansions. This includes the formalization of concepts such as digit shifts and carries. For a base that is a power of 2 we formalize the binary AND, binary orthogonality and binary masking of two natural numbers. This library on digit expansions builds the basis for the formalization of the DPRM theorem.

Contents

1	Digit functions	2
1.1	Simple properties and equivalences	2
2	Carries in base-b expansions	5
2.1	Definition of carry received at position k	5
2.2	Properties of carries	6
3	Digit-wise Operations	7
3.1	Binary AND	7
3.2	Binary orthogonality	8
3.3	Binary masking	9

1 Digit functions

theory *Bits-Digits*
imports *Main*
begin

We define the n-th bit of a number in base 2 representation

definition *nth-bit* :: $nat \Rightarrow nat \Rightarrow nat$ (**infix** \mid 100) **where**
 $nth\text{-bit}\ num\ k = (num\ div\ (2\ ^\ k))\ mod\ 2$

as well as the n-th digit of a number in an arbitrary base

definition *nth-digit* :: $nat \Rightarrow nat \Rightarrow nat \Rightarrow nat$ **where**
 $nth\text{-digit}\ num\ k\ base = (num\ div\ (base\ ^\ k))\ mod\ base$

In base 2, the two definitions coincide.

lemma *nth-digit-base2-equiv*: $nth\text{-bit}\ a\ k = nth\text{-digit}\ a\ k\ (2::nat)$
<proof>

lemma *general-digit-base*:
assumes $t1 > t2$ **and** $b > 1$
shows $nth\text{-digit}\ (a * b^\ t1)\ t2\ b = 0$
<proof>

lemma *nth-bit-bounded*: $nth\text{-bit}\ a\ k \leq 1$
<proof>

lemma *nth-digit-bounded*: $b > 1 \implies nth\text{-digit}\ a\ k\ b \leq b - 1$
<proof>

lemma *obtain-smallest*: $P\ (n::nat) \implies \exists k \leq n. P\ k \wedge (\forall a < k. \neg(P\ a))$
<proof>

1.1 Simple properties and equivalences

Reduce the *nth-digit* function to (j) if the base is a power of 2

lemma *digit-gen-pow2-reduct*:
 $\langle nth\text{-digit}\ a\ t\ (2\ ^\ c) \mid k = a \mid (c * t + k) \rangle$ **if** $\langle k < c \rangle$
<proof>

Show equivalence of numbers by equivalence of all their bits (digits)

lemma *aux-even-pow2-factor*: $a > 0 \implies \exists k\ b. ((a::nat) = (2^\ k) * b \wedge odd\ b)$
<proof>

lemma *aux0-digit-wise-equiv*: $a > 0 \implies (\exists k. nth\text{-bit}\ a\ k = 1)$
<proof>

lemma *aux1-digit-wise-equiv*: $(\forall k. (nth\text{-bit}\ a\ k = 0)) \longleftrightarrow a = 0$ (**is** $?P \longleftrightarrow ?Q$)
<proof>

lemma *aux2-digit-wise-equiv*: $(\forall r < k. \text{nth-bit } a \ r = 0) \longrightarrow (a \bmod 2^k = 0)$
 ⟨proof⟩

lemma *digit-wise-equiv*: $(a = b) \longleftrightarrow (\forall k. \text{nth-bit } a \ k = \text{nth-bit } b \ k)$ (is ?P \longleftrightarrow ?Q)
 ⟨proof⟩

Represent natural numbers in their binary expansion

lemma *aux3-digit-sum-repr*:
 assumes $b < 2^r$
 shows $(a * 2^r + b) \text{ i } r = (a * 2^r) \text{ i } r$
 ⟨proof⟩

lemma *aux2-digit-sum-repr*:
 assumes $n < 2^c \ r < c$
 shows $(a * 2^{c+n}) \text{ i } r = n \text{ i } r$
 ⟨proof⟩

lemma *aux1-digit-sum-repr*:
 assumes $n < 2^c \ r < c$
 shows $(\sum_{k < c} ((n \text{ i } k) * 2^k)) \text{ i } r = n \text{ i } r$
 ⟨proof⟩

lemma *digit-sum-repr*:
 assumes $n < 2^c$
 shows $n = (\sum_{k < c} ((n \text{ i } k) * 2^k))$
 ⟨proof⟩

lemma *digit-sum-repr-variant*:
 $n = (\sum_{k < n} ((\text{nth-bit } n \ k) * 2^k))$
 ⟨proof⟩

lemma *digit-sum-index-variant*:
 $r > n \longrightarrow ((\sum_{k < n} ((n \text{ i } k) * 2^k)) = (\sum_{k < r} (n \text{ i } k) * 2^k))$
 ⟨proof⟩

Digits are preserved under shifts

lemma *digit-shift-preserves-digits*:
 assumes $b > 1$
 shows $\text{nth-digit } (b * y) \ (Suc \ t) \ b = \text{nth-digit } y \ t \ b$
 ⟨proof⟩

lemma *digit-shift-inserts-zero-least-siginificant-digit*:
 assumes $t > 0$ and $b > 1$
 shows $\text{nth-digit } (1 + b * y) \ t \ b = \text{nth-digit } (b * y) \ t \ b$
 ⟨proof⟩

Represent natural numbers in their base-b digitwise expansion

lemma *aux3-digit-gen-sum-repr*:

assumes $d < b^{\wedge}r$ **and** $b > 1$

shows $\text{nth-digit } (a * b^{\wedge}r + d) \text{ } r \text{ } b = \text{nth-digit } (a * b^{\wedge}r) \text{ } r \text{ } b$

<proof>

lemma *aux2-digit-gen-sum-repr*:

assumes $n < b^{\wedge}c$ $r < c$

shows $\text{nth-digit } (a * b^{\wedge}c + n) \text{ } r \text{ } b = \text{nth-digit } n \text{ } r \text{ } b$

<proof>

lemma *aux1-digit-gen-sum-repr*:

assumes $n < b^{\wedge}c$ $r < c$ **and** $b > 1$

shows $\text{nth-digit } (\sum k < c. ((\text{nth-digit } n \text{ } k \text{ } b) * b^{\wedge}k)) \text{ } r \text{ } b = \text{nth-digit } n \text{ } r \text{ } b$

<proof>

lemma *aux-gen-b-factor*: $a > 0 \implies b > 1 \implies \exists k \text{ } c. ((a::\text{nat}) = (b^{\wedge}k) * c \wedge \neg(c \text{ mod } b = 0))$

<proof>

lemma *aux0-digit-wise-gen-equiv*:

assumes $b > 1$ **and** *a-geq-0*: $a > 0$

shows $(\exists k. \text{nth-digit } a \text{ } k \text{ } b \neq 0)$

<proof>

lemma *aux1-digit-wise-gen-equiv*:

assumes $b > 1$

shows $(\forall k. (\text{nth-digit } a \text{ } k \text{ } b = 0)) \longleftrightarrow a = 0$ (**is** $?P \longleftrightarrow ?Q$)

<proof>

lemma *aux2-digit-wise-gen-equiv*: $(\forall r < k. \text{nth-digit } a \text{ } r \text{ } b = 0) \longrightarrow (a \text{ mod } b^{\wedge}k = 0)$

<proof>

Two numbers are the same if and only if their digits are the same

lemma *digit-wise-gen-equiv*:

assumes $b > 1$

shows $(x = y) \longleftrightarrow (\forall k. \text{nth-digit } x \text{ } k \text{ } b = \text{nth-digit } y \text{ } k \text{ } b)$ (**is** $?P \longleftrightarrow ?Q$)

<proof>

A number is equal to the sum of its digits multiplied by powers of two

lemma *digit-gen-sum-repr*:

assumes $n < b^{\wedge}c$ **and** $b > 1$

shows $n = (\sum k < c. ((\text{nth-digit } n \text{ } k \text{ } b) * b^{\wedge}k))$

<proof>

lemma *digit-gen-sum-repr-variant*:

assumes $b > 1$

shows $n = (\sum k < n. ((\text{nth-digit } n \text{ } k \text{ } b) * b^{\wedge}k))$

<proof>

lemma *digit-gen-sum-index-variant*:

assumes $b > 1$ **shows** $r > n \implies$

$(\sum_{k < n}. ((nth_digit\ n\ k\ b) * b^k)) = (\sum_{k < r}. (nth_digit\ n\ k\ b) * b^k)$

<proof>

nth-digit extracts coefficients from a base-b digitwise expansion

lemma *nth-digit-gen-power-series*:

fixes $c\ b\ k\ q$

defines $b \equiv 2^{(Suc\ c)}$

assumes *bound*: $\forall k. (f\ k) < b$

shows *nth-digit* $(\sum_{k=0..q}. (f\ k) * b^k)\ t\ b = (if\ t \leq q\ then\ (f\ t)\ else\ 0)$

<proof>

Equivalence condition for the *nth-digit* function [1] (see equation 2.29)

lemma *digit-gen-equiv*:

assumes $b > 1$

shows $d = nth_digit\ a\ k\ b \iff (\exists x.\exists y.(a = x * b^{(k+1)} + d * b^k + y \wedge d < b \wedge y < b^k))$

(is $?P \iff ?Q$)

<proof>

end

theory *Carries*

imports *Bits-Digits*

begin

2 Carries in base-b expansions

Some auxiliary lemmas

lemma *rev-induct*[*consumes 1, case-names base step*]:

fixes $i\ k :: nat$

assumes *le*: $i \leq k$

and *base*: $P\ k$

and *step*: $\bigwedge i. i \leq k \implies P\ i \implies P\ (i - 1)$

shows $P\ i$

<proof>

2.1 Definition of carry received at position k

When adding two numbers m and n, the carry is *introduced* at position 1 but is *received* at position 2. The function below accounts for the latter case.

```

      k: 6 5 4 3 2 1 0
      c:           1
      - - - - -

```

$$\begin{array}{r}
m: \quad 1 \ 0 \ 1 \ 0 \ 1 \ 0 \\
n: \quad \quad \quad \quad 1 \ 1 \\
\hline
m + n: 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0
\end{array}$$

definition *bin-carry* :: *nat* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *nat* **where**
bin-carry *a b k* = (*a mod 2^k* + *b mod 2^k*) *div 2^k*

Carry in the subtraction of two natural numbers

definition *bin-narry* :: *nat* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *nat* **where**
bin-narry *a b k* = (if *b mod 2^k* > *a mod 2^k* then 1 else 0)

Equivalent definition

definition *bin-narry2* :: *nat* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *nat* **where**
bin-narry2 *a b k* = ((*2^k* + *a mod 2^k* - *b mod 2^k*) *div 2^k* + 1) *mod 2*

lemma *bin-narry-equiv*: *bin-narry* *a b c* = *bin-narry2* *a b c*
<proof>

2.2 Properties of carries

lemma *div-sub*:
fixes *a b c* :: *nat*
shows (*a - b*) *div c* = (if (*a mod c* < *b mod c*) then *a div c - b div c - 1* else *a div c - b div c*)
<proof>

lemma *dif-digit-formula*: *a* \geq *b* \longrightarrow (*a - b*)*ik* = (*a_ik* + *b_ik* + *bin-narry* *a b k*) *mod 2*
<proof>

lemma *dif-narry-formula*:
a \geq *b* \longrightarrow *bin-narry* *a b* (*k + 1*) = (if (*a_ik* < *b_ik* + *bin-narry* *a b k*) then 1 else 0)
<proof>

lemma *sum-digit-formula*: (*a + b*)*ik* = (*a_ik* + *b_ik* + *bin-carry* *a b k*) *mod 2*
<proof>

lemma *sum-carry-formula*: *bin-carry* *a b* (*k + 1*) = (*a_ik* + *b_ik* + *bin-carry* *a b k*) *div 2*
<proof>

lemma *bin-carry-bounded*:
shows *bin-carry* *a b k* = *bin-carry* *a b k mod 2*
<proof>

lemma *carry-bounded*: *bin-carry* *a b k* \leq 1

<proof>

lemma *no-carry*:

$(\forall r < n. ((\text{nth-bit } a \ r) + (\text{nth-bit } b \ r) \leq 1)) \implies$
 $(\text{nth-bit } (a + b) \ n) = (\text{nth-bit } a \ n + \text{nth-bit } b \ n) \bmod 2$

(is ?P \implies ?Q n)

<proof>

lemma *no-carry-mult-equiv*: $(\forall k. \text{nth-bit } a \ k * \text{nth-bit } b \ k = 0) \longleftrightarrow (\forall k. \text{bin-carry } a \ b \ k = 0)$

(is ?P \longleftrightarrow ?Q)

<proof>

lemma *carry-digit-impl*: $\text{bin-carry } a \ b \ k \neq 0 \implies \exists r < k. a \ \downarrow \ r + b \ \downarrow \ r = 2$

<proof>

end

theory *Binary-Operations*

imports *Bits-Digits Carries*

begin

3 Digit-wise Operations

3.1 Binary AND

fun *bitAND-nat* :: $\text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat}$ (**infix** && 64) **where**

$0 \ \&\& \ - = 0 \ |$

$m \ \&\& \ n = 2 * ((m \ \text{div} \ 2) \ \&\& \ (n \ \text{div} \ 2)) + (m \ \text{mod} \ 2) * (n \ \text{mod} \ 2)$

lemma *bitAND-zero[simp]*: $n = 0 \implies m \ \&\& \ n = 0$

<proof>

lemma *bitAND-1*: $a \ \&\& \ 1 = (a \ \text{mod} \ 2)$

<proof>

lemma *bitAND-rec*: $m \ \&\& \ n = 2 * ((m \ \text{div} \ 2) \ \&\& \ (n \ \text{div} \ 2)) + (m \ \text{mod} \ 2) * (n \ \text{mod} \ 2)$

<proof>

lemma *bitAND-commutes*: $m \ \&\& \ n = n \ \&\& \ m$

<proof>

lemma *nth-digit-0*: $x \leq 1 \implies \text{nth-bit } x \ 0 = x$ *<proof>*

lemma *bitAND-zeroone*: $a \leq 1 \implies b \leq 1 \implies a \ \&\& \ b \leq 1$

<proof>

lemma *aux1-bitAND-digit-mult*:

fixes $a\ b\ c :: \text{nat}$

shows $k > 0 \wedge a \bmod 2 = 0 \wedge b \leq 1 \implies (a + b) \text{ div } 2^k = a \text{ div } 2^k$

<proof>

lemma *bitAND-digit-mult*: $(\text{nth-bit } (a \ \&\& \ b) \ k) = (\text{nth-bit } a \ k) * (\text{nth-bit } b \ k)$

<proof>

lemma *bitAND-single-bit-mult-equiv*: $a \leq 1 \implies b \leq 1 \implies a * b = a \ \&\& \ b$

<proof>

lemma *bitAND-mult-equiv*:

$(\forall k. (\text{nth-bit } c \ k) = (\text{nth-bit } a \ k) * (\text{nth-bit } b \ k)) \longleftrightarrow c = a \ \&\& \ b$ (**is** $?P \longleftrightarrow ?Q$)

<proof>

lemma *bitAND-linear*:

fixes $k :: \text{nat}$

shows $(b < 2^k) \wedge (d < 2^k) \implies (a * 2^k + b) \ \&\& \ (c * 2^k + d) = (a \ \&\& \ c) * 2^k + (b \ \&\& \ d)$

<proof>

3.2 Binary orthogonality

cf. [1] section 2.6.1 on "Binary orthogonality"

The following definition differs slightly from the one in the paper. However, we later prove the equivalence of the two definitions.

fun *orthogonal* :: $\text{nat} \Rightarrow \text{nat} \Rightarrow \text{bool}$ (**infix** \perp 49) **where**

$(\text{orthogonal } a \ b) = (a \ \&\& \ b = 0)$

lemma *ortho-mult-equiv*: $a \perp b \longleftrightarrow (\forall k. (\text{nth-bit } a \ k) * (\text{nth-bit } b \ k) = 0)$ (**is** $?P \longleftrightarrow ?Q$)

<proof>

lemma *aux1-1-digit-lt-linear*:

assumes $b < 2^r \ k \geq r$

shows $\text{bin-carry } (a * 2^r) \ b \ k = 0$

<proof>

lemma *aux1-digit-lt-linear*:

assumes $b < 2^r$ **and** $k \geq r$

shows $(a * 2^r + b) \ i \ k = (a * 2^r) \ i \ k$

<proof>

lemma *aux-digit-shift*: $(a * 2^t) \ i \ (l+t) = a \ i \ l$

<proof>

lemma *aux-digit-lt-linear*:
assumes $b: b < (2::nat)^{\wedge}t$
assumes $d: d < (2::nat)^{\wedge}t$
shows $(a * 2^{\wedge}t + b) \dot{\leq} k \leq (c * 2^{\wedge}t + d) \dot{\leq} k \longleftrightarrow ((a * 2^{\wedge}t) \dot{\leq} k \leq (c * 2^{\wedge}t) \dot{\leq} k \wedge b \dot{\leq} k \leq d \dot{\leq} k)$
 $\langle proof \rangle$

lemma *aux2-digit-lt-linear*:
fixes $a b c d t l :: nat$
shows $\exists k. (a * 2^{\wedge}t) \dot{\leq} k \leq (c * 2^{\wedge}t) \dot{\leq} k \longrightarrow a \dot{\leq} l \leq c \dot{\leq} l$
 $\langle proof \rangle$

lemma *aux3-digit-lt-linear*:
fixes $a b c d t k :: nat$
shows $\exists l. a \dot{\leq} l \leq c \dot{\leq} l \longrightarrow (a * 2^{\wedge}t) \dot{\leq} k \leq (c * 2^{\wedge}t) \dot{\leq} k$
 $\langle proof \rangle$

lemma *digit-lt-linear*:
fixes $a b c d t :: nat$
assumes $b: b < (2::nat)^{\wedge}t$
assumes $d: d < (2::nat)^{\wedge}t$
shows $(\forall k. (a * 2^{\wedge}t + b) \dot{\leq} k \leq (c * 2^{\wedge}t + d) \dot{\leq} k) \longleftrightarrow (\forall l. a \dot{\leq} l \leq c \dot{\leq} l \wedge b \dot{\leq} l \leq d \dot{\leq} l)$
 $\langle proof \rangle$

Sufficient bitwise (digitwise) condition for the non-strict standard order of natural numbers

lemma *digitwise-leq*:
assumes $b > 1$
shows $\forall t. nth_digit\ x\ t\ b \leq nth_digit\ y\ t\ b \implies x \leq y$
 $\langle proof \rangle$

3.3 Binary masking

Preliminary result on the standard non-strict of natural numbers

lemma *bitwise-leq*: $(\forall k. a \dot{\leq} k \leq b \dot{\leq} k) \longrightarrow a \leq b$
 $\langle proof \rangle$

cf. [1] section 2.6.2 on "Binary Masking"

Again, the equivalence to the definition there will be proved in a later lemma.

fun *masks* :: $nat \implies nat \implies bool$ (**infix** \preceq 49) **where**
 $masks\ 0 = True$ |
 $masks\ a\ b = ((a \div 2 \preceq b \div 2) \wedge (a \bmod 2 \leq b \bmod 2))$

lemma *masks-substr*: $a \preceq b \implies (a \div (2^{\wedge}k) \preceq b \div (2^{\wedge}k))$
 $\langle proof \rangle$

lemma *masks-digit-leq*: $(a \preceq b) \implies (\text{nth-bit } a \ k) \leq (\text{nth-bit } b \ k)$

<proof>

lemma *masks-leq-equiv*: $(a \preceq b) \iff (\forall k. (\text{nth-bit } a \ k) \leq (\text{nth-bit } b \ k))$ (**is** $?P \iff ?Q$)

<proof>

lemma *masks-leq*: $a \preceq b \implies a \leq b$

<proof>

lemma *mask-linear*:

fixes $a \ b \ c \ d \ t :: \text{nat}$

assumes $b < (2 :: \text{nat})^t$

assumes $d < (2 :: \text{nat})^t$

shows $((a * 2^t + b) \preceq (c * 2^t + d)) \iff (a \preceq c \wedge b \preceq d)$ (**is** $?P \iff ?Q$)

<proof>

lemma *aux1-lm0241-pow2-up-bound*: $(\exists (p :: \text{nat}). (a :: \text{nat}) < 2^{\lceil \text{Suc } p \rceil})$

<proof>

lemma *aux2-lm0241-single-digit-binom*:

assumes $1 \geq (a :: \text{nat})$

assumes $1 \geq (b :: \text{nat})$

shows $\neg(a = 1 \wedge b = 1) \iff ((a + b) \text{ choose } b) = 1$ (**is** $?P \iff ?Q$)

<proof>

lemma *aux3-lm0241-binom-bounds*:

assumes $1 \geq (m :: \text{nat})$

assumes $1 \geq (n :: \text{nat})$

shows $1 \geq m \text{ choose } n$

<proof>

lemma *aux4-lm0241-prod-one*:

fixes $f :: (\text{nat} \Rightarrow \text{nat})$

assumes $(\forall x. (1 \geq f \ x))$

shows $(\prod_{k \leq n. (f \ k)} = 1) \implies (\forall k. k \leq n \implies f \ k = 1)$ (**is** $?P \implies ?Q$)

<proof>

lemma *aux5-lm0241*:

$(\forall i. (\text{nth-bit } (a + b) \ i) \text{ choose } (\text{nth-bit } b \ i) = 1) \implies$

$\neg(\text{nth-bit } a \ i = 1 \wedge \text{nth-bit } b \ i = 1)$

(**is** $?P \implies ?Q \ i$)

<proof>

end

References

- [1] Y. Matiyasevich. On Hilbert's tenth problem. In M. Lamoureux, editor, *PIMS Distinguished Chair Lectures*, volume 1. Pacific Institute for the Mathematical Sciences, 2000.