Tnum <AND, OR xor AND> Proof

2.24.21

a) Let A be the set of concrete values represented by tnum x.

b) Let a_{and}, a_{or} represent functions that perform bitwise AND and bitwise OR, respectively, on all members of set A:

$$A = \{a_1, a_2, a_3, ..., a_n\}$$

$$a_{and} = a_1 \land a_2 \land a_3 \land ... \land a_n$$

$$a_{or} = a_1 \lor a_2 \lor a_3 \lor ... \lor a_n$$

Observation 0.1 *If all members of set A contain 1 in the ith bit, then a_{and} will return 1 in the ith bit. This corresponds to the known 1's in the tnum.*

Observation 0.2 *If all members of set A contain 0 in the ith bit, then a_{or} will return 0 in the ith bit. This corresponds to the known 0's in the tnum.*

Observation 0.3 *Any 1 in the ith bit of the resulting bitvector $a_{or} \oplus a_{and}$ corresponds to uncertain bits in the tnum. Let $a_{uncertain} = a_{or} \oplus a_{and}$ (where $\oplus$ represents bitwise xor)*

Observation 0.4 *Any 0 in the ith bit of the resulting bitvector $a_{or} \oplus a_{and}$ corresponds to certain bits in the tnum.*

Observation 0.5 *Any 1 in the ith bit of the resulting bitvector $\neg(a_{or} \oplus a_{and})$ corresponds to certain bits in the tnum. Let $a_{certain} = \neg(a_{or} \oplus a_{and})$*

Definition 1 (Well-formed tnum) $x.value \land x.mask = 0$.

Definition 2 (Tnum membership) *Let y be a concrete value, then $y \in x \iff y \land \neg x.mask = x.value$.*

Theorem 3 *Given a set of concrete values A, a correct and maximally precise tnum x can be derived with the following formulation: $< a_{and}, a_{or} \oplus a_{and} >$ where $a_{and} = x.value$ and $a_{or} \oplus a_{and} = x.mask$.*

Proof: [Soundness]. First, we can show that our formulation produces a well formed tnum in the following way :

$$\text{x.value} \wedge \text{x.mask} = 0$$

$$= a_{and} \wedge (a_{or} \oplus a_{and}) = 0$$

$$= (a_{and} \wedge a_{or}) \oplus (a_{and} \wedge a_{and}) = 0$$

$$= (a_{and} \wedge a_{or}) \oplus a_{and} = 0$$

$$= a_{and} \oplus a_{and} = 0$$

*note that, by definition, $a_{and} \subseteq a_{or}$ which implies that $a_{and} \wedge a_{or} = a_{and}$.

Let A_k be an arbitrary member of A and $A_k[i]$ denote the ith bit of member A_k. Now, using case analysis, we show that all members of A are represented by the tnum $< a_{and}, a_{or} \oplus a_{and} >$ by satisfying the definition of tnum membership:

$$A_i \wedge \neg x.mask = x.value$$

$$= A_k \wedge \neg(a_{or} \oplus a_{and}) = a_{and}$$

$$= A_k[i] \wedge a_{certain}[i] = a_{and}[i]$$

1) $A_k[i] = 0$. This implies that $a_{and}[i] = 0$ since a_{and} will capture any 0 in the ith of a member of A if such exists. Thus the bitwise operation $A_k[i] \wedge a_{certain}[i] = a_{and}[i]$ holds regardless of the value of $a_{certain}[i]$.

2) $A_k[i] = 1$ and $a_{and}[i] = 1$. In this case the ith bit must be a certain 1 since it is contained by a_{and}. This implies that $a_{certain}[i] = 1$, which means that $A_k[i] \wedge a_{certain}[i] = a_{and}[i]$ must also be true.

3) $A_k[i] = 1$ and $a_{and}[i] = 0$. In this case, the 1 in the ith bit is uncertain since it is not present in a_{and} which implies that $a_{certain} = 0$. Thus, $A_k[i] \wedge a_{certain}[i] = a_{and}[i]$ must hold in this case as well.

∎

Proof: [Maximal precision]. Here we show that $< a_{and}, a_{or} \oplus a_{and} >$ is also maximally precise.

Definition 4 *Given set A and tnum x which represents it, we call the ith bit of tnum x certain if it has the same value across all members of set A. Then tnum x is maximally precise if it captures all certain bits in set A:*

$$\forall a \in A, a_1[i] = a_2[i] = a_3[i] ... = a_n[i] = P \implies x.mask[i] = 0, x.value[i] = P$$

Lemma 5 *Given an n-bit tnum x, let $|x|$ denote the size of the set of concrete values tnum x represents. Then $|x| = 2^{n-k}$ where k denotes the number of certain bits in tnum x.*

Proof: The number of possible values we can represent with an n-bit number is $2^n = 2 \cdot 2 \cdot 2 \cdot ... \cdot 2$ n times. Let k denote the number of certain bits in tnum x. Then $n - k$ must be the number of uncertain bits in tnum x. From Observation 0.4 above, we know that every 0 in the ith bit of the tnum mask $(a_{or} \oplus a_{and})$ represents a certain bit, meaning this bit can only take one value. Likewise, from Observation 0.3, we know that every 1 in the ith bit of the tnum mask $(a_{or} \oplus a_{and})$ represents an uncertain bit, meaning this bit can take two values. This gives us a piecewise function where the ith bit of tnum x can represent either one option or two options as follows:

$$\text{x[i]} = \begin{cases} 2 & \text{if bit i is uncertain} \\ 1 & \text{if bit i is certain} \end{cases}$$

Then the number of values tnum x can represent is

$$|x| = x[0] \cdot x[1] \cdot x[2] \cdot ... \cdot x[n] = 2^{n-k} 1^k = 2^{n-k}.$$

∎

Lemma 6 *$< a_{and}, a_{or} \oplus a_{and} >$ will capture all certain bits q, if such exist, in set A.*

Proof: There are only two ways in which the ith bit of tnum x can be deemed certain - the ith bit across all members of set A is set to 1, or the ith bit across all members of set A is set to 0. The following cases show how our formulation captures both instances:

1) $a_{or}[i] = 0$. This implies that $a_{and}[i] = 0$ which means that all members of set A contain 0 in the ith bit. Then $a_{or}[i] \oplus a_{and}[i] = 0$, meaning that the ith bit of tnum x must be certain.
2) $a_{and}[i] = 1$. This implies that $a_{or} = 1$ which means that all members of set A contain 1 in the ith bit. Then $a_{or}[i] \oplus a_{and}[i] = 0$, meaning that the ith bit of tnum x must be certain in this case as well.

Then we can say that all q certain bits, if such exist, in set A must be accounted for.

∎

By Lemma 6, $< a_{and}, a_{or} \oplus a_{and} >$ will capture all q certain bits in set A - this corresponds to the maximal amount of certain bits present in set A. This satisfies Definition 4 and implies that $< a_{and}, a_{or} \oplus a_{and} >$ must produce a maximally precise tnum x. By Lemma 5, the maximally precise tnux x will represent 2^{n-q} values. ∎