Tnum <AND, OR xor AND> Proof

2.24.21

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

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

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

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

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

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 1 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) $a.value \wedge a.mask = 0$.

Definition 2 (Tnum membership) $x \in a \iff x \wedge \neg a.mask = a.value$.

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

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

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

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

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

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

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

*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:

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

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

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

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 *Let* $\min(|a|)$ *denote the minimal cardinality of tnum a that can represent all members of set A. Then tnum a is maximally precise if* $|a| = \min(|a|)$.

Lemma 5 *Given an n-bit tnum a,* $|a| = 2^{n-k}$ *where k denotes the number of certain bits in tnum a.*

Proof: Given an n-bit number, the number of possible values we can represent is 2^n. Let k denote the number of certain bits in tnum a. From our observations above, we know that every 0 in the ith bit of the tnum mask $(a_{or} \oplus a_{and})$ represents a certain bit. It follows that every such certain bit must be present in all members of tnum a. Then the combinatorial question we seek to answer is: how many values out of the 2^n possible representable values contain k certain bits. Therefore, $|a| = \dfrac{2^n}{2^k} = 2^{n-k}$. ∎

It follows from Lemma 5 that in order to find the maximally precise tnum, $\min(|a|) = \min(2^{n-k})$, k must be the maximum value possible given set A - meaning that all certain bits of set A must be captured in the mask. If this is not the case, then tnum a is not maximally precise.

Lemma 6 *$a_{or} \oplus a_{and}$ will capture all certain bits in a given set A.*

Proof: By contradiction, let's assume $a_{or} \oplus a_{and}$ does not capture all certain bits in a given set A. Then a_{and} will return 0 in the ith bit if all set members contain 1 in the ith bit and a_{or} will return 1 in the ith bit if all set members contain 0 in the ith bit. This, however, is contradicting to the nature of the bitwise operations defined above and thus cannot be true. ∎

By Lemma 5 and Lemma 6 and Definition 4, we have that $< a_{and}, a_{or} \oplus a_{and} >$ must be maximally precise and we conclude that Theorem 3 must hold. ∎