Correcting Theorem 6.4.2

There's a subtlety in Theorem 6.4.2 that no one quite got. (Don't feel bad; we didn't either when we wrote the book.) We're not very formal in the book when we define σ^{sba}. There is also some room for confusion about when $decided_i$ should hold. Let's consider the second issue first. On p. 192, we define π in ba-compatible interpreted contexts so that $\pi(decided_i(y))$ is true at (r, m) if the action decide_i was performed in any previous round. Now suppose that i is correct at $(r, m-1)$ and is supposed to perform the $\mathsf{decide}_i(y)$ action according to its protocol at round m, but the environment also makes i faulty in round m. Should $decided_i(y)$ hold at $(r, m+1)$? I want the answer to be yes. Since γ is a recording context, the environment state records whether process i attempted to perform $\mathsf{decide}_i(y)$, so the truth of $decided_i(y)$ at a point can be determined by looking at the environment state at that point.

Now consider σ^{sba}. In particular, look at the Agreement and Simultaneity Properties, as informally defined on p. 192. Agreement, for example, says that "the nonfaulty processes all decide on the same value". What does that mean? It could mean:

(a) If i and j are nonfaulty throughout run r, then if i decides on y, then so does j.

(b) If i is nonfaulty at (r, m) and has decided y by (r, m) (note that we avoid the problems above of what "decide" really means here, since i is nonfaulty after it has already decided) and j is nonfaulty at (r, m') and has decided y' by (r, m'), then $y = y'$.

(c) If i is nonfaulty at (r, m), and i is about decide y at (r, m), and j is nonfaulty at (r, m') and is about to decide y', then $y = y'$.

We have analogous problems with Simultaneity. It could mean:

(a) If i and j are nonfaulty throughout run r, then if i decides at round m, then so does j.

(b) If i and j are nonfaulty at (r, m) and i has decided by (r, m), then so has j

(c) If i and j are nonfaulty at (r, m) is i about decide at (r, m), then so is j.

The literature is fuzzy on which intepretation to take. The only papers that address the issue carefully that I'm aware of use (a) (although I'll admit I haven't exactly scoured the literature to check). Actually, it turns out that, by and large, the fuzziness in the literature doesn't matter. No matter which definition of Agreement and Simultaneity we use, and no matter what we assume about $decided_i$, as long as the environment's protocol is such that it is always possible that no further processes fail (in a sense I'll explain below), then a protocol that guarantees SBA under one interpretation guarantees SBA under all of them.

However, given that w're putting so few constraints on the environment protocol in the definition of ba-compatible contexts, we need to be more careful. Definition (a) is not in the spirit of our usage of $\mathcal{N}(r,m)$, where we consider which processes are correct/faulty at a point, not throughout the run. Definition (b) is probably closest to how we interpret things in the book. However, both definitions (a) and (b) are somewhat problematic, due to the fact that ba-compatible contexts give a lot of freedom to the environment protocol. To see why, suppose we have a system with two processes (i.e., $n = 2$). The environment makes exactly one of them faulty in round 1 of every run. (This is allowed by the definition.) Now consider the following trivial protocol. At round 1, each process decides on its preferred value (i.e., process i performs the action $\mathsf{decide}(x_i)$). There is no communication at all. Since at time 1 exactly one of the processes will have failed, it will be true that all the correct processes will agree on the same value (since there is only one correct process). Under interpretation (a) or (b), this protocol attains SBA. Now consider a run r where process 1 starts with a 0, process 2 starts with a 1, and process 2 is the one that fails in round 1. It is easy to see that $(\mathcal{I}, r, 0) \models \mathit{deciding}_{\mathcal{N}}(0)$. But there is another run r' where, again, process 1 starts with a 0, process 2 starts with a 1, but now process 1 fails in round 1. Since $1 \in \mathcal{N}(r,0) \cap \mathcal{N}(r',1)$ and $(\mathcal{I}, r', 0) \models \mathit{deciding}_{\mathcal{N}}(1) \wedge \neg \mathit{deciding}_{\mathcal{N}}(0)$, we have $(\mathcal{I}, r, 0) \models \neg C_{\mathcal{N}}(\mathit{deciding}_{\mathcal{N}}(0))$, contradicting Theorem 6.4.2.

However, under interpretation (c) the protocol above is *not* an SBA protocol at all. In a run where process 1 starts with initial value 0 and process 2 starts with initial value 1, then $\mathit{deciding}_1(1) \wedge \mathit{deciding}_2(0)$ holds at time 0, contradicting Agreement under interpretation (c) (since both 1 and 2 are nonfaulty at time 0). Moreover, we can show that under interpretation (c) for Agreement and Simultaneity, Theorem 6.4.2 is correct provided we take $(\mathcal{I}, r, m) \models \mathit{deciding}_{\mathcal{N}}(y)$ iff $(\mathcal{I}, r, m) \models \mathit{deciding}_i(y)$ for all $i \in \mathcal{N}(r,m)$. Pretty much all of you who got full credit for Problem 6.1.5 took this definition of $\mathit{deciding}_{\mathcal{N}}$ and were implicitly assuming interpretation (c). (I was quite lenient on grading this though.) Interestingly, the Dwork-Moses paper on which much of this discussion is based, seems to be using interpretation (c) in the proof of their analogue of Theorem 6.4.2.

Notice that if we use interpretation (c), then we should also add a few other observations to the book. In particular, we should observe that if $\mathcal{I}$ satisfies σ^{sba}, by Simultaneity, we have

$$\mathcal{I} \models \mathit{deciding}_i(y) \Rightarrow B_i^{\mathcal{N}}(\mathit{deciding}_{\mathcal{N}}(y)).$$

Thus, using Theorem 6.4.2, we get the corollary

$$\mathcal{I} \models \mathit{deciding}_i(y) \Rightarrow B_i^{\mathcal{N}} C_{\mathcal{N}}(\mathit{deciding}_{\mathcal{N}}(y)).$$

This is the fact we actually use in our knowledge-based program for SBA.