Our textbook proves that TQBF is PSPACE-complete. The same proof is replicated all over the Web. Every treatment that I've found glosses over too many details for my taste, especially in the base case. So I've written out these notes, to make the proof more explicit.

Recall that TQBF is the set of all true, fully quantified Boolean formulas. We have already demonstrated that TQBF is in PSPACE. So it remains to prove that TQBF is PSPACE-hard.

Theorem 0.1. *Any language $L \in PSPACE$ is polynomial-time-reducible to TQBF.*

Proof. Let M be a deterministic Turing machine that decides L in polynomial space. There exists some constant k such that, for all n and all strings w of length n, M uses at most n^k space to decide whether $w \in L$. Let Q be the state set of M and Γ the tape alphabet of M. A configuration of M is a string $a_0 a_1 \cdots a_{n^k}$ over $Q \cup \Gamma$ in the usual way. When M is given input $w = w_1 \cdots w_n$, its starting configuration is

$$c_{\text{start}} = q_{\text{start}} w_1 \cdots w_n \textvisiblespace \cdots \textvisiblespace.$$

Without loss of generality, we assume that M, before halting, empties its tape and parks its tape head at the far left, so that its unique accepting configuration is

$$c_{\text{accept}} = q_{\text{accept}} \textvisiblespace \cdots \textvisiblespace.$$

Notice that $T = |Q \cup \Gamma|^{n^k+1}$ is an upper bound on the number of configurations of M, and hence on the time that M requires to accept or reject w. For any input w, we will recursively construct a fully quantified Boolean formula that describes the operation of M on w. The formula will be true if and only if M accepts w.

We need to establish a preliminary concept. For any symbol c, we can form a set

$$\{c_{i,s} : i = 0, \ldots, n^k, s \in Q \cup \Gamma\}$$

of $|Q \cup \Gamma| \cdot (n^k + 1)$ variables. Any given configuration of M corresponds to a unique assignment of truth values to these variables: Namely, $c_{i,s}$ is true in the assignment if and only if cell i of the configuration contains symbol s. On the other hand, there are some assignments of truth values that do not correspond to any configuration of M — for example, an assignment in which $c_{3,a}$ and $c_{3,b}$ are both true, or an assignment in which $c_{4,q}$ and $c_{6,q}$ are both true (for $q \in Q$).

Given two symbols c and d, we have two variable sets $\{c_{i,s}\}$ and $\{d_{i,s}\}$, and we can write the Boolean formula

$$\phi_{c,d,0} = \bigwedge_{i=0}^{n^k} \bigwedge_{s \in Q \cup \Gamma} (c_{i,s} \wedge d_{i,s}) \vee (\overline{c_{i,s}} \wedge \overline{d_{i,s}}).$$

Because the formula is unquantified, its truth or falsity depends on an assignment of truth values to its variables; under some assignments it is true, and under other assignments it is false. What we can say is this: If we assign truth values to the c-variables based on a certain configuration of M, and we assign truth values to the d-variables based on another configuration of M, then the formula is true if and only if the two configurations are identical — that is, iff M can go from the first configuration to the second in zero steps.

We now wish to write a formula that is true iff M can go from one configuration to another in one step. This formula will incorporate information about M's transition function $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\mathcal{L}, \mathcal{R}\}$.

Namely, if $\delta(q,a) = (r,b,\mathcal{L})$, then consider the formula

$$\begin{aligned}\phi_{c,d,1}^{j,(q,a,r,b,\mathcal{L})} &= c_{j,q} \wedge c_{j+1,a} \wedge d_{j-1,r} \wedge d_{j+1,b} \wedge \left(\bigvee_{e \in \Gamma} c_{j-1,e} \wedge d_{j,e} \right) \\ &\wedge \left(\bigwedge_{i \in \{0,\ldots,j-2,j+2,\ldots,n^k\}} \bigwedge_{s \in Q \cup \Gamma} (c_{i,s} \wedge d_{i,s}) \vee (\overline{c_{i,s}} \wedge \overline{d_{i,s}}) \right).\end{aligned}$$

If we assign truth values to the c- and d-variables based on two configurations of M, then $\phi_{c,d,1}^{j,(q,a,r,b,\mathcal{L})}$ is true if and only if the first configuration has its state marker in cell j and M moves from the first configuration to the second using the transition $\delta(q,a) = (r,b,\mathcal{L})$. The first line of the formula expresses the change of state, the movement of the tape head, and the modification of the tape around the tape head; the second line of the formula expresses the fact that the rest of the tape remains unchanged. One can invent a similar formula to express transitions of the form $\delta(q,a) = (r,b,\mathcal{R})$. Then let

$$\phi_{c,d,1} = \phi_{c,d,0} \vee \bigvee_{j} \bigvee_{(q,a,r,b,\mathcal{D}) \in \delta} \phi_{c,d,1}^{j,(q,a,r,b,\mathcal{D})}.$$

The variables in $\phi_{c,d,1}$ are all free (unquantified). If we assign truth values to the c- and d-variables based on two configurations of M, then $\phi_{c,d,1}$ is true if and only if M can transition from the first configuration to the second in one or fewer steps. This is the base case of our recursion.

Now suppose that $t > 1$. Consider the formula

$$\phi_{c,d,t} = \exists m \left(\phi_{c,m,t/2} \wedge \phi_{m,d,t/2} \right),$$

where m is a symbol distinct from c and d, and "$\exists m$" is shorthand for the concatenation of the $|Q \cup \Gamma| \cdot (n^k + 1)$ quantifiers $\exists m_{i,s}$, for $i = 0, \ldots, n^k$ and $s \in Q \cup \Gamma$. This formula is free on the c- and d-variables but quantified on all of its other variables. If we assign truth values to the c-variables based on one configuration of M, and truth values to the d-variables based on another configuration of M, then we obtain a fully quantified Boolean formula that is true if and only if M can go from the first configuration to the second in t or fewer steps. If we use c_{start} and c_{accept} as our two configurations, and $t = T$ for our time bound, then $\phi_{c,d,t}$ is true if and only if M accepts w. In other words, this algorithm is a reduction from L to TQBF:

(1) For some symbols c and d, recursively compute $\phi_{c,d,T}$.
(2) Replace the d-variables with their truth values determined from c_{accept}.
(3) For the given w, replace the c-variables with their truth values determined from the starting configuration c_{start} of M on input w.

The problem with this reduction is that the resulting formula is too large. On each step of the recursion, we halve t, but we double the number of ϕ-formulas involved. In the end, the number of subformulas of the form $\phi_{c,d,1}$ will be proportional to T, and hence exponential in n. So the space required will be exponential in n, and the time required must be (at least) exponential in n. We wanted a polynomial-time reduction.

We can do better by replacing the conjunction in $\phi_{c,d,t}$ with a universal quantifier. Let $\phi_{c,d,t}$ be the formula

$$\exists m \, \forall (c', d') \in \{(c,m),(m,d)\} \; \phi_{c',d',t/2}.$$

This formula is really shorthand for

$$\exists m \; \forall c' \; \forall d' \; ((\phi_{c',c,0} \wedge \phi_{d',m,0}) \vee (\phi_{c',m,0} \wedge \phi_{d',d,0})) \rightarrow \phi_{c',d',t/2}.$$

Keep in mind that each quantifier here is actually shorthand for $|Q \cup \Gamma| \cdot (n^k + 1)$ quantifiers. How long is this formula? Well, before the $\rightarrow$ symbol, there are three sets of $\mathcal{O}(n^k)$ quantifiers each, and there are four formulas of the form $\phi_{c,d,0}$, each of which is $\mathcal{O}(n^k)$. So each level of recursion adds $\mathcal{O}(n^k)$ symbols to the formula. There are

$$\log_2 T = \log_2 |Q \cup \Gamma|^{n^k+1} = (n^k + 1) \log_2 |Q \cup \Gamma| \in \mathcal{O}(n^k)$$

levels of recursion. So the total length of the formula is $\mathcal{O}(n^{2k})$. In addition, computing this polynomial-length formula requires only polynomial time, because at each level of recursion the algorithm simply selects three new symbols m, c', d', writes some quantifiers and $\phi_{c,d,0}$-terms based on those symbols, and divides t by two.

□

A detail: The Turing machine that computes the reduction is not given k. So it might have to try various values of k, until it finds one that works. This inflates the running time by a constant factor.

Another detail: The divisions by two are a bit messy, unless T is a power of two. So make T be the smallest power of two that is at least $|Q \cup \Gamma|^{n^k+1}$.

Another detail: Writing a symbol with subscripts requires more than one tape cell. The size of the subscripts is logarithmic in n^k, so this detail can't ruin the running time.