

A Patriotic Analysis of Programming Paradigms

Jacob Weiner
Abraham Lincoln Chair for Patriotism
Founding Fathers University

SIGBOVIK 2022

Abstract

In this paper, we will examine why dependent type theory is the most patriotic paradigm for programming. We will examine several key founding documents to draw our conclusion. We will view many American flag images and feel patriotic.

1 Introduction

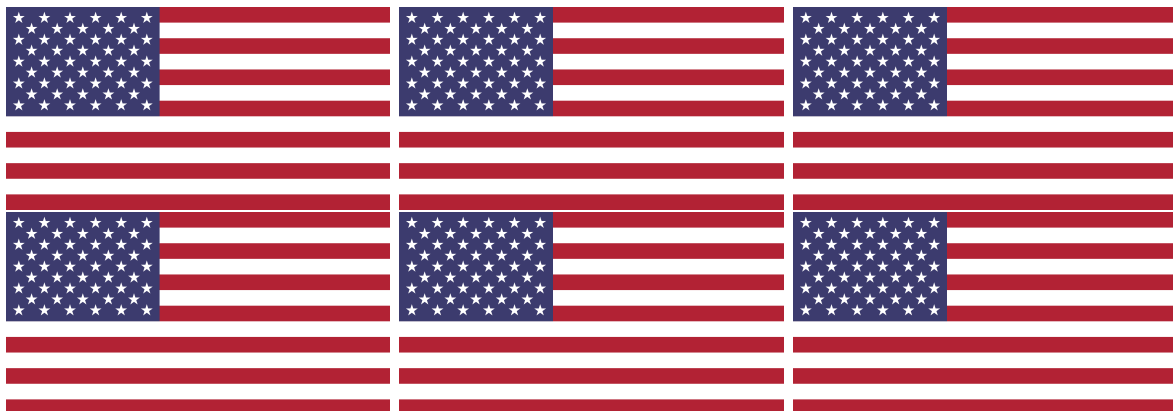
It is universally accepted among Americans that patriotism is the highest and most important virtue¹. Hence, it is critical to the overall well-being of our nation and the people within it to identify what paradigms and techniques in computer science are deserving of our glorious star-spangled flag, and which are being secretly promulgated by evil enemy spies. Thus, we can rid ourselves of those elements which hold us back from the glorious future which our founders intended for us.

To quote John Winthrop, "...we shall be as a [shining] city upon a hill" if all our code is truly *American*.

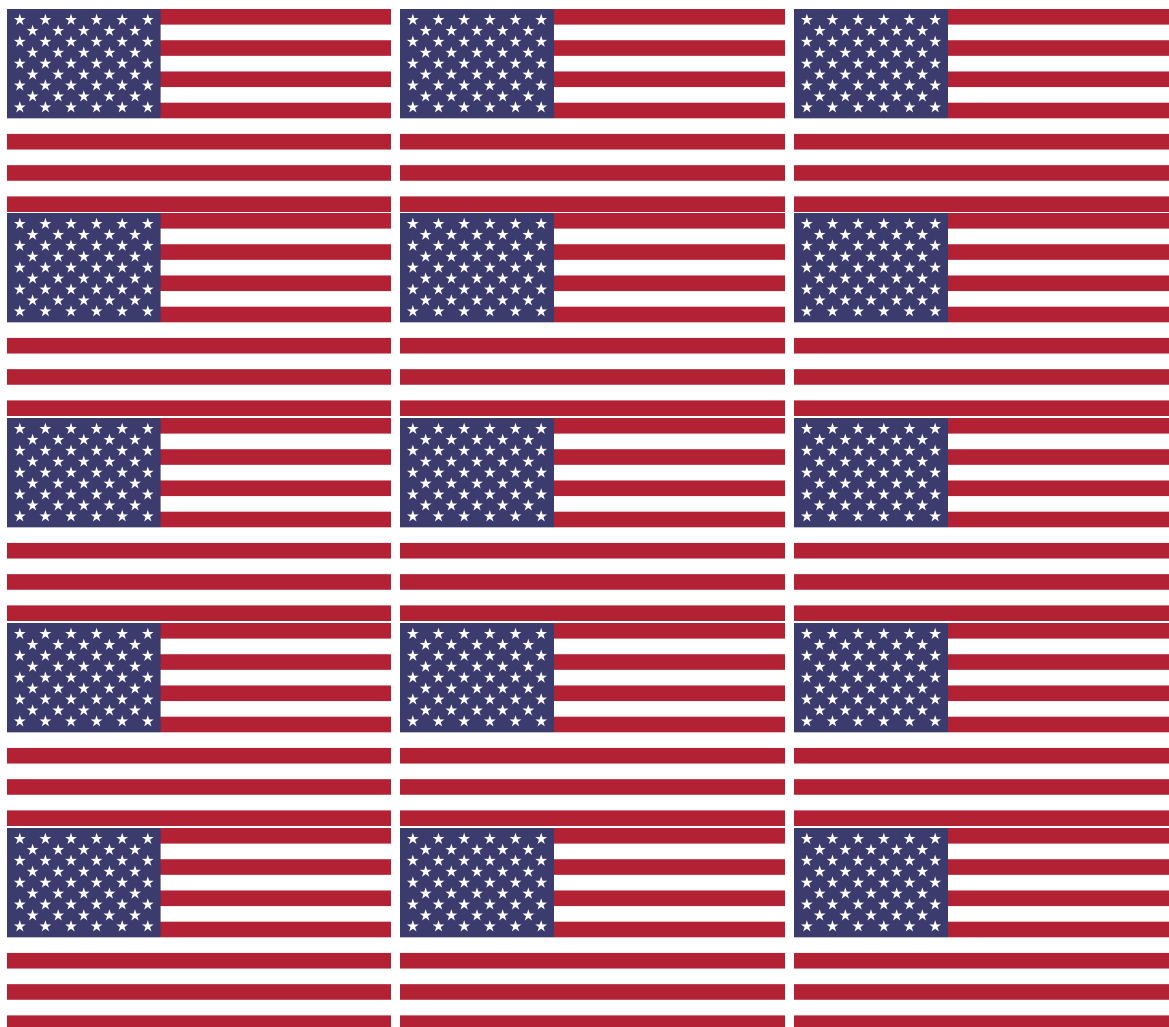


Figure 1: AMERICA

2 Motivation



¹If you dissent, you are an **unpatriotic traitor!!!!!!**



My coding 'tis of thee
Sweet land of liberty
Of thee I sing...

Land where the simply typed
Lambda calculus thrives,
From every shell, a cry:
"Let FP ring!"

3 Methods

I examined founding documents and common sayings for out-of-context quotations to support my thesis.

I used American flags to distract from several questionable logical leaps.



Figure 2: AMERICA

4 Analysis of Data and Results

The first hint on our quest to find the most patriotic paradigm is in our core document, the Consitution. The Constitution defines what it means to be American², and no part of it more so than the Preamble.

The Preamble states that the goal of America is to form a more perfect **union**. Therefore, any paradigm which hopes to be patriotic must support union types. This allows for C, which features **union** types, as well as functional languages, which allow for datatypes with multiple constructors.

The importance of union types can be seen in Article 1, Section 1. This section clearly describe “member of congress” a union type: either a **Senator** or a **Representative**.



Figure 3: AMERICA

If any document hopes to come close to the Constitution in patriotism³, it is the Declaration of

²Obviously.

³Yes, I just created that word.

Independence, crafted by the great Thomas Jefferson. It notes that all human beings have certain “unalienable rights”.

Note the importance of the word **unalienable** – it means that these rights can neither be altered nor removed. Hence, we *must*⁴ conclude that immutability of data is a core patriotic value, without which our country would fall to ruins.

Only one paradigm offers immutability, along with the union types so highly prized by the Constitution. This is **functional programming**.

One curious note about the Constitution is that Congress is described in article 1, while the Presidency is delayed until article 2. This is because the founders believed that deliberation is more important than action. By extensionality, we must conclude that they would regard imperative programming to be inferior, as it is a paradigm of action.



Figure 4: AMERICA

So we are now absolutely sure that the paradigm of our founding fathers is functional. But what flavor of functional programming? Another look at the Declaration of Independence will illuminate us to the true intentions of Washington, Franklin, Jefferson, and all the other great ones.

“We hold these *truths* to be self-evident: that all men are created *equal*...”

Note the three key words in this patriotic phrase:

- **Truth** and **evident** indicate that we ought to have a way to create evidence that our programs are working correctly.
- The mention of **equality** indicates that all types should be equality types.

The latter condition is rather stringent, but it can be done if we allow for function extensionality. But to use function extensionality, we need to express propositions with universal quantifiers within our language! How can this be done?

There is only one answer: we must use dependent type theory with the Curry-Howard Correspondence⁵.

In fact, we could have seen this from the beginning, for it is often said that there is nothing as American as apple pie – and dependent type theory includes at its core Pi-types!

One piece of further evidence that dependent type theory is what our founders would have supported comes from the 9th amendment in the Bill of Rights. It notes that the enumeration of rights “should not be construed” to imply that there were not further rights. This clearly implies

⁴At least, those of us who are truly AMERICAN.

⁵Both Curry and Howard are good old American citizens.

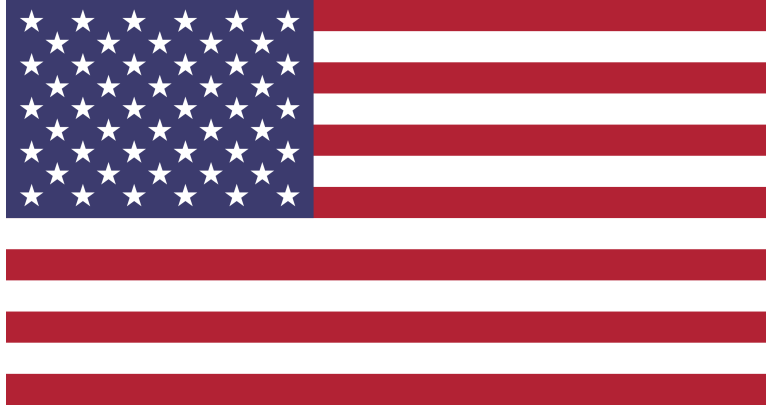


Figure 5: AMERICA

that the set of all rights to which American citizens are entitled is in fact an uncountable set, as they cannot be enumerated!

Dependent type theory easily handles creating and reasoning about uncountable types, while few other paradigms truly support uncountability – even so-called “real number types” in many languages, from C to SML, are limited by machine precision. Yet using dependent type theory, one must represent real numbers precisely – or else not at all!

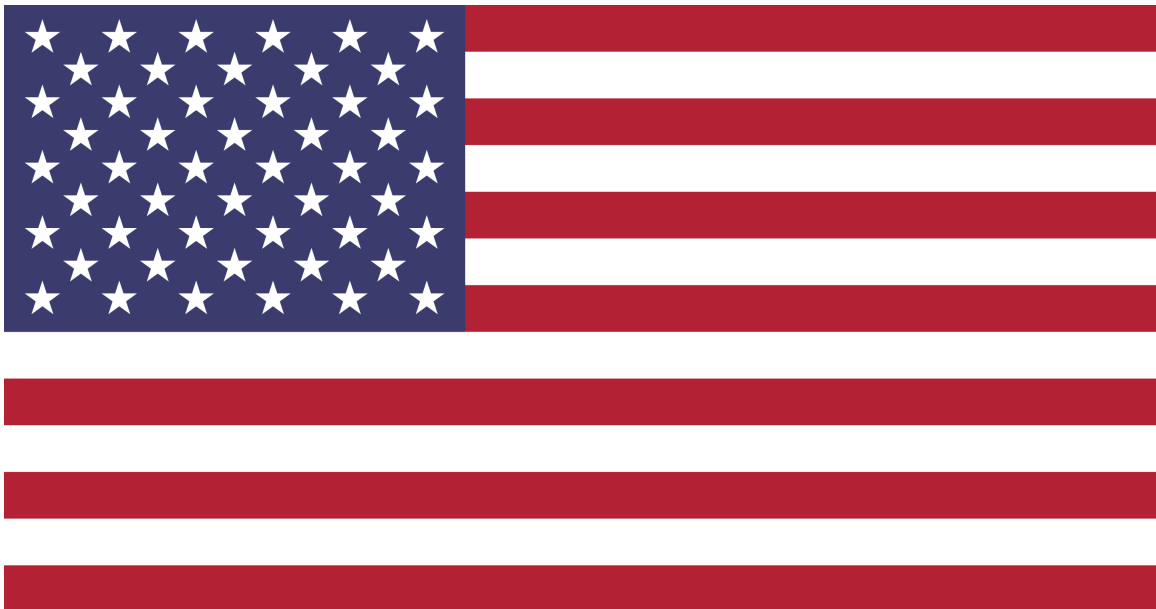


Figure 6: AMERICA

5 Conclusions

Dependent type theory should be the only paradigm used by true American computer scientists and programmers. Any other paradigm would cause the founding fathers to roll over in their graves.

6 Recommendations

Ban all other paradigms from American soil immediately.

7 Future Research

This highly innovative method can be used to evaluate differing techniques in many other areas of human effort and ingenuity, including science, mathematics, art, literature, social sciences, business, basket weaving, red herring catching, and WiFi customer service.

8 Impact

America will be better because of this research. I am sure of it.



Figure 7: AMERICA

9 Acknowledgments

George Washington is to be acknowledged.



Figure 8: AMERICA