Blockchain Lectures Notes (BLN) - README.md

The Second Cohort Of The Plutus Pioneer Programme

In my humble opinion, Cardano is to Blockchains what Linux is to Kernels.

Cardano has essentially become my job... I do love this community though :)

Currently Working On: Alonzo Purple Test Net & Proof-Of-Burn Challenge

1. Authors Note

I have made the decision to begin making notes and documenting all lectures and exercises found within the Plutus Pioneer Program (cohort two). The writing style will be two-fold: technical, but creative. This is for numerous reasons. Firstly, writing does aid the cognitive ability to solve problems [1], Plutus is a new platform, Plutus-core is a new language, Haskell is somewhat unique and understanding aspects of how this whole system fits together requires an element of creative thinking and problem solving. Similarly, these technologies [1] are novel in nature, as such, it may be required to do some creative writing in order to explore new potential ideas.

I've been using distributed ledgers and blockchain consensus algorithms for over ten years. I remember buying BTC at $3 and seeing it hit $7 was mind blowing for me (at the time). It's a shame I never held on to many, but I just look back and laugh. Why? Because it's not about money, although most people would disagree (this is one of the reasons why I feel somewhat at home within the Cardano community). It's about technology and ultimately: changing the world. It's a strange thought: we are individuals, we have choices, and if we decide to do so, we can make a real difference here. If there is anything worth pursuing, it's making a difference. This is why I decided to devote as much time as possible to the difficult task of participating within the Plutus Pioneer Program and brushing up on Haskell.

I would like to express my gratitude to IOHK for constructing this program and accepting me as a candidate. Furthermore, I would like to thank my family for their continual support in my efforts to pursue a career path which isn't thought to be conventional.

I believe I can speak for the entire community when I extend my sincerest thanks to the world class engineers and scientists who have contributed to building Cardano as of current. Charles, you will are and will continue to be an inspiration to us all and to the world. I hope we can all collectively strive to make positive societal change.

1.1 Updates

14th AUG: Exciting times! Alonzo Purple Test Net is available to a subset of the pioneers. The Hard Fork on the Main Net is due 12th of September (what a great birthday present). We will see the implementation of MetaData, Native Assets and SMART CONTRACT FUNCTIONALITY. The one thing I would warn people about, and this is purely my own subjective opinion, is: don't expect too much to happen all at once. I get this feeling that people think that as soon as smart contracts are available the price of ADA is going to explode. Firstly, it's never been about the price. Secondly, we're writing Smart Contracts in Haskell (which isn't HUGELY adopted within industry [9]), I believe we've chosen the right tool for the job, but adoption may take some time. Furthermore, the Catalyst projects may take a while to get going and finally, there are literally only a couple of thousand developers (at most) that can write Plutus. Plutus is basically Haskell, so maybe I'm overblowing it here, but you do still need to know the gotchas. For example: compiling parameterised validators, if I remember correctly, requires a specific knowledge of Plutus, because standard oxford brackets simply will not compile. It would be nice to see hundreds of new products and platforms instantaneously using Cardano Smart Contracts, but I'm just a little skeptical. However, I do hold the following opinion:

There is only one way from here on, and that's up.

2. Preface

This set if notes, written in a casual, formal, yet creative style has been cultivated through the activity of participating within the second cohort of the Plutus Pioneer Program. I felt it to be important to document my own experience with the program in order to gain the most from it. Furthermore, it is my hope that the notes themselves (and perhaps even through the contribution of others) will be able to help other aspiring developers for time to come.

To be completely honest, I'm not hugely qualified to write any of this! Which is why I would hugely appreciate any corrections via contribution. I do however hold the following qualifications:

- A First Class Honours Degree in Computer Science from the University of Manchester.
- A Postgraduate Diploma in Artificial Intelligence / Informatics (I hope to write my thesis on this very subject after COVID-19 has 'blown over', thus obtaining my MSc) from the University of Edinburgh.

Note: for what it's worth I hold a good few years experience in industry writing web *applications... Not exactly what you would call academic.*

I wrote this set of notes in an ad-hoc style manner. As the lectures were released, I've done my best to keep up with work, in addition to following the lectures, implementing the homework's and keeping these detailed notes.

3. Foreword

To any and all IOHK employees: feel free to contribute by adding a foreword via a pull request.

4. Formatting

As these are technical notes, the format for each set of lecture notes will be written to a pseudo-technical specification. This means each set of notes should follow a template, making it easier for readers to break down and understand. Each set of notes will be presented in accordance with the following schema:

- Introductory Lecture Information
- Incremental Sections (1. → N.)
- Subsections (1.1 → N.X)
- Possible *Coffee Time* Sections — Thought Experiments, or Exercises (These Will Be Italicised) and will be specified as 'COFFEE TIME!'
- End Of Section Notes — Optional, Only If Required
- Exercises and Associated Comments
- Questions and Comments on Possible Problems (Things I May Be Uncertain Of)
- Lecture Summary

I will be doing my best to keep to the above schema for each set of notes, but this isn't a publishable book or formal technical documentation, so expect a degree of variance in style.

Note: both footnotes and references will be used within this document set, which may seem confusing at times. However, please be aware that footnotes are formatted as enumerated superscript values, whereas footnotes are enumerated bracketed values.

Quotes: You will see various quotes throughout the text, this is simply how I like to write.

5. Content Within This Repo

The contents of this repo is two-fold. Firstly (and most importantly), the content is a set of pseudo-academically written notes for the Plutus Pioneer Program [3], which follow the formatting as outlined in §5. Secondly, a set of academic papers which I have deemed to be of greatest importance **to me** is also provided. In addition, these papers will (slowly, but surly) be printed out, annotated and scanned. Shortly thereafter, these scans will be reuploaded to this repo.

This repo will **forever and always** remain open, unless instructed otherwise by a member of the Cardano organisation or by an employee of IOHK.

6. Intended Audience

These notes are for anybody and everybody (somewhat vague, I know), but the audience could range from those who are current Plutus Pioneers, individuals who are just following the programme, developers from other communities, or even legislators (however, I highly doubt they would be reading my notes / documentation on Cardano). But ultimately, we are all here for the same reason, we believe in what we're building and the best way to approach adoption is to allow everybody to see what you're doing, at least, that would be my intuition.

But, to reduce the scope a little bit and be somewhat more realistic, the intended audience is really for those who are looking to develop on Cardano.

7. Why These Notes May Be Helpful

First and foremost, these notes are likely most helpful to me. You do not understand something unless you can explain it. So, if you can write it, you can explain **something**, whether or not that **something** is correct is another matter. My hope is that most of what I do write **is correct** or that somebody reads it and corrects me (either via a direct message, but I would encourage open collaboration). However, once I have finished this program (and I will), and once I have got a **real firm grasp** on Haskell (and I will) and finally, after I have meticulously written and re-written every set of lecture notes (and I will), then this repo may be hugely benefitial to other people. That is the aim with this paricular set of notes and that is why I believe these notes may be hepful.

8. Contributing

Anyone is free to contribute via a pull request.

9. Table of Contents

- Lecture Six: Oracles, Core module, Swap module, Funds Module, Test Module, PAB & Front-ends
- Lecture Seven: Committal Schemes, State Machines, Maintaining State Using Datum, NFTs and 'Thread'
- Lecture Eight: More State Machines? And Testing... *Note: Not Yet Quite There...*
- Lecture Nine: Marlowe
- Lecture Ten: -

Note: Currently Unwritten Notes are Marked With '-' unless stated otherwise.

10. Summary

As an accepted candidate for the second cohort (my first cohort e-mail landed in my junk mail), I have decided to write a set of notes for each and every lecture in the hope that it will help me learn, but in addition, it will help others. These are not strictly technical notes, there is some creative flair (at least I would like to think so), so you may even enjoy the read. I have a few years experience in industry as a web developer, software engineer and machine learning engineer. I got my UG @ the University of Manchester and my PG @ the University of Edinburgh. Formatting is suppose to be (be isn't hugely) consistent. My intended audience is mainly other Plutus Pioneers or those who wish to learn and are outside of the program. Anyone is free to contribute and feel free to flick through each lecture as is listed above.

11. Nomenclature

- UTxO: A model of accounting used to identify how much 'money' (in this case: a digital 'currency') any 'wallet' has access to (in the context of 'cryptocurrencies'[1].
- EUTxO: An extended model of UTxO. The fundamentals remain the same. Thus, transactions are made up of numerous inputs, which themselves are unspent transaction outputs. However, there are some modifications to the model which are important. These modifications allow for more general transactions through the use of arbitrary logic [8].
- Cardano-node: syncs to the Cardano Blockchain and talks between all of the Cardano subcomponents. It doesn't have to be set up as a stake pool, however, it is possible to set up a stake pool. It may be a core component for facilitating PoS consensus through the operation of a `pool` . It is important to note that others can delegate to a `pool` , but as of right now, will not earn the same kind of annual RoA (return on ada) as an operator.
- cardano-cli: command line interface for Cardano.
- cardano-wallet: HTTP server and command line for managing on-chain UTxOs.
- cardano-db-sync: postgreSQL for storing blocks and transactions.
- cardano-graphql: similar to FB api, it's easy to understand why. You have a load of wallets with keys in them (think of these as 'people' - they're not though, they're wallets)

and you have a bunch of connections between them called Unspent Transaction Outputs. You also have a load of transactions, so a graph seems like an appropriate data structure to me to represent this kind of data (wallets and Txs are nodes, UTxOs are edges).

- cardano-rosetta: DevOps for Blockchain.
- cardano-addresses: for backing up wallets (key phrases) - a module.
- cardano-ledger-specs: formal specification for current release.
- bech32: Haskell implementation of bech32.
- smash: powers SPOs - metadata aggregation server and keeps track of whos staking what.
- ouroboros-network: they used fancy words, I'm just going to call this the Cardano Blockchain Consensus Algorithm.
- Plutus-Platform: an application development platform for developing distributed applications using the Cardano blockchain [5].
- Compiler: A compiler takes some source code, and produces an output in another language, while the retaining meaning of the source.
  `Source -> (analysis) -> Compiler -> (Synthesis) -> Target.` [4]
- plutus-tx: The compiler which is responsible for transforming Plutus (which is a subset of Haskell) into Plutus-core [6].
- plutus-core: What high level languages are to assembly language, Plutus is to Plutus-core [7]. From what I understand, plutus-core is essentially: System F omega with (equi-)recursive types.
- AssetClass: A native token that exists on the Cardano blockchain.
- Minting: the process of creating a set of native tokens or an NFT.
- Burning: the process of destroying a native token.
- Bytestring: A datatype which may contain 8bit bytes.
- XOverloadStrings: A Haskell setting that allows a Bytestring to contain a String literal.

More Items Are Continued To Be Added As Time Elapses.

12. StackExchange (Cardano) Answers:

1. What exactly is a redeemer?
2. How to compile plutus source code locally?

13. References

[1] Kellogg, R.T., 1999.
The psychology of writing.
Oxford University Press.

[2] Jones, M.P., 2021.
Plutus Tx: compiling Haskell into Plutus Core. Viewed 2nd August 2021.

https://iohk.io/en/blog/posts/2021/02/02/plutus-tx-compiling-haskell-into-plutus-core/

[3] The Cardano Foundation, July 2021.
Input Output Hong Kong (IOHK).
Last Updated: Early July 2021.
https://testnets.cardano.org/en/plutus-pioneer-program/

[4] Jonathon Dilworth, 18 May 2015.
The University of Manchester Alumni.
Last Updated: 7 August 2021.
https://github.com/jonathondilworth/compilers

[5] IOHK GitHub.
Input Output Hong Kong (IOHK).
https://github.com/input-output-hk/plutus

[6] IOHK GitHub.
Input Output Hong Kong (IOHK).
https://github.com/input-output-hk/plutus/tree/master/plutus-tx

[7] IOHK GitHub.
Input Output Hong Kong (IOHK).
https://github.com/input-output-hk/plutus/tree/master/plutus-core

[8] Chakravarty, M.M., Chapman, J., MacKenzie, K., Melkonian, O., Jones, M.P. and Wadler, P., 2020, February.
The extended UTXO model.
In International Conference on Financial Cryptography and Data Security (pp. 525-539).
Springer, Cham.

[9] Rabai, B.A., 2015.
Programming Language Use in US Academia and Industry.
Informatics in Education, 14(2), pp.143-160.

14. Footnotes

1. The technologies I am referencing are Distributed Ledgers and Blockchain Consensus Algorithms — 'cryptocurrencies'.

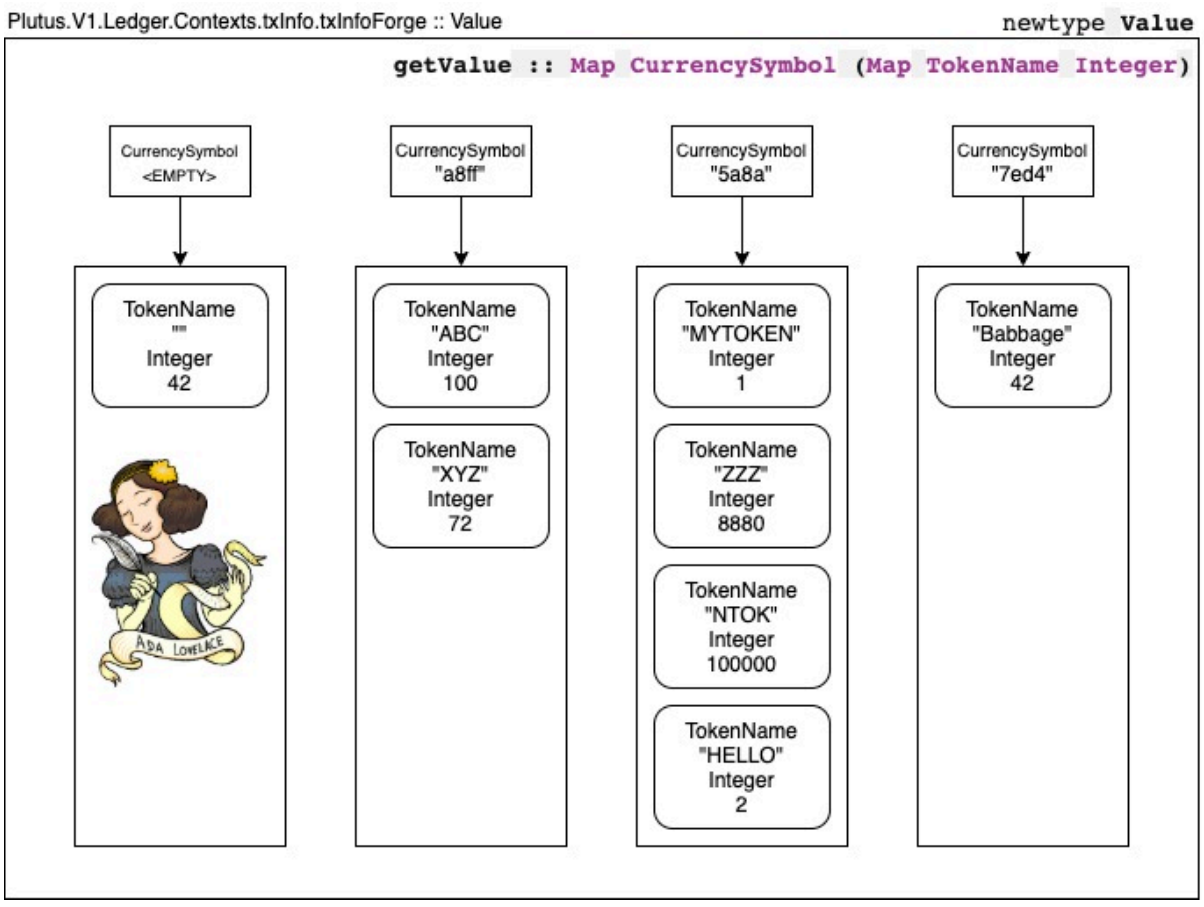
15. Appendix A

15.1: Plutus-Platform: The 'on-chain' and 'off-chain' portion of Cardano's smart(er) contract implementation (from this point onwards, smart contracts simply refer to the extended UTxO model that implements Plutus). This facilitates smart contract transactions (which may be complex in nature) using off-chain code (implemented in 'raw' Haskell, plus some libraries made available via Cardano. *note: it does not require compiling to Plutus-core*). Off-chain code performs transactional validation checks (some of which are time-based) to ensure any such transaction can be legitimately executed (or the UTxO can be 'spent,' or as it were, 'not-spent'). Furthermore, on-chain code performs the execution of transaction via active nodes operating on the network (when prompted to) and writes to the (collective) Blockchain accordingly.

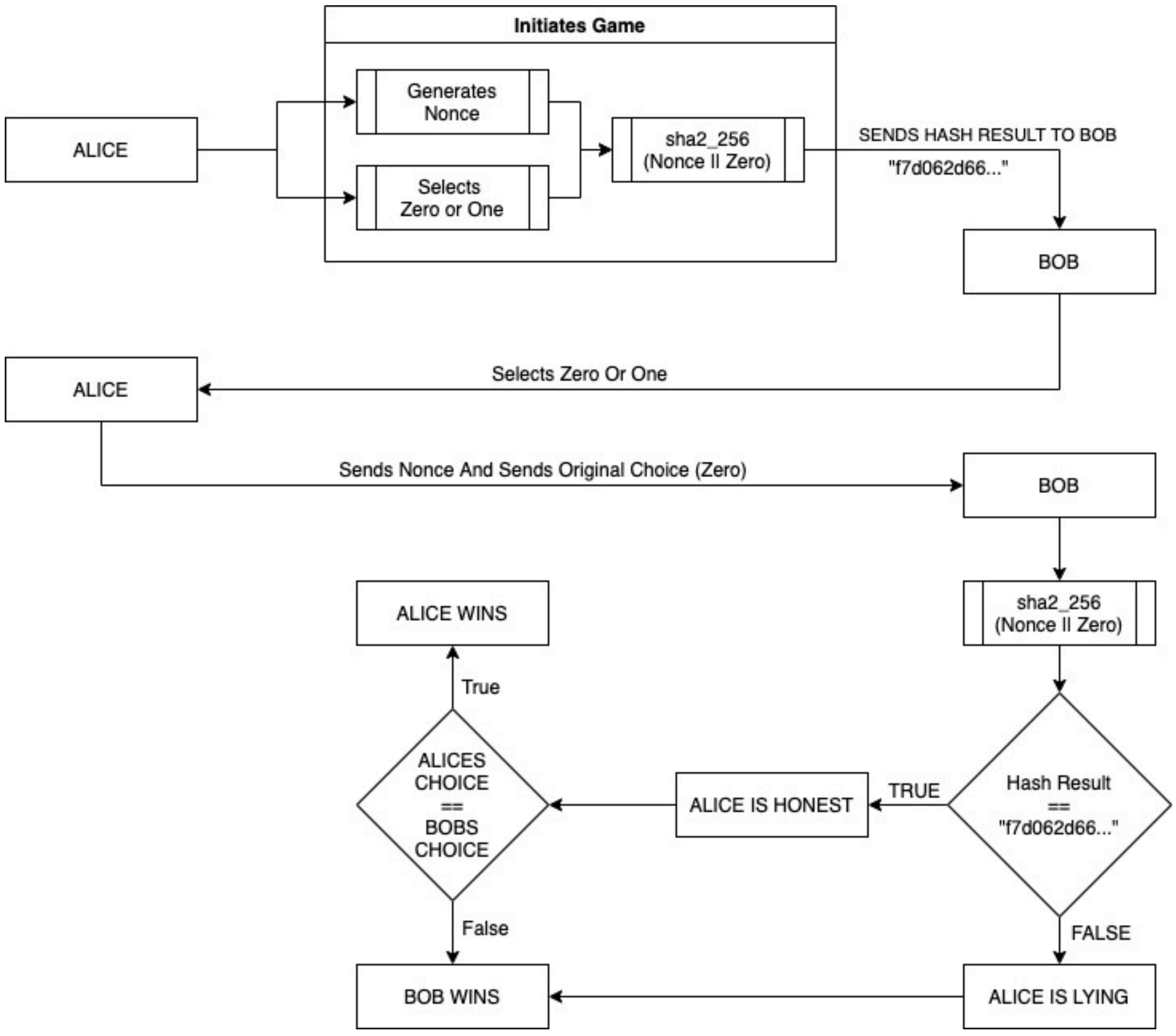
15.2: Contracts can remain 'stale' forever (if nobody changes their state by initiating a change of state at that 'script' or EUTxO address).

16. Appendix B: Images

16.1: Plutus/V1/Ledger/Value.hs Constructor:


Plutus.V1.Ledger.Contexts.txInfo.txInfoForge :: Value
newtype Value
getValue :: Map CurrencySymbol (Map TokenName Integer)
CurrencySymbol <EMPTY>
CurrencySymbol "a8ff"
CurrencySymbol "5a8a"
CurrencySymbol "7ed4"
TokenName "" Integer 42
TokenName "ABC" Integer 100
TokenName "XYZ" Integer 72
TokenName "MYTOKEN" Integer 1
TokenName "ZZZ" Integer 8880
TokenName "NTOK" Integer 100000
TokenName "HELLO" Integer 2
TokenName "Babbage" Integer 42
ADA LOVELACE

16.2: Committal Scheme:



17. Appendix: Additional Tables

In Progress

18. Appendix: Additional Literature

In Progress

19. TODO

1. Correct any errors within any of my current lecture notes.
2. Continue to include references and footnotes within all lecture notes.
3. Catch up to and finish Lecture Ten ASAP.
4. Continue to work on Plutus' Alonzo Purple Test Net.
5. Continue to work on submission for Catalyst.