Using how to “read a paper” by S. Kshav as a guide

First pass

* Carefully read
  + Title: the Model checker spin(not much here)
  + Abstract
    - Efficient verification for distributed software systems
      * Detect design errors
      * Works on high level modeling
      * Works on detailed code modeling
    - Overview of design, structor, theoretical foundation, of verifier
    - Overview of example applications
  + Introduction
    - Generic support for verification of asyncrouns systems
    - Specificationwith
      * Randezwous primitives
      * Asynchronous message passing through buffered channels
      * Shared variables
      * Combinitions of above
    - Distingueses itself with focus on asynch software instead of sync in hardware
    - Main goals
      * Aimes
        + Intuitive
        + program like notation
        + Unambiguous
        + Details not neccicary (does this mean higher level not needing lower?)
      * Powerful concise notation for correctness requirements
      * Methodology for achieving aims and matching them with correctness verification
    - Structure
      * Start with high level specification of the model or algorithm
        + Spin checks for syntax errores which you fix
      * Use interactive till you feel things work as you want
      * Spin is used to generate omtimized on the fly verification program from high level specifications(what does this mean)
      * Runes the verification and spits out a counter example if applicable
* Read section and subsection headings but nothing else
  + Foundation
    - Temporal logic requirements
    - Domain of Application
  + Algorithms
    - Nested Depth-first Search
    - From LTL Forula to Buchi Automata
    - Partial Order Reduction
    - Memory Management
      * State Compression
      * Bit-State Hashing
  + Practical Applications
    - Process Scheduling
    - Leader Election
      * A modification
    - Flow Control
      * Safety Properties
      * Liveness Properties
    - Other Application
* Read conclusion
  + Scarce capapbilites for conccuurrent systems
  + Still build on the tools for this from before
  + Design methodology as follows
    - Distinction from behavor and requirements on behavior
      * Unambiguously defining the two aspects in a verification prototype in PROMELA
    - Prototype is verified using spin. Both requirements and behavier are checked for internal and mutal consistency.
    - Revise design until successful design, then refine design further towards full implementation

Answer

* Category(measurement, analysis, description, prototype)
  + Description of a “new” approach to formalizing distributed systems
* Context (related papers)
  + Pass for now
* Correctness
  + Seems correct mostly though I do want to look back at the logic and see if it is technically LTLf
* Contributions
  + Shows a new tool for proving distributed system specification
* Clarity
  + Felt very clear

Needed for other presentation.

Determan

* Background info
* Motivation(problem being soved or question answered)
  + Modeling system for distributed software systems
  + Generic support for verification of asyncrouns systems
  + Notation that follow aims as above
    - Promila used
  + Expressive notation for correctness
    - LTL
      * Technically wouldn’t it be LTLf since it can only work on finite traces?
* Products of the paper(specifically novel), tools, solutions, artifacts
* Reproducable
* Is it correct
* Buidable(exdent or utilize ofr different projects)
* Future work