BindingRestElement Binding Pattern

A Comedy of Errors? A useful pattern?
ES6 Grammar

ArrayBindingPattern[Yield] :
  `[
    Elision? BindingRestElement?[?Yield] `]
  `[
    BindingElementList?[?Yield] `]
  `[

BindingRestElement[Yield] :
  `... BindingIdentifier[?Yield]

ArrayAssignmentPattern[Yield] :
  `[
    Elision? AssignmentRestElement?[?Yield] `]
  `[
    AssignmentElementList?[?Yield] `]
  `[
    AssignmentElementList?[Yield] `, Elision? AssignmentRestElement?[?Yield] `]

AssignmentRestElement[Yield] :
  `... DestructuringAssignmentTarget[?Yield]`
What does it mean?

[ ... [ ... x ] ] = [1, 2, 3];
// Ok, AssignmentRestElement allows a
DestructuringAssignmentTarget

let [ ... [ ... x ] ] = [1, 2, 3];
// Syntax error, BindingRestElement only allows a BindingIdentifier
Why?

• Historical accident?
• Bug filed on error thrown by having an array literal after a rest.
  • https://bugs.ecmascript.org/show_bug.cgi?id=3361
• Bug "fixed" by allowing AssignmentRestElement to be a binding pattern.
• Did not update BindingRestElement at the same time.
Possible Avenues

1. Keep as is.
2. Remove support for a binding pattern in a rest element.
3. Align AssignmentPattern with BindingPattern to allow further destructuring of the rest element.

```
BindingRestElement[Yield] : 
  `...` BindingIdentifier[?Yield]
  `...` BindingPattern[?Yield]
```

Plus necessary static and runtime semantics to align with BindingPattern