Notes to Aycock and Horspool 2002

Jeffrey Kegler

June 2, 2010

Abstract

The 2002 paper by Aycock and Horspool, on a modification to Jay Earley's parser, has been essential to my work on general BNF parser. It is a difficult read for several reasons. First, the paper is dense with important, but difficult, insights. Second, a reader needs to be familiar with much previous work, including in the now-neglected field of general BNF parsing. Finally, there are a number of obscurities and some errors. This paper is a series of notes and errate intended to address the last problem.

1   THIS IS A DRAFT

This paper is in late stages, but not yet finished. It will contain many errors of the kind that the last drafts should remove.

2   Introduction

The 2002 paper by Aycock and Horspool contains a number of obscurities and some errors. This paper is a series of notes and errata intended to address the last problem. Notes are not distinguished from errata in what follows. The difference is a fine one, and it is not clear that it would serve the reader to know where this author draws the line in this particular cloud.

The discerning reader will realize that an exercise in nit-picking like this article is an especially sincere form of praise. Only if this author truly thought Aycock and Horspool's 2002 paper rewarded careful and repeated study, would he have collected these notes.

3   Lemma 5.2 is Incomplete

The proof of Lemma 5.2 seems incomplete. In case 4 on page 622, the last sentence begins "If $I'' \in S_i''$". There is no justification given in the proof for the assertion that $I'' \in S_i'$. A justification is necessary in order for Lemma 5.2 and Theorem 5.1 to go through.

Case 4 can be shown, and the proof successfully completed, if all 4 cases are wrapped in an induction. This induction could be on completion depth, where the completion depth of an item is defined as the number of completion steps (E_c or E_c') within the current Earley set needed to add that item to the Earley set. For example, a scanned item would have zero completion depth.

Lemma 5.3, case 4 relies on Lemma 5.2's case 4 for its justification, and therefore has the same issue.

4   Parent Pointers in Theorem 6.1

In the definitions leading up to Theorem 6.1 on page 624, $l \sqsubset S$ is stated to be equivalent to $[A \rightarrow \alpha\bullet\beta,\ j] \in S_i$, where j is never defined. Similarly, $L \sqsubset S_i$ is stated to be equivalent to $l \sqsubset S_i$ for all $l \in L$.

LR(0) items never appear alone as member of Earley sets. They are always in a duple with a parent pointer. Many of the subsequent statements using the $l \sqsubset S$ and $L \sqsubset S_i$ notations are only true for the correct choice of parent pointer.

The problem can be resolved, and all statements involving the square subset ($\sqsubset$) notation become true, if it is revised to include the parent pointer. This can be done by setting $l@j \sqsubset S$ to the equivalent of $[A \rightarrow \alpha\bullet\beta,\ j] \in S_i$, and $L@j \sqsubset S_i$ to be equivalent to $l@j \sqsubset S_i$ for all $l \in L$. The square subset notation is only used in column one on page 624. If "$@j$" is introduced before the square subset symbol everywhere it occurs, the expectations for the parent pointer are made clear. This change in notation leaves Theorem 6.1 true, and its proof correct.

5   LR(0) State versus LR(0) Item

On page 624, the notation for LR(0) states and LR(0) items, while not incorrect, is inconsistent is a way this is very likely to be confusing. In column one, LR(0) states are always designated with a capital letter (L), while LR(0) items are always designated with lowercase letters (l). But in the second paragraph of column two on the same page, LR(0) states are designated with the lowercase letter l , previously reserved for LR(0) items.

6   Prediction Items are included in LR(0) States

NOTE: In the second paragraph of page 624, this sentence occurs: "All items $[A \rightarrow \bullet\alpha]$ must be in l too". As pointed out above, use of L for the LR(0) state would be more consistent than the use of l.

Also, I found it necessary to remind myself why the assertion is true: By the definition of LR(0) states, whenever an item $[B \rightarrow ...\bullet A..., k]$ is in an LR(0) state, all prediction items for A must also be in that same LR(0) state.

7   `foreach` over Worklists

NOTE: In the pseudocode on page 625 and on page 627 it's important to realize that, in the `foreach` loops over the contents of Earley sets, the Earley sets should be implemented as "worklists". This is stated on page 625, but is easy to miss. Implementation as worklists means the the loop must be able to add new items as it proceeds, and that the iteration includes must include those newly added items. A naive implementation of a foreach loop would usually not provide worklist semantics.