b)

|  |  |  |  |
| --- | --- | --- | --- |
| **Stack** | **Buffer** | **New Dependedncy** | **Transition** |
| ROOT | I parsed this sentence correctly | - | Init |
| ROOT I | Parsed this sentence correctly | - | Shift |
| ROOT I parsed | This sentence correctly | - | Shift |
| ROOT I parsed this | Sentence correctly | - | Shift |
| ROOT I parsed this sentence | Correctly | - | Shift |
| ROOT I parsed sentence | Correctly | Sentence 🡪this | Left-arc |
| ROOT I parsed | correctly | Parsed 🡪 sentence | Right-arc |
| ROOT parsed | correctly | Parsed 🡪 I | Left-arc |
| ROOT parsed correctly | - | - | Shift |
| ROOT parsed | - | Parsed 🡪correctly | Right-arc |
| ROOT | - | ROOT 🡪parsed | Right-arc |

c) A sentence containing n words will be parsed in O)n) steps (assuming we know a sequence of steps that leads to a correct parsing):  
Each word is inserted exactly once to the buffer and extracted from it (using SHIFT) to the stack exactly once, so those steps take 2n steps.

Each word in the stack is extracted from it exactly once, using LEFT-ARC or RIGHT-ARC, so this add n steps.

All together we got 3n steps.

g) Best UAS on dev set: 88.63

Best UAS on test set: 89.29

h)   
i.   
**Error type:** Prepositional phrase attachment error  
**Incorrect dependency:** named 🡪 Midland  
**Correct dependency:** guy 🡪 Midland

ii.  
**Error type:** Modifier attachment error  
**Incorrect dependency:** elements 🡪 most  
**Correct dependency:** crucial 🡪 most

iii.  
**Error type:** Verb phrase attachment error  
**Incorrect dependency:** wedding 🡪 fearing  
**Correct dependency:** heading 🡪 fearing

iv.  
**Error type:** Coordination attachment error  
**Incorrect dependency:** makes 🡪 rescue  
**Correct dependency:** rush 🡪 rescue