BNF Tools Version 2.0: Migration Notes

Discovered features in version 1.0 and their migration status, as well as further migration notes are summarised below.

Core Features:

- 1. BNF front-end lexer, parser, linker, etc.
- 2. Model serialisation translates a textual representation to and from an Ecore model
- 3. IDE provides editors and other IDE features, e.g. outline, auto-completion, syntax highlighting, quick-fixing, etc.

Validation Rules:

These rules will give warnings in the editor if they find inconsistencies and constraint violations in the code.

- 1. *checkUpdateGrammarConsistency:* checks if the grammar is an updated grammar, for consistency validation between the original grammar and the new grammar
 - ♦ status: not migrated
- 2. *checkReferencedOnlyOnce:* checks if there are rules that are only referenced only once, this is an inlining hint
 - ♦ status: migrated
- 3. checkPassthroughRule: checks if a rule gets just passed through, e.g. a ::= b, b ::= "literal" <=> a
 - ::= "literal", which is better for EBNF
 - ♦ status: migrated
- 4. checkEqualAlternative: checks if a rule has equal alternatives, e.g.: a: := d + d
 - ♦ status: migrated
- 5. *checkDuplicateRules:* checks if there is another rule that has the same right hand side, e.g.: a ::= "literal" | b, c ::= "literal" | b

♦ status: migrated

- 6. *checkSubruleDuplicates:* checks if there are identical parts of right hand sides, this is an extract subrule hint e.g. a::= b c d and f::= e c d <=> a::= b g and f::= e g with g::= c d
 - ♦ status: not migrated
- 7. *checkUnusedRule:* checks if a rule never used (except for the first rule)
 - ♦ status: migrated
- 8. checkNameIsUnique: checks if the name of a rule is already in use e.g. a ::= "end", a ::= "fin"
 - ♦ status: migrated

Generation:

Generating an intermediate xsl-fo representation for all BNF grammars, which can then be rendered to PDF or RTF using apache-fop

• status: partially migrated, merge or delta grammars not supported (were they in the original implementation?)

Formatting:

Automated formatting support

- *legacy:* had some formatting support, but it was commented out in the main project
- *migrated:* line-break after every rule, white-spaces after all literals, possible update to line-break and double tab after all >l< bars

Refactoring and Quick-fixing:

Automated restructuring and quick fixing support.

- 1. auto numbering of rules
 - ♦ status: not migrated
- 2. remove unused rules
 - ♦ status: migrated
- 3. replacement of pass-through rules
- status: not migrated
- 4. uppercase token rules

♦ status: not migrated

- 5. find references
 - status: provided by xtext2

6. renaming

♦ status: provided by xtext2

Grammar Composition:

Three grammar types:

- grammar: normal grammar
- delta grammar: different notation for an extension grammar,
 - adds new rules to a grammar (regular rules) and adds changes to rules of the extended grammar (extension rules) (can be automatically created)
- merge grammar: for merging extensions

The legacy version had an automatic composition process for merge grammars with a small pop-up-menu.

• status: at the moment no support for grammar composition in the migrated version