Stuart Bourne 10091057
Andrew Storus 10103737
Ben Grady 10067106

Analysis of the Difference File

The table below summarizes the changes reflected in the corresponding files, along with a detailed explanation as to why each file was modified. Further details about the specific lines that were removed or changed from the original files can be found in the files IdentifiersDiff.txt, parserPtDiff.txt, ScandefDiff.txt and scanDiff.txt. These files are the direct output of the diff command being run on stdIdentifiers, parser.pt, scan.def and scan.ssl, respectively.

File Name	Line (Final)	Why We Changed it	What we Changed it to/from
Scan.ssl	37-40	We added new Character classes for /,#,!, and @	lSlash ‘/’ lHash ‘#’ lExclamation ‘!’ lAt ‘@’
Dsfd\Scan.ssl	59-84	Removing the list of old PT keywords tokens	pThen pEnd pUntil pDo pConst pProcedure pBegin pWhile pRepeat lastKeywordToken = pRepeat
Scan.ssl	70-84	Adding the list of new JT keywords tokens	pLet pFunction pSwitch pDefault pElsif pLoop pBreak pWhen pClass pPublic pHash pAt pLeftBrace pRightBrace lastKeywordToken = pRightBrace
Scan.ssl	103	Deleting the old Assignment operator tokens	pColonEquals
Scan.ssl	110	Adding the new Assignment operator tokens	pDoubleEquals
scan.ssl	158-164	Commenting out the old S/SL logic for comments	%[% | ‘*’ : % @AlternateComment % | * : % .pLeftParen %]
scan.ssl	174	Removed old Assignment operator S/SL logic so that colon would always emit pColon and not pColonEquals	.pColon
scan.ssl	184-188	Adding the S/SL logic for pHash and pAt	|’#’: .pHash |’@’: .pAt
scan.ssl	190-195	Added double equals and assignment operator S/SL logic	[|’=’ :

<table>
<tr><td></td><td></td><td></td><td>.pDoubleEquals
|*:
.pEquals
]</td></tr>
<tr><td>scan.ssl</td><td>209-217</td><td>Added notEquals S/SL logic and edited lessThan logic</td><td>| '!':
[
| '=':
.pNotEqual
| *:
#eIllegalChar
]
| '<':
[
| '=':
.pLessEqual
| *:
.pLess
]</td></tr>
<tr><td>scan.ssl</td><td>230-243</td><td>Changed old comment styles and added new commenting S/SL logic</td><td>| '{':% Don't need old comment styles
.pLeftBrace
| '}':
.pRightBrace
| '/': % new comment styles
[
| '*': %multiline comments
@BlockComment
| '/':%single line comments
@Comment
| *:%single slash does not compute
#eIllegalChar
]</td></tr>
<tr><td>scan.ssl</td><td>301-310</td><td>Implemented an inline commenting subroutine</td><td>Comment :
% Discard the contents of a comment
{[
| lEndFile: %Comment could be last line of the program
.pEndFile
>
| lNewLine: %comment in most cases ended by a new line
.pNewLine
>
| *: % comment may contain any other character
?
]};</td></tr>
<tr><td>scan.ssl</td><td>311-327</td><td>Implemented a block commenting subroutine</td><td>BlockComment :
{[
| '*':
[
| '/':
></td></tr>
</table>

			| *:] | lNewLine: .pNewLine | lEndFile: #eCommentEOF .pEndFile > | *: ?]};

parser.pt	53-83	Updating integer values of input tokens to match scan.def	lLetter = 0;
			firstCharClass = 0;
			lDigit = 1;
			lPlus = 2;
			lMinus = 3;
			lStar = 4;
			lColon = 5;
			lEquals = 6;
			lDot = 7;
			lComma = 8;
			lSemicolon = 9;
			lQuote = 10;
			lLeftAngle = 11;
			lRightAngle = 12;
			lLeftParen = 13;
			lRightParen = 14;
			lLeftBracket = 15;
			lRightBracket = 16;
			lLeftBrace = 17;
			lRightBrace = 18;
			lSlash = 19;
			lHash = 20;
			lExclamation = 21;
			lAt = 22;
			lBlank = 23;
			lIllegal = 24;
			lNewLine = 25;
			lTab = 26;
			lNewPage = 27;
			lEndFile = 28;
			lastCharClass = 28;

parser.pt	96-154	Updating integer values of input tokens to match scan.def	pDiv = 0;
			firstKeywordToken = 0;
			pMod = 1;
			pOr = 2;
			pAnd = 3;
			pNot = 4;
			pElse = 5;
			pOf = 6;
			pArray = 7;
			pFile = 8;

			pPacked = 9;
			pProgram = 10;
			pVar = 11;
			pType = 12;
			pIf = 13;
			pCase = 14;
			pLet = 15;
			pFunction = 16;
			pSwitch = 17;
			pDefault = 18;
			pElsif = 19;
			pLoop = 20;
			pBreak = 21;
			pWhen = 22;
			pClass = 23;
			pPublic = 24;
			lastKeywordToken = 24;
			pIdentifier = 25;
			firstCompoundInputToken = 25;
			pInteger = 26;
			pLiteral = 27;
			lastCompoundInputToken = 27;
			pNewLine = 28;
			pEndFile = 29;
			pPlus = 30;
			pMinus = 31;
			pStar = 32;
			pDot = 33;
			pComma = 34;
			pSemicolon = 35;
			pColon = 36;
			pEquals = 37;
			pNotEqual = 38;
			pDoubleEquals = 39;
			pLess = 40;
			pLessEqual = 41;
			pGreaterEqual = 42;
			pGreater = 43;
			pLeftParen = 44;
			pRightParen = 45;
			pLeftBracket = 46;
			pRightBracket = 47;
			pDotDot = 48;
			pHash = 49;
			pAt = 50;
			pLeftBrace = 51;
			pRightBrace = 52;
			lastSyntaxToken = 52;

parser.pt	308	Updating the numberStdIdentifiers	numberStdIdentifiers = 44;
parser.pt	332	Changed what is recognised as a quote from ‘ to “	quote = '"';
parser.pt	829-832	Added the character class maps for	charClassMap[ord('/')] := lSlash;

		/, !, #, @	charClassMap[ord('!')] := lExclamation;
			charClassMap[ord('#')] := lHash;
			charClassMap[ord('@')] := lAt;

parser.pt	1768-1686	Added necessary logic to break the parser.	parseInputToken := pNewLine;
			newInputLine := false;
			while parseInputToken <> pEndFile do
			AcceptSyntaxToken;
			{Parser;}

stdIdentifiers	13-21	Removed then, end, until, do, const, procedure, begin, while, repeat and char keywords and replaced with let, function, switch, default, elsif, loop, break, when, class, public and string keywords.	let function switch default elsif loop break when class public

Scan.def	25-120	Added necessary corresponding integers by running ssl scan.ssl	lSlash = 19;
			lHash = 20;
			lExclamation = 21;
			lAt = 22;
			lBlank = 23;
			lIllegal = 24;
			lNewLine = 25;
			lTab = 26;
			lNewPage = 27;
			lEndFile = 28;
			lastCharClass = 28;
			
			{ Output Tokens }
			pDiv = 0;
			firstKeywordToken = 0;
			pMod = 1;
			pOr = 2;
			pAnd = 3;
			pNot = 4;
			pElse = 5;
			pOf = 6;
			pArray = 7;
			pFile = 8;
			pPacked = 9;
			pProgram = 10;
			pVar = 11;
			pType = 12;
			pIf = 13;
			pCase = 14;
			pLet = 15;
			pFunction = 16;
			pSwitch = 17;
			pDefault = 18;
			pElsif = 19;

			pLoop = 20;
			pBreak = 21;
			pWhen = 22;
			pClass = 23;
			pPublic = 24;
			lastKeywordToken = 24;
			pIdentifier = 25;
			firstCompoundInputToken = 25;
			pInteger = 26;
			pLiteral = 27;
			lastCompoundInputToken = 27;
			pNewLine = 28;
			pEndFile = 29;
			pPlus = 30;
			pMinus = 31;
			pStar = 32;
			pDot = 33;
			pComma = 34;
			pSemicolon = 35;
			pColon = 36;
			pEquals = 37;
			pNotEqual = 38;
			pDoubleEquals = 39;
			pLess = 40;
			pLessEqual = 41;
			pGreaterEqual = 42;
			pGreater = 43;
			pLeftParen = 44;
			pRightParen = 45;
			pLeftBracket = 46;
			pRightBracket = 47;
			pDotDot = 48;
			pHash = 49;
			pAt = 50;
			pLeftBrace = 51;
			pRightBrace = 52;
			lastSyntaxToken = 52;
			
			{ Input/Output Tokens }
			
			{ Error Codes }
			eIllegalChar = 5;
			firstScanError = 5;
			eLiteralLine = 6;
			eLiteralEOF = 7;
			eCommentEOF = 8;
			eNoSeparator = 9;
			lastScanError = 9;
			
			{ Type Values }
			
			{ S/SL Rule Table Addresses } {
			Scan = 0;

			Identifier = 252;
			Integer = 269;
			Literal = 284;
			BlockComment = 351;
			Comment = 328;
			}
			
			{ S/SL Table Parameters }
			sslTblSize = 387;