Grammar for Language Describing Game Structures

Brooks MacLachlan

March 27, 2019

Pretty-printed EBNF grammar:

⟨*game*⟩ ::= 'Players:' ⟨*teamList*⟩ 'Rounds:' ⟨*roundList*⟩ 'Win:' ⟨*winConditionList*⟩

⟨*teamList*⟩ ::= (⟨*team*⟩ {',' ⟨*team*⟩} | ⟨*playerList*⟩ | 'randomly divide' ⟨*playerList*⟩ 'into' ⟨*nameList*⟩) ['all with' ⟨*attributeList*⟩]

⟨*team*⟩ ::= 'team' ⟨*name*⟩ ':' ⟨*playerList*⟩

⟨*playerList*⟩ ::= ⟨*player*⟩ {',' ⟨*player*⟩}

⟨*player*⟩ ::= ⟨*name*⟩ ['with' ⟨*attributeList*⟩]

⟨*attributeList*⟩ ::= ⟨*attribute*⟩ {',' ⟨*attribute*⟩}

⟨*attribute*⟩ ::= 'affiliation called' ⟨*name*⟩ | ('score' | 'resource' | 'counter') 'called' ⟨*name*⟩ ['starting at' ⟨*number*⟩] ['with minimum of' ⟨*number*⟩] ['with maximum of' ⟨*number*⟩]

⟨*roundList*⟩ ::= ⟨*round*⟩ ';' {⟨*round*⟩}

⟨*round*⟩ ::= ⟨*phaseList*⟩ ['repeated' ⟨*number*⟩ 'times' ['with modifications' ':' ⟨*modifierList*⟩]]

⟨*modifierList*⟩ ::= ⟨*modifier*⟩ ';' {⟨*modifier*⟩}

⟨*modifier*⟩ ::= ['just' | 'from'] (⟨*roundReference*⟩ ('before' | 'after' | 'instead of') ⟨*phaseReference*⟩ 'insert' ⟨*phase*⟩)

⟨*roundReference*⟩ ::= 'round' ⟨*number*⟩

⟨*phaseReference*⟩ ::= 'phase' ⟨*number*⟩

⟨*phase*⟩ ::= ⟨*action*⟩ | ⟨*progression*⟩

$\langle phaseList \rangle ::= \langle phase \rangle$ {'`\n`' $\langle phase \rangle$}

$\langle action \rangle ::= \langle competition \rangle \mid \langle decision \rangle$

$\langle competition \rangle ::= $ ['`scored`'] ['`team`'] '`competition between`' $\langle identifierList \rangle$

$\langle decision \rangle ::= $ ['`self-included`'] '`vote by`' $\langle identifierList \rangle$ '`between`' $\langle identifierList \rangle$
    $\mid$ ['`self-included`'] '`nomination of`' $\langle number \rangle$ '`by`' $\langle identifierList \rangle$ '`between`'
    $\langle identifierList \rangle \mid$ '`allocation of`' $\langle name \rangle$ '`by`' $\langle identifierList \rangle \mid$ ['`self-included`']
    '`directed vote by`' $\langle identifierList \rangle$ '`between`' $\langle identifierList \rangle \mid$ '`uses?`' $\langle identifier \rangle$
    '`then`' '`(`' $\langle phaseList \rangle$ '`)`' ['`otherwise`' '`(`' $\langle phaseList \rangle$ '`)`']

$\langle progression \rangle ::= (\langle affiliationUpdate \rangle \mid \langle counterUpdate \rangle)$ ('`for`' $\mid$ '`of`') $\langle identifierList \rangle$

$\langle affiliationUpdate \rangle ::= $ '`elimination`' $\mid$ ('`add`' $\mid$ '`remove`') $\langle name \rangle \mid$ ['`number preserving`']
    '`swap`' $\langle nameList \rangle$ ['`adding`' $\langle nameList \rangle$] $\mid$ '`change`' $\langle name \rangle$ '`to`' $\langle name \rangle \mid$
    '`merge`' $\langle nameList \rangle$ [$\langle name \rangle$]

$\langle counterUpdate \rangle ::= $ ('`increase`' $\mid$ '`decrease`') $\langle name \rangle$ '`by`' $\langle value \rangle \mid$ '`set`' $\langle name \rangle$
    '`to`' $\langle value \rangle$

$\langle value \rangle ::= \langle number \rangle \mid \langle name \rangle \mid$ '`results`' '`of`' ($\langle compReference \rangle \mid \langle allocateReference \rangle$
    $\mid \langle voteReference \rangle$)

$\langle identifierList \rangle ::= \langle idValList \rangle$ ['`except`' $\langle idList \rangle$]

$\langle idList \rangle ::= \langle identifier \rangle$ {'`,`' $\langle identifier \rangle$}

$\langle idValList \rangle ::= \langle identifierVal \rangle$ {'`,`' $\langle identifierVal \rangle$}

$\langle identifierVal \rangle ::= \langle identifier \rangle$ ['`*`' $\langle value \rangle$]

$\langle identifier \rangle ::= $ '`everyone`' $\mid \langle name \rangle \mid$ '`chance`' '`(`' $\langle identifierList \rangle$ '`)`' $\mid$ '`nominated`'
    $\mid$ '`tied`' $\mid$ '`eliminated`' $\mid$ ('`winner`' $\mid$ '`loser`') '`of`' $\langle compReference \rangle \mid$ ('`majority`'
    $\mid$ '`minority`') '`of`' $\langle voteReference \rangle$ [$\langle tiebreaker \rangle$] $\mid$ ('`highest`' $\mid$ '`lowest`' $\mid$
    '`most`' $\mid$ '`least`') $\langle name \rangle$ '`(`' $\langle identifierList \rangle$ '`)`' [$\langle tiebreaker \rangle$]

$\langle compReference \rangle ::= $ '`competition`' [$\langle number \rangle$]

$\langle allocateReference \rangle ::= $ '`allocation`' [$\langle number \rangle$]

$\langle voteReference \rangle ::= $ '`vote`' [$\langle number \rangle$]

$\langle tiebreaker \rangle ::= $ '`tiebroken by`' ([$\langle action \rangle$] $\langle identifier \rangle$)

$\langle name \rangle ::= $ ('`a`' $\mid ... \mid$ '`z`' $\mid$ '`A`' $\mid ... \mid$ '`Z`') {'`a`' $\mid ... \mid$ '`z`' $\mid$ '`A`' $\mid ... \mid$ '`Z`' $\mid$ '`0`' $\mid ... \mid$ '`9`'}

$\langle nameList \rangle ::= \langle name \rangle$ {',' $\langle name \rangle$}

$\langle number \rangle ::=$ ['-'] ('0' | ... | '9')+

$\langle winCondition \rangle ::=$ ['team'] 'reach' $\langle name \rangle$ 'of' $\langle number \rangle$ | ['team'] 'competition'
    | ['team'] ('highest' | 'lowest' | 'most' | 'least') $\langle name \rangle$ | $\langle number \rangle$ 'member jury vote'
    | 'survive'

$\langle winConditionList \rangle ::= \langle winCondition \rangle$ {',' $\langle winCondition \rangle$}

Notes:

- make grammar unambiguous

- should allocate be a decision and progression or just progression? Both, but decision will automatically also do the progression. The progression exists for non-decision related resource updates, such as awarding a player more of a resource for winning a competition.