Taylor's LaTeXTest Document

Taylor DeMint

April 25, 2018

Algorithm 1: STR($\mathcal{P}$)

```
Input: $\mathcal{P}$: a CSP instance
Output: $\mathcal{P}$ after enforcing GAC
$domains \leftarrow domains(\mathcal{P})$
$\mathcal{Q} \leftarrow constraints(\mathcal{P})$
while $\mathcal{Q} \neq \emptyset$ do
    $constraint \leftarrow$ Pop($\mathcal{Q}$)
    $newDomain \leftarrow$ Revise($constraint$)
    foreach $var \in newDomain$ do
        if $newDomain[var] \neq domains[var]$ then
            $domains[var] \leftarrow domains[var] \cap newDomain[var]$
            $\mathcal{Q} \leftarrow \mathcal{Q} \cup ((constraints(\mathcal{P}) \ni var) \setminus constraint)$
        if $domains[var] = \emptyset$ then
            throw inconsistent
return $\mathcal{P}$
```

Algorithm 2: STR($\mathcal{P}$)

```
Input: $\mathcal{P}$: a CSP instance
Output: $\mathcal{P}$ after enforcing GAC, $null$ if inconsistent
$\mathcal{Q} \leftarrow constraints(\mathcal{P})$
while $\mathcal{Q} \neq \emptyset$ do
    $C_i \leftarrow$ Pop($\mathcal{Q}$)
    foreach $var \in scope(C_i)$ do
        if ReviseDomain($var, C_i$) then
            if $domain(var) = \emptyset$ then
                return $null$
            else
                $\mathcal{Q} \leftarrow \mathcal{Q} \cup ((constraints(\mathcal{P}) \ni var) \setminus C_i)$
        else if ReviseConstraint($C_i, var$) then
            if $relation(C_i) = \emptyset$ then
                return $null$
return $\mathcal{P}$
```

Algorithm 3: ReviseConstraint(C_i, var)

```
Input: $C_i$: a table constraint
Input: $var$: a CSP variable such that $var \in scope(C_i)$
Output: $true$ if the relation of the constraint is revised, $false$
        otherwise
$revised \leftarrow false$
foreach $tuple \in relation(C_i)$ do
    if $isAlive(tuple)$ then
        if $\pi_{var}(tuple) \notin domain(var)$ then
            $isAlive(tuple) \leftarrow false$
            $count(C_i) \leftarrow count(C_i) - 1$
            $revised \leftarrow true$
return $revised$
```

Algorithm 4: ReviseDomain(var, C_i)

```
Input: $var$: a CSP variable
Input: $C_i$: a table constraint such that $var \in scope(C_i)$
Output: $true$ if the domain is revised, $false$ otherwise
$revised \leftarrow false$
$domain \leftarrow \emptyset$
foreach $tuple \in relation(C_i)$ do
    if $isAlive(tuple)$ then
        $domain \leftarrow domain \cup \pi_{var}(tuple)$
$newDomain \leftarrow domain \cap domain(var)$
if $newDomain \neq domain(var)$ then
    $domain(var) \leftarrow newDomain$
    $revised \leftarrow true$
return $revised$
```