glpPB ILP-based Pseudo-Boolean SAT Solver

Hossein M. Sheini hsheini@umich.edu
Electrical Engineering and Computer Science Department,
University of Michigan, Ann Arbor, MI

This document is a brief description of the glpPB ILP-based Pseudo-Boolean (PB) SAT Solver version 0.2. This solver is developed at the University of Michigan, Ann Arbor, MI by Hossein Sheini, supervised by Professor Karem A. Sakallah. The solver is available for download at

http://www.eecs.umich.edu/~hsheini/pueblo.

glpPB is an application of GNU Linear Programming Kit (GLPK) version 4.10 and is capable to read OPB format. Details on GLPK can be obtained from:

http://www.gnu.org/software/glpk/

The glpPB's interfacing code using GLPK API is as follow (excluding parsing routines):

```
LPX *XPS;
XPS = lpx_create_prob();
lpx_set_class(XPS, LPX_MIP);
lpx_set_int_parm(XPS, LPX_K_MSGLEV, 0);

// parse OPB input file and write to XPS

lpx_order_matrix(XPS);
lpx_scale_prob(XPS);
lpx_adv_basis(XPS);
if(lpx_simplex(XPS) != LPX_E_OK) {
  printf("c ERROR in LA Solver [1]\n"); exit(0);
}
int res = lpx_get_status(XPS);

if(res == LPX_INFEAS || res == LPX_NOFEAS) {
  fprintf(stdout, "s UNSATISFIABLE\n"); exit(20);
}
if(res == LPX_UNBND || res == LPX_UNDEF) {
  printf("c ERROR in LA Solver [2]\n"); exit(0);
}

if(lpx_integer(XPS) != LPX_E_OK) {
  printf("c ERROR in MIP Solver [1]\n"); exit(0);
}
res = lpx_mip_status(XPS);
```

```
if(res == LPX_I_UNDEF) {
  printf("c ERROR in MIP Solver [2]\n"); exit(0);
}

if(res == LPX_I_NOFEAS) {
  fprintf(stdout, "s UNSATISFIABLE\n"); exit(20);
}

LPXKKT *kkt = new LPXKKT;
lpx_check_int(XPS, kkt);
printf("c quality of integer solution is %c\n", kkt->pe_quality);

if(kkt->pe_quality != 'H') {
  printf("c REJECTING the solution due to poor quality\n");
  fprintf(stdout, "s UNKNOWN\n"); exit(0);
}

fprintf(stdout, "s OPTIMUM FOUND\n");
fprintf(stdout, "v ");
for(int i=0; i<lpx_get_num_cols(XPS); i++) {
  if( lpx_mip_col_val(XPS, i+1) == 0.0 ) fprintf(stdout, "-");
  fprintf(stdout, "%s ", lpx_get_col_name(XPS, i+1));
}
fprintf(stdout, "\n");
exit(30);
```