The Medicaid Notch

TheoryGuru applications

Load Economicreasoning package only if it is not already loaded

```
If[Length@Names["PLTools`*"] < 10,
 Get["http://economicreasoning.com"]]
```

Notes

c and *n* denote consumption and amount worked, respectively.
u[*c*,*n*] is the household utility function.

c' and *n'* denote ANY alternative that is feasible, but not optimal, when there is no Medicaid program.

c_m denotes the consumption that a household working *n'* would have if there were a Medicaid program.

FPL refers to the household-income cutoff for receiving benefits under the Medicaid program. Its dollar amount depends on household composition and state of residence, but is roughly the Federal Poverty Line.

Definitions

```
povertycheck[income_] :=
  {income ≤ FPL ⇒ poverty[income] == 1, income > FPL ⇒ poverty[income] == 0};

OnBudgetContraintwoMedicaid[c_, n_] := c == w n

OnBudgetContraintwMedicaid[c_, n_] := c == w n + poverty[w n] m

ConsumptionisaGood =
  (u[cm, n'] - u[c', n']) (cm - c') > 0 || (cm == c' && u[cm, n'] == u[c', n']);
```

```
cnistheOptimumwoMedicaid = u[c, n] > u[c', n'];
```

Results

It is feasible for Medicaid to raise utility, but it must do so by pushing (or keeping) the household in poverty

```
TheoryGuru[{cnistheOptimumwoMedicaid,
  m > 0, povertycheck[w n'],
  ConsumptionisaGood,
  OnBudgetContraintwoMedicaid[c', n'],
  OnBudgetContraintwMedicaid[cm, n'],
  u[cm, n'] > u[c, n]},
 poverty[w n'] == 1]
```

True

I.e., it is necessary (but not sufficient) for the household to choose poverty in order to benefit from Medicaid

```
TheoryOverlap[{cnistheOptimumwoMedicaid,
  m > 0, povertycheck[w n'],
  ConsumptionisaGood,
  OnBudgetContraintwoMedicaid[c', n'],
  OnBudgetContraintwMedicaid[cm, n']},
 u[cm, n'] > u[c, n],
 poverty[w n'] == 1]
```

poverty(w n′) = 1 is necessary but not sufficient for $u(c_m, n') > u(c, n)$

Medicaid cannot induce a household to exit poverty

```
AnyAlternativecmIseitherPovertyorUndesirable =
  poverty[w n'] == 1 || u[cm, n'] < u[c, n];

TheoryGuru[{cnistheOptimumwoMedicaid,
  m > 0, povertycheck[w n'],
  ConsumptionisaGood,
  OnBudgetContraintwoMedicaid[c', n'],
  OnBudgetContraintwMedicaid[cm, n']},
 AnyAlternativecmIseitherPovertyorUndesirable]
```

True

A Medicaid household that is indifferent to earning above the poverty line must have income exactly at FPL

```
QuasiConcaveUtility =
  0 ≤ x ≤ 1 ⇒ (u[c, n] ≤ u[cm, nm] && u[x c + (1 - x) cm, nFPL] ≥ u[c, n] ||
     u[c, n] ≥ u[cm, nm] && u[x c + (1 - x) cm, nFPL] ≥ u[cm, nm]);

ConsumptionisaGood = (u[x c + (1 - x) cm, nFPL] - u[cFPL, nFPL]) (x c + (1 - x) cm - cFPL) > 0 ||
   (cFPL == x c + (1 - x) cm && u[x c + (1 - x) cm, nFPL] == u[cFPL, nFPL]);

DefineWorkScenarios = {w nFPL == FPL, 0 ≤ nm ≤ nFPL < n, nFPL == x n + (1 - x) nm};

IndifferenttoEarningAboveFPL =
  u[c, n] == u[cm, nm](* best that can be done on Medicaid *) ≥ u[cFPL, nFPL];

TheoryGuru[{m > 0, IndifferenttoEarningAboveFPL,
  DefineWorkScenarios,
  povertycheck[w nm], povertycheck[w nFPL],
  ConsumptionisaGood,
  QuasiConcaveUtility,
  OnBudgetContraintwoMedicaid[c, n],
  OnBudgetContraintwMedicaid[cm, nm],
  OnBudgetContraintwMedicaid[cFPL, nFPL]},
 w nm == FPL]
```

```
True
```

If the household chooses poverty *because* of the Medicaid program, its income can be strictly below FPL

c and *n* denote consumption and amount worked, respectively, without Medicaid.

c_m and n_m denote consumption and amount worked, respectively, with Medicaid.

c' and *n'* denote ANY alternative that is feasible, but not optimal, when there is no Medicaid program.

```
SetOptions[TheoryGuru, keepall → True];
```

```
TheoryGuru[{m > 0, u[cm, nm] > u[c, n] ≥ u[c', n'],
  u[cm, nm] ≥ u[cFPL, nFPL],
  DefineWorkScenarios,
  povertycheck[w nm], povertycheck[w n'], povertycheck[w nFPL],
  ConsumptionisaGood,
  QuasiConcaveUtility,
  OnBudgetContraintwoMedicaid[c, n],
  OnBudgetContraintwoMedicaid[c', n'],
  OnBudgetContraintwMedicaid[cm, nm],
  OnBudgetContraintwMedicaid[cFPL, nFPL], w > 0},
 w nm < FPL]
```

The mutually consistency of the assumptions needs to be verified.

True for some, False for others

```
Column@Sort@TheoryInstanceR[]
```

Using MostRecentGuruTheory.

$c \to 1$
$FPL \to \frac{1}{2}$
$m \to 1$
$n \to 1$
$w \to 1$
$x \to \frac{1}{2}$
$\text{poverty}[w\, n_{FPL}] \to 1$
$\text{poverty}[w\, n_m] \to 1$
$\text{poverty}[w\, n'] \to 0$
$c_{FPL} \to \frac{3}{2}$
$c_m \to 1$
$n_{FPL} \to \frac{1}{2}$
$n_m \to 0$
$u[c, n] \to -1$
$u[c_{FPL}, n_{FPL}] \to 0$
$u[c_m, n_m] \to 1$
$u[c\, x + (1 - x)\, c_m, n_{FPL}] \to -1$
$u[c', n'] \to -1$
$c' \to 1$
$n' \to 1$

```
SetOptions[TheoryGuru, keepall → False];
```

Variable interpretations